POLITECHNIKA RZESZOWSKA
im. IGNACEGO ŁUKASIEWICZA

WYDZIAŁ ELEKTROTECHNIKI I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ

ROZPRAWA DOKTORSKA

mgr inż. Dawid Kalandyk

Aplikacje metod sztucznej inteligencji ze szczególnym uwzględnieniem algorytmu uczenia się ze wzmocnieniem

w formie cyklu publikacji naukowych

Promotor

dr hab. inż. Roman Zajdel, prof. PRz

Rzeszów, wrzesień 2024

Podziękowania

Pragnę serdecznie podziękować wszystkim osobom, które przyczyniły się do powstania tej rozprawy. Szczególne podziękowania składam na ręce Profesora Romana Zajdla, promotora mojej pracy doktorskiej, za poświęcony czas, nieustanne motywowanie do rozwoju i cenne rady. Dziękuję promotorowi pomocniczemu doktorowi Bogdanowi Kwiatkowskiemu, za inspirację tematyką związaną z maszynami CNC. Chciałbym również podziękować Profesorowi Tomaszowi Kapuścińskiemu oraz Profesorowi Mariuszowi Oszustowi za życzliwość i służenie dobrą radą. Dziękuję też wszystkim współautorom publikacji wchodzących w skład tego cyklu. Szczególne podziękowania składam najbliższym oraz przyjaciołom, którzy zawsze mnie wspierali.

Spis treści

Wykaz symboli, oznaczeń i akronimów

ASIMO – Advanced Step in Innovative Mobility

CNC – Computer Numeric Control

CNN – Convolutional Neural Network

CRNN – Convolutional Recurrent Neural Network

DDQN – Double Deep Q-Learning

DRL – Deep Reinforcement Learning

DTW – Dynamic Time Warping

FIS – Fuzzy Inference System

flops – Floating Point Operations Per Second

FPS – First Player Shooter

GANs – Generative Adversarial Networks

GSADGM – Gray Scale Absolute Difference Gradient Method

HMM – Hidden Markov Model

k-NN – k-Nearest Neighbours

LOSO – Leave One Subject Out

LSTM – Long Short-Term Memory

PJM - Polski Język Migowy

PSO – Particle Swarm Optimization

RL – Reinforcement Learning

RPRO – Reference Points Realization Optimization

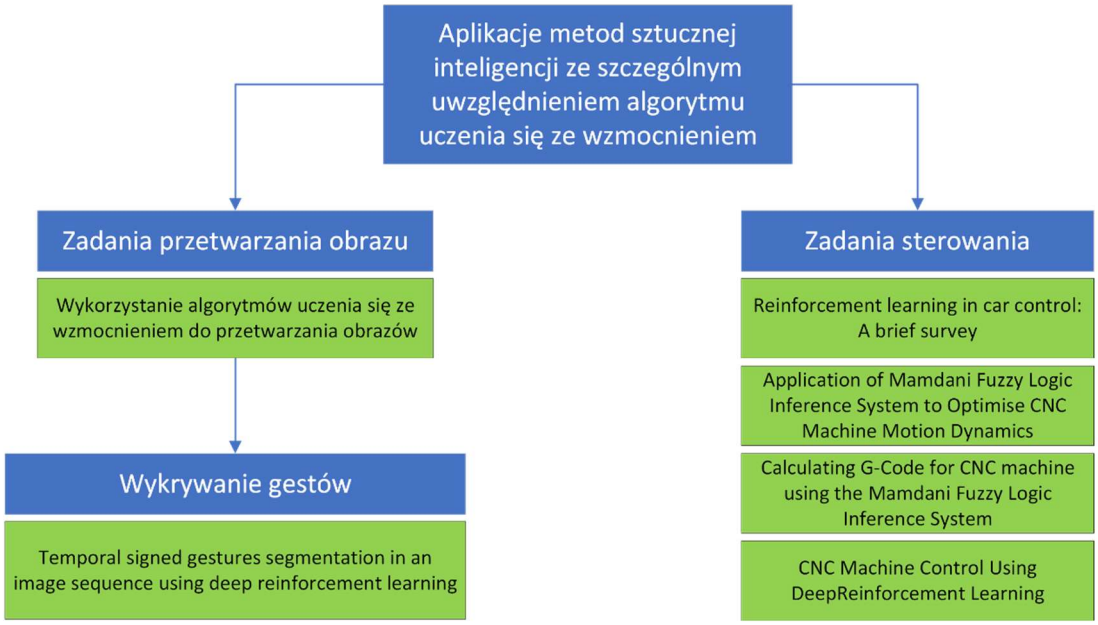
SI – Sztuczna Inteligencja

SOM – Self-Organizing Map

SRI – Stanford Research Institute

1. Wprowadzenie

Niniejsza rozprawa doktorska stanowi monotematyczny cykl publikacji naukowych dotyczących aplikacji metod sztucznej inteligencji ze szczególnym uwzględnieniem algorytmu uczenia się ze wzmocnieniem. Praca osadzona jest w dyscyplinie Informatyka Techniczna i Telekomunikacja, a część omówionych prac porusza tematykę sterowania związaną z dyscypliną Automatyka, Elektronika, Elektrotechnika i Technologie Kosmiczne. Praca jest podzielona na dwie główne gałęzie tematyczne: zadania przetwarzania obrazu oraz zadania sterowania. Pierwsza z nich dotyczy przetwarzania danych wizualnych z propozycją algorytmu wykrywania gestów poprzez czasową segmentację strumienia wideo. W drugiej gałęzi skupiono się na sterowaniu dynamiką ruchu wrzeciona maszyny sterowanej numerycznie (ang. Computer Numeric Control – CNC). Artykuły ujęte w cyklu zostały zebrane na diagramie przedstawionym na Rys. 1. Kolorem niebieskim oznaczono tematykę badań, natomiast zielone pola zawierają tytuły prac omówionych w dalszej części niniejszej dysertacji. Większa część opisanych badań została zaprezentowana w trakcie międzynarodowych konferencji naukowych.



Rys. 1 Artykuły ujęte w cyklu [opracowanie własne]

W skład niniejszej rozprawy doktorskiej wchodzi sześć publikacji:

[A-1] Kalandyk, D. (2021). Rozdział monografii „Nowoczesne technologie – strategie, rozwiązania i perspektywy rozwoju. Tom 1" pt. „Wykorzystanie algorytmów uczenia się ze wzmocnieniem do przetwarzania obrazów", str. 180-223; http://bc.wydawnictwo-tygiel.pl/publikacja/1A77370A-0E87-0A33-F834-334902710840; wkład: 100%, (autor korespondencyjny)

[A-2] Kalandyk, D. (2021). Reinforcement learning in car control: A brief survey. 2021 Selected Issues of Electrical Engineering and Electronics (WZEE), 1-8. https://doi.org/10.1109/WZEE54157.2021.9576838; wkład: 100%; (autor korespondencyjny)

[A-3] Kalandyk, D., Kwiatkowski, B., & Mazur, D. (2023, August). Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics. In 2023 IEEE International Conference on Fuzzy Systems (FUZZ) (pp. 1-4). IEEE. https://doi.org/10.1109/FUZZ52849.2023.10309802; wkład 70%; liczba punktów czerwiec 2023: 140; liczba punktów lipiec 2024: 70; (autor korespondencyjny)

[A-4] Kalandyk, D., Kwiatkowski, B., & Mazur, D. CNC Machine Control Using Deep Reinforcement Learning. Bulletin of the Polish Academy of Sciences Technical Sciences, e148940-e148940. https://doi.org/10.24425/bpasts.2024.148940; wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.8

[A-5] Kalandyk, D., & Kapuściński, T. (2024). Temporal signed gestures segmentation in an image sequence using deep reinforcement learning. Engineering Applications of Artificial Intelligence, 131, 107879. https://doi.org/10.1016/j.engappai.2024.107879; wkład 90%; liczba punktów: 140; IF: 8.0; CS: 12.3; (autor korespondencyjny)

[A-6] Kalandyk, D., Kwiatkowski, B., & Mazur, D. Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System. Archives of Control Sciences, artykuł przyjęty do publikacji, wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.7; (autor korespondencyjny)

Pięć z pośród sześciu prac zostało opublikowanych w bazach SCOPUS oraz Web Of Science. Wszystkie prace zostały opublikowane w latach 2021-2024. Sumaryczny Impact Factor (zgodnie z rokiem ukazania się publikacji) wynosi 11.6, sumaryczny wskaźnik Cite Score wynosi 20.5, a liczba punktów MNiSW (punktacja czasopism naukowych zgodnie z aktualnym wykazem MNiSW) wynosi 510.

1.1. Motywacja oraz stan wiedzy

Sztuczna Inteligencja (SI) jest definiowana[1] jako zdolność maszyn do wykazywania ludzkich umiejętności m.in.: wnioskowania, uczenia się, planowania, czy nawet kreatywności. Początki Sztucznej Inteligencji przypadają na 1950 rok, kiedy to jej prekursor oraz pionier w dziedzinie komputerów i logiki Alan Turing przedstawił artykuł pt. „Computing Machinery and Intelligence" [1], w którym postawił bardzo ogólne, lecz kluczowe pytanie: „Czy maszyny mogą myśleć?". Można powiedzieć, że tym samym rozpoczął on pewną epokę w dziejach świata. Sam termin Sztuczna Inteligencja formalnie pojawił się sześć lat później, tj. w 1956 roku na konferencji w Dartmouth[2], którą zorganizowało czterech naukowców: John McCarthy, Marvin Minsky, Nathaniel Rochester oraz Claude Shannon.

Jednym z pierwszych sukcesów na polu „myślących maszyn" był program o nazwie ELIZA [2] utworzony przez Josepha Weizenbauma w 1966 roku, który poprzez analizę słów kluczowych oraz kontekstu wypowiedzi potrafił sprawić wrażenie rozmowy z człowiekiem. Jego celem była pomoc w terapii osobom z zaburzeniami psychicznymi. Niespełna trzy lata później Instytut Badań Stanford (ang. Stanford Research Institute - SRI[3]) zbudował pierwszego autonomicznego robota o nazwie Shakey[4]. Autonomia robota była określona poprzez jego możliwości percepcji rzeczywistości, umiejętność planowania trasy pomiędzy kolejnymi punktami, a nawet zdolność do reorganizacji obiektów w przestrzeni, w której się poruszał. Rozwój badań nad SI nie hamował, co potwierdza system ekspercki o nazwie MYCIN [3], stworzony w 1973 roku przez Edwarda Shortliffe. Zadaniem tego systemu była pomoc lekarzowi w identyfikacji rodzaju bakterii wywołujących chorobę krwi, a także dobraniu określonego antybiotyku. System składał się z około 600 reguł, operujących na

[1] https://www.europarl.europa.eu/topics/pl/article/20200827STO85804/sztuczna-inteligencja-co-to-jest-i-jakie-ma-zastosowania
[2] https://home.dartmouth.edu/
[3] https://www.sri.com/
[4] http://ai.stanford.edu/users/nilsson/OnlinePubs-Nils/shakey-the-robot.pdf

odpowiedziach z około 50-60 pytań zadanych lekarzowi opisujących stan pacjenta. Jego największymi zaletami, poza krótkim czasem odpowiedzi, była możliwość działania w sytuacji braku części informacji oraz możliwość uzasadnienia udzielonej odpowiedzi. W tym samym czasie Seppo Linnainmaa zaproponował efektywne wykorzystanie reguły łańcuchowej Leibniza, tworząc szeroko znaną metodę wstecznej propagacji[5].

W kolejnych latach, dzięki rozwojowi technologii umożliwiającej prowadzenie zdecydowanie bardziej złożonych obliczeń, można było obserwować wzmożony rozwój algorytmów sztucznej inteligencji i ich zastosowań. W latach 90-tych firma IBM® zaprezentowała komputer o nazwie Deep Blue[6], który był w stanie wygrać partię szachów z ówczesnym mistrzem świata Garrym Kasparovem. Jednostka liczyła 32 procesory zdolne wykonać niespełna 1.4 mld operacji zmiennoprzecinkowych na sekundę (ang. Floating Point Operations Per Second – flops), co przekładało się na ewaluację około 200 mln ustawień figur na planszy. Innymi badaniami, które prekurowały późniejszą ścieżkę rozwoju dziedziny, były te dotyczące autonomicznych pojazdów do transportu ludzi. Pierwszym systemem był ALVINN [4] zaproponowany w 1988 roku przez Deana A. Pomerleau na konferencji Advances in Neural Information Processing Systems. Ważnym punktem na osi czasu można również określić powstanie algorytmu NETtalk[7], autorstwa Lindy Watson [5]. Jego zadaniem było przetwarzanie tekstu pisanego na mowę. Stanowiło to kolejny krok w stronę usprawnienia interakcji pomiędzy człowiekiem a maszyną. Ostatnim przywołanym przykładem niech będzie sieć LeNet-5 [6] służąca do rozpoznawania cyfr na czekach bankowych. Yan LeCun, który opracował tą architekturę, zapoczątkował tym samym dynamiczny rozwój sieci splotowych (ang. Convolutional Neural Network – CNN) stosowanych dziś szeroko.

Lata 2000-2020 można określić jako następny etap rozwoju Sztucznej Inteligencji jako dziedziny. Podobnie jak wcześniej rozwój technologii, w tym przypadku ogromnych baz danych (ang. Big Data), umożliwił utworzenie wielu coraz lepiej działających rozwiązań. Wśród osiągnięć tego etapu można wyróżnić:

1) Rozwój projektów autonomicznych robotów i pojazdów – w roku 2000 w muzeum Miraikan[8] firma Honda prezentuje humanoidalnego robota o nazwie

[5] Linnainmaa, S. (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Doctoral dissertation, Master's Thesis (in Finnish), Univ. Helsinki).
[6] https://www.ibm.com/history/deep-blue
[7] http://www.ntalk.de/Nettalk/en/
[8] https://www.miraikan.jst.go.jp/en/

ASIMO[9] (ang. Advanced Step in Innovative Mobility), który miał być krokiem w kierunku utworzenia robotycznego asystenta życia codziennego. Robot miał możliwość poruszania się w przód, do tyłu oraz na boki, a także potrafił planować trasę w inteligentny sposób. Jego atutem było również posiadanie wyświetlacza imitującego ludzką twarz oraz wydawanie różnych dźwięków, dzięki czemu dość łatwo było mu nawiązać kontakt z człowiekiem. Nacisk na analizę języka naturalnego oraz zrozumienie kontekstu wypowiedzi położyła firma IBM®, której komputer o nazwie Watson[10] pokonał w 2011 roku w teleturnieju Jeopardy![11] wszystkich konkurentów. Trzecim przykładem jest autonomiczny samochód o nazwie Stanley[12] skonstruowany w 2005 przez zespół Stanford Racing Team. Wygrał on ówczesną edycję zawodów The Grand Challenge[13] znanych też jako DARPA Challenge, przejeżdżając jako pierwszy aż 132 mile kalifornijskiej pustyni w czasie 6 godzin i 54 minut. Całość trasy została pokonana bez żadnej pomocy człowieka, co przenosiło odpowiedzialność za manewry oraz nawigację na robota.

2) Dynamiczny rozwój modeli klasyfikatorów opartych na głębokich sieciach konwolucyjnych – został on rozpoczęty w 2012 roku na konferencji Advances in Neural Information Processing Systems, gdzie Alex Krizhevsky, Ilya Sutskever oraz Geoffrey E. Hinton zaproponowali architekturę sieci o nazwie AlexNet [7]. Jej zadaniem było klasyfikowanie około 1.3 mln zdjęć wysokiej rozdzielczości ze zbioru LSVRC-2010 ImageNet[14] [8]. Docelowa liczba klas, które miały być rozróżniane, wynosiła 1000. Zaproponowana sieć neuronowa liczyła łącznie 500 tys. neuronów oraz około 60 mln współczynników poddawanych optymalizacji. Była ona rozszerzeniem wcześniej wspomnianej architektury LeNet-5. Różnicę stanowiła zarówno zmiana funkcji jądra warstwy puli ze średniej na maksimum, jak również dołożenie trzech następujących po sobie warstw konwolucyjnych. Dwa lata później w 2014 roku zaproponowana została sieć GoogleNet [9] oparta o tak zwane jednostki Inception module [10]. Ich wyróżnikiem jest jednoczesne uzyskanie informacji z obrazu przy pomocy warstw konwolucyjnych o różnym

[9] https://global.honda/en/robotics/asimo/
[10] https://www.ibm.com/watson
[11] https://www.jeopardy.com/jeoparday
[12] https://cs.stanford.edu/group/roadrunner/stanley.html
[13] https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles
[14] https://www.image-net.org/

rozmiarze filtra, które w następnym kroku są łączone jako kolejne kanały obrazu wyjściowego. W tym samym roku Karen Simonyan oraz Andrew Zisserman wprowadzają architekturę sieci o nazwie VGGNet [11]. Jej założenia polegają na wielokrotnym powtórzeniu grup warstw, z których każda składa z dwóch lub trzech warstw konwolucyjnych o rozmiarze filtra 3x3, a na końcu umieszczona jest warstwa puli o funkcji jądra maksimum. Warstwy konwolucyjne każdej kolejnej z grup zwracają coraz głębsze wyjście, zaczynając od głębokości 64 a kończąc na głębokości 512. Po pięciu takich grupach występują dwie warstwy w pełni połączone o dużej liczbie neuronów, a cała sieć kończy się warstwą Softmax. Ostatnim przykładem w tej grupie jest architektura o nazwie ResNet [12] przedstawiona w 2015 roku na konferencji Computer Vision and Pattern Recognition przez Kaiming He, Xiangyu Zhang, Shaoqing Ren oraz Jian Sun. Jest to architektura oparta na rezyduach, które są zdefiniowane jako grupy dwóch lub trzech warstw konwolucyjnych o odpowiednio dobranych parametrach. Ważnym aspektem działania tej sieci jest też dodawanie wejścia danego rezyduum do jego wyjścia. Na przestrzeni lat powstały kolejne rozszerzenia oraz modyfikacje tej architektury mianowicie: ResNet-18, ResNet-34, ResNet-50, ResNet-101 oraz ResNet-152. Ich nazwy wskazują zwiększającą się sumaryczną liczbę wszystkich warstw. Dodatkowo kolejne architektury mają odpowiednio zwiększone liczby filtrów w kolejnych warstwach. Wymienione architektury w oryginale lub z pewnymi modyfikacjami są szeroko stosowane we wszelkich zadaniach związanych z przetwarzaniem obrazów, [13] takich jak: klasyfikowanie obrazów retinopatii cukrzycowej [14], klasyfikowanie zapisów EEG pod kątem choroby epileptycznej [15], rozpoznawanie chorób kur [16], monitorowanie jakości produktów żywnościowych [17], rozpoznawanie słów arabskich pisanych odręcznie [18], kontrola jakości działania linii produkcyjnej [19], autonomiczne sterowanie pojazdami [20] oraz automatyczne wykrywanie przeszkód na drodze [21], rozpoznawanie uszu [22], wykrywanie i lokalizowanie obiektów na zdjęciach [23][24], czy nawet wykrywanie gestów [25].

3) Wprowadzenie asystentów głosowych – oprogramowanie rozpoznające mowę naturalną w języku angielskim oraz wykonujące odpowiednie zadania przypisane do konkretnych komend, które z czasem było rozwijane, by analizować również kontekst wypowiedzi i wyświetlać odpowiednie informacje zgodnie

z wprowadzonym głosowo zapytaniem. Najbardziej popularnymi rozwiązaniami są: Siri[15] zaproponowana przez firmę Apple®, GoogleNow[16] zwany też Google Assistant zaprojektowany przez firmę Google®, a także Alexa[17] oferowana przez firmę Amazon®.

4) Generatywne Sieci Adwersaryjne (ang. Generative Adversarial Networks [26] - GANs) – zostały zaproponowane przez Iana Goodfellow'a oraz jego współpracowników w 2014 roku. Główna zasada działania opiera się na dwóch konkurujących ze sobą sieciach – jednej zwanej generatorem, odpowiedzialnej za tworzenie próbek w jak największym stopniu podobnych do danych rzeczywistych oraz drugiej, zwanej dyskryminatorem, której zadanie polega na rozpoznaniu, czy przedstawione dane są rzeczywiste, czy też zostały spreparowane. Siła tego rozwiązania polega na tworzeniu dwóch rozwiązań jednocześnie, przy czym postęp jednej z sieci napędza rozwój drugiej.

5) Głębokie uczenie się ze wzmocnieniem (ang. Deep Reinforcement Learning - DRL) – algorytm, który został rozpropagowany szczególnie dzięki pracy zespołu Google DeepMind. W 2015 roku opublikował on pracę, w której wykorzystał wspomniany algorytm do nauki podejmowania decyzji w trakcie rozgrywki dla 49 gier platformy Atari [27]. Wcześniej podejmowali oni już podobne próby [28]. Uczenie się ze wzmocnieniem (ang. Reinforcement Learning - RL) jest oparte na paradygmacie zakładającym wykorzystanie do nauki niegotowych odpowiedzi, których powinien nauczyć się algorytm, lecz raczej sygnału wartościującego odpowiedzi algorytmu. Dzięki takiemu podejściu jest on w stanie nauczyć się takiego sposobu podejmowania decyzji, który będzie lepszy od tego zaproponowanego przez eksperta. Sam paradygmat zakłada istnienie agenta działającego w pewnym środowisku, mogącego podejmować interakcje z tym środowiskiem, otrzymując przy tym pewne informacje zwane sygnałem wzmocnienia. Istota algorytmu głębokiego uczenia się ze wzmocnieniem polega na wykorzystaniu głębokiej sieci konwolucyjnej jako funkcji analizującej aktualny stan środowiska, w którym aktualnie znajduje się agent. Podejście to pozwoliło autorom na utworzenie systemu osiągającego poziom rozgrywki gracza ludzkiego lub niekiedy przewyższający go. Z biegiem

[15] https://www.apple.com/siri/
[16] https://assistant.google.com/
[17] https://alexa.com/

czasu algorytm z sukcesem został również zastosowany w trudniejszym zadaniu jakim jest granie w grę typu FPS (ang. First Player Shooter) [29]. Samo uczenie się ze wzmocnieniem i jego głębokie odmiany [30] zostały z powodzeniem zastosowane do wielu zadań, [31][32] jak np.: usuwanie szumów oraz napisów z obrazów przy pomocy algorytmu PixelRL [33], poprawianie kolorów obrazu [34], aktywne wykrywanie obiektów zwykłych [35] oraz połączonych [36], śledzenie obiektów [37][38], czy rozpoznawanie zachowań [39].

6) Algorytmy grupy Alpha – zostały zaprojektowane przez zespół DeepMind, a pierwszym z nich jest AlphaGo[18] opublikowany w 2015 roku, któremu udało się pokonać mistrza świata w grze Go. Było to duże osiągnięcie, szczególnie z uwagi na fakt ogromnej złożoności gry mimo jej pozornej prostoty – liczba możliwych układów na planszy to 10^{170}. Początkowo autorzy uczyli model prowadzenia rozgrywki w sposób podobny do ludzkich graczy [40]. Dalsza nauka modelu była prowadzona poprzez rywalizację algorytmu z samym sobą [41]. Rozwiązanie opierało się na wykorzystaniu potencjału algorytmu głębokiego uczenia się ze wzmocnieniem oraz zaawansowanych algorytmów przeszukiwania. Dwa lata później w 2017 roku zespół ten opublikował rozwiązanie AlphaZero.[19], Było ono jeszcze bardziej skuteczne w grze Go [42], a ponadto potrafiło wygrywać także w grze w szachy. Pokonało ono wówczas mistrza z roku 2016, czyli algorytm Stockfish[20] oraz Shogi, czyli japońską wersję szachów, w której pokonało dotychczas najlepszy algorytm o nazwie Elmo. Dodatkowo ważnym aspektem jest również odkrycie nowych możliwości prowadzenia rozgrywki, jak np. utrzymywanie króla w centrum planszy (gra w Shogi), co początkowo wzbudzało zdziwienie profesjonalnych graczy, ale przy utrzymaniu odpowiedniej strategii pozwoliło na zwiększenie szans na zwycięstwo. Trzeci przykład nie odnosi się już do problemu opracowania strategii wygrywającej w grze planszowej, lecz zgoła innego tematu, jakim jest przewidywanie struktury cząsteczek białkowych. System nosi nazwę AlphaFold[21]. Jest to niezwykłe osiągnięcie ułatwiające, między innymi, opracowywanie nowych leków.

[18] https://deepmind.google/technologies/alphago/
[19] https://deepmind.google/discover/blog/alphazero-shedding-new-light-on-chess-shogi-and-go/
[20] https://stockfishchess.org/
[21] https://deepmind.google/discover/blog/putting-the-power-of-alphafold-into-the-worlds-hands/

Omówione rozwiązania to tylko niewielki wycinek osiągnięć dziedziny, jaką jest Sztuczna Inteligencja oraz jej praktycznych zastosowań. Niniejsza praca jest wynikiem inspiracji tymi osiągnięciami i chęcią rozwoju istniejących metod. Cel ten wydaje się być realny, biorąc pod uwagę ogromną liczbę potencjalnych możliwości zastosowania istniejących oraz nowych metod.

1.2. Hipoteza badawcza oraz cele pracy

Głównym celem pracy jest przedstawienie możliwości aplikacji metod sztucznej inteligencji ze szczególnym uwzględnieniem algorytmu uczenia się ze wzmocnieniem. Opierając się na tym założeniu sformułowano następującą hipotezę badawczą:

> *Możliwa jest aplikacja różnych metod sztucznej inteligencji, a w szczególności algorytmu uczenia się ze wzmocnieniem, zarówno do zadań przetwarzania obrazu, jak i do zadań sterowania, celem uzyskania rezultatów nie gorszych niż przy pomocy innych metod znanych z literatury.*

W celu potwierdzenia postawionej hipotezy sformułowano następujące zadania szczegółowe:

Zadanie 1. Studia literaturowe dotyczące wykorzystania algorytmu uczenia się ze wzmocnieniem do:
- a) rozwiązywania zadań przetwarzania obrazów,
- b) rozwiązywania zadań sterowania.

Zadanie 2. Zebranie niezbędnych danych oraz utworzenie zbioru pozwalającego na trenowanie oraz weryfikowanie poprawności działania badanych metod:
- a) do zadania wykrywania gestów,
- b) do zadania sterowania dynamiką ruchu wrzeciona maszyny CNC.

Zadanie 3. Zaproponowanie metody pozwalającej na:
- a) czasową segmentację ciągłego strumienia gestów,
- b) optymalizację sterowania dynamiką ruchu wrzeciona maszyny CNC z wykorzystaniem logiki rozmytej,
- c) optymalizację sterowania dynamiką ruchu wrzeciona maszyny CNC z wykorzystaniem paradygmatu uczenia się ze wzmocnieniem.

Osiągnięcie celu dysertacji polegało na realizacji wyszczególnionych zadań, które wymagały: pozyskania, przetworzenia i analizy danych, sformułowania odpowiednich pomniejszych celów, wyboru i wykorzystania odpowiednich narzędzi informatycznych, implementacji oraz weryfikacji proponowanych rozwiązań, a także interpretacji wyników oraz sformułowania wniosków.

2. Aplikacje metod sztucznej inteligencji ze szczególnym uwzględnieniem algorytmu uczenia się ze wzmocnieniem

Rozdział ten przedstawia zaproponowane przez autora aplikacje metod sztucznej inteligencji ze szczególnym uwzględnieniem algorytmu uczenia się ze wzmocnieniem. W poszczególnych podrozdziałach opisano kolejno przeprowadzony przegląd powiązanej literatury, która stanowiła inspirację własnych pomysłów autora. Dalej przedstawiono propozycję autorskiej metody czasowej segmentacji strumienia gestów w postaci wideo. Poruszono również tematykę optymalizacji dynamiki pracy maszyny CNC z wykorzystaniem zarówno systemu opartego na logice rozmytej, jak i systemu bazującego na paradygmacie uczenia się ze wzmocnieniem.

2.1. Uczenie się ze wzmocnieniem w zadaniach przetwarzania obrazu

Zagadnienie przetwarzania obrazów jest aktualnie niemal nieodzownym elementem ogromnej części systemów komputerowych. Jak już wspomniano, informacje zapisane w obrazie oraz umiejętność ich odczytania, a także wnioskowania na ich podstawie pozwalają na tworzenie rozwiązań ułatwiających życie codzienne. Na przykład wykonanie zdjęcia w aplikacji po wykryciu uśmiechniętej twarzy bez potrzeby korzystania z przycisku na ekranie. Bezpośrednio wpływają one także na sferę biznesową, gdzie przykładem może być uwierzytelnianie przy pomocy twarzy. Natomiast w sferze gospodarczo-produkcyjnej najbardziej popularnym zastosowaniem jest szeroko rozumiany monitoring hal produkcyjnych oraz magazynów obejmujący zarówno zwykły dozór terenu zakładu, jak również samą analizę jakości działania linii produkcyjnej, czy jakości gotowych produktów. Realizując Zadanie 1.a) przeprowadzono analizę literatury pod kątem zastosowania algorytmu uczenia się ze wzmocnieniem do realizacji szeroko rozumianych zadań przetwarzania obrazu. Wyróżniono trzy główne kategorie istniejących rozwiązań, z których każda zawiera kilka węższych podkategorii realizowanych przez wyspecjalizowane algorytmy służące konkretnemu zadaniu. Podział przedstawia się następująco:

1) Modyfikacja obrazu – rozumiana jako ingerencja w konkretne piksele obrazu. Kategoria zawiera algorytmy realizujące zadania takie jak:
    a) redukcja szumów:
        i. poprzez budowanie łańcucha różnego rodzaju przekształceń obrazu [43],
        ii. z wykorzystaniem algorytmu PixelRL, gdzie modyfikacja konkretnych pikseli obrazu jest realizowana przez przypisanych do nich agentów [33],
    b) korekcja kolorów:
        i. uwzględniająca kolejne modyfikacje całego obrazu przez agenta dysponującego analizą kontekstu dzięki sieci głębokiej VGG16 oraz informacją o kolorach obrazu na podstawie histogramu [34]
        ii. oparta o ustalenie wartości parametrów wejściowych do procesu przetwarzania obrazu w programach Adobe Lightroom® oraz Adobe Photoshop® [44],
    c) odzyskiwanie bloków:
        i. rozumiane jako rekonstrukcja obrazów pochodzących z tomografii komputerowej z wykorzystaniem algorytmu Alternating Direction Method of Multipliers [45] lub algorytmu Magnetic Resonance Imaging [46],
        ii. rozumiane jako usuwanie napisów zasłaniających obraz z wykorzystaniem wspomnianej wcześniej metody PixelRL [33],
    d) wyostrzanie obrazów oparte na metodzie PixelRL wykorzystującej możliwości algorytmów: Enhanced Deep Super-Resolution network, Enhanced Super-Resolution Generative Adversarial Network oraz Progressive Perception-Oriented Network [47],
    e) scalanie obrazów w programie Adobe Photoshop®, gdzie zadaniem zaproponowanego przez autorów algorytmu był odpowiedni dobór parametrów procesu blendowania [48].

2) Detekcja oraz śledzenie obiektów – algorytmy realizujące jedno z zadań:
    a) detekcja obiektów:
        i. przy założeniu podziału obrazu na określoną liczbę sektorów [49],
        ii. poprzez modyfikację rozmiaru oraz położenia ramki okalającej obiekt [35],

iii. z przyspieszeniem wyszukiwania poprzez odpowiedni dobór kolejnych punktów skupienia na obrazie [50],
iv. przy pomocy dwóch współpracujących agentów wykrywających obiekty połączone [36];

b) śledzenie obiektów:
i. w oparciu o procesy decyzyjne markowa [37]
ii. poprzez modyfikację rozmiaru oraz położenia ramki okalającej obiekt do momentu osiągnięcia odpowiedniego poziomu pewności agenta [38]
iii. na podstawie uaktualniania mapy ciepła wskazującej przekonanie agenta o miejscu przebywania obiektu [51].

3) Różne – zadania bardziej złożone:
a) rozpoznawanie gestów [39] i zachowań [52] z wykorzystaniem obrazów pochodzących z kamery szerokokątnej zapewniającej kontekst sceny oraz kamery o wysokiej jakości obrazu analizującej konkretne części postaci,
b) rozpoznawanie twarzy na podstawie sekwencyjnego porównania klatek dwóch materiałów wideo [53],
c) szeroko rozumiane zadania sterowania:
i. utrzymywanie pojazdu w obrębie drogi z wykorzystaniem samoorganizujących się map (ang. Self-Organizing Maps - SOM) w celu analizowania obrazu z kamery czołowej [54],
ii. osiąganie strategii wygrywającej w 49 grach platformy Atari poprzez odpowiednią imitację ruchów drążka sterującego [27],
iii. sterowanie czasem otwarcia przesłony oraz parametrem ISO poprzez zarządzanie wartością ekspozycji w celu uzyskania żądanego przez użytkownika efektu wizualnego [55],
d) ekstrakcja istotnych klatek filmu w celu przewijania w przód materiału wideo [56],
e) rekonstrukcja sceny 3D z wykorzystaniem sieci MVCNN, SSCNet oraz 2DCNN [57].

Przeprowadzona analiza przytoczonych prac pozwoliła na zapoznanie się z różnymi podejściami do zastosowania paradygmatu uczenia się ze wzmocnieniem. Szczególną inspiracją do tworzenia własnych rozwiązań stały się prace [33][38][27]. Na ich

przykładzie zauważono, że wykorzystując algorytm uczenia się ze wzmocnieniem, można zdecydowanie podnieść poziom interpretowalności działań podejmowanych przez zaproponowany system oraz ułatwić proces ich wizualizacji. Algorytm ten jest również względnie prosty w implementacji, co sprawiło, że doczekał się wielu realizacji w różnych językach programowania. W pewnych warunkach proces jego nauki może przebiegać zdecydowanie bardziej efektywnie, co wyraża się poprzez szybsze osiągnięcie zadowalającego rozwiązania. Dodatkowym atutem jest również możliwość jego zastosowania w sytuacji braku klasycznych danych uczących w postaci par (przykład, pożądana odpowiedź) – do nauki wystarczy mu sygnał wartościujący podjętą akcję, który może być nawet odsunięty w czasie. Analiza wykazała również rosnące zainteresowanie paradygmatem uczenia się ze wzmocnieniem, co można zaobserwować na Rys. 2. zaczerpniętym z [A-1]. Nie do pominięcia jest także interdyscyplinarny charakter analizowanych prac, które poruszają aspekty z zakresu: fotografii, grafiki komputerowej, bezpieczeństwa, socjologii, psychologii czy robotyki.

Rys. 2 Zestawienie przeanalizowanych prac uwzględniające datę publikacji [A-1]

Udział własny autora niniejszej rozprawy doktorskiej w przygotowaniu pracy [A-1] wynosił 100% i polegał na: analizie badanych prac, opracowaniu wyników oraz redakcji treści pracy. Wprowadzona systematyka pozwoliła na zidentyfikowanie potencjalnych obszarów rozwoju dziedziny, natomiast praca w kole naukowym Interakcji Człowiek-Komputer GEST oraz chęć odpowiedzi na zapotrzebowanie otoczenia społecznego

zorientowały dalsze prace w kierunku propozycji rozwiązania realizującego wykrywanie gestów języka migowego.

2.2. Czasowa segmentacja strumienia gestów

Od ponad 25 lat zadanie automatycznego interpretowania ludzkich gestów w celu rozwijania integracji ludzi z różnymi systemami komputerowymi stanowiło wyzwanie dla wielu badaczy. Wyróżnić można następujący podział tematyczny badań:

1) Zliczanie palców dłoni – głównym zadaniem jest poprawne zidentyfikowanie liczby widocznych (wyciągniętych) palców dłoni w celu podjęcia na tej podstawie odpowiedniej akcji. Przykładem może być praca [58], której autorzy chcieli uzyskać informację o geście wykonywanym przez osobę prowadzącą samochód, w celu wykonania przypisanej do danego gestu akcji.

2) Rozpoznawanie statycznych gestów dłoni – zadanie to polega na rozpoznaniu pokazywanego gestu na podstawie zdjęcia bądź wybranej klatki filmu wideo. Wymienić można tutaj dwa przykłady. Pierwszym z nich jest próba utworzenia systemu analizy gestu w czasie rzeczywistym [59], drugim natomiast jest wykorzystanie ukrytych modeli Markov'a (ang. Hidden Markov Model - HMM) [60].

3) Rozpoznawanie dynamicznych gestów dłoni – zadanie polegające na rozpoznaniu pokazywanego gestu na podstawie filmu wideo. Można tutaj wymienić szereg istotnych prac prezentujących różne podejścia do tego zadania: system działający w czasie rzeczywistym [61], system sterowania grą komputerową przy pomocy gestów, system wykorzystujący splotową rekurencyjną sieć neuronową (ang. Convolutional Recurrent Neural Network - CRNN) w połączeniu z jednostkami Long Short-Term Memory (LSTM) [62], system oparty o kamery typu RGB-D oraz algorytm k-najbliższych sąsiadów (ang. k-Nearest Neighbours – k-NN) w połączeniu z algorytmem Dynamic Time Warping (DTW) i HMM [63], a także prace wykorzystujące algorytm uczenia się ze wzmocnieniem [64] oraz jego modyfikację w postaci głębokiego uczenia się ze wzmocnieniem. (ang. Deep Reinforcement Learning - DRL) [65].

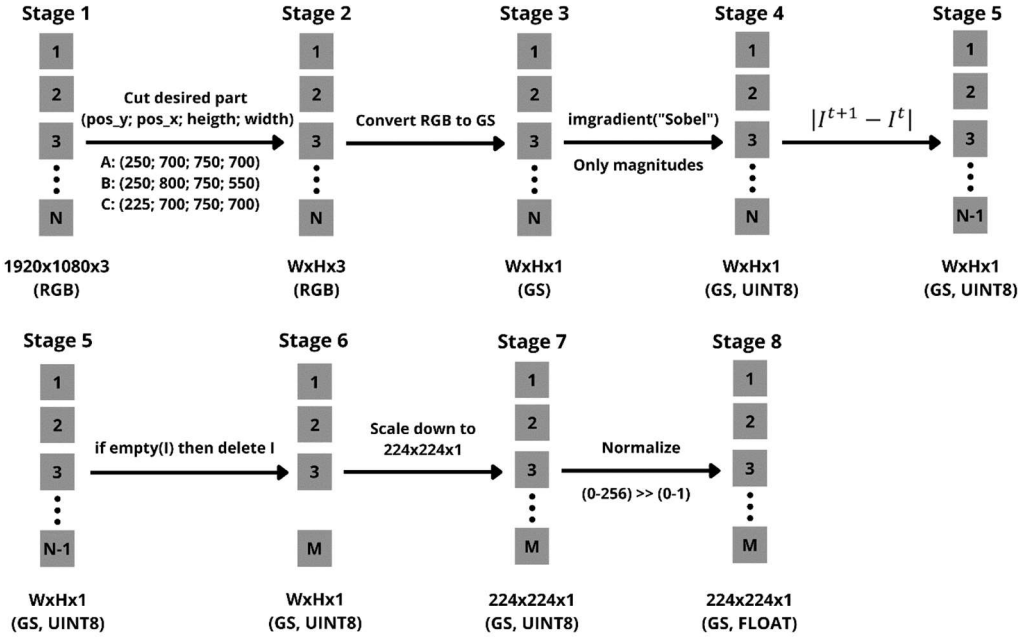
Na przestrzeni kolejnych lat rozwoju dziedziny szeroko pojętego rozpoznawania gestów powstały prace przeglądowe analizujące konkretne jej aspekty. Ogólną analizę problemu

przedstawili autorzy [66]. Kolejno w pracach [67] (2007), [68] oraz [69] (2012) zostały przeanalizowane metody rozpoznawania statycznych gestów dłoni. Dużo bardziej obszerną analizę, bo nie kilkunastu, lecz niespełna 90 prac, zaprezentowano rok później w [70]. W następnych latach powstały kolejne prace analizujące nowo powstałe algorytmy, jak np. [71]. Na szczególną uwagę zasługują też prace [72] oraz [73], które oprócz analizy samych algorytmów do rozpoznawania statycznych gestów dłoni zawierają także podsumowanie informacji o dostępnych bazach danych do nauki tych algorytmów. W 2021 roku opublikowano natomiast kompleksową analizę [74] zarówno wspomnianych algorytmów, jak i baz danych. Jej dodatkowym atutem, oprócz analizy ponad 200 prac, jest również usystematyzowanie zagadnień związanych z procesem rozpoznawania gestów statycznych oraz dynamicznych. Problematyka rozpoznawania dynamicznych gestów dłoni również była szeroko analizowana. Podczas, gdy jedni autorzy skupiali się na analizie algorytmów rozpoznawania gestów języka migowego w kontekście aplikacji mobilnych [75], kolejni rozważali aspekt złożoności obliczeniowej [76], a inni skupiali się na analizie metod opartych na uczeniu sieci głębokich [77]. Podobnie dużym zainteresowaniem cieszyła i nadal cieszy się tematyka tworzenia odpowiednich i coraz bardziej bogatych w różnego rodzaju gesty baz danych [78].

Przeprowadzenie analizy wspomnianych prac przeglądowych pozwoliło na dostrzeżenie pewnych zależności, które zostały opisane w publikacji [A-5]. Zidentyfikowano dwa główne podejścia do zadania rozpoznawania gestów dynamicznych. Pierwsze z nich skupia się na rozpoznawania wyodrębnionych wcześniej gestów (ang. Isolated Sign Language Recognition), natomiast drugie polega na analizie kolejnych klatek filmu oraz jednoczesnym wykryciu i klasyfikacji danego gestu (ang. Continuous Sign Language Recognition). Metody należące do pierwszej grupy osiągają zdecydowanie wyższe wyniki poprawności klasyfikacji od tych z grupy drugiej, natomiast należy w tym miejscu stwierdzić fakt, że w codziennym życiu osoby posługujące się językiem migowym nie wstrzymują ruchów po każdym znaku, lecz ich ruchy płynnie przechodzą z jednego w drugi. Jest to problem koartykulacji obecny też w zadaniu analizy mowy ludzkiej. Autor niniejszej pracy zauważył też, że wszystkie dostępne bazy danych wydają się w ogóle nie uwzględniać tego zjawiska, co jest widoczne w sposobie tworzenia etykiet kolejnych klatek nagrań w tych bazach. Kierując się chęcią utworzenia połączenia między obiema grupami algorytmów i umożliwienia nauki rozpoznawania gestów na podstawie danych oczyszczonych ze zbędnego w jego przekonaniu szumu informacyjnego

wywołanego przez zjawisko koartykulacji, autor w artykule [A-5] zaproponował metodę zdolną do czasowej segmentacji strumienia wideo.

W pierwszej kolejności należało przygotować dedykowaną pod kątem nowego rozwiązania bazę danych. Do tego celu wykorzystano bazę danych przygotowaną w Katedrze Informatyki i Automatyki Politechniki Rzeszowskiej we współpracy z Podkarpackim Stowarzyszeniem Głuchych rozszerzając ją o odpowiednio przygotowane etykiety określające miejsce występowania gestów (wartość 1) oraz miejsce występowania przejść bądź przerw w pokazywaniu gestów (wartość 0). Nagrania były realizowane przez członków stowarzyszenia pod opieką zawodowych tłumaczy. Do nagrań wykorzystano kamerę Microsoft Kinect Xbox One 2.0 działającą z klatkarzem równym 29.98. W ten sposób uzyskano odpowiednie serie zdjęć RGB-D o rozdzielczości 1920x1080 pikseli zapisanych w formacie JPEG. Wspomniane adnotacje wykonano przy pomocy oprogramowania ELAN[22]. Ostateczna forma bazy danych zawierała nagrania 33 sekwencji gestów (zwanych również wyrażeniami) wykonywanych 5-krotnie przez 3 osoby.



Rys. 3 Schemat przetwarzania wideo przez algorytm GSADGM [A-5]

W kolejnym kroku autor niniejszej pracy na podstawie wykonanej analizy danych zdefiniował szereg czynników mających wpływ na przetwarzanie strumienia wideo

[22] Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands (https://archive.mpi.nl/tla/elan)

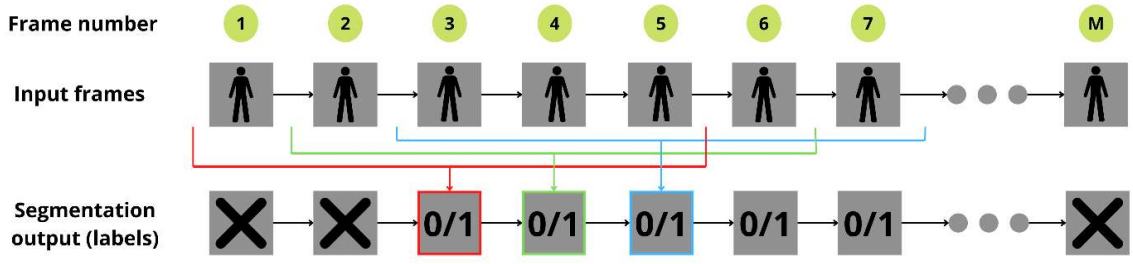
mogących potencjalnie utrudniać zadanie rozpoznawania gestów. Były to: możliwość zbieżności koloru skóry lub włosów z elementami tła bądź garderoby osoby pokazującej gesty, cienie wywołane różnym sposobem oświetlenia, a także sama jasność obrazu. Chcąc zminimalizować potencjalne negatywne efekty wywołane tymi czynnikami, a także dążąc do zmniejszenia wymiarowości przetwarzanych danych, autor niniejszej pracy zaproponował autorski algorytm wstępnego przetwarzania materiału wideo o nazwie Gray Scale Absolute Difference Gradient Method (GSADGM). Kolejne kroki jego działania oraz rozmiar danych w każdym z etapów (Stage 1 – Stage 8) zostały zobrazowane na Rys. 3.

W pierwszym kroku będącym przejście z etapu pierwszego (Stage 1) do etapu drugiego (Stage 2) każda z klatek filmu jest kadrowana w celu odrzucenia zbędnych informacji oraz zapewnienia centralnego ustawienia postaci w kadrze. Później (przejście Stage 2 – Stage 3) klatki te są konwertowane do odcieni szarości. Dzięki takiemu działaniu zyskuje się minimum 3-krotne zmniejszenie wielkości danych bez utraty istotnych informacji. Następnie (przejście Stage 3 – Stage 4) obliczane są gradienty pomiędzy pikselami każdej klatki, a do dalszych obliczeń pozostawia się tylko informację o ich wielkościach. W kolejnym kroku (przejście Stage 4 – Stage 5) obliczana jest wartość bezwzględna z różnicy między kolejnymi klatkami każdego z filmów. Tak uzyskane obrazy są weryfikowane – jeżeli są całkiem czarne (wszystkie piksele są równe 0), to są one usuwane (przejście Stage 5 – Stage 6). Następnym krokiem jest skalowanie obrazów do wymiarów umożliwiających podanie ich do zaprojektowanej sieci głębokiej (przejście Stage 6 – Stage 7). Ostatnim krokiem jest normalizacja wartości pikseli w celu stabilizacji procesu nauki (przejście Stage 7 – Stage 8). Opisany proces pozwala na uzyskanie informacji o ruchu postaci, redukując potencjalne negatywne efekty wywołane przez wspomniane wcześniej czynniki. Na Rys. 4 została przedstawiona przykładowa sekwencja klatek stanowiących dane wejściowe przekazane do zaproponowanego algorytmu. Rys. 5 prezentuje natomiast przykładowy rezultat działania zaproponowanego algorytmu. Dla poprawy czytelności przykładowe dane wyjściowe zostały przedstawione w postaci negatywu.

Rys. 4 Przykładowy strumień danych wejściowych dla algorytmu GSADGM [A-5]

Rys. 5 Przykładowy strumień danych wyjściowych dla algorytmu GSADGM przedstawiony w negatywie dla poprawy czytelności [A-5]

Następnym krokiem w trakcie projektowania rozwiązania było zaproponowanie modelu sieci neuronowej oraz sposobu jej nauki. Autor niniejszej pracy zdecydował się wykorzystać połączenie głębokich sieci konwolucyjnych ze względu na ich możliwości analizy skomplikowanych danych wielowymiarowych oraz algorytm uczenia się ze wzmocnieniem, a dokładnie jego odmiany zwanej Double Deep Q-Learning (DDQN) [79], która jest szeroko stosowana w literaturze dla zapewnienia poprawy zbieżności i stabilności procesu nauki. W pracy [A-5] przebadane zostało siedem autorskich architektur, z których część była wzorowana na modelu ResNet. Autor przyjął, że analiza jednego filmu zawierającego sekwencję gestów będzie określała ramy epizodu. Agent w każdym kroku epizodu korespondującym z klatką filmu może podejmować jedną z dwóch akcji będącą decyzją, czy w danej klatce trwa gest (etykieta 1), czy nie (etykieta 0). Jako obserwacje pochodzące ze środowiska posiada on zestaw przygotowanych wcześniej klatek filmu. Warto zauważyć, że liczba kroków w epizodzie jest uzależniona od długości filmu i odpowiednio pomniejszona w zależności od wielkości zestawu klatek, który może obserwować agent. Diagram przedstawiający symbolicznie opisaną sytuację został zamieszczony na Rys. 6.

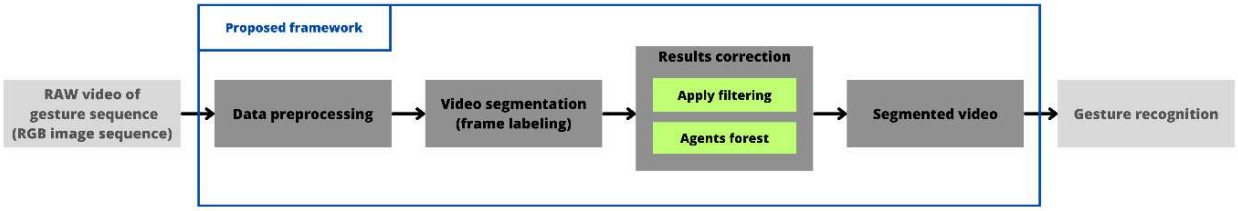


Rys. 6 Schemat wiedzy i działań agenta [A-5]

Jako sygnał wartościujący akcje podejmowane przez agenta zdecydowano się zastosować następującą zależność: „Jeżeli agent w danej klatce wybrał etykietę zgodną z informacjami zawartymi w bazie, wtedy otrzymuje wzmocnienie równe 0.0, natomiast w przypadku błędnej decyzji otrzymuje on karę w postaci wzmocnienia o wartości -0.2”. W ramach końcowej obróbki wyznaczonych przez agenta etykiet kierując się analizą

uzyskanych wyników zdecydowano się przebadać wpływ trzech zaproponowanych filtrów oraz metodę głosowania opartą na algorytmie lasu drzew. Zastosowanie obu metod jednocześnie zapewniło zdecydowanie lepsze wyniki pracy zaproponowanego rozwiązania.

Autorzy [A-5] zdecydowali się zaproponować rozbudowane testy oraz analizę proponowanego przez siebie rozwiązania. Oprócz klasycznych testów w formule LOSO (ang. Leave One Subject Out) polegającej na traktowaniu jako zbioru testowego nagrań konkretnej osoby, przygotowano również protokoły sprawdzające oparte na wyborze 1, 2 lub 3 osób we wszystkich kombinacjach oraz na 5-krotnej walidacji krzyżowej względem powtórzeń nagrań wykonanych przez każdą z osób. W artykule przedstawiono również omówienie wpływu omówionych parametrów oraz przeanalizowano działanie wyuczonych filtrów sieci. Zaproponowane rozwiązanie osiąga wskaźnik poprawności na poziomie 0.76 podczas testów zgodnych z protokołem LOSO oraz wartość 0.89 podczas testów pojedynczych osób.



Rys. 7 Zakres działań zaproponowanego algorytmu [A-5]

Udział własny autora niniejszej rozprawy doktorskiej w przygotowaniu pracy [A-5] wynosił 90% i polegał na: wspólnej konceptualizacji proponowanego rozwiązania, opracowaniu metodyki, przygotowaniu danych, implementacji niezbędnych do nauki oraz testów kodów aplikacji, opracowaniu wyników oraz współredakcji treści pracy. Praca zbliża autora do osiągnięcia celu, jakim jest utworzenie aplikacji mobilnej pozwalającej w czasie rzeczywistym tłumaczyć mowę Polskiego Języka Migowego (PJM), co będzie miało pozytywny wpływ na komfort codziennej komunikacji międzyludzkiej. Zakres rozwiązania zaproponowanego w artykule [A-5] został schematycznie przedstawiony na Rys. 7. Aktualnie trwają badania nowych propozycji architektur oraz intensywne prace w kierunku rozwinięcia bazy danych o nagrania dla większej liczby osób pokazujących gesty, co będzie miało pozytywny wpływ na jakość działania rozwiązań proponowanych w przyszłości, a także podniesie możliwości weryfikacji istniejących rozwiązań. Warto również zauważyć, że opracowana baza

danych została udostępniona publicznie w trybie „na żądanie”, co niewątpliwie ułatwia innym badaczom weryfikację własnych propozycji rozwiązań. Omówiona publikacja realizuje nie tylko Zadanie 2.a) lecz także Zadanie 3.a).

2.3. Zadania sterowania

Drugą tematyką podejmowaną w niniejszej rozprawie doktorskiej jest zastosowanie metod sztucznej inteligencji do rozwiązywania zadań sterowania. Cieszą się one szerokim zainteresowaniem ze względu na swoją wartość wyrażoną poprzez częściową bądź pełną automatyzację różnego rodzaju procesów. Pozwala to zredukować ryzyko błędu ludzkiego operatora w przypadku chociażby kierowania wszelkiego rodzaju pojazdami oraz pozwala mu skupić się na innych bardziej wymagających zadaniach. Realizując Zadanie 1.b) przeprowadzono analizę literatury pod kątem zastosowania algorytmu uczenia się ze wzmocnieniem, do realizacji szeroko rozumianych zadań sterowania pojazdami autonomicznymi. W artykule [A-2] autor niniejszej rozprawy doktorskiej usystematyzował oraz scharakteryzował szereg aspektów związanych z zastosowaniem algorytmu uczenia się ze wzmocnieniem w zadaniach sterowania pojazdami. Przygotowane opracowanie było wynikiem analizy blisko 50 prac związanych z omawianą tematyką:

1) **Rodzaj zadania do realizacji** – pierwsza wprowadzona systematyka dotyczyła rodzaju zadania postawionego przed proponowanym przez badaczy systemem. W pracy wyróżniono poniższe kategorie umieszczając odpowiednie odwołania do konkretnych analizowanych prac:
    a) parkowanie,
    b) utrzymywanie docelowej prędkości pojazdu,
    c) utrzymywanie bezpiecznej odległości od poprzedzającego pojazdu,
    d) utrzymywanie się w obrębie danego pasa ruchu,
    e) włączanie się do ruchu,
    f) wyprzedzanie,
    g) przejazd przez skrzyżowanie,
    h) kierowanie pojazdem w przypadku występowania utrudnień z obserwacją otoczenia,
    i) reagowanie na nagłe i niespodziewane zdarzenia drogowe,
    j) przejazd trasy w możliwie najkrótszym czasie,
    k) zarządzanie zasobami energetycznymi pojazdu,

l) zarządzanie przepływem informacji,
m) reagowanie na rozpoznane znaki drogowe,
n) zarządzanie ruchem pojazdów (przepustowość skrzyżowania lub przepustowość drogi).

Największym zainteresowaniem wśród badaczy cieszą się zadania: utrzymywanie się w obrębie danego pasa ruchu (23 prace), wyprzedzanie (13 prac), przejazd trasy w możliwie najkrótszym czasie (12 prac), przejazd przez skrzyżowanie (6 prac).

2) **Symulator ruchu** – druga systematyka dotyczyła różnego rodzaju środowisk, a w szczególności symulatorów wykorzystanych przez badaczy w analizowanych pracach. W artykule [A-2] autor niniejszej rozprawy doktorskiej omawia następujące symulatory: SUMO, CARLO, CARLA, TORCS, WRC 6, GTS, Microsoft AirSim, Open DS-CTS 1.0, UnityML Agents. Dodatkowo opisy ubogacono o odwołania do konkretnych analizowanych prac oraz odnośniki do miejsc, z których można pobrać wspomniane symulatory. Warto zauważyć, że ogromna większość prac, z wyjątkiem sześciu, korzystała z symulatorów. Tylko dwie analizowane prace były wykonane przy użyciu rzeczywistych danych, a cztery korzystały z danych rzeczywistych oraz symulowanych jednocześnie.

3) **Rodzaj danych wejściowych** – trzecia systematyka dotykająca tematu związanego z charakterystyką rozwiązania oraz jego implementacją. W trakcie analizy wyróżniono następujące kategorie:

   a) Dane pochodzące z wszelkiego rodzaju czujników (prędkościomierz, LIDAR, RADAR, itp.), które były wykorzystane w różnych kombinacjach.

   b) Dane w postaci obrazów z kamer, które były usytuowane w różnych miejscach:

      i. widok pierwszoosobowy,
      ii. widok z perspektywy trzeciej osoby,
      iii. widok z lotu ptaka.

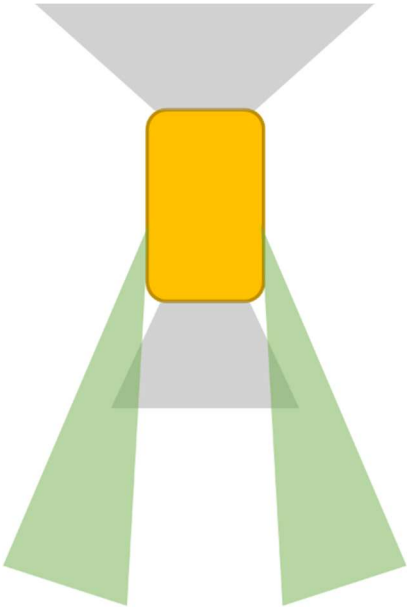
   c) Dane mieszane łączące zarówno informacje z różnych czujników, jak i obrazy z kamer:

i. widok pierwszoosobowy,
ii. widok z lotu ptaka.

Badacze w ponad połowie prac skorzystali tylko i wyłącznie z danych pochodzących z różnych czujników. Kolejnymi w równej mierze chętnie wykorzystywanymi danymi były obrazy z kamer w widoku pierwszej osoby oraz dane mieszane uwzględniające obrazy z kamer również w widoku pierwszoosobowym.

4) **Rodzaj zbioru dostępnych akcji** – ostatnia zaproponowana systematyka dotyczy istotnej, z punktu widzenia algorytmu uczenia się ze wzmocnieniem kwestii, mianowicie definicji typu możliwych do podjęcia przez agenta akcji. W analizowanych pracach wyróżnić można trzy podejścia badaczy wyrażone poprzez zastosowanie zbioru akcji o charakterze dyskretnym, ciągłym lub mieszanym. Tylko dwie prace wykorzystywały zbiór akcji typu mieszanego, natomiast pozostałe w równej części korzystały z typu dyskretnego lub ciągłego.

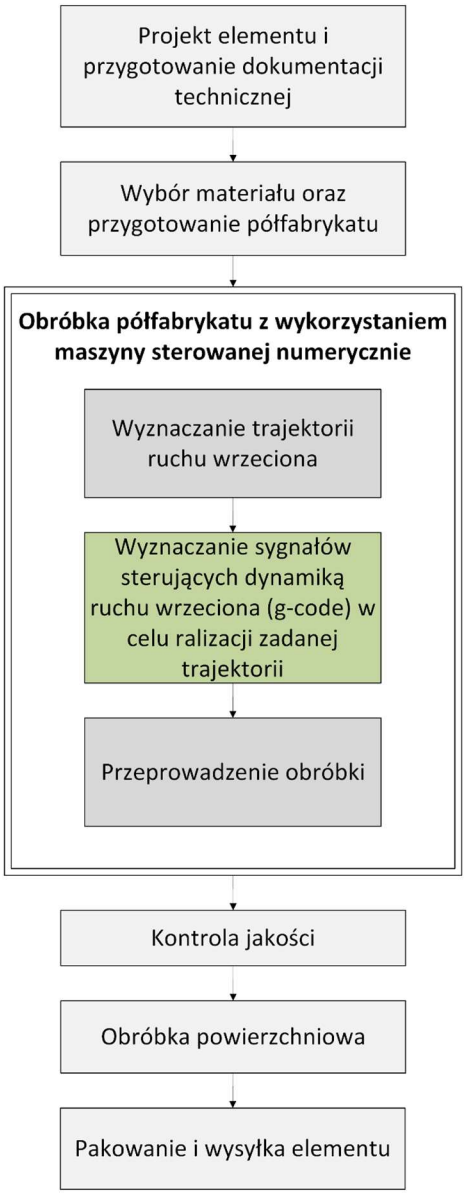
Wszystkie zaproponowane systematyki zostały uzupełnione o odpowiednie tabele zawierające odnośniki do konkretnych analizowanych prac, co ułatwi czytelnikowi dotarcie do interesującego go źródła, a także pokaże opracowania, do których wyników można odnieść swoje badania. Dodatkowo w pracy przedstawiony został również szkic autorskiego rozwiązania związanego z zadaniem wyprzedzania. W zamierzeniu wykorzysta on sygnały z czujników wzbogacone o informacje z szeregu kamer, których pokrycie zostało przedstawione na Rys. 8. Kolorem żółtym oznaczony został rozważany pojazd, kolorem szarym oznaczone zostało pole widzenia kamer przedniej oraz tylnej, których zadaniem jest obserwacja najbliższego otoczenia pojazdu. Natomiast kolorem zielonym zaznaczony został obszar pokrycia kamer tylnych-skośnych, które w zamierzeniu powinny wykrywać pojazdy jadące z tyłu, ale z dużo większą prędkością oraz pozwalać na uniknięcie niebezpiecznej sytuacji. Takie ustawienie kamer zakłada, że wyprzedzanie następuje na drodze posiadającej minimum dwa pasy w każdą stronę. W trakcie analiz zauważono, że żaden z badaczy nie próbował rozważać wyprzedzania, w trakcie którego koniecznym jest zjechanie na przeciwny pas ruchu. Po uzyskaniu pomyślnych rezultatów z badań w aktualnie zaproponowanym układzie (Rys. 8), kolejnym krokiem będzie podjęcie wyzwania realizacji zadania wyprzedzania w trudniejszej z opisanych wersji. Omówiona publikacja realizuje Zadanie 1.b).



Rys. 8 Schemat układu kamer w proponowanym rozwiązaniu [opracowanie własne]

2.4. Systemy logiki rozmytej w zadaniach sterowania maszynami CNC

Na podstawie informacji zdobytych podczas wykonanych przeglądów literatury autor niniejszej rozprawy podjął działania mające na celu wykorzystanie metod sztucznej inteligencji do realizacji zadania optymalizacji pracy maszyny CNC poprzez sterowanie dynamiką ruchu jej wrzeciona. Prace te były realizowane i są nadal rozwijane w zespole, którego członkami są dr inż. Bogdan Kwiatkowski oraz dr hab. inż. Damian Mazur z Politechniki Rzeszowskiej. Wspólnie postanowiono, by opracowywane metody weryfikować, porównując je z algorytmem Reference Points Realization Optimization (RPRO) [80]. Należy również dodać, że algorytm RPRO jest wykorzystywany w przemyśle. Jego zadaniem jest generowanie tzw. g-code'u, czyli ciągu instrukcji reprezentujących zmiany dynamiki wrzeciona w kolejnych dyskretnych krokach czasowych. Algorytm RPRO skupia się na minimalizacji czasu procesu obróbczego z jednoczesnym umożliwieniem określenia żądanej dokładności, która jest rozumiana jako średnia dokładność osiągnięcia każdego z punktów referencyjnych należących do założonej przez operatora trajektorii. Rolę algorytmu RPRO oraz opracowywanych metod w procesie produkcji elementów maszyn przedstawiono na Rys. 9 (kolor zielony).



Rys. 9 Rola projektowanych metod w procesie produkcji elementów maszyn. [opracowanie własne]

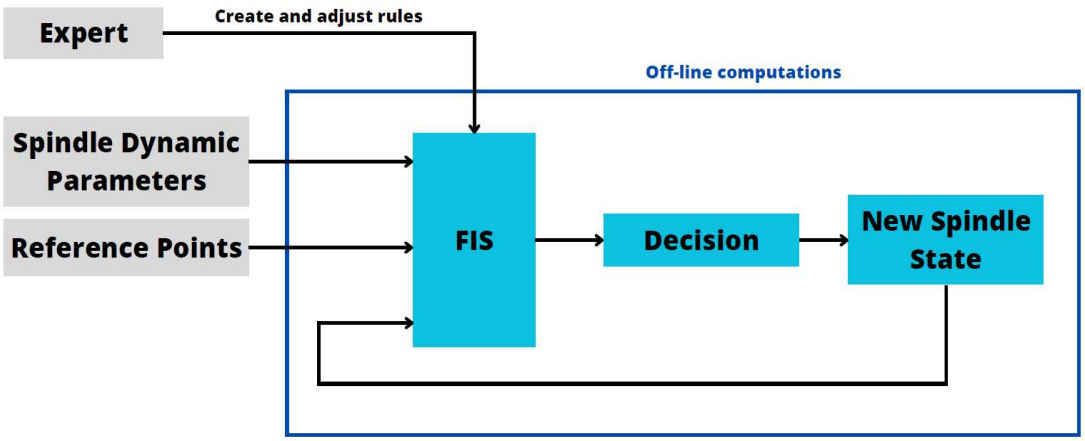
Prace rozpoczęto od zaprojektowania dedykowanej bazy danych pozwalającej na weryfikowanie jakości działania proponowanych później algorytmów. Postanowiono, że baza będzie składała się z szeregu symulowanych procesów obróbczych o różnych trajektoriach wykonywanych przez maszyny o różnych parametrach dynamiki. Przyjęto następujące możliwe wartości parametrów:

- liczba punktów trajektorii $\{15, 50, 100\}$
- gęstość punktów referencyjnych

$$\begin{cases} Duża & - & odległości\ mniejsze\ niż\ 1mm \\ Średnia & - & odległości\ między\ 1mm\ a\ 10mm \\ Mała & - & odległości\ między\ 10mm\ a\ 100mm \end{cases}$$

- maksymalna prędkość wrzeciona wyrażona w $[m/min]$ $\{2.5, 4.0, 6.0, 8.0\}$
- maksymalne przyspieszenie wrzeciona wyrażone w $[m/s^2]$ $\{1.5, 1.8, 2.0, 2.5, 3.0\}$
- maksymalny zryw wrzeciona wyrażony w $[m/s^3]$ $\{10, 20, 30\}$

Ze względu na charakterystykę konstrukcji i oprogramowania maszyn CNC długość dyskretnego kroku czasowego podczas wszystkich symulacji wynosi 2ms. Dodatkowo w celu zapewnienia możliwości przeprowadzenia każdego procesu obróbki zdecydowano się na to, by podczas rejestrowania symulacji sterowanych algorytmem RPRO żądana dokładność obróbki wynosiła 0.01mm. Postanowiono również, by dla każdej z dziewięciu grup, rozumianych jako kombinacje długości trajektorii oraz jej gęstości, wylosować po dziesięć różnych trajektorii. Finalnie utworzona baza danych składa się z łącznie 5400 zarejestrowanych procesów obróbczych, których zestawy parametrów są między sobą parami różne. Jednocześnie omówiona praca jest realizacją Zadanie 2.b).



Rys. 10 Schemat działania systemu zaproponowanego w pracy [A-6]

Po pomyślnym utworzeniu niezbędnej bazy danych autor niniejszej rozprawy wraz z zespołem podjął się wyzwania utworzenia systemu opartego na systemie wnioskowania rozmytego, którego zadaniem było sterowanie dynamiką wrzeciona maszyny CNC. Schemat działania rozwiązania zaproponowanego w pracy [A-6] został przedstawiony na Rys. 10. Jego kluczowym elementem jest wspomniany już system sterowania rozmytego (ang. Fuzzy Inference System - FIS). Wejście rzeczonego systemu stanowią zarówno zlecona przez operatora trajektoria jak również parametry dynamiki maszyny oraz aktualna dynamika ruchu wrzeciona. Na podstawie tych informacji system wyznacza decyzję określającą, która z możliwych akcji ma zostać podjęta, mianowicie:

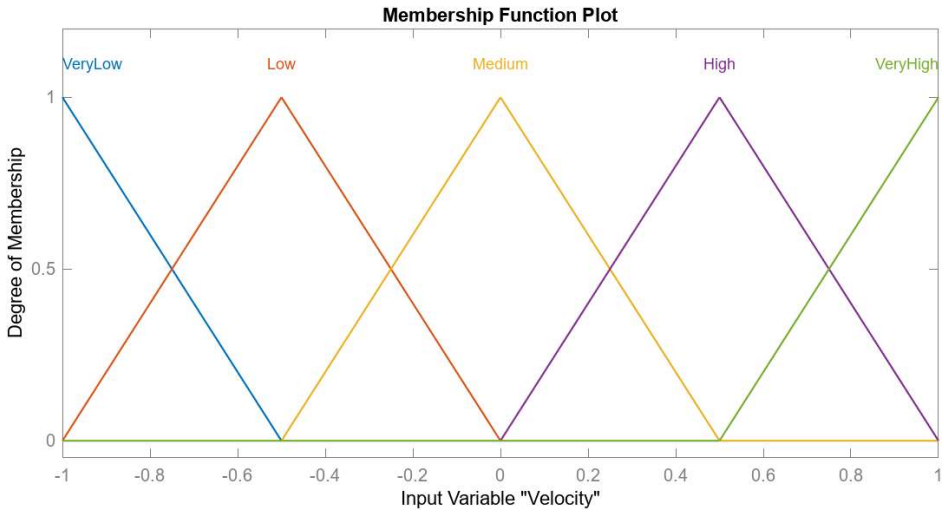
- należy spowolnić wrzeciono,
- należy utrzymać aktualną prędkość wrzeciona,
- należy zwiększyć prędkość wrzeciona.

Po wykonaniu akcji wskazanej przez system symulator wyznacza nowe położenie wrzeciona, czym zamyka cykl. Kluczową rolę pełni również ekspert. Opracowuje on reguły, na których operuje system rozmyty.

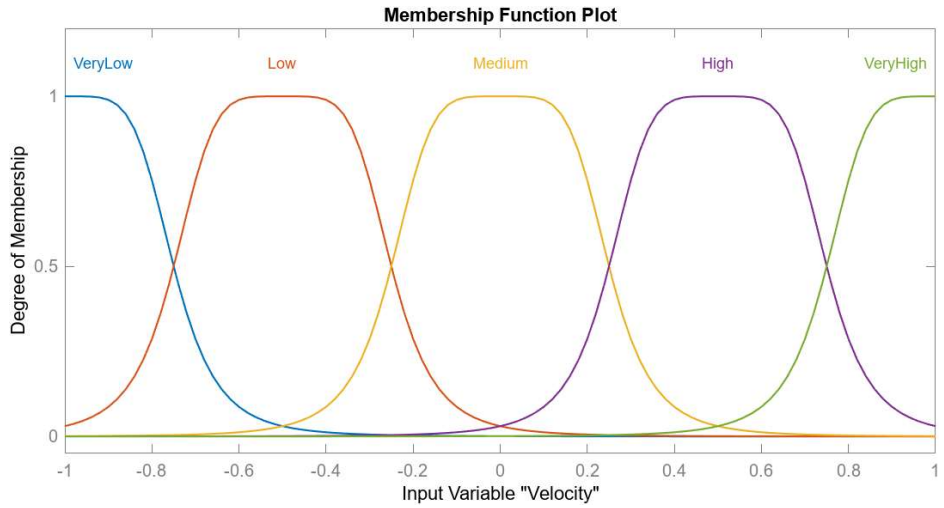
Badania obejmowały dwa modele – system o dwóch wejściach oraz system o trzech wejściach. Każdy z modeli zaprojektowany został w dwóch wariantach określających rodzaj funkcji przynależności: funkcja trójkątna lub krzywa dzwonowa. Wejścia stanowiły odpowiednio przeskalowane wielkości:

- **znormalizowana prędkość wrzeciona** określona jako stosunek aktualnej wartości prędkości oraz maksymalnej możliwej prędkości wrzeciona,
- **znormalizowany dystans do następnego punktu referencyjnego** określony jako stosunek odległości do następnego punktu referencyjnego oraz maksymalnej odległości między dwoma sąsiednimi punktami referencyjnymi w rozważanej trajektorii,
- **znormalizowane przyspieszenie wrzeciona** określone jako stosunek aktualnej wartości przyspieszenia oraz maksymalnego możliwego przyspieszenia wrzeciona.

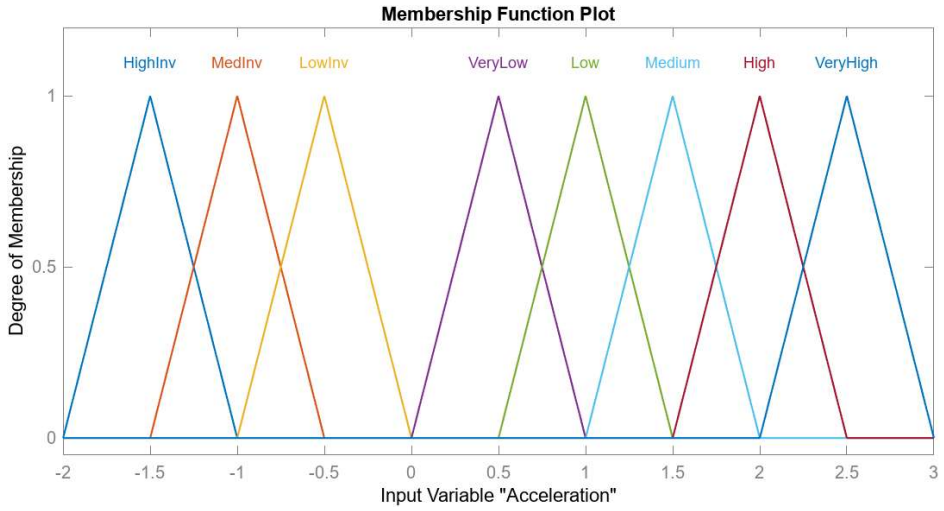
Pierwsze dwa z wymienionych wejść zostały opisane pięcioma równomiernie rozłożonymi funkcjami reprezentującymi rozmyte wielkości: *bardzo małe*, *małe*, *średnie*, *duże* lub *bardzo duże*. Dokładny kształt funkcji opisujących wejście reprezentujące znormalizowaną prędkość wrzeciona został przedstawiony odpowiednio dla funkcji o kształcie trójkątnym na Rys. 11, a dla krzywej dzwonowej na Rys. 12. Wejście opisujące znormalizowane przyspieszenie wrzeciona opisane zostało natomiast przez osiem funkcji z uwagi na możliwość hamowania wrzeciona: *wsteczne duże*, *wsteczne średnie*, *wsteczne małe*, *bardzo małe*, *małe*, *średnie*, *duże* lub *bardzo duże*. Dokładne pokrycie tego wejścia funkcjami zostało przedstawione na Rys. 13 oraz Rys. 14. Wyjście systemu opisane zostało zaś trzema funkcjami reprezentującymi możliwe do wykonania akcje: *spowolnienie ruchu wrzeciona*, *utrzymanie prędkości wrzeciona* lub *przyspieszenie ruchu wrzeciona*. Pokrycie funkcjami wyjścia dla obydwu wariantów zostało przedstawione na Rys. 15 oraz Rys. 16.



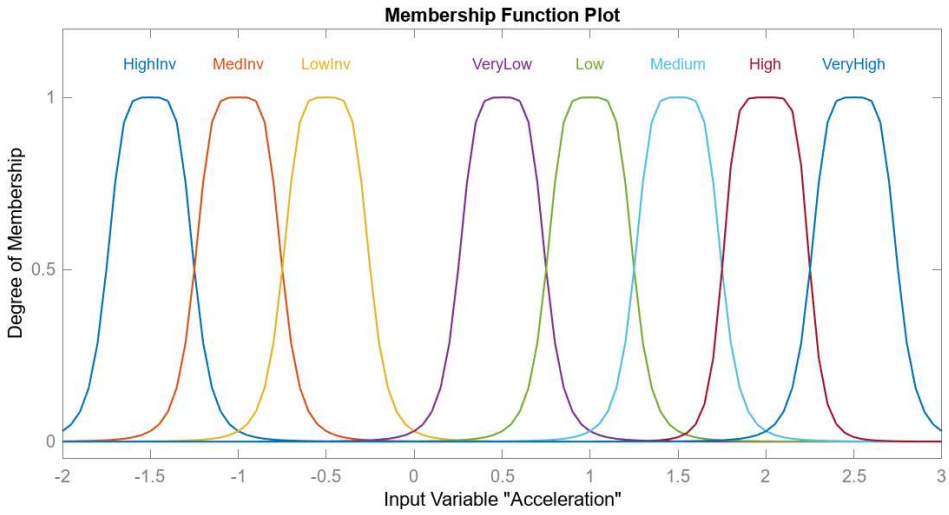
Rys. 11 System Mamdaniego - pokrycie wejścia "znormalizowana prędkość" - funkcje trójkątne [opracowanie własne]



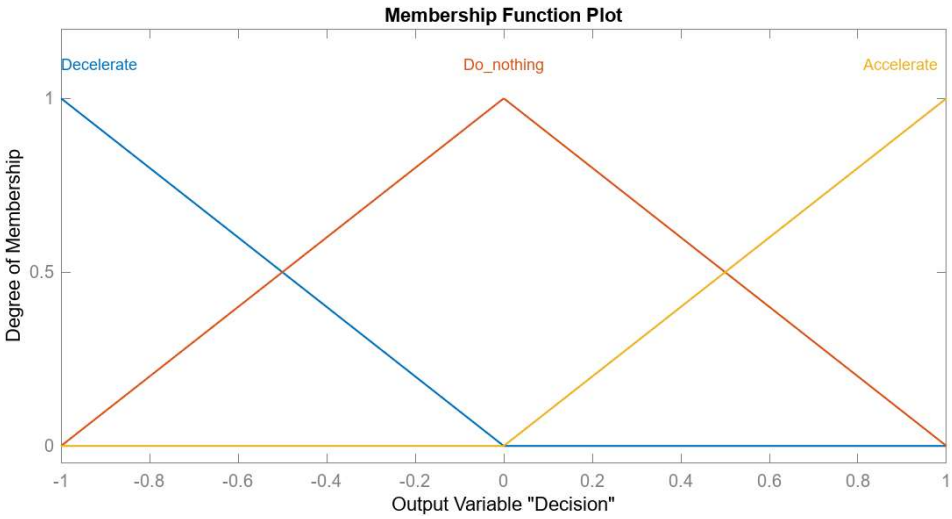
Rys. 12 System Mamdaniego - pokrycie wejścia "znormalizowana prędkość" - krzywe dzwonowe [opracowanie własne]



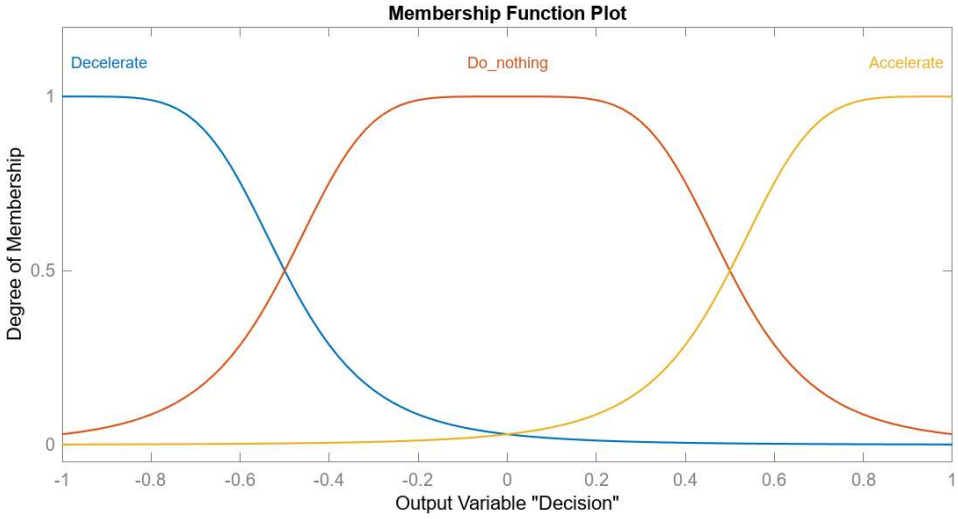
Rys. 13 System Mamdaniego - pokrycie wejścia "znormalizowane przyspieszenie" - funkcje trójkątne [opracowanie własne]



Rys. 14 System Mamdaniego - pokrycie wejścia "znormalizowane przyspieszenie" – krzywe dzwonowe [opracowanie własne]

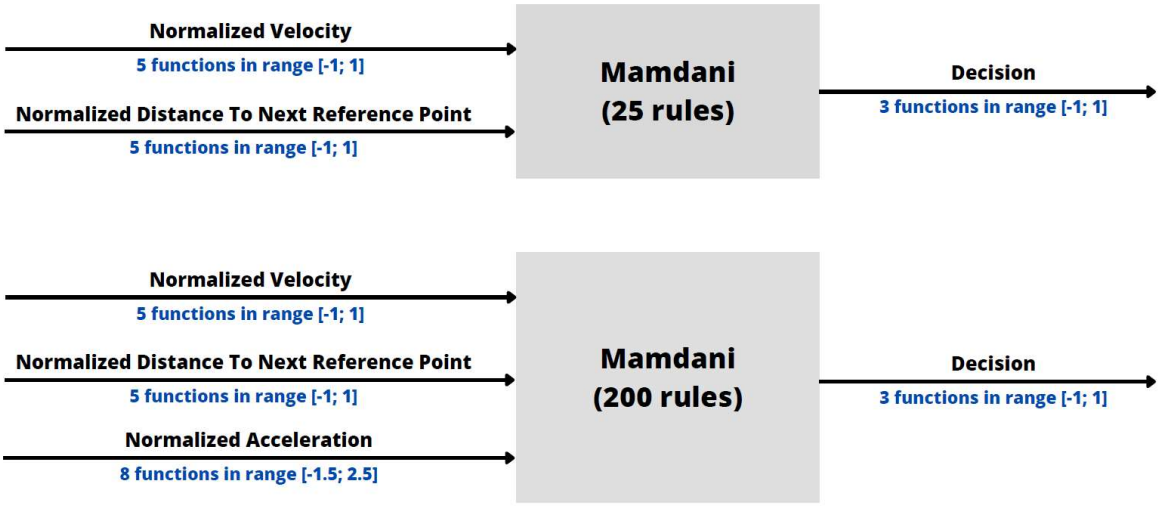


Rys. 15 System Mamdaniego - pokrycie wyjścia - funkcje trójkątne [opracowanie własne]



Rys. 16 System Mamdaniego - pokrycie wyjścia - krzywe dzwonowe [opracowanie własne]

Pierwszy z zaproponowanych modeli został uzupełniony o 25 reguł zapewniających pełne pokrycie sytuacji, które mogą wystąpić podczas procesu obróbki. Analogicznie dla drugiego modelu autorzy zaproponowali 200 reguł. Schemat opisanych modeli został przedstawiony na Rys. 17.



Rys. 17 Budowa modeli wykorzystanych w pracy [A-6]

Każda z zaproponowanych reguł miała postać koniunkcji, co można zauważyć na Rys. 18, gdzie zostały przedstawione wszystkie reguły dla pierwszego z opisywanych modeli. W trakcie badań, chcąc zoptymalizować czas wykonywania obliczeń, autorzy postanowili również przebadać modele o regułach skróconych w oparciu o logikę boolowską. Zaproponowane reguły skrócone dla pierwszego modelu zostały przedstawione na Rys. 19. Statystykę każdego z zaproponowanych modeli podsumowuje Tab. 1. Dzięki zaproponowanemu skróceniu reguł udało się zmniejszyć liczbę obliczeń wykonywanych przy podjęciu każdej z decyzji o odpowiednio 36% dla modelu pierwszego oraz aż 62% dla drugiego modelu. Można też zauważyć, że w przypadku modelu o trzech wejściach skrócenie reguł wprowadziło balans pomiędzy liczbami reguł o poszczególnych następnikach.

Tab. 1 Statystyki modeli badanych w pracy [A-6]

Rules type	System type	Decelerate	Do nothing	Accelerate	Total rules number
full cover	**2-variable**	4	8	13	25
	3-variable	56	47	97	200
reduced	**2-variable**	4	8	4	16
	3-variable	21	23	32	76

	Rule
1	If Velocity is VeryLow and DistanceToNextRefPoint is VerySmall then Decision is Do_nothing
2	If Velocity is VeryLow and DistanceToNextRefPoint is Small then Decision is Accelerate
3	If Velocity is VeryLow and DistanceToNextRefPoint is Medium then Decision is Accelerate
4	If Velocity is VeryLow and DistanceToNextRefPoint is Big then Decision is Accelerate
5	If Velocity is VeryLow and DistanceToNextRefPoint is VeryBig then Decision is Accelerate
6	If Velocity is Low and DistanceToNextRefPoint is VerySmall then Decision is Do_nothing
7	If Velocity is Low and DistanceToNextRefPoint is Small then Decision is Do_nothing
8	If Velocity is Low and DistanceToNextRefPoint is Medium then Decision is Accelerate
9	If Velocity is Low and DistanceToNextRefPoint is Big then Decision is Accelerate
10	If Velocity is Low and DistanceToNextRefPoint is VeryBig then Decision is Accelerate
11	If Velocity is Medium and DistanceToNextRefPoint is VerySmall then Decision is Decelerate
12	If Velocity is Medium and DistanceToNextRefPoint is Small then Decision is Decelerate
13	If Velocity is Medium and DistanceToNextRefPoint is Medium then Decision is Do_nothing
14	If Velocity is Medium and DistanceToNextRefPoint is Big then Decision is Accelerate
15	If Velocity is Medium and DistanceToNextRefPoint is VeryBig then Decision is Accelerate
16	If Velocity is High and DistanceToNextRefPoint is VerySmall then Decision is Decelerate
17	If Velocity is High and DistanceToNextRefPoint is Small then Decision is Do_nothing
18	If Velocity is High and DistanceToNextRefPoint is Medium then Decision is Do_nothing
19	If Velocity is High and DistanceToNextRefPoint is Big then Decision is Accelerate
20	If Velocity is High and DistanceToNextRefPoint is VeryBig then Decision is Accelerate
21	If Velocity is VeryHigh and DistanceToNextRefPoint is VerySmall then Decision is Decelerate
22	If Velocity is VeryHigh and DistanceToNextRefPoint is Small then Decision is Do_nothing
23	If Velocity is VeryHigh and DistanceToNextRefPoint is Medium then Decision is Do_nothing
24	If Velocity is VeryHigh and DistanceToNextRefPoint is Big then Decision is Accelerate
25	If Velocity is VeryHigh and DistanceToNextRefPoint is VeryBig then Decision is Accelerate

Rys. 18 System Mamdaniego o dwóch wejściach - zaproponowane reguły [opracowanie własne]

	Rule
1	If Velocity is VeryLow and DistanceToNextRefPoint is VerySmall then Decision is Do_nothing
2	If Velocity is Low and DistanceToNextRefPoint is VerySmall then Decision is Do_nothing
3	If Velocity is Low and DistanceToNextRefPoint is Small then Decision is Do_nothing
4	If Velocity is Low and DistanceToNextRefPoint is Medium then Decision is Accelerate
5	If Velocity is Medium and DistanceToNextRefPoint is VerySmall then Decision is Decelerate
6	If Velocity is Medium and DistanceToNextRefPoint is Small then Decision is Decelerate
7	If Velocity is Medium and DistanceToNextRefPoint is Medium then Decision is Do_nothing
8	If Velocity is High and DistanceToNextRefPoint is VerySmall then Decision is Decelerate
9	If Velocity is High and DistanceToNextRefPoint is Small then Decision is Do_nothing
10	If Velocity is High and DistanceToNextRefPoint is Medium then Decision is Do_nothing
11	If Velocity is VeryHigh and DistanceToNextRefPoint is VerySmall then Decision is Decelerate
12	If Velocity is VeryHigh and DistanceToNextRefPoint is Small then Decision is Do_nothing
13	If Velocity is VeryHigh and DistanceToNextRefPoint is Medium then Decision is Do_nothing
14	If Velocity is VeryLow and DistanceToNextRefPoint is not VerySmall then Decision is Accelerate
15	If DistanceToNextRefPoint is Big then Decision is Accelerate
16	If DistanceToNextRefPoint is VeryBig then Decision is Accelerate

Rys. 19 System Mamdaniego o dwóch wejściach - reguły skrócone [opracowanie własne]

W trakcie kreowania reguł autorzy skupili się na osiągnięciu dość dobrej dokładności, czyli takiej na poziomie około 20 mikrometrów, która pozwala na zastosowanie algorytmu w przemyśle. Dodatkowym aspektem była chęć uzyskania wygładzonego ruchu wrzeciona, który pozytywnie wpływa nie tylko na dokładność procesu obróbczego, lecz także zmniejsza ilość energii potrzebnej do poruszania wrzecionem oraz zwiększa żywotność wiertła, ponieważ nie naraża go na nadmierne naprężenia [81].

Tab. 2 Średnia dokładność procesu obróbki [µm] – RPRO [opracowanie własne]

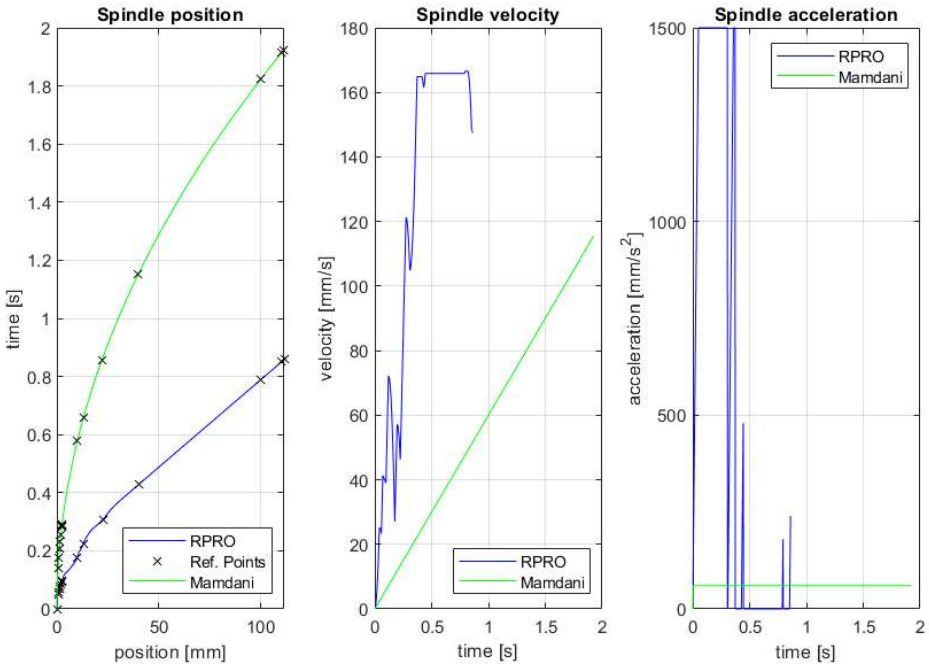
RPRO – średnia dokładność procesu obróbki [µm]		**Gęstość punktów referencyjnych w trajektorii**		
		Duża	**Średnia**	**Mała**
Liczba punktów referencyjnych	**15**	21.36 ± 8.74	40.54 ± 17.10	49.72 ± 22.17
	50	15.39 ± 15.85	19.21 ± 16.80	16.23 ± 14.32
	100	3.67 ± 1.47	3.88 ± 1.27	3.95 ± 01.29

Tab. 3 Średnia dokładność procesu obróbki [µm] – Mamdani [A-6] [opracowanie własne]

Mamdani – średnia dokładność procesu obróbki [µm]		**Gęstość punktów referencyjnych w trajektorii**		
		Duża	**Średnia**	**Mała**
Liczba punktów referencyjnych	**15**	15.28 ± 4.34	27.29 ± 8.15	32.07 ± 9.94
	50	28.54 ± 10.39	33.69 ± 12.06	32.96 ± 11.02
	100	25.29 ± 11.16	28.12 ± 11.99	28.68 ± 11.74

W Tab. 2 przedstawiono średnią dokładność procesu obróbki dla algorytmu RPRO stanowiącego odniesienie, natomiast w Tab. 3 zebrano najlepsze z uzyskanych przez zaproponowane modele średnich dokładności procesu obróbki. Obydwie zaprezentowane tabele zawierają wyniki z wyszczególnieniem każdej z 9 podgrup w bazie danych reprezentujących zbiory trajektorii o konkretnej liczbie punktów oraz gęstości ich rozłożenia. Otrzymane wartości średniej dokładności procesu obróbki znalazły się w zakresie od 15 do 35 mikrometrów, co spełniło założenia zespołu projektowego. Należy również zaznaczyć, że obydwa badane modele we wszystkich wariantach spełniły oczekiwania, natomiast najlepiej radził sobie model z trzema wejściami w wariancie z większą liczbą reguł. Różnica ta, choć nie była duża, to jest istotna z punktu widzenia złożoności obliczeniowej rozwiązania i wskazuje na konieczność szukania

alternatywnych metod redukcji tej złożoności. Wyższość modelu z większą liczbą wejść była zgodna z przewidywaniami autorów pracy – większa ilość informacji o aktualnej sytuacji powinna skutkować zwiększoną jakością pracy systemu. Kolejnym aspektem poddanym weryfikacji jest zamierzone wygładzenie pracy wrzeciona. W tym celu przeanalizowano przebieg pracy algorytmu, a przykładową dynamikę ruchu wrzeciona podczas procesu obróbczego, przedstawiono na Rys. 20. Analiza wykazała, że udało się osiągnąć zamierzony efekt redukcji szarpnięć wrzeciona, a w konsekwencji obniżyć wydatek energetyczny oraz wydłużyć żywotność wiertła i całej maszyny w porównaniu do algorytmu RPRO.



Rys. 20 Porównanie działania algorytmu zaproponowanego w pracy [A-6] oraz algorytmu RPRO

Zaproponowane rozwiązanie przedstawione w artykule [A-6] wprowadza nowe podejście do zadania optymalizacji procesu obróbczego poprzez generowanie określonego g-code'u z wykorzystaniem metod sztucznej inteligencji. Głównymi jego zaletami są zarówno łatwość implementacji jak również łatwość interpretacji sterujących modelem reguł. Dodatkowo daje ono możliwość prostej parametryzacji oraz personalizacji polegającej na optymalizacji konkretnych aspektów procesu obróbczego poprzez odpowiednie zarządzanie zestawem reguł. Omówiona publikacja realizuje Zadanie 3.b).

Drugie opracowanie [A-3] zawiera opis innej metody zaproponowanej przez autora niniejszej rozprawy wraz z zespołem. Wprowadzoną nowością był odmienny sposób przygotowania reguł systemu, które w tym podejściu są modelowane przez algorytm optymalizacji rojem cząstek (ang. Particle Swarm Optimization - PSO). Głównym celem było nauczenie zaprojektowanego systemu naśladowania pracy istniejącego algorytmu. Jako algorytm źródłowy wybrano wspomniany już wcześniej algorytm RPRO. Chcąc sprawdzić zdolności uogólniające tworzonego systemu postanowiono do nauki wykorzystać jedynie dane z symulacji procesów obróbczych dla najkrótszych trajektorii (15 punktów) o gęstościach punktów referencyjnych *Dużej* oraz *Średniej*. Dodatkowo, żeby sprawdzić stabilność rozwiązania, proces nauki powtarzano 5-krotnie wybierając do nauki dane z odpowiedniej grupy reprezentujące kolejne pary trajektorii, czyli te o numerach 1-2, 3-4, 5-6, 7-8, oraz 9-10. Dane testowe stanowiły informacje o procesach obróbczych dla pozostałych ośmiu trajektorii w danej grupie (przykładowo dla pary 1-2 znajdującej się w zbiorze trajektorii uczących trajektorie 3-10 stanowiły zbiór testujący).

Tab. 4 Statystyki i rezultaty procesu nauki systemów logiki rozmytej [A-3]

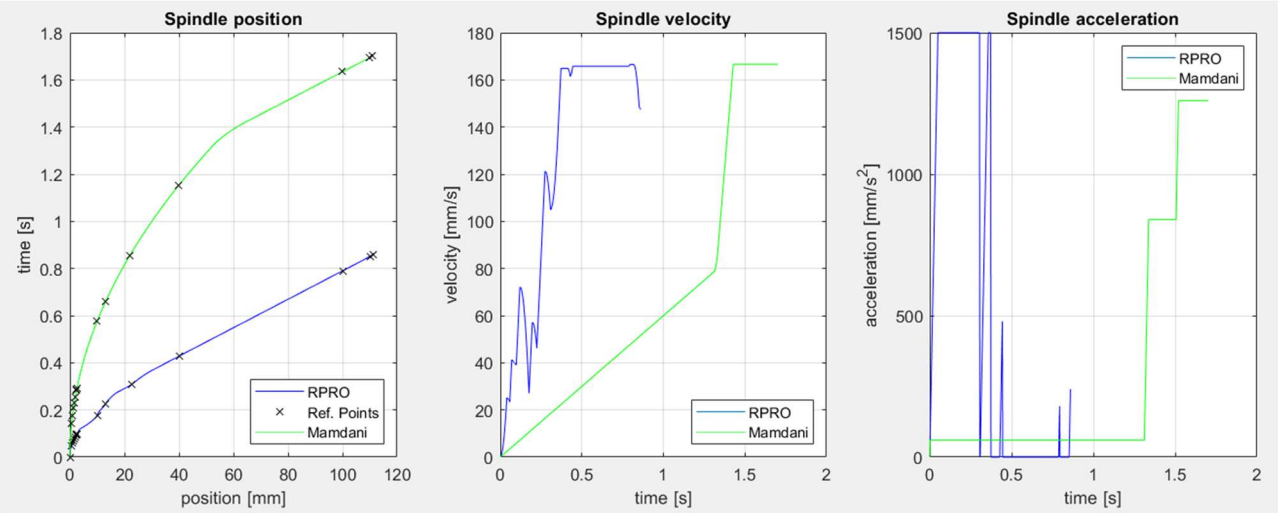
Ref. Pts. Density	Training Pair Number	Max. Iter. Num.	Training Time	RMSE Training Set [mm]	RMSE Testing Set [mm]	Bad Decission Fraction	Train Records Number	Test Records Number	Rules Number
High	1	60	1h 8min 43s	0.341	0.338	0.128	13628	53912	52
	2	60	52min 8s	0.355	0.361	0.134	12759	54781	54
	3	60	1h 19min 43s	0.334	0.337	0.116	14552	52988	60
	4	60	1h 14min 15s	0.333	0.337	0.110	12673	54867	81
	5	60	1h 17min 18s	0.325	0.334	0.114	13928	53612	79
Medium	1	60	56min 28s	0.407	0.408	0.093	60623	286040	78
	2	60	1h 36min 9s	0.495	0.441	0.168	77910	268753	145
	3	60	1h 21min 7s	0.501	0.446	0.166	69874	276789	122
	4	60	52min 29s	0.393	0.405	0.089	67603	279060	56
	5	60	1h 28min 33s	0.517	0.473	0.199	70653	276010	136

Rezultaty badań zebrano w Tab. 4. W każdym z badanych przypadków ustalono, że proces nauki będzie trwał maksymalnie 60 iteracji. Rzeczywisty czas nauki oscylował od około 52 minut do maksymalnie 1 godziny 36 minut. W 9 z 10 przypadków metryka RMSE dla danych treningowych spadła poniżej wartości 0.5, co oznacza, że system poprawnie reaguje na większość sytuacji – poprawnie przypisuje wartość -1, 0, lub 1 w zależności od wybranej akcji. Osiągnięte wartości metryki RMSE dla danych testowych w każdym przypadku osiągnęły zadowalający poziom. Sukces nauki potwierdza również metryka informująca o ułamku błędnych decyzji, czyli tych niezgodnych z zachowaniem algorytmu RPRO. Bezpośrednio przekłada się to na skuteczność proponowanej metody od 80.1% w najgorszym badanym przypadku do 91.1% w najlepszej sytuacji. Ostatnim istotnym elementem jest liczba reguł, która oscyluje w zakresie od wartości 52 aż do 145, przy czym głównie przyjmuje wartości poniżej 80. Należy w tym miejscu zauważyć, że liczba ta jest bliska liczbie zredukowanych reguł dla systemu z pracy [A-6]. Podobieństwo odnaleziono również w samej treści reguł, mianowicie tak, jak uprzednio, mają one postać koniunkcji poprzedników.

Tab. 5 Wyniki badania porównawczego średniej dokładności [A-3]

Ref. Pts. Density	**Reference Points Number in Trajectory**	**RPRO**	**Mamdani**				
			1	**2**	**3**	**4**	**5**
High	**15**	0.0214 ± 0.0087	0.0255 ± 0.0061	0.0255 ± 0.0061	0.0256 ± 0.0062	0.0255 ± 0.0061	0.0232 ± 0.0063
	50	0.0405 ± 0.0172	0.0314 ± 0.0088	0.0314 ± 0.0088	0.0313 ± 0.0088	0.0314 ± 0.0088	0.0315 ± 0.0095
	100	0.0497 ± 0.0222	0.0327 ± 0.0100	0.0327 ± 0.0099	0.0327 ± 0.0099	0.0327 ± 0.0100	0.0332 ± 0.0102
Medium	**15**	0.0154 ± 0.0158	0.0162 ± 0.0029	0.0166 ± 0.0031	0.0270 ± 0.0109	0.0273 ± 0.0103	0.0162 ± 0.0030
	50	0.0192 ± 0.0169	0.0265 ± 0.0056	0.0269 ± 0.0060	0.0325 ± 0.0109	0.0326 ± 0.0109	0.0269 ± 0.0060
	100	0.0162 ± 0.0143	0.0303 ± 0.0080	0.0311 ± 0.0087	0.0323 ± 0.0101	0.0322 ± 0.0099	0.0305 ± 0.0082

Autorzy zweryfikowali również potencjalne możliwości uogólniające przeprowadzając symulację procesów obróbczych dla wszystkich kombinacji w bazie danych o wymienionych wcześniej gęstościach. Wyniki średniej dokładności wyrażonej w milimetrach wraz z odchyleniem standardowym odpowiednio pogrupowano, a także porównano z korespondującymi wynikami dla wzorcowego algorytmu RPRO. Tab. 5 przedstawia zebrane rezultaty. Analizując przedstawione dane można dostrzec, że proponowane rozwiązanie osiąga wyniki na podobnym poziomie co algorytm RPRO, a czasami nawet nieco lepsze, jak w przypadku trajektorii o wysokiej gęstości i liczbie punktów równej 50 oraz 100. Drugim ważnym aspektem jest stabilność proponowanego rozwiązania, którą potwierdzają niskie wartości odchyleń standardowych.

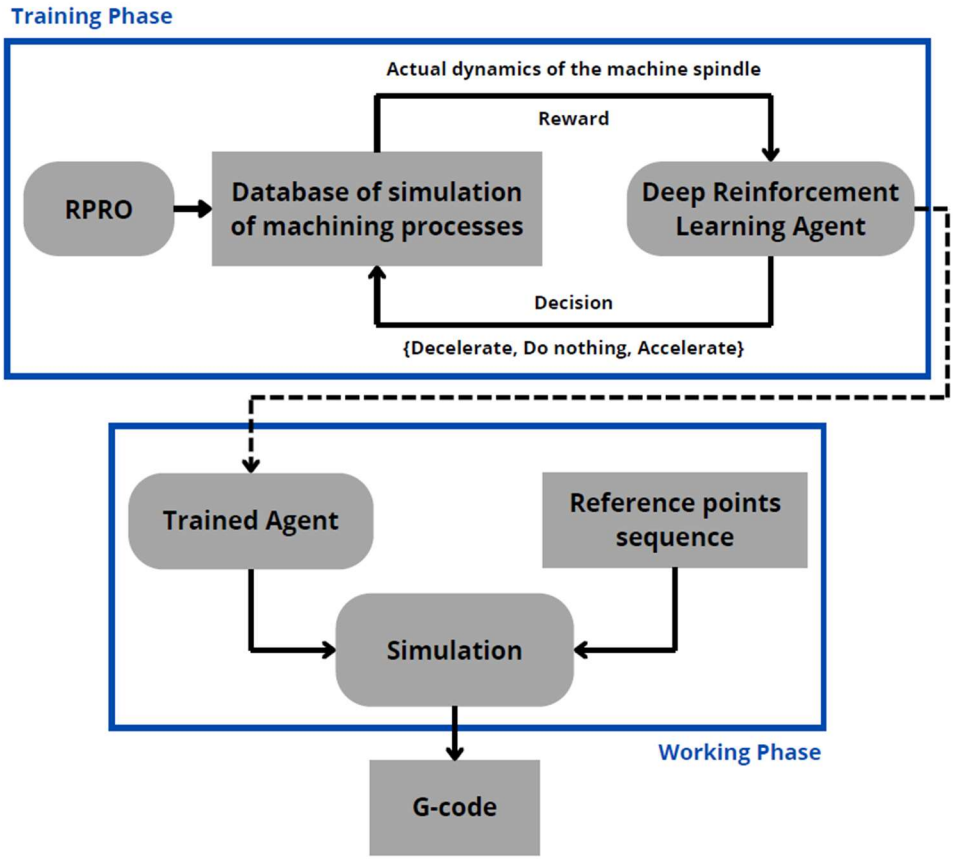


Rys. 21 Porównanie działania algorytmu zaproponowanego w pracy [A-3] oraz algorytmu RPRO

Uzyskane wyniki wskazują na zadowalające odwzorowanie jakości pracy algorytmu RPRO. Dodatkowo potwierdzona została hipoteza stawiana przez autorów o możliwościach uogólniających zaproponowanego systemu, który dla nowych danych radził sobie na podobnym poziomie, uzyskując w pełni akceptowalne rezultaty. Chcąc uzupełnić analizę, autorzy [A-3] ponownie porównali też dokładny przebieg procesów obróbczych, z których jeden został przedstawiony na Rys. 21. Analizując wykresy, można zauważyć, że mimo braku zachowania pełnej wierności w stosunku do algorytmu źródłowego, zaproponowany system stabilizował pracę wrzeciona. Stabilizacja jest tutaj, jak uprzednio, rozumiana jako redukcja szarpnięć wywołanych naprzemiennym przyspieszaniem oraz spowalnianiem wrzeciona. Omówiona publikacja realizuje Zadanie 3.b).

2.5. Algorytm uczenia się ze wzmocnieniem w zadaniach sterowania maszynami CNC

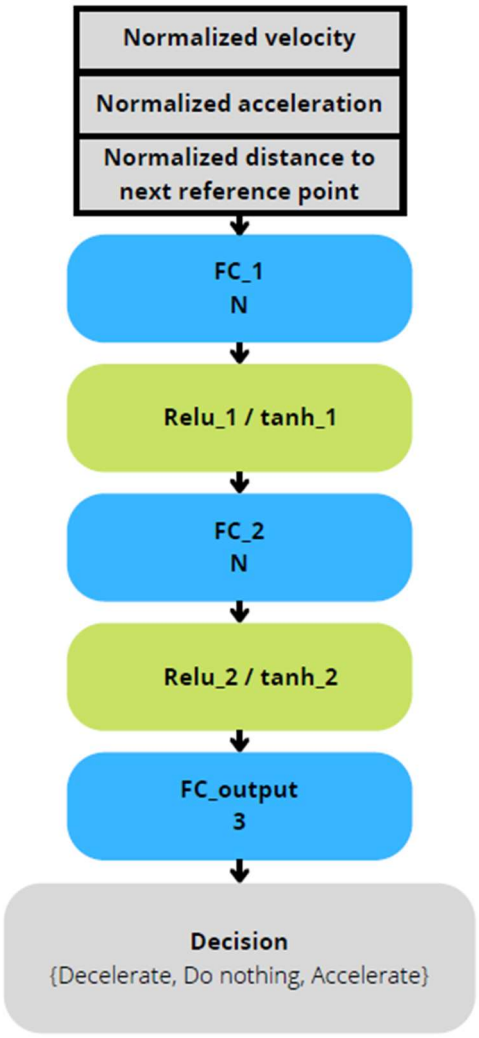
Autor niniejszej rozprawy podjął również działania mające na celu wykorzystanie metod sztucznej inteligencji do realizacji zadania optymalizacji pracy maszyny CNC poprzez sterowanie dynamiką ruchu jej wrzeciona z wykorzystaniem paradygmatu uczenia się ze wzmocnieniem. Prace te były realizowane i są nadal rozwijane we wspomnianym zespole, którego członkami są dr inż. Bogdan Kwiatkowski oraz dr hab. inż. Damian Mazur. W artykule [A-4] zaprezentowana została metoda nauki naśladowania algorytmu źródłowego z wykorzystaniem głębokiego uczenia się ze wzmocnieniem.



Rys. 22 Schemat działania systemu zaproponowanego w pracy [A-4]

W pracy przyjęto, że każdy proces obróbki można określić mianem epizodu, który składa się z odpowiedniej liczby kroków pozwalającej na realizację wszystkich punktów referencyjnych znajdujących się w zadanej trajektorii. W każdym kroku agent ma możliwość podjęcia jednej z trzech akcji: *spowolnienie wrzeciona*, *utrzymanie prędkości wrzeciona* lub *przyspieszenie wrzeciona*. Decyzja agenta jest oparta na sygnałach otrzymywanych ze środowiska a mianowicie: znormalizowanej prędkości, znormalizowanemu przyspieszeniu oraz znormalizowanej odległości od następnego

punktu referencyjnego. Sposób wyznaczania tych wielkości został już omówiony w rozdziale 2.4. Wartościowanie akcji agenta realizowane jest przez regułę: „*Jeżeli agent w danym kroku wybrał akcję zgodną z informacjami zawartymi w bazie, czyli taką samą jak algorytm RPRO, wtedy otrzymuje wzmocnienie równe 0.0, natomiast w przypadku podjęcia innej decyzji otrzymuje on karę w postaci wzmocnienia o wartości -0.1*”. Schemat nauki oraz pracy systemu został przedstawiony w postaci diagramu na Rys. 22.



Rys. 23 Struktura sztucznej sieci neuronowej wykorzystanej w pracy [A-4]

W zaproponowanym rozwiązaniu rolę aproksymatora funkcji wartości akcji pełniła sztuczna sieć neuronowa, której schematyczna budowa została przedstawiona na Rys. 23. Składała się ona z dwóch warstw w pełni połączonych o liczbie neuronów oznaczonych przez N, po których następowała ostatnia warstwa złożona z trzech neuronów reprezentujących wymienione wcześniej akcje. W trakcie badań autorzy sprawdzali różne kombinacje parametrów sieci, mianowicie testowane były wartości N ze zbioru {6, 8, 12,

16, 20, 24}, a także funkcje aktywacji neuronów pierwszej i drugiej warstwy, takie jak: Rectified Linear Unit – ReLU oraz tangens hiperboliczny – tanh. Badane architektury zostały wyszczególnione w Tab. 6.

Tab. 6 Badane architektury sieci neuronowych [A-4]

Evaluated Architecture Number	Layerwise Neuron Count (N)	Layerwise Activation Function
1	24	ReLU
2	24	tanh
3	20	ReLU
4	20	tanh
5	16	ReLU
6	16	tanh
7	12	ReLU
8	12	tanh
9	8	ReLU
10	8	tanh
11	6	ReLU
12	6	tanh

Badaniu poddano również różne wartości tzw. współczynnika dalekowzroczności agenta zwanego też współczynnikiem dyskontowania (ang. Discount Factor) oraz różne wartości współczynnika szybkości uczenia (ang. Learning Rate). Badania zdecydowano się przeprowadzić na dwóch podgrupach bazy danych odnoszących się do trajektorii o 15 punktach referencyjnych oraz odpowiednio gęstym i średnim rozłożeniu punktów referencyjnych. Podobnie jak wcześniej, w celu sprawdzenia stabilności rozwiązania oraz jego możliwości uogólniających, dla każdej z grup badania 5-krotnie powtarzano. Każde z powtórzeń określało wybór odpowiedniej pary trajektorii jako tych, które stanowić będą dane uczące. Naturalnie dane testujące stanowiły wtedy zapisy procesów obróbczych rozważających pozostałe osiem trajektorii (np. podczas drugiego powtórzenia trajektorie 3 oraz 4 stanowią dane uczące, co z kolei determinuje trajektorie 1-2 oraz 5-10 jako dane testujące). W każdym z powtórzeń liczba zapisów procesów obróbczych wykorzystywanych do nauki agenta wynosiła 120, natomiast liczba zapisów w zbiorze testującym wynosiła 480.

Tab. 7 Wyniki badań - współczynnik korelacji Pearson'a dla liczby kroków czasowych [A-4]

Steps Count Correlation		Discount Factor					
		0.99			0.999		
Ref. Points Density	Network Arch. Number	Learning Rate			Learning Rate		
		1e-3	1e-4	1e-5	1e-3	1e-4	1e-5
High	1	0.847	0.750	0.848	0.647	0.844	0.844
	2	0.855	0.852	0.855	0.847	0.842	0.846
	3	0.844	0.855	0.848	0.839	0.777	0.847
	4	0.852	0.851	0.849	0.687	0.848	0.849
	5	0.849	0.856	---	0.844	0.853	---
	6	0.849	0.853	---	0.855	0.852	---
	7	0.825	0.857	---	0.847	0.846	---
	8	0.850	0.851	0.849	0.850	0.836	0.848
	9	0.821	0.856	---	0.845	0.850	---
	10	0.852	0.855	0.846	0.846	0.848	0.845
	11	0.842	0.851	---	0.848	0.848	---
	12	0.855	0.849	---	0.844	0.845	---
Medium	1	0.773	0.777	0.774	0.772	---	---
	2	0.777	0.774	0.774	---	0.472	0.538
	3	0.774	0.774	0.774	---	---	---
	4	0.771	0.777	0.775	---	0.773	0.726
	5	0.774	0.774	---	---	---	---
	6	0.771	0.775	0.664	---	0.386	0.237
	7	0.777	0.774	0.578	---	---	0.041
	8	0.775	0.776	0.774	---	0.171	0.172
	9	0.774	0.774	---	---	---	---
	10	0.774	0.775	0.774	---	0.009	0.112
	11	0.772	0.774	0.543	---	---	0.054
	12	0.774	0.775	---	---	---	---

Autorzy zaproponowali trzy metryki opisujące działanie wyuczonych agentów. Pierwszą z nich jest współczynnik korelacji Pearson'a określający związek pomiędzy liczbą kroków czasowych każdego z procesów obróbczych zaplanowanych przez algorytm RPRO, a liczbą kroków wyznaczonych przez wyuczonego agenta. Wartość 0 zaproponowanej metryki wskazywać będzie na brak jakiejkolwiek korelacji, natomiast wartości bliskie 1 informować będą o ścisłym związku między liczbami kroków. Druga zaproponowana metryka to średnia dokładność realizacji punktów referencyjnych wyrażona w mikrometrach. Uzupełnieniem tych informacji jest trzecia metryka, czyli mediana liczby kroków czasowych w procesach obróbczych wygenerowanych przez wyuczonego agenta. Omówione metryki opisujące wyniki przeprowadzonych eksperymentów zostały przestawione odpowiednio: współczynnik korelacji w Tab. 7, średnia dokładność procesu obróbki w Tab. 8, a mediana liczby kroków czasowych w Tab. 9.

Tab. 8 Wyniki badań - średnia dokładność [A-4]

Mean Error		Discount Factor					
		0.99			0.999		
Ref. Points Density	Network Arch. Number	Learning Rate			Learning Rate		
		1e-3	1e-4	1e-5	1e-3	1e-4	1e-5
High	1	2.38	2.32	2.56	2.47	2.46	2.55
	2	2.45	2.43	2.51	2.46	2.44	2.63
	3	2.4	2.46	2.49	2.44	2.45	2.58
	4	2.45	2.49	2.51	2.52	2.52	2.46
	5	2.56	2.5	3.11	2.41	2.48	3.29
	6	2.47	2.43	4.69	2.48	2.35	4.81
	7	2.32	2.37	4.45	2.46	2.52	4.57
	8	2.49	2.47	2.56	2.49	2.43	2.57
	9	2.4	2.5	3.16	2.41	2.51	3.38
	10	2.45	2.48	2.54	2.39	2.55	2.54
	11	2.47	2.51	3.86	2.47	2.49	4.48
	12	2.41	2.51	435	2.46	2.48	437
Medium	1	5.53	5.9	5.41	2708	4496	4450
	2	5.79	5.31	5.14	4835	2661	2926
	3	5.54	5.03	1561	4481	4799	3580
	4	4.99	5.47	5.34	4708	2531	816
	5	5.37	6.2	7.08	4653	4742	4599
	6	5.08	5.56	2445	4852	1814	3607
	7	5.02	5.19	2414	3727	4741	3581
	8	5.42	5.14	5.06	4698	3369	2443
	9	4.83	5.14	3344	3568	3742	4455
	10	5.4	5.34	5.13	4677	4286	4366
	11	5.55	5.91	2446	4450	4622	3472
	12	5.21	5.32	4421	4736	2635	4832

Analizując przedstawione współczynniki korelacji (Tab. 7) zauważono, że większość spośród badanych modeli osiągnęła wartości tej metryki większe od 0.75, a prawie połowa wartości większe od 0.84, co może wskazywać, że działanie wyuczonych agentów jest podobne do działania wzorcowego algorytmu RPRO. Zauważono również, że dla pewnych kombinacji badanych parametrów, wartości współczynnika korelacji są bliższe wartości 0 lub w skrajnych przypadkach nie było możliwe ich policzenie (miejsca oznaczone potrójną pauzą „---”). W tych miejscach dokładność procesu obróbki często była nieakceptowalna, natomiast liczba kroków w pewnych przypadkach wynosiła 1, co wskazuje na całkowity brak realizacji zamierzonej trajektorii. Kolejnym wnioskiem, jaki nasunął się autorom podczas analizy Tab. 7, jest informacja o preferowanych wartościach współczynnika dyskontowania równego 0.99 oraz współczynnika szybkości uczenia równego 0.001. Analizując otrzymane wartości średnich dokładności (Tab. 8), autorzy zauważyli, że agenci osiągają jej bardzo dobry poziom – dokładność rzędu 2.0–5.0 mikrometrów pozwala na zastosowanie systemu w przemyśle. Dodatkowo potwierdzono

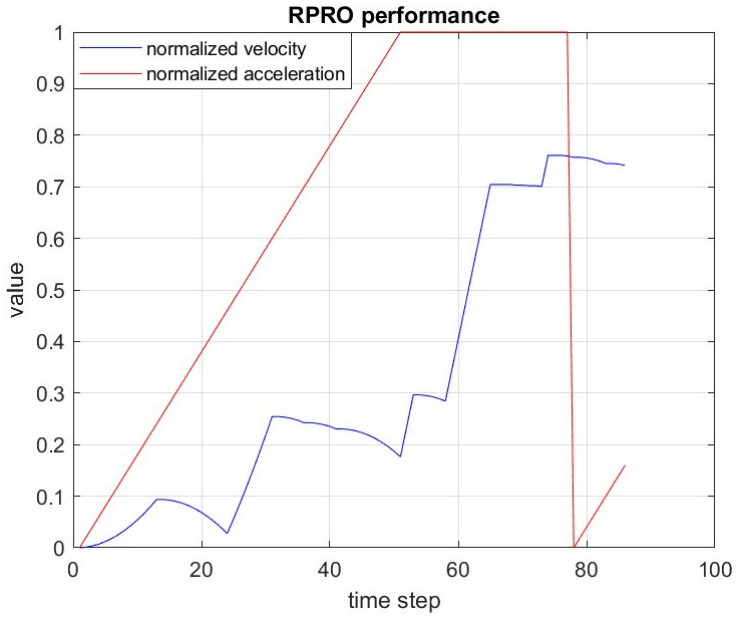
wnioski wysnute wcześniej – agenci o słabej korelacji liczby kroków charakteryzują się nieakceptowalnym poziomem osiąganej dokładności procesu obróbki. Ponadto autorzy zauważyli również, że agenci oparci na architekturach o sigmoidalnej funkcji aktywacji osiągają nieco lepsze wyniki od tych opartych na architekturach z funkcją RELU. Uzupełniając zebrane informacje o mediany liczby kroków (Tab. 9), autorzy zauważyli również, że zaproponowani agenci realizują procesy obróbki nie tylko dokładniej niż algorytm RPRO, lecz także wykorzystują do tego mniejszą liczbę kroków czasowych. Algorytm RPRO potrzebował odpowiednio 110 kroków czasowych na realizację przykładowej trajektorii ze zbioru o dużej gęstości punktów referencyjnych oraz 500 kroków na wykonanie trajektorii ze zbioru o średniej gęstości punktów referencyjnych. Uzyskane rozwiązanie pozwoliło skrócić czas realizacji procesu obróbczego o ok. 10%.

Tab. 9 Wyniki badań - mediana kroków czasowych [A-4]

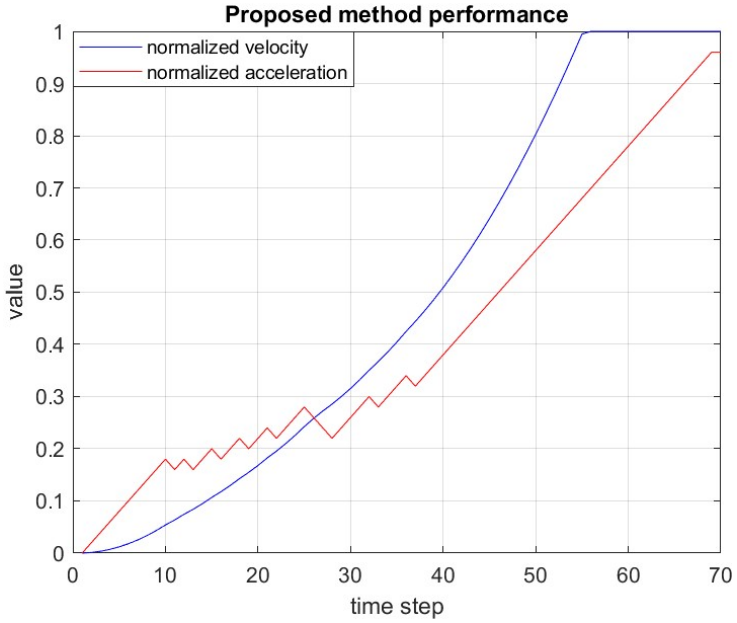
Steps Count		**Discount Factor**					
		0.99			**0.999**		
Ref. Points Density	**Network Arch. Number**	**Learning Rate**			**Learning Rate**		
		1e-3	**1e-4**	**1e-5**	**1e-3**	**1e-4**	**1e-5**
High	**1**	85	88	80	86	85	80
	2	83	83	81	83	82	82
	3	85	83	82	84	85	83
	4	82	82	80	84	80	82
	5	82	81	74	87	82	75
	6	83	84	69	82	84	72
	7	89	82	71	85	86	71
	8	82	83	80	81	88	81
	9	86	80	75	85	80	75
	10	83	81	80	83	83	82
	11	85	80	76	84	81	72
	12	84	80	1	83	80	1
Medium	**1**	471	469	476	179	71	101
	2	475	482	484	1	293	204
	3	478	482	446	73	1	101
	4	489	467	478	1	322	453
	5	482	465	431	1	1	101
	6	486	480	410	1	263	234
	7	487	482	406	87	1	101
	8	481	485	483	1	290	472
	9	488	486	367	159	82	109
	10	479	481	482	1	280	243
	11	476	471	487	68	1	258
	12	478	481	71	14	297	1

Autorzy [A-4], tak jak w poprzednich pracach, porównali też dokładny przebieg procesów obróbczych, a przykład takiego porównania został przedstawiony na Rys. 24 oraz Rys. 25. Analizując wykresy, można zauważyć, że ponownie proponowane

rozwiązanie stabilizuje proces obróbczy poprzez redukcję szarpnięć wywołanych naprzemiennym przyspieszaniem oraz spowalnianiem wrzeciona. Zauważyć należy również fakt, że dodatkowym pozytywnym aspektem proponowanych w pracach [A-6], [A-3] oraz [A-4] rozwiązań jest linearyzacja procesu generowania g-cod'u – podczas, gdy algorytm RPRO wielokrotnie cofa się do analizy poprzednich punktów referencyjnych, by wyznaczyć lepszy g-code, przedstawione rozwiązania generują jego kolejne kroki w sposób iteracyjny. Omówiona w tym rozdziale publikacja [A-4] realizuje Zadanie 3.c).



Rys. 24 Porównanie działania algorytmów – głębokie uczenie się ze wzmocnieniem [A-4]



Rys. 25 Porównanie działania algorytmów – algorytm RPRO [A-4]

3. Podsumowanie

Niniejsza praca koncentruje się na wykorzystaniu metod sztucznej inteligencji, a w szczególności algorytmu uczenia się ze wzmocnieniem, do zadań przetwarzania obrazów oraz do zadań sterowania dynamiką ruchu wrzeciona maszyn CNC. Postawiona hipoteza mówiąca, że *możliwa jest aplikacja różnych metod sztucznej inteligencji, a w szczególności algorytmu uczenia się ze wzmocnieniem, zarówno do zadań przetwarzania obrazu, jak i do zadań sterowania, celem uzyskania rezultatów nie gorszych niż przy pomocy innych metod znanych z literatury*, została uprawdopodobniona przez realizację następujących zadań.

Zadanie 1. Studia literaturowe dotyczące wykorzystania algorytmu uczenia się ze wzmocnieniem:

a. **do rozwiązywania zadań przetwarzania obrazów** – Zadanie zostało zrealizowanie poprzez przygotowanie przeglądu literatury [A-1], który został zaprezentowany podczas konferencji krajowej oraz przedstawiony w formie publikacji. Zgromadzona wiedza stanowiła inspirację do dalszych prac.

b. **do rozwiązywania zadań sterowania** – Zadanie zostało zrealizowanie poprzez przygotowanie przeglądu literatury [A-2], który został zaprezentowany podczas konferencji międzynarodowej oraz przedstawiony w formie publikacji. Zgromadzona wiedza stanowiła inspirację podczas dalszych prac.

Pierwsze z opracowań wykonane w języku polskim miało na celu upowszechnienie wiedzy o tego rodzaju zastosowaniach algorytmu uczenia się ze wzmocnieniem oraz o istnieniu samego algorytmu. Drugie opracowanie wprowadzało systematykę opisywanego zagadnienia oraz obejmowało identyfikację potencjalnych punktów rozwoju dziedziny.

Zadanie 2. Zebranie niezbędnych danych oraz utworzenie zbioru pozwalającego na trenowanie oraz weryfikowanie poprawności działania badanych metod:

a. **do zadania wykrywania gestów** – Zadanie zostało wykonane poprzez utworzenie dedykowanej bazy danych wykorzystanej podczas badań nad

propozycją algorytmu, którego zadaniem było wykrywanie gestów w ciągłym strumieniu wideo poprzez jego czasową segmentację [A-5].

b. **do zadania sterowania dynamiką ruchu wrzeciona maszyny CNC** – Zadanie zostało wykonane poprzez utworzenie dedykowanej bazy danych wykorzystanej podczas badań nad propozycją algorytmu, którego zadaniem była optymalizacja pracy maszyny CNC poprzez odpowiednie sterowanie dynamiką ruchu wrzeciona [A-6] [A-3] [A-4].

Zadanie 3. Zaproponowanie autorskiej metody pozwalającej na:

a. **czasową segmentację ciągłego strumienia gestów** – Zadanie zostało wykonane poprzez publikację artykułu [A-5] zawierającego opis proponowanej metody czasowej segmentacji strumienia gestów. W pracy zaproponowano również autorski sposób przetwarzania wstępnego klipów wideo w celu minimalizacji niekorzystnego wpływu szeregu czynników. W pracy wykorzystano zarówno głębokie sieci neuronowe jak również paradygmat uczenia się ze wzmocnieniem.

b. **optymalizację sterowania dynamiką ruchu wrzeciona maszyny CNC z wykorzystaniem logiki rozmytej** – Zadanie zostało wykonane poprzez publikację dwóch artykułów naukowych [A-6] [A-3], w których prezentowane są proponowane rozwiązania oparte na systemach eksperckich logiki rozmytej. W trakcie badań do nauki zbioru reguł wykorzystano też algorytm optymalizacji rojem cząstek oraz algorytm genetyczny. Wyniki badań zostały przedstawione w ramach konferencji.

c. **optymalizację sterowania dynamiką ruchu wrzeciona maszyny CNC z wykorzystaniem paradygmatu uczenia się ze wzmocnieniem** – Zadanie zostało wykonane poprzez publikację artykułu naukowego [A-4], w którym prezentowane jest proponowane rozwiązanie oparte na sieci neuronowej oraz na paradygmacie uczenia się ze wzmocnieniem. Wyniki badań zostały przedstawione w ramach konferencji.

Główny wkład autora rozprawy w działalność naukową w dyscyplinie Informatyka Techniczna i Telekomunikacja obejmują:

- przeprowadzenie przeglądu literatury w zakresie wykorzystania algorytmu uczenia się ze wzmocnieniem zarówno do zadań przetwarzania obrazu jak również do zadań sterowania przedstawiając aktualny stan wiedzy technicznej,

- identyfikację oraz sformułowanie potencjalnych punktów rozwoju opisywanych dziedzin,

- identyfikację oraz sformułowanie problemów badawczych, które są ważne z punktu widzenia zarówno zadania wykrywania gestów na podstawie ciągłego strumienia wideo jak również zadania optymalizacji sterowania dynamiką ruchu wrzeciona maszyny CNC,

- udział w opracowaniu dedykowanej bazy danych pozwalającej na trenowanie oraz weryfikację algorytmu czasowej segmentacji strumienia wideo uwzględniającej problem koartykulacji,

- opracowanie autorskiej metody wstępnego przetwarzania strumienia wideo złożonej z szeregu operacji wykonywanych na każdej z klatek filmu,

- opracowanie autorskiej metody pozwalającej na wykrywanie dynamicznych gestów polskiego języka migowego poprzez czasową segmentację strumienia wideo opartej o głęboką sieć splotową trenowaną algorytmem uczenia się ze wzmocnieniem,

- opracowanie dedykowanej bazy danych pozwalającej na trenowanie oraz weryfikację algorytmów optymalizujących pracę maszyny CNC poprzez sterowanie dynamiką ruchu wrzeciona,

- znaczny udział w opracowaniu autorskich metod optymalizujących pracę maszyny CNC poprzez sterowanie dynamiką ruchu wrzeciona,

- sformułowanie wniosków wynikających z przeprowadzonych eksperymentów oraz identyfikację dalszych kierunków rozwoju,

- znaczny udział w opracowaniu publikacji naukowych omawiających zagadnienia wymienione w niniejszej rozprawie.

Autor niniejszej rozprawy doktorskiej kontynuuje oraz rozszerza omówione w jej treści badania w następujący sposób:

- **Optymalizacja pracy maszyn CNC** – aktualnie finalizowane są prace badawcze mające na celu porównanie efektywności nauki reguł dla systemu Mamdaniego przy pomocy algorytmu genetycznego, a także efektywność pracy systemu po

konwersji na system Takani-Sugeno, co w zamierzeniu powinno znacząco zminimalizować czas potrzebny na wygenerowanie kodu sterującego. Dodatkowo w najbliższym czasie planowane jest zakończenie fazy konceptualizacji systemu opartego o algorytm uczenia się ze wzmocnieniem stosowany do sterowania wieloma osiami maszyny jednocześnie. Badania te będą również uwzględniały szeroko zakrojone porównanie dokładności oraz czasu obróbki półfabrykatu pomiędzy symulacją a rzeczywistym procesem.

- **Wykrywanie gestów** – aktualnie trwa proces organizowania zasobów niezbędnych do rozwoju utworzonej bazy danych GEST oraz planowanie integracji utworzonego rozwiązania z propozycją narzędzia rozpoznającego wyizolowane gesty. Celem prac jest utworzenie stabilnego systemu tłumaczącego nagranie na chmurę słów reprezentujących znaki pokazywane w materiale wideo. Docelowo planuje się utworzenie aplikacji mobilnej pozwalającej zredukować poziom wykluczenia społecznego osób głuchych. Dodatkowo na podstawie zidentyfikowanych w trakcie badań potencjalnych punktów rozwoju odkryto potrzebę skonstruowania bazy danych o charakterze benchmarkowym pozwalającej na rzetelną analizę proponowanych w literaturze algorytmów. Aktualne prace skupiają się na zdobyciu zasobów niezbędnych do wykonania rzeczonej bazy danych. Kolejnym krokiem będzie implementacja minimum 35 z istniejących metod i poddania ich głębokiej analizie.

- **Ocena jakości (zadania przetwarzania obrazów)** – aktualnie autor niniejszej rozprawy doktorskiej wraz z mgr. inż. Igorem Stępniem zaproponował system, którego zadaniem jest ocena jakości przedstawionego nagrania wideo. Nauka oraz weryfikacja systemu zostały przeprowadzone z wykorzystaniem bazy danych LIVE-YT-Gaming [82] zawierającej nagrania streamów gier komputerowych. Algorytm znajduje potencjalne zastosowanie jako system wstępnej weryfikacji filmów dla platform streamingowych. Aktualnie praca opisująca badania została zgłoszona do recenzji. Dodatkowo w tym samym zespole został zaprojektowany algorytm oceny jakości zdjęć pochodzących z rezonansu magnetycznego [83][84]. Dotychczasowe badania wykazują współczynnik korelacji Pearson'a na poziomie niemal 0.97, co wskazuje na niezwykle wysoką zgodność z opiniami ekspertów. Aktualnie prowadzone są dalsze eksperymenty, a w niedługim czasie opracowana zostanie treść publikacji.

Literatura

[1] A. M. Turing, *Computing machinery and intelligence*. Springer, 2009.

[2] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Commun ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[3] W. Van Melle, "MYCIN: a knowledge-based consultation program for infectious disease diagnosis," *Int J Man Mach Stud*, vol. 10, no. 3, pp. 313–322, 1978.

[4] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Adv Neural Inf Process Syst*, vol. 1, 1988.

[5] L. Watson, "Net Talk.," *Vocational Education Journal*, vol. 69, no. 6, p. 41, 1994.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv Neural Inf Process Syst*, vol. 25, 2012.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[9] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] M. Z. Alom *et al.*, “The history began from alexnet: A comprehensive survey on deep learning approaches,” *arXiv preprint arXiv:1803.01164*, 2018.

[14] T. Shanthi and R. S. Sabeenian, “Modified Alexnet architecture for classification of diabetic retinopathy images,” *Computers & Electrical Engineering*, vol. 76, pp. 56–64, 2019.

[15] H. Ke, D. Chen, X. Li, Y. Tang, T. Shah, and R. Ranjan, “Towards brain big data classification: Epileptic EEG identification with a lightweight VGGNet on global MIC,” *IEEE Access*, vol. 6, pp. 14722–14733, 2018.

[16] L.-D. Quach, N. Pham-Quoc, D. C. Tran, and M. Fadzil Hassan, “Identification of chicken diseases using VGGNet and ResNet models,” in *International Conference on Industrial Networks and Intelligent Systems*, Springer, 2020, pp. 259–269.

[17] J. Ni, J. Gao, L. Deng, and Z. Han, “Monitoring the change process of banana freshness by GoogLeNet,” *IEEE Access*, vol. 8, pp. 228369–228376, 2020.

[18] R. Almodfer, S. Xiong, M. Mudhsh, and P. Duan, “Enhancing AlexNet for arabic handwritten words recognition using incremental dropout,” in *2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI)*, IEEE, 2017, pp. 663–669.

[19] R. Ozdemir and M. Koc, “A quality control application on a smart factory prototype using deep learning methods,” in *2019 IEEE 14th international conference on computer sciences and information technologies (CSIT)*, IEEE, 2019, pp. 46–49.

[20] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, “Deep learning algorithm for autonomous driving using googlenet,” in *2017 IEEE intelligent vehicles symposium (IV)*, IEEE, 2017, pp. 89–96.

[21] P. Salavati and H. M. Mohammadi, “Obstacle detection using GoogleNet,” in *2018 8th international conference on computer and knowledge engineering (ICCKE)*, IEEE, 2018, pp. 326–332.

[22] A. Abd Almisreb, N. Jamil, and N. M. Din, “Utilizing AlexNet deep transfer learning for ear recognition,” in *2018 fourth international conference on information retrieval and knowledge management (CAMP)*, IEEE, 2018, pp. 1–5.

[23] K. Sathish, S. Ramasubbareddy, and K. Govinda, “Detection and localization of multiple objects using VGGNet and single shot detection,” in *Emerging Research in Data Engineering Systems and Computer Communications: Proceedings of CCODE 2019*, Springer, 2020, pp. 427–439.

[24] M. F. Haque, H.-Y. Lim, and D.-S. Kang, “Object detection based on VGG with ResNet network,” in *2019 International conference on electronics, information, and communication (ICEIC)*, IEEE, 2019, pp. 1–3.

[25] Y. Zhiqi, “Gesture recognition based on improved VGGNET convolutional neural network,” in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, 2020, pp. 1736–1739.

[26] I. Goodfellow *et al.*, “Generative adversarial nets,” *Adv Neural Inf Process Syst*, vol. 27, 2014.

[27] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[28] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[29] G. Lample and D. S. Chaplot, “Playing FPS games with deep reinforcement learning,” in *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[30] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Process Mag*, vol. 34, no. 6, pp. 26–38, 2017.

[31] Y. Li, “Deep reinforcement learning: An overview,” *arXiv preprint arXiv:1701.07274*, 2017.

[32] S. S. Mousavi, M. Schukat, and E. Howley, “Deep reinforcement learning: an overview,” in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2*, Springer, 2018, pp. 426–440.

[33] R. Furuta, N. Inoue, and T. Yamasaki, “PixelRL: Fully convolutional network with reinforcement learning for image processing,” *IEEE Trans Multimedia*, vol. 22, no. 7, pp. 1704–1719, 2019.

[34] J. Park, J.-Y. Lee, D. Yoo, and I. S. Kweon, “Distort-and-recover: Color enhancement using deep reinforcement learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5928–5936.

[35] J. C. Caicedo and S. Lazebnik, “Active object localization with deep reinforcement learning,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2488–2496.

[36] X. Kong, B. Xin, Y. Wang, and G. Hua, “Collaborative deep reinforcement learning for joint object search,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1695–1704.

[37] Y. Xiang, A. Alahi, and S. Savarese, “Learning to track: Online multi-object tracking by decision making,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4705–4713.

[38] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi, “Action-decision networks for visual tracking with deep reinforcement learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2711–2720.

[39] T. Darrell, “Reinforcement learning of active recognition behaviors,” *Portions of this paper previously appeared in Advances in Neural Information Processing Systems (NIPS 1995)*, vol. 8, no. 1997, pp. 73–80, 1997.

[40] D. Silver *et al.*, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[41] D. Silver *et al.*, “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[42] D. Silver *et al.*, “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[43] K. Yu, C. Dong, L. Lin, and C. C. Loy, “Crafting a toolchain for image restoration by deep reinforcement learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2443–2452.

[44] S. Kosugi and T. Yamasaki, “Unpaired image enhancement featuring reinforcement-learning-controlled image editing software,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 11296–11303.

[45] C. Shen, Y. Gonzalez, L. Chen, S. B. Jiang, and X. Jia, “Intelligent parameter tuning in optimization-based iterative CT reconstruction via deep reinforcement learning,” *IEEE Trans Med Imaging*, vol. 37, no. 6, pp. 1430–1439, 2018.

[46] W. Li, X. Feng, H. An, X. Y. Ng, and Y.-J. Zhang, “MRI reconstruction with interpretable pixel-wise operations using reinforcement learning,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 792–799.

[47] K. Vassilo, C. Heatwole, T. Taha, and A. Mehmood, “Multi-step reinforcement learning for single image super-resolution,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 512–513.

[48] W.-C. Hung, J. Zhang, X. Shen, Z. Lin, J.-Y. Lee, and M.-H. Yang, “Learning to blend photos,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 70–86.

[49] S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell, “Timely object recognition,” *Adv Neural Inf Process Syst*, vol. 25, 2012.

[50] S. Mathe, A. Pirinen, and C. Sminchisescu, “Reinforcement learning for visual object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2894–2902.

[51] J. Supancic III and D. Ramanan, “Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 322–331.

[52] T. Darrell and A. Pentland, “Active gesture recognition using partially observable Markov decision processes,” in *Proceedings of 13th International Conference on Pattern Recognition*, IEEE, 1996, pp. 984–988.

[53] Y. Rao, J. Lu, and J. Zhou, “Attention-aware deep reinforcement learning for video face recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3931–3940.

[54] G. Yu and I. K. Sethi, “Road-following with continuous learning,” in *Proceedings of the Intelligent Vehicles’ 95. Symposium*, IEEE, 1995, pp. 412–417.

[55] H. Yang, B. Wang, N. Vesdapunt, M. Guo, and S. B. Kang, “Personalized exposure control using adaptive metering and reinforcement learning,” *IEEE Trans Vis Comput Graph*, vol. 25, no. 10, pp. 2953–2968, 2018.

[56] S. Lan, R. Panda, Q. Zhu, and A. K. Roy-Chowdhury, “Ffnet: Video fast-forwarding via reinforcement learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6771–6780.

[57] X. Han *et al.*, “Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 234–243.

[58] S. Ahmed, F. Khan, A. Ghaffar, F. Hussain, and S. H. Cho, “Finger-counting-based gesture recognition within cars using impulse radar with convolutional neural network,” *Sensors*, vol. 19, no. 6, p. 1429, 2019.

[59] Y. Zhu, G. Xu, and D. J. Kriegman, “A real-time approach to the spotting, representation, and recognition of hand gestures for human–computer interaction,” *Computer Vision and Image Understanding*, vol. 85, no. 3, pp. 189–208, 2002.

[60] M. Elmezain, A. Al-Hamadi, and B. Michaelis, “Hand trajectory-based gesture spotting and recognition using HMM,” in *2009 16th IEEE international conference on image processing (ICIP)*, IEEE, 2009, pp. 3577–3580.

[61] P. Neto, D. Pereira, J. N. Pires, and A. P. Moreira, “Real-time and continuous hand gesture spotting: An approach based on artificial neural networks,” in *2013 IEEE international conference on robotics and automation*, IEEE, 2013, pp. 178–183.

[62] E. Tsironi, P. Barros, C. Weber, and S. Wermter, “An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition,” *Neurocomputing*, vol. 268, pp. 76–86, 2017.

[63] T. Kapuscinski and M. Wysocki, “Recognition of signed expressions in an experimental system supporting deaf clients in the city office,” *Sensors*, vol. 20, no. 8, p. 2190, 2020.

[64] Z. Zhang, J. Pu, L. Zhuang, W. Zhou, and H. Li, “Continuous sign language recognition via reinforcement learning,” in *2019 IEEE international conference on image processing (ICIP)*, IEEE, 2019, pp. 285–289.

[65] W. Seok, Y. Kim, and C. Park, “Pattern recognition of human arm movement using deep reinforcement learning,” in *2018 International Conference on Information Networking (ICOIN)*, IEEE, 2018, pp. 917–919.

[66] S. Anwar, S. K. Sinha, S. Vivek, and V. Ashank, “Hand gesture recognition: a survey,” in *Nanoelectronics, Circuits and Communication Systems: Proceeding of NCCS 2017*, Springer, 2019, pp. 365–371.

[67] S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[68] P. Trigueiros, F. Ribeiro, and L. P. Reis, “A comparison of machine learning algorithms applied to hand gesture recognition,” in *7th Iberian conference on information systems and technologies (CISTI 2012)*, IEEE, 2012, pp. 1–6.

[69] H. S. Hasan and S. A. Kareem, “Human computer interaction for vision based hand gesture recognition: A survey,” in *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, IEEE, 2012, pp. 55–60.

[70] A. R. Sarkar, G. Sanyal, and S. Majumder, “Hand gesture recognition systems: a survey,” *Int J Comput Appl*, vol. 71, no. 15, 2013.

[71] S. Singh, A. K. Gupta, and T. Singh, “Computer vision based hand gesture recognition: A survey,” *Int. J. Comput. Sci. Eng*, vol. 7, no. 5, pp. 548–556, 2019.

[72] P. K. Pisharady and M. Saerbeck, “Recent methods and databases in vision-based hand gesture recognition: A review,” *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015.

[73] K. M. Sagayam and D. J. Hemanth, “Hand posture and gesture recognition techniques for virtual reality applications: a survey,” *Virtual Real*, vol. 21, pp. 91–107, 2017.

[74] D. Sarma and M. K. Bhuyan, “Methods, databases and recent advancement of vision-based hand gesture recognition for hci systems: A review,” *SN Comput Sci*, vol. 2, no. 6, p. 436, 2021.

[75] D. H. Neiva and C. Zanchettin, “Gesture recognition: A review focusing on sign language in a mobile context,” *Expert Syst Appl*, vol. 103, pp. 159–183, 2018.

[76] N. Aloysius and M. Geetha, “Understanding vision-based continuous sign language recognition,” *Multimed Tools Appl*, vol. 79, no. 31, pp. 22177–22209, 2020.

[77] R. Jain, R. K. Karsh, and A. A. Barbhuiya, “Literature review of vision-based dynamic gesture recognition using deep learning techniques,” *Concurr Comput*, vol. 34, no. 22, p. e7159, 2022.

[78] S. Ruffieux, D. Lalanne, E. Mugellini, and O. Abou Khaled, “A survey of datasets for human gesture recognition,” in *Human-Computer Interaction. Advanced Interaction Modalities and Techniques: 16th International Conference, HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014, Proceedings, Part II 16*, Springer, 2014, pp. 337–348.

[79] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Proceedings of the AAAI conference on artificial intelligence*, 2016.

[80] B. Kwiatkowski, T. Kwater, D. Mazur, and J. Bartman, “An offline application that determines themaximum accuracy of the realization ofreference points from G-code for givenparameters of CNC machine dynamics,” *Bulletin of the Polish Academy of Sciences Technical Sciences*, pp. e147345–e147345.

[81] M. Gavlas, M. Drbul, V. Dekys, and M. Saga, “Effect of Vibration on Machine Tool Accuracy and Lifetime,” in *MATEC Web of Conferences*, EDP Sciences, 2022, p. 05003.

[82] X. Yu, Z. Ying, N. Birkbeck, Y. Wang, B. Adsumilli, and A. C. Bovik, “Subjective and objective analysis of streamed gaming videos,” *IEEE Trans Games*, 2023.

[83] M. Oszust, A. Piórkowski, and R. Obuchowicz, “No-reference image quality assessment of magnetic resonance images with high-boost filtering and local features,” *Magn Reson Med*, vol. 84, no. 3, pp. 1648–1660, 2020.

[84] I. Stpień, R. Obuchowicz, A. Piórkowski, and M. Oszust, “Fusion of deep convolutional neural networks for no-reference magnetic resonance image quality assessment,” *Sensors*, vol. 21, no. 4, p. 1043, 2021.

Dorobek naukowy autora

W niniejszej rozprawie omówiono wybrane z publikacji autora. Ten rozdział zawiera informacje o całym aktualnym dorobku naukowym autora dysertacji. Dodatkowo autor ten pełnił rolę recenzenta materiałów w czasopiśmie Journal of Sensors, a także materiałów konferencyjnych.

Artykuły naukowe

[A-1] Kalandyk, D. (2021). Rozdział monografii „Nowoczesne technologie – strategie, rozwiązania i perspektywy rozwoju. Tom 1" pt. „Wykorzystanie algorytmów uczenia się ze wzmocnieniem do przetwarzania obrazów", str. 180-223; http://bc.wydawnictwo-tygiel.pl/publikacja/1A77370A-0E87-0A33-F834-334902710840; wkład: 100%, (autor korespondencyjny)

[A-2] Kalandyk, D. (2021). Reinforcement learning in car control: A brief survey. 2021 Selected Issues of Electrical Engineering and Electronics (WZEE), 1-8. https://doi.org/10.1109/WZEE54157.2021.9576838; wkład: 100%; (autor korespondencyjny)

[A-3] Kalandyk, D., Kwiatkowski, B., & Mazur, D. (2023, August). Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics. In 2023 IEEE International Conference on Fuzzy Systems (FUZZ) (pp. 1-4). IEEE. https://doi.org/10.1109/FUZZ52849.2023.10309802; wkład 70%; liczba punktów czerwiec 2023: 140; liczba punktów lipiec 2024: 70; (autor korespondencyjny)

[A-4] Kalandyk, D., Kwiatkowski, B., & Mazur, D. CNC Machine Control Using Deep Reinforcement Learning. Bulletin of the Polish Academy of Sciences Technical Sciences, e148940-e148940. https://doi.org/10.24425/bpasts.2024.148940; wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.8

[A-5] Kalandyk, D., & Kapuściński, T. (2024). Temporal signed gestures segmentation in an image sequence using deep reinforcement learning.

Engineering Applications of Artificial Intelligence, 131, 107879. https://doi.org/10.1016/j.engappai.2024.107879; wkład 90%; liczba punktów: 140; IF: 8.0; CS: 12.3; (autor korespondencyjny)

[A-6] Kalandyk, D., Kwiatkowski, B., & Mazur, D. Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System. Archives of Control Sciences, artykuł przyjęty do publikacji, wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.7; (autor korespondencyjny)

[A-7] Shchur, I., Mazur, D., Makarchuk, O., Bilyakovskyy, I., Turkovskyi, V., Kwiatkowski, B., & Kalandyk, D. (2022). Improved Matlab/Simulink model of dual three-phase fractional slot and concentrated winding PM motor for EV applied brushless DC drive. Archives of Control Sciences, 677-707. wkład 14.3%; liczba punktów: 100; IF: 1.2; CS: 2.7

Wystąpienia konferencyjne

[K-1] Kalandyk, D. *"Wykorzystanie algorytmów uczenia się ze wzmocnieniem do przetwarzania obrazów"*, XIII Interdyscyplinarna Konferencja Naukowa TYGIEL 2021 „Interdyscyplinarność kluczem do rozwoju", marzec 2021, Lublin

[K-2] Kalandyk, D. *"Reinforcement learning in car control: A brief survey"*, 2021 Selected Issues of Electrical Engineering and Electronics (WZEE), wrzesień 2021, Rzeszów, przedstawiona praca została wyróżniona, **praca w komitecie organizacyjnym**

[K-3] Kalandyk, D. *"Uczenie się ze wzmocnieniem jako alternatywne podejście do rozwiązywania trudnych zadań"*, Warszawskie Dni Informatyki 2022, kwiecień 2022, Warszawa

[K-4] Kalandyk, D. Kwiatkowski, B. Mazur, M. *"Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System"*, XI Konferencja Naukowa pt. „Symbioza Techniki i Informatyki", czerwiec 2023, Kiry

[K-5] Kalandyk, D. Kwiatkowski, B. Mazur, M. *"Application of Fuzzy Logic to optimise CNC Machine motion dynamics"*, 2023 IEEE International Conference on Fuzzy Systems, sierpień 2023, Songdo Incheon, Korea Południowa

[K-6] Kalandyk, D. Kwiatkowski, B. Mazur, M. *"Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System"*, Konferencja Postępy w Elektrotechnice Stosowanej PES - 17, sierpień 2023, Kiry

[K-7] Kalandyk, D. Stępień, I. *"Reinforcement Learning with Hand Crafted and Deep Learning features for Video Quality Assessment Approach"*, Selected Issues in Power Engineering, Electrical Engineering and Industry 4.0, grudzień 2023, Rzeszów, **praca w komitecie organizacyjnym**

[K-8] Kalandyk, D. Sołtys, K. Ciećko, K. *"Propozycja inteligentnego urządzenia rehabilitacji ruchowej"*, IV Konferencja Kół Naukowych w ramach Politechnicznej sieci Via Carpatia im. Prezydenta Lecha Kaczyńskiego, maj 2024, Lublin

[K-9] Kalandyk, D. *"Application of Fuzzy Inference Systems to generate g-code optimizing Computerized Numerical Control machine motion dynamics*, Postępy w Elektrotechnice Stosowanej PES-18, czerwiec 2024, Kościelisko

Artykuły naukowe wchodzące w skład cyklu (opublikowane w latach 2020-2024)

W niniejszym rozdziale zamieszczono pełną treść opublikowanych prac wchodzących w skład cyklu publikacji:

[A-1] Kalandyk, D. (2021). Rozdział monografii „Nowoczesne technologie – strategie, rozwiązania i perspektywy rozwoju. Tom 1" pt. „Wykorzystanie algorytmów uczenia się ze wzmocnieniem do przetwarzania obrazów", str. 180-223; http://bc.wydawnictwo-tygiel.pl/publikacja/1A77370A-0E87-0A33-F834-334902710840; wkład: 100%, (autor korespondencyjny)

[A-2] Kalandyk, D. (2021). Reinforcement learning in car control: A brief survey. 2021 Selected Issues of Electrical Engineering and Electronics (WZEE), 1-8. https://doi.org/10.1109/WZEE54157.2021.9576838; wkład: 100%; (autor korespondencyjny)

[A-3] Kalandyk, D., Kwiatkowski, B., & Mazur, D. (2023, August). Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics. In 2023 IEEE International Conference on Fuzzy Systems (FUZZ) (pp. 1-4). IEEE. https://doi.org/10.1109/FUZZ52849.2023.10309802; wkład 70%; liczba punktów czerwiec 2023: 140; liczba punktów lipiec 2024: 70; (autor korespondencyjny)

[A-4] Kalandyk, D., Kwiatkowski, B., & Mazur, D. CNC Machine Control Using Deep Reinforcement Learning. Bulletin of the Polish Academy of Sciences Technical Sciences, e148940-e148940. https://doi.org/10.24425/bpasts.2024.148940; wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.8

[A-5] Kalandyk, D., & Kapuściński, T. (2024). Temporal signed gestures segmentation in an image sequence using deep reinforcement learning. Engineering Applications of Artificial Intelligence, 131, 107879. https://doi.org/10.1016/j.engappai.2024.107879; wkład 90%; liczba punktów: 140; IF: 8.0; CS: 12.3; (autor korespondencyjny)

[A-6] Kalandyk, D., Kwiatkowski, B., & Mazur, D. Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System. Archives of Control Sciences, artykuł przyjęty do publikacji, wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.7; (autor korespondencyjny)

Dawid Kalandyk[1]

Wykorzystanie algorytmów uczenia się ze wzmocnieniem do przetwarzania obrazów

1. Wprowadzenie

Ostatnią dekadę można nazwać czasem dynamicznego rozwoju wizji komputerowej. Wprowadzone technologie, wykorzystywane przez marki takie, jak: Google Glass (2010) czy Kinect (2010) służą nie tylko rozrywce, ale także mogą pomagać w nauce, np. poprzez symulacje wycieczek po muzeach bądź zapewnienie zróżnicowanej aktywności ruchowej. Ponadto rozwijane są też systemy inteligentnego sterowania autonomicznymi pojazdami poziomu 5, czyli pojazdy nieposiadające przyrządów sterowania manualnego, lecz będące w pełni autonomiczne – realizujące swoje funkcje automatycznie. Jednym z kluczowych problemów trapiących naukowców rozwijających wymienione technologie jest efektywne przetwarzanie obrazów. Naturalnie, oprócz klasycznych metod analizy i przetwarzania obrazów, istnieją też podejścia wykorzystujące sztuczną inteligencję oraz uczenie maszynowe. Głównymi narzędziami sztucznej inteligencji, analizującymi obrazy, są głębokie sieci neuronowe oraz wielowarstwowe splotowe sieci neuronowe. Najczęściej służą one do klasyfikacji obrazów oraz ekstrakcji interesujących cech obrazu. Wykorzystujący je paradygmat uczenia się ze wzmocnieniem, a także konkretne przykłady przetwarzania obrazu z jego pomocą, zostaną omówione w kolejnych rozdziałach. Celem niniejszego rozdziału jest zaprezentowanie różnych sposobów aplikacji algorytmu uczenia się ze wzmocnieniem do zadań przetwarzania obrazów. Ponadto wprowadzona zostanie pewna systematyka omówionych rozwiązań wyróżniająca dwie podstawowe grupy. Do pierwszej grupy zadań modyfikujących obraz należą: redukcja szumów, korekcja kolorów, odzyskiwanie bloków obrazu, konwersja obrazów do tzw. super rozdzielczości potocznie zwana poprawą rozdzielczości obrazu, a także scalanie obrazów. Druga opisana grupa to algorytmy, których zadaniem jest detekcja oraz śledzenie obiektów. Ponadto rozważone zostaną również przykłady algorytmów niewpisujących się do żadnej z wymienionych grup. Rozwiązania te są jednak ciekawe z uwagi na sposób analizy obrazów prowadzący do ekstrakcji pożądanych cech, a w efekcie do osiągnięcia strategii optymalnej.

2. Uczenie się ze wzmocnieniem

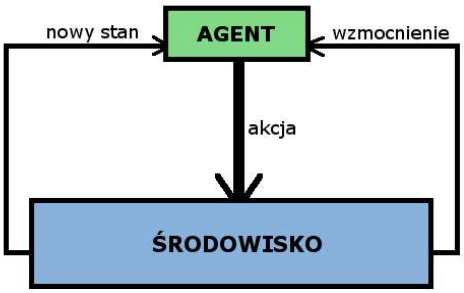
Podstawowym sposobem nauki (zwanym uczeniem z nauczycielem) jest oglądanie wzorców, a następnie próba ich odtworzenia. Dzięki porównaniu wartości wytworzonych z wartościami wzorcowymi system uczący się (uczeń) ma możliwość poprawy swojego zachowania, co prowadzi go do osiągnięcia efektów bliskich wzorcom. Alternatywnym podejściem jest algorytm uczenia się ze wzmocnieniem. W tym modelu agent pragnący wypracować strategię optymalną w trakcie interakcji ze środowiskiem, w którym się porusza, otrzymuje sygnały wzmocnienia od tzw. krytyka,

[1] dawid.kalandyk@gmail.com, Szkoła Doktorska Nauk Inżynieryjno-Technicznych na Politechnice Rzeszowskiej.

będącego często integralną częścią środowiska. Agent, znajdując się w stanie s_1, podejmuje jedną z możliwych do wykonania w tym stanie akcji ze zbioru A. Następnie agent obserwuje nowy stan s_2, w którym się znalazł, a także wzmocnienie towarzyszące podjętej akcji. Sygnał ten pełni rolę sprzężenia zwrotnego informującego agenta o jakości akcji, którą ten wykonał. Jak nietrudno zauważyć, wzmocnienie nie informuje agenta o tym, jak daleko znalazł się on od poprawnego rozwiązania, a jedynie pełni rolę wartościującą. W omówionym modelu agent nigdy nie musi znać prawidłowego rozwiązania problemu. Jego zachowanie można porównać do dziecka otwierającego prezent – każda nieskuteczna próba nie pozostaje tylko stratą czasu, ale przynosi wiedzę, którą można wykorzystać przy kolejnym zetknięciu się z podobnym problemem. Można zatem łatwo zauważyć, że, gdy badany problem można rozwiązać metodą prób i błędów, stosowanie algorytmu uczenia się ze wzmocnieniem jest jak najbardziej wskazane. Dokładny sposób działania agenta został przedstawiony na listingu 1, natomiast ilustracja modelu opisanego procesu została przedstawiona na rysunku 1 [1, 2].

```
1   zainicjuj strategię losowo
2   dopóki nie osiągnięto wyznaczonego celu wykonuj
3           wybierz akcję do wykonania
4           wykonaj wybraną akcję
5           obserwuj wzmocnienie i następny stan
6           modyfikuj strategię
7   jeżeli nie wyznaczono strategii optymalnej to
8           wróć do stanu początkowego
9           przejdź do kroku 2
```

Listing 1. Schemat procesu nauki agenta, [opracowanie własne]



Rysunek 1. Model interakcji agenta ze środowiskiem, [opracowanie własne]

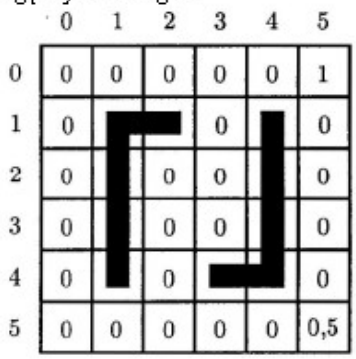
Kolejne akcje podejmowane przez agenta można utożsamić z dyskretnymi krokami czasowymi, natomiast ciąg akcji prowadzący w efekcie do osiągnięcia stanu końcowego zwykło się nazywać epizodem. Naturalną wydaje się myśl, by móc wyposażyć agenta w mechanizm pozwalający sterować jego tzw. „dalekowzrocznością". Chodzi tu o zdolność agenta do rozważania daleko idących skutków aktualnie podejmowanej akcji. Rolę taką pełni współczynnik dyskontowania γ. Dzięki niemu agent jest w stanie obliczyć oczekiwaną sumę nagród, które zbierze, posługując się daną strategią π: $S \rightarrow A$, będącą w rzeczywistości funkcją wybierającą akcję do wykonania w danym stanie. Wykorzystując współczynnik dyskontowania oraz wartości otrzymywanych przez

agenta wzmocnień, można zdefiniować nie tylko funkcję wartości V, określoną przez oczekiwaną zdyskontowaną sumę wzmocnień określoną dla każdego stanu, ale także funkcję wartości akcji Q, określoną poprzez oczekiwaną zdyskontowaną sumę wzmocnień otrzymanych przez agenta od momentu podjęcia określonej akcji w danym stanie. Wzory (1) oraz (2) opisują wspomniane funkcje.

$$V^{\pi}(s) = E_{\pi}\left[\sum_{t=0}^{t_{max}} (\gamma^{t} \cdot r_t) \;\; dla\, s_0 = s\right] \tag{1}$$

$$Q^{\pi}(s,a) = E_{\pi}\left[r_0 + \sum_{t=1}^{t_{max}} (\gamma^{t} \cdot r_t) \;\; dla\, s_0 = s, a_0 = a\right] \tag{2}$$

Przykładowym środowiskiem, w którym porusza się agent, może być przedstawiony na rysunku 2. prosty labirynt. W każdym polu widnieje liczba będąca wartością wzmocnienia otrzymywanego przez agenta w momencie, gdy ten znajdzie się w danym miejscu. Przy założeniu punktu startowego agenta w lewym dolnym rogu oraz jego celu – osiągnięcia komórki w prawym górnym bądź prawym dolnym rogu, można nakreślić prosty schemat procesu nauki. Epizodem określane będzie 30 kroków agenta wykonane przez niego od punktu początkowego. Epizod kończył się będzie po osiągnięciu pozycji o współrzędnych (0,5) lub (5,5) albo po przekroczeniu wspomnianych 30 kroków. Zgodnie z zasadą dyskontowania wzmocnienie w kroku t ma wartość $r_t = \gamma^t r_{ij}$. Jak nietrudno zauważyć, im dłużej agent będzie ociągał się z dotarciem do wyjścia, tym mniejszą zdyskontowaną sumę wzmocnień otrzyma. Drugim, szczególnie ważnym faktem, jest dobór wartości współczynnika dyskontującego, gdyż to od niego zależy, jaką drogę wybierze agent.



Rysunek 2. Środowisko komórkowe – labirynt [1]

3. Algorytmy modyfikujące obraz

Mimo rozwoju systemów rejestracji obrazu oraz urządzeń wizji komputerowej, wciąż zachodzi potrzeba szybkiej poprawy zgromadzonych obrazów. Korekta ta może być niezbędna, w celu umożliwienia lub ułatwienia odczytania z obrazu pewnych informacji, bądź też może mieć charakter czysto estetyczny. W pracy [3] z 2018 roku autorzy podsumowali rozwój nowej gałęzi uczenia maszynowego – algorytmy skupiające się na rekonstrukcji obrazów. W licznej grupie różnych rozwiązań znalazło się tylko jedno wykorzystujące paradygmat uczenia się ze wzmocnieniem. W roku 2018 oraz latach kolejnych powstały nowe, ciekawe rozwiązania, również zasługujące

na uwagę. Niniejszy rozdział poświęcony będzie więc grupie algorytmów korzystających z paradygmatu uczenia się ze wzmocnieniem, których zadaniem jest różnego rodzaju modyfikacja obrazu tak, aby podnieść jego jakość lub poprawić walory czysto estetyczne. W kolejnych podrozdziałach przedstawiono wybrane zadania, tj. redukcja szumów, korekcja kolorów, odzyskiwanie bloków, wyostrzanie oraz scalanie obrazów.

3.1. Redukcja szumów

Pierwszą z omawianych metod przetwarzania obrazów jest redukcja lub całkowita eliminacja zakłóceń oraz szumów. Tradycyjne sposoby polegają na zastosowaniu jednego z dostępnych filtrów lub też kilkukrotne ich użycie, aż do uzyskania zamierzonego efektu. Istnieją również podejścia wykorzystujące sztuczne sieci neuronowe m. in. zaproponowane w pracy [4] w 2012 roku przez badaczy Xie, Xu oraz Chen oparte o zwielokrotnione warstwy DA (ang. *Denoising Auto-encoder*). W artykule [5] opublikowanym w 2019 roku Furuta, Inoue oraz Yamasaki przedstawili podejście wykorzystujące algorytm uczenia się ze wzmocnieniem. Zakłada ono wykonywanie jednej z wymienionych w tabeli 1. operacji niezależnie dla każdego piksela obrazu. W modelu tym zakłada się istnienie liczby agentów równej liczbie pikseli obrazu. Ponadto funkcja wartości oraz funkcja wartości akcji są aproksymowane za pomocą neuronowych sieci splotowych, odpowiednio sieć wartości (ang. *Value network*) oraz sieć polityki (ang. *Policy network*), będącej reprezentacją schematu decyzyjnego agenta. Ilustrację opisanego modelu nazwanego przez autorów PixelRL przedstawiono na rysunku 3. Wzmocnienie dla każdego agenta (piksela) w kolejnych krokach jest obliczane zgodnie ze wzorem (3).

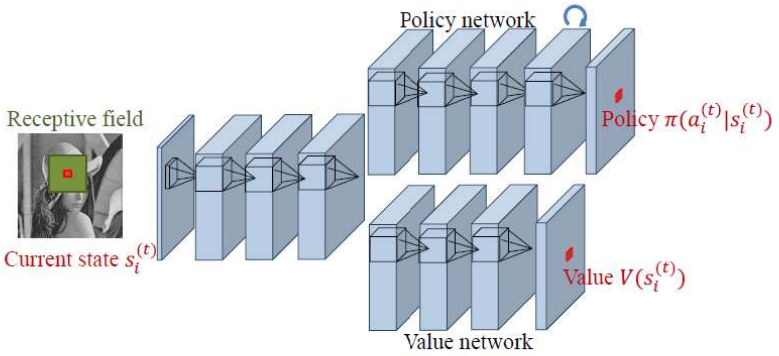
$$r_i^{(t)} = \left(I_i^{target} - s_i^{(t)}\right)^2 - \left(I_i^{target} - s_i^{(t+1)}\right)^2 \qquad (3)$$

gdzie:

- $r_i^{(t)}$ – wzmocnienie przydzielone i-temu agentowi w kroku t
- I_i^{target} – wartość docelowa i-tego piksela
- $s_i^{(t)}$ – wartość i-tego piksela wyznaczona przez agenta w kroku t

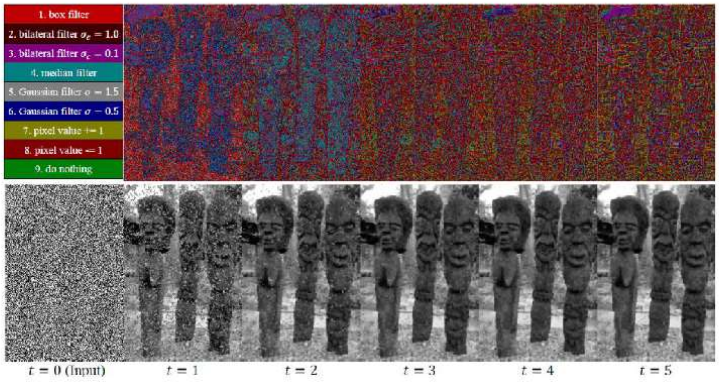
Tabela 2. PixelRL – opis dostępnych akcji [5]

numer akcji	akcja	rozmiar filtra	parametry filtra
1	filtr pudełkowy	5x5	-
2	filtr bilateralny	5x5	$\sigma_c = 1.0$, $\sigma_S = 5.0$
3	filtr bilateralny	5x5	$\sigma_c = 0.1$, $\sigma_S = 5.0$
4	filtr medianowy	5x5	-
5	filtr Gaussa	5x5	$\sigma = 1.5$
6	filtr Gaussa	5x5	$\sigma = 0.5$
7	wartość piksela +1	-	-
8	wartość piksela -1	-	-
9	brak akcji	-	-



Rysunek 3. PixelRL – schemat budowy sieci głębokiej [5]

Przykład skuteczności działania omówionego algorytmu został zilustrowany na rysunku 4. Przedstawia on obraz wejściowy oraz efekt działania agentów po każdym z pięciu kroków. Dodatkowo przy użyciu kolorów przedstawione zostały akcje wykonane przez każdego agena, co pokazuje niezrównaną zaletę metody, jaką jest interpretowalność jej pracy. Obraz wejściowy jest wynikiem aplikacji szumu typu sól i pieprz o gęstości 0.9.



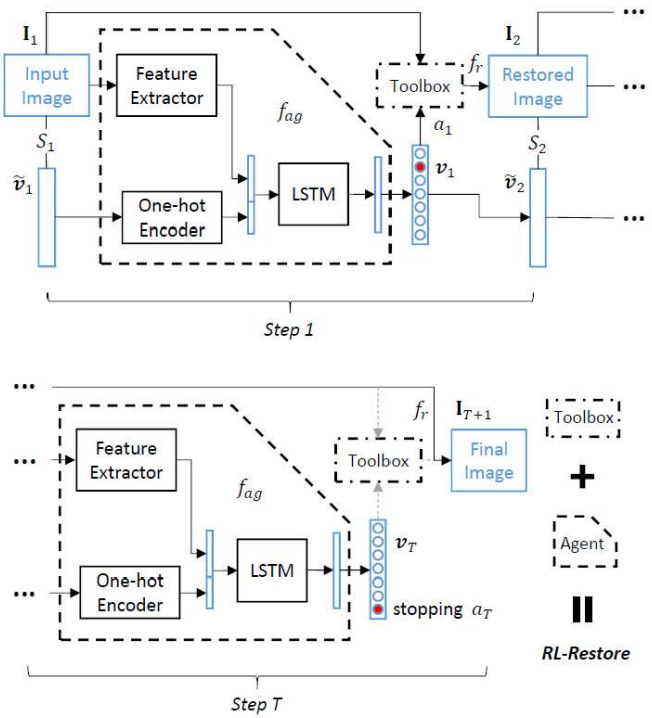
Rysunek 4. PixelRL - przykład działania (redukcja szumów) [5]

Odmiennym podejściem jest algorytm zaproponowany rok wcześniej w [6] przez Yu i in. Przyjmuje on istnienie agenta modyfikującego obraz sekwencyjnie. Agent ten ma do dyspozycji odpowiednio określony koszyk narzędzi zawierający stosowne filtry. Filtry te są różnej głębokości sieciami konwolucyjnymi służącymi redukcji efektu szumu Gaussa, efektu rozmycia Gaussa, bądź zniekształceń spowodowanych operacją kompresji JPEG. Informacje na temat możliwych do podjęcia przez agenta akcji przedstawione zostały w tabeli 2. Należy zauważyć, że do redukcji zakłóceń o większej sile autorzy zaproponowali zdecydowanie głębsze sieci konwolucyjne pozwalające na dokładniejszą analizę obrazu.

Tabela 3. Model Yu i in. – opis dostepnych akcji [6]

numer akcji	akcja	siła zakłócenia	głębokość sieci CNN
1	redukcja szumu Gaussa	słabe	3
2	redukcja szumu Gaussa	mocne	8
3	redukcja rozmycia Gaussa	słabe	3
4	redukcja rozmycia Gaussa	mocne	8
5	redukcja zniekształceń po kompresji JPEG	słabe	3
6	redukcja zniekształceń po kompresji JPEG	mocne	8
7	zakończenie procesu modyfikacji	-	-

Stan, w którym znalazł się agent, określony jest przez dwa tory. Pierwszy z nich to zbiór 32 cech uzyskanych poprzez analizę obrazu, przez sieć głęboką. Składa się on z 4 warstw splotowych, po których następuje warstwa w pełni połączona. Drugi tor to wektor wartości binarnych wskazujący poprzednio wykonaną przez agenta akcję. W wektorze tym pominięta zostaje akcja o zakończeniu pracy. W wyniku połączenia obydwu torów powstaje wektor 40 wartości będący jednocześnie argumentem sieci LSTM (ang. *Long Short-Term Memory*). Wyjście sieci LSTM określa stan, w którym znajduje się agent. Funkcja wartości akcji jest tu aproksymowana poprzez warstwę w pełni połączoną o liczbie neuronów równej liczbie możliwych do wykonania akcji.



Rysunek 5. Model Yu i in. – schemat działania [6]

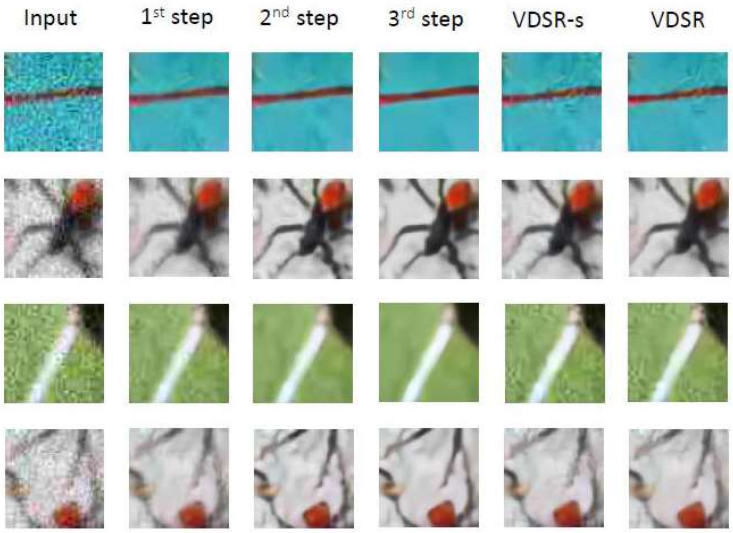
Po dokonaniu wyboru agent modyfikuje obraz przy użyciu odpowiedniego filtra (wyłączając oczywiście akcję zakończenia procesu modyfikacji). Następnie bada on uzyskany w ten sposób obraz i cały schemat powtarzany jest raz jeszcze. Ilustracja opisanego systemu została przedstawiona na rysunku 5. W trakcie procesu nauki agentowi przydzielane jest wzmocnienie opisane wzorem (4).

$$r_t = P_{t+1} - P_t \tag{4}$$

gdzie:

- r_t – wzmocnienie przydzielone agentowi w kroku t
- P_t – wartość metryki PSNR uwzględniającej różnice między obrazem w kroku t a obrazem docelowym

Metryka PSNR (ang. *Peak Signal-to-Noise Ratio*) jest definiowana jako logarytm dziesiętny ze stosunku kwadratu najwyższej wartości piksela obrazu z sumą kwadratów różnic wartości pikseli pomiędzy obrazem docelowym a obrazem zniekształconym. Na rysunku 6. zostało przedstawione porównanie działania zaproponowanej metody RL-Restore z dwiema innymi mianowicie VDSR (ang. *Very Deep Super Resolution*) oraz VDSR-s. Można bez trudu zauważyć, że już w pierwszym kroku, RL-Restore osiągał porównywalne a czasem nawet lepsze wyniki.



Rysunek 6. Model Yu i in. – przykład działania [6]

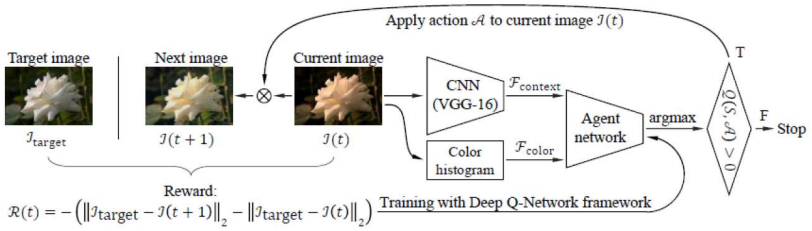
3.2. Korekcja kolorów i upiększanie

Drugim, wyróżnionym w grupie algorytmów modyfikujących obraz zadaniem, jest korekcja kolorów obrazu w celu uzyskania odpowiedniego efektu wizualnego. Omówiony w rozdziale 3.1 algorytm PixelRL jest w stanie wykonać również takie zadanie. W tym celu niezbędna była modyfikacja zbioru możliwych akcji. Określono 6

działań: zmianę kontrastu, zmianę saturacji, zmianę jasności oraz zmianę nasycenia parami kolorów RG, GB i RB. Każde z tych działań mogło być wykonane na dwa różne sposoby, mianowicie mogło wzmocnić lub osłabić dany wskaźnik o 5%. Dodatkowo agent mógł nie podejmować żadnego działania. Liczba możliwych do podjęcia akcji dla tak zdefiniowanego zbioru wynosi zatem 13. Tak przygotowany agent był w stanie modyfikować obraz wejściowy, uzyskując przeróżne efekty wizualne. Na rysunku 7. porównano (kolejno od lewej) obrazy wejściowe, obrazy uzyskane za pomocą metody DNN, obrazy uzyskane przez zaproponowany system oraz obraz będący efektem pracy eksperta.

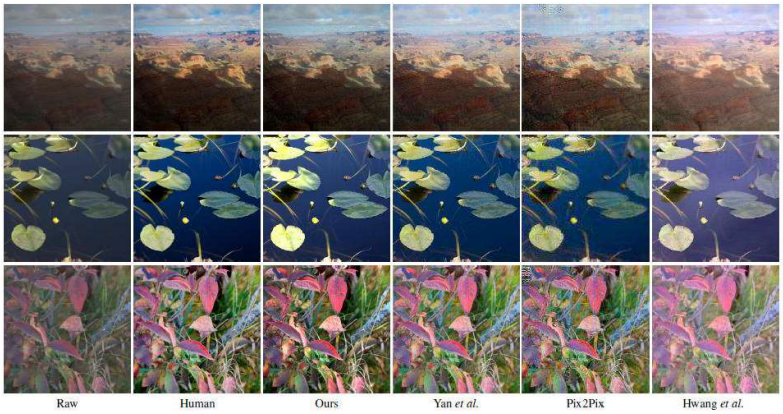
Rysunek 7. PixelRL – przykład działania (korekcja kolorów) [5]

Niemal identyczny zbiór akcji przyjęli w 2018 roku badacze Park i in. W [7] zaproponowali oni zgoła odmienny system. Zakładał on istnienie jednego agenta modyfikującego cały obraz w każdym kroku czasowym. Na początku każdego kroku obraz był analizowany przez sieć konwolucyjną VGG-16, która odzwierciedlała funkcję kontekstu $F_{context}$. Dodatkowo wyznaczany był kolorowy histogram obrazu, który był reprezentacją funkcji koloru F_{color}. Obydwie te funkcje stanowiły reprezentację stanu, w którym znalazł się agent. Następnie informacje te były przetwarzane przez sieć głęboką, będącą aproksymacją funkcji wartości akcji odzwierciedlającej strategię, którą posługiwał się agent. Po ocenie wartości akcji agent podejmował wybraną akcję, dokonując jednocześnie zmiany obrazu. Algorytm kończył działanie w momencie, gdy wartość każdej z akcji spadała poniżej 0. Wzmocnienie było wyliczane jako różnica odległości euklidesowej (norma L2) obrazu przed wykonaniem akcji, a obrazem docelowym z odległością euklidesową obrazu po wykonaniu wybranej akcji, a obrazem docelowym. Opisany schemat działania został przedstawiony w formie graficznej na rysunku 8.



Rysunek 8. Model Park i in. – schemat działania [7]

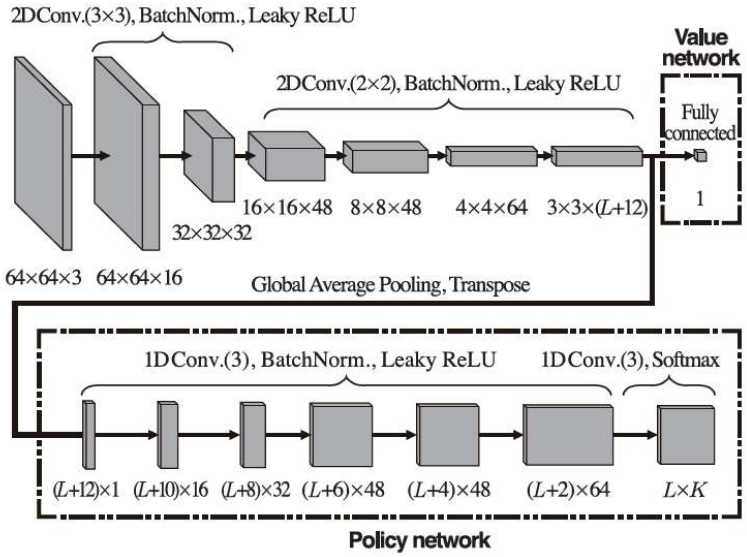
Rezultat pracy zbudowanego agenta w porównaniu z wynikami innych metod przetwarzania obrazu został przedstawiony na rysunku 9. W kolejnych kolumnach od lewej strony znajdują się odpowiednio obrazy: oryginalne, edytowane przez eksperta, uzyskane przy pomocy zaproponowanego rozwiązania oraz uzyskane trzema innymi metodami niewykorzystującymi uczenia się ze wzmocnieniem.



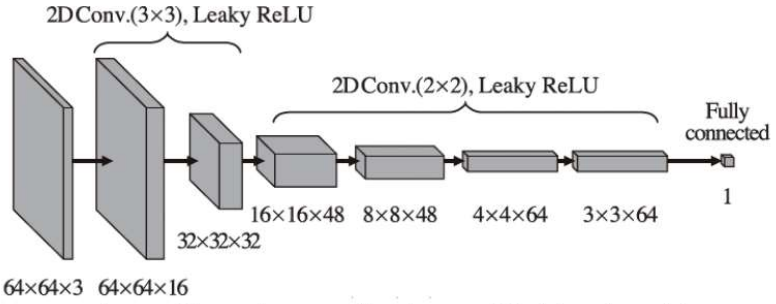
Rysunek 9. Model Park i in. – przykład działania [7]

Kolejną propozycję aplikacji uczenia się ze wzmocnieniem do przetwarzania obrazu wysunęli w [8] w 2020 roku badacze Kosugi oraz Yamasaki. Zadaniem agenta był dobór parametrów wejściowych do procesu przetwarzania obrazu w programie Adobe Lightroom® lub Adobe Photoshop®. Twórcy wykorzystali algorytm uczenia agenta opierający się zarówno na funkcji wartości, jak i funkcji wartości akcji. Do aproksymacji tych funkcji zbudowali oni moduł tzw. generatora będącego złożoną siecią neuronową. Jej schemat został przedstawiony na rysunku 10.

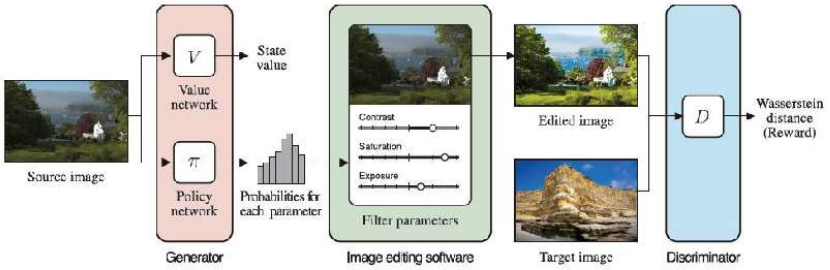
Szczególną uwagę należy zwrócić na ostatnią warstwę sieci reprezentującej strategię agenta. Z uwagi na fakt ciągłości argumentów wejściowych programów przetwarzających obraz autorzy postanowili dokonać ich dyskretnej reprezentacji. Agent ma zatem możliwość wybrania jednej z równo rozłożonych wartości pośrednich każdego z parametrów. Liczba wartości pośrednich każdego z parametrów wynosi L, natomiast ich ilość K. Dzięki zastosowaniu funkcji SoftMax uzyskiwane jest prawdopodobieństwo opisujące każdą z możliwych do wykonania akcji. Po dokonaniu wyboru wartości parametrów następuje modyfikacja obrazu przy pomocy jednego z wymienionych wcześniej programów. W trakcie procesu nauki, po każdym kroku modyfikującym obraz obliczana jest również wartość wzmocnienia przydzielana agentowi. Do tego celu został zaprojektowany moduł tzw. dyskryminatora będącego głęboką siecią neuronową. Schemat budowy tej sieci został przedstawiony na rysunku 11, natomiast ilustrację całego modelu zaproponowanego przez autorów przedstawia rysunek 12.



Rysunek 10. Model Kosugi & Yamasaki – schemat modułu generatora [8]

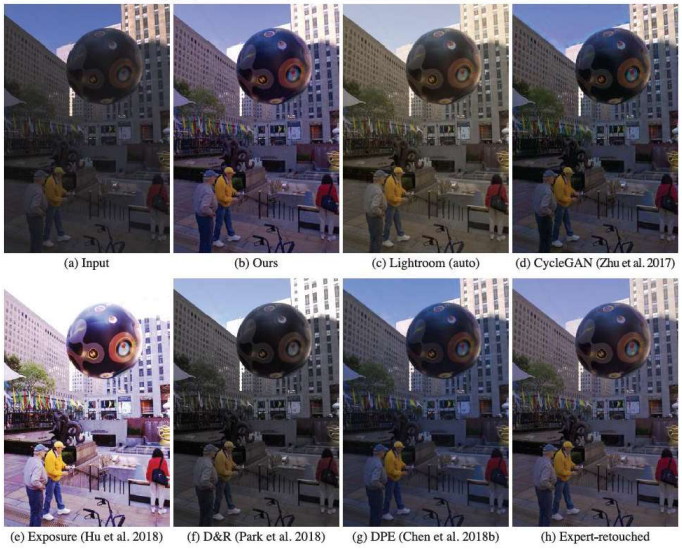


Rysunek 11. Model Kosugi & Yamasaki – schemat modułu dyskryminatora [8]

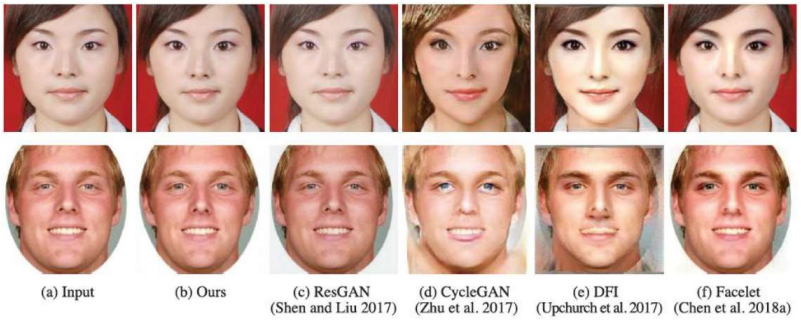


Rysunek 12. Model Kosugi & Yamasaki – schemat działania [8]

Na rysunku 13 zostały zestawione: (a) obraz wejściowy, (b) obraz uzyskany zaproponowaną metodą, (c) obraz automatycznie poprawiony przez program Adobe Lightroom®, (d)(e)(g) obrazy uzyskane innymi metodami niewykorzystującymi uczenia się ze wzmocnieniem, (f) obraz uzyskany metodą omówioną wcześniej, zaproponowaną przez Park i in., (h) obraz poprawiony przez eksperta. Można zauważyć, że obraz uzyskany przy pomocy omawianego algorytmu, mimo iż różni się od obrazu poprawionego przez eksperta, jest wizualnie przystępny. Pozwala on dostrzec wszystkie detale, a przy tym nie wygląda nienaturalnie. Zaproponowane rozwiązanie zostało również przetestowane, jako narzędzie dobierające parametry algorytmu upiększania twarzy w programie Adobe Photoshop®. Na rysunku 14 przedstawiono porównanie portretu oryginalnego (a) z działaniem zaproponowanego modelu (b) oraz działaniem innych metod niewykorzystujących uczenia się ze wzmocnieniem (c-f). Z łatwością można zauważyć, że omawiany algorytm osiągnął zdecydowanie lepsze wyniki pracy od algorytmów konkurencyjnych.



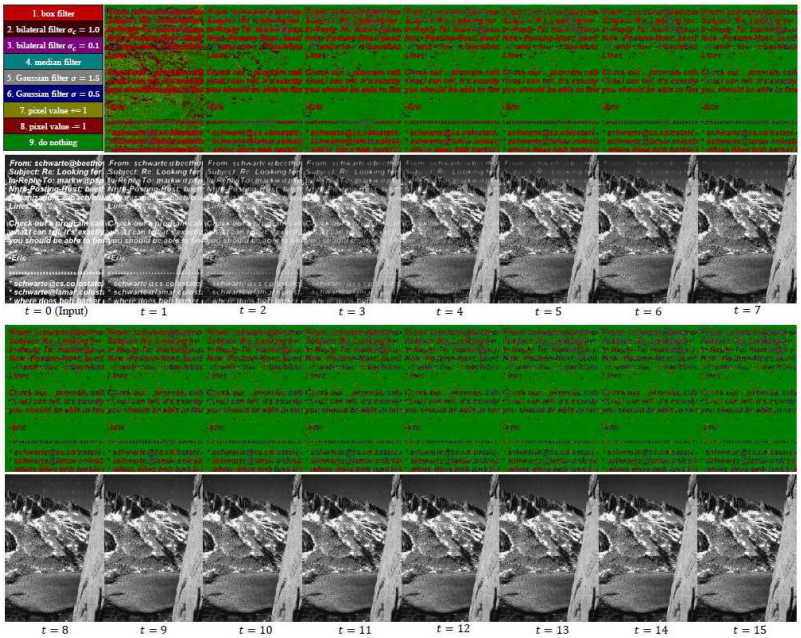
Rysunek 13. Model Kosugi & Yamasaki – przykład działania (korekcja kolorów) [8]



Rysunek 14. Model Kosugi & Yamasaki – przykład działania (upiększanie twarzy) [8]

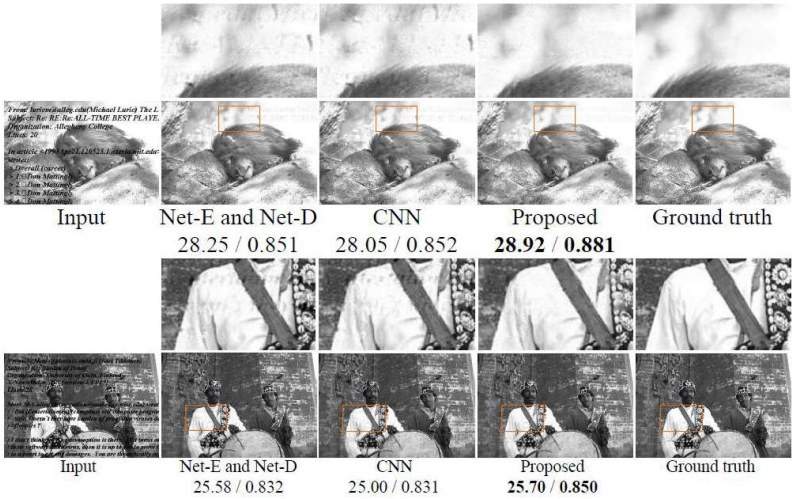
3.3. Odzyskiwanie bloków

Trzecim z kolei zadaniem jest odzyskiwanie utraconych bloków obrazu. Mogą to być zarówno nieprawidłowości w danych spowodowane np. uszkodzeniem nośnika lub elementy dodane, takie jak np. napisy. Na początku należy przytoczyć wspomniany już dwa razy algorytm PixelRL, który został wykorzystany również do wykonania zadania odzyskiwania utraconych bloków. Przyjmując zbiór możliwych do wykonania akcji przedstawiony w tabeli 1, można z powodzeniem uzyskać zamierzony efekt. Na rysunku 15 został zilustrowany proces usuwania napisów z przykładowego obrazu. Ponownie górny wiersz to mapa akcji podejmowanych przez agentów w każdym kroku czasowym. Mimo, iż algorytm potrzebował zdecydowanie więcej czasu (3 razy więcej kroków czasowych), to zadanie zostało wykonane pomyślnie. Ponadto nadal istnieje możliwość odtworzenia drogi, jaką przebył każdy z agentów.



Rysunek 15. PixelRL – przykład działania (odzyskiwanie bloków cz. 1) [5]

Na rysunku 16 natomiast porównano działanie algorytmu PixelRL z dwoma konkurencyjnymi algorytmami. Dodatkowo w pierwszym oraz trzecim wierszu zostały przedstawione powiększone fragmenty obrazów w celu łatwiejszego porównania pracy każdego z algorytmów. Jak nietrudno zauważyć, algorytm PixelRL osiągnął zdecydowanie najlepsze efekty prawie całkowicie eliminując napisy, a tym samym odzyskując utracone wcześniej bloki.



Rysunek 16. PixelRL - przykład działania (odzyskiwanie bloków cz.2.) [5]

Inną formą odzyskiwania bloków obrazu mogą być metody pośrednie. Może być to np. rekonstrukcja obrazów tomografii komputerowej (ang. CT – *Computer Tomography*). Metoda automatyzacji tego procesu została zaproponowana w [9] w 2018 roku przez Shen i in. jako narzędzie, którego parametrami wejściowymi steruje agent, wykorzystali oni algorytm ADMM (ang. *Alternating Direction Method of Multipliers*). Wspomniany algorytm modyfikuje obraz wejściowy zgodnie z wektorem parametrów otrzymanym od użytkownika. Wektor zawiera po jednym parametrze na każdy piksel obrazu. Celem agenta było określenie wielkości zmiany każdego z elementów składowych wektora parametrów. Autorzy zdecydowali się na wybór zbioru akcji przedstawionego w tabeli 3 jako aproksymator funkcji wartości akcji autorzy zaproponowali sieć głęboką, której schemat został przedstawiony na rysunku 17. Za stan, w którym znalazł się agent, przyjęte zostało najbliższe otoczenie wybranego piksela, czyli kwadrat 9 x 9, gdzie badany piksel znajduje się w centrum. Początkowe warstwy konwolucyjne analizują treść badanego wycinka obrazu. Zadaniem następujących po nich warstw w pełni połączonych jest regresja informacji uzyskanych podczas analizy, by w efekcie otrzymać wartość każdej z możliwych do podjęcia akcji. Wzmocnienie zostało zaś zdefiniowane wzorem (5).

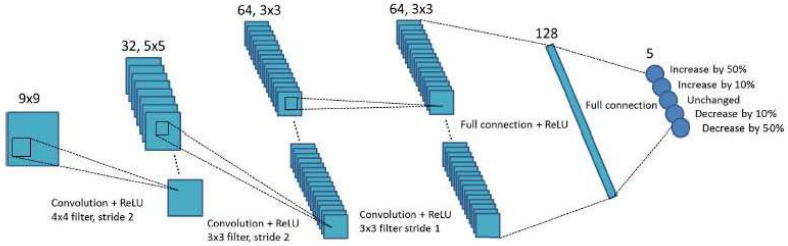
$$r_t = \frac{\|S^*(x)\|}{\|S_{t+1}(x) - S^*(x)\|} - \frac{\|S^*(x)\|}{\|S_t(x) - S^*(x)\|} \tag{5}$$

gdzie:

- r_t – wzmocnienie przydzielone agentowi w kroku t
- $S_t(x)$ – grupa pikseli (9×9) o numerze x należąca do modyfikowanego obrazu przedstawiona w postaci wektora
- $S^*(x)$ – grupa pikseli (9×9) o numerze x należąca do docelowego obrazu przedstawiona w postaci wektora
- $\|\cdot\|$ - operator obliczenia normy L2 (odległości euklidesowej)

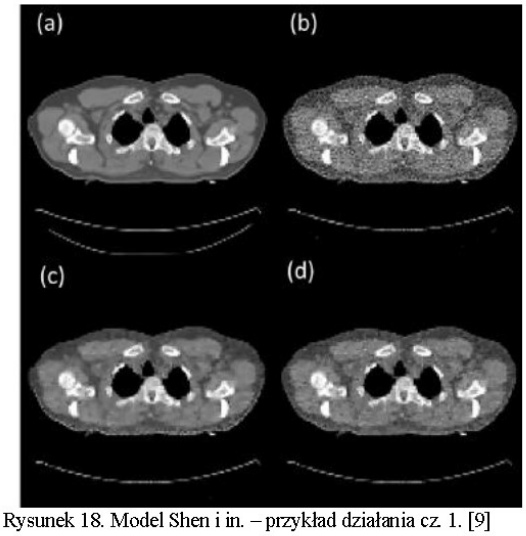
Tabela 4. Model Shen i in. – opis dostępnych akcji [9]

numer akcji	akcja
1	obniżenie wartości parametru o 50%
2	obniżenie wartości parametru o 10%
3	zwiększenie wartości parametru o 50%
4	zwiększenie wartości parametru o 10%
5	pozostawienie wartości parametru bez zmian

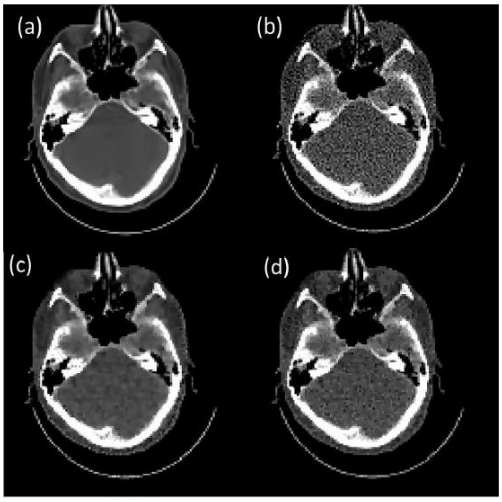


Rysunek 17. Model Shen i in. – schemat działania [9]

Na rysunku 18 oraz 19 zestawiono: (a) obraz wzorcowy, (b) obraz zrekonstruowany dla arbitralnej wartości parametrów równej w obu przypadkach 0.005, (c) obraz zrekonstruowany przy pomocy zaproponowanego modelu, (d) obraz zrekonstruowany dla ustawionej przez eksperta wartości, odpowiednio 0.05 oraz 0.12.



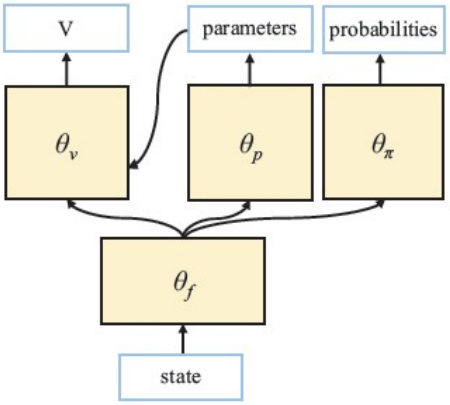
Rysunek 18. Model Shen i in. – przykład działania cz. 1. [9]



Rysunek 19. Model Shen i in. – przykład działania cz. 2 [9]

Z łatwością można dostrzec, że najlepiej obraz wzorcowy odwzorowany został przez zaproponowany model wykorzystujący uczenie się ze wzmocnieniem. Należy więc zauważyć, że możliwość automatyzacji procesu przetwarzania obrazu połączona z osiągnięciem bardziej satysfakcjonujących rezultatów, jest ogromną zaletą wykorzystania algorytmu uczenia się ze wzmocnieniem.

Trend rozwijający różne aplikacje algorytmu uczenia się ze wzmocnieniem do obróbki obrazów medycznych został utrzymany. Dwa lata później, w roku 2020, [10] badacze Li i in. zaproponowali system służący rekonstrukcji wyników obrazowania rezonansem magnetycznym (ang. MRI – *Magnetic Resonance Imaging*). Zakłócenia, rozmycia oraz efekt aliasingu powstają w uzyskanych obrazach na skutek obniżenia liczby próbek w czasie wykonywania badania. Taka redukcja ma na celu przyspieszenie tego procesu, co przekłada się korzystnie na komfort pacjenta czy też umożliwia wykonanie badań większej liczbie osób. Badacze, wzorując się na omówionym wcześniej modelu PixelRL, utworzyli własny system, którego schemat został zilustrowany na rysunku 20. Stanem, w jakim znalazł się agent, określony został cały obraz. Obraz ten był przetwarzany przez sieć konwolucyjną, której zadaniem była ekstrakcja jego cech. Wyodrębnione cechy stanowiły wejście trzech pozostałych modułów. Pierwszym z nich jest sieć głęboka będąca aproksymacją funkcji wartości akcji. Zbiór możliwych do wykonania akcji został przedstawiony w tabeli 4. Należy w tym miejscu zauważyć, że autorzy, dążąc do uzyskania ciągłości w doborze parametrów filtrów określonych dla odpowiednich akcji, założyli, że są one również wyliczane przez agenta. Oznaczono je symbolami: p_L, p_{S1}, p_{S2}, p_{S3}, p_{S4}, p_u.



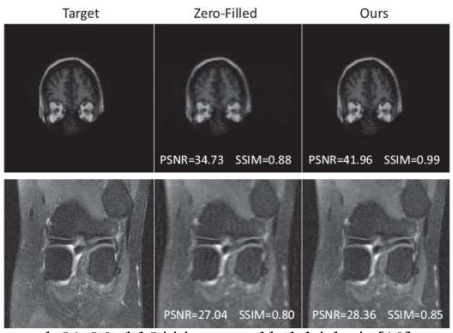
Rysunek 20. Model Li i in. – schemat działania [10]

Za ich wyznaczenie odpowiada drugi z modułów, również będący siecią głęboką. Na końcu informacje o wyodrębnionych cechach oraz o wyznaczonych wartościach parametrów są łączone i podawane na wejście ostatniego z modułów odpowiedzialnego za aproksymację funkcji wartości.

Tabela 5. Model Li i in. – opis dostępnych akcji [10]

numer akcji	akcja	rozmiar filtra	parametry filtra
0	brak akcji	-	-
1	filtr pudełkowy	5x5	-
2	filtr bilateralny	5x5	$\sigma_c = 1.0$, $\sigma_s = 5.0$
3	filtr medianowy	5x5	-
4	filtr Gaussa	5x5	$\sigma = 0.5$
5	filtr Laplace'a	3x3	p_L
6	filtr Sobel'a (lewo)	3x3	p_{S1}
7	filtr Sobel'a (prawo)	3x3	p_{S2}
8	filtr Sobel'a (góra)	3x3	p_{S3}
9	filtr Sobel'a (dół)	3x3	p_{S4}
10	nieostre maskowanie	5x5	$\sigma = 0.5$, p_u
11	wartość piksela -3	-	-

Na rysunku 21. zostały porównane (odpowiednio od lewej): obraz docelowy, obraz zniekształcony (wejście algorytmu) oraz obraz zrekonstruowany przez zaproponowany system. Ponadto obliczono również metryki PSNR oraz SSIM określające odpowiednio poziom szumu oraz podobieństwo dwóch obrazów. Analizując zestawienie, można zauważyć znaczącą poprawę jakości obrazu po rekonstrukcji.



Rysunek 21. Model Li i in. – przykład działania [10]

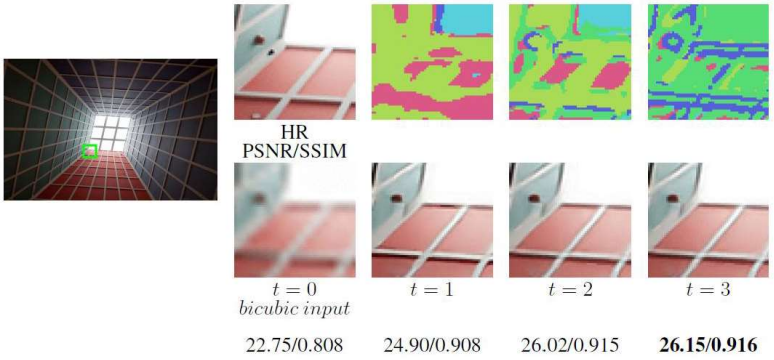
3.4. Wyostrzanie

Czwartym rodzajem zadań jest wyostrzanie obrazów. Oprócz oczywistej korzyści, jaką jest bogatsze wrażenie wizualne, działanie to może również pozytywnie wpływać na prace innych algorytmów przetwarzających obrazy (np. ułatwione wykrywanie krawędzi). W 2020 roku Vassilo i in.zaproponowali [11] – rozwiązanie oparte na omówionym modelu PixelRL. Zmodyfikowali oni zbiór akcji, wykorzystując istniejące rozwiązania oparte na sztucznych sieciach neuronowych, takich jak: EDSR (ang. *Enhanced Deep Super-Resolution network*), ESRGAN (ang. *Enhanced Super-Resolution Generative Adversarial Network*), ukierunkowaną na metrykę PSNR sieć ESRGAN-PSNR oraz PPON (ang. *Progressive Perception-Oriented Network*). Spis wszystkich dostępnych akcji przedstawiono w tabeli 5.

Tabela 6. PixelRL (Vassilo i in.) – opis dostępnych akcji [11]

numer akcji	Akcja
1	wartość piksela -1
2	brak akcji
3	wartość piksela +1
4	filtr EDSR
5	filtr ESRGAN
6	filtr ESRGAN–PNSR
7	filtr PPON

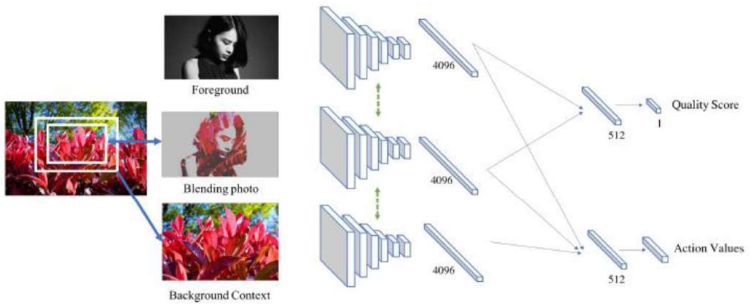
Model sieci będących aproksymacją funkcji wartości oraz funkcji wartości akcji został zaimplementowany analogicznie, jak w oryginale. Wartość wzmocnienia również była obliczana zgodnie z pierwotną wersją algorytmu – wzór (1). Efekt działania zaproponowanej metody został przedstawiony na rysunku 22. Po lewej stronie widnieje obraz całościowy, górny wiersz zawiera mapy podejmowanych przez agentów akcji dla wybranego fragment obrazu (dla czytelności), natomiast w dolnym wierszu przedstawiony został wybrany fragment obrazu wejściowego oraz obrazów uzyskanych podczas kolejnych kroków czasowych działania algorytmu. Pod tymi ostatnimi przedstawione zostały równie obliczone metryki PSNR oraz SSIM. Z łatwością można zauważyć, że każdy kolejny krok wykonany przez agenta przybliża go do osiągnięcia obrazu zgodnego z obrazem wzorcowym. Tak, jak w wyżej omówionych przykładach, ważnymi zaletami są zarówno dobrej jakości obrazy, jak również możliwość ilustracji oraz późniejszych analiz ścieżek decyzyjnych każdego z agentów.



Rysunek 22. PixelRL (Vassilo i in.) – przykład działania [11]

3.5. Scalanie

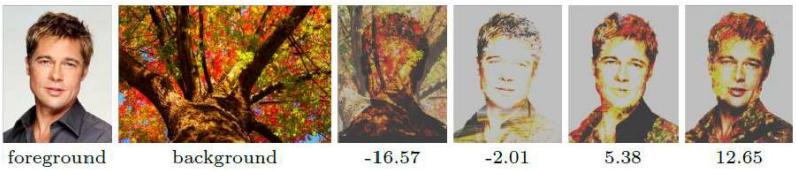
Piątym i zarazem ostatnim wyróżnionym zadaniem z grupy algorytmów modyfikujących obraz jest scalanie dwóch obrazów. Operacja taka nie ułatwia analizy obrazu, lecz jej rezultatem jest obraz wywołujący pożądany efekt wizualny. W 2018 roku w [12] Hung i in. zaproponowali system, którego zadaniem był dobór parametrów dla narzędzia scalającego dwa obrazy w programie Adobe Photoshop®. Jako wejście przyjęto dwa obrazy: obraz tła oraz obraz pierwszoplanowy. Początkowe wartości parametrów oraz lokalizacja i rozmiar ramki ograniczającej wybrany z obrazu tła fragment wyznaczane są w sposób losowy. Wyznaczany jest również pierwszy z kolei obraz wynikowy. W kolejnych krokach czasowych wszystkie 3 obrazy analizowane są z wykorzystaniem głębokiej sieci konwolucyjnej VGG16 zwieńczonej warstwą w pełni połączoną o liczbie neuronów równej 4096. Tak przygotowane informacje na temat trzech obrazów stanowią wejście kolejnej nieco mniejszej warstwy w pełni połączonej o liczbie neuronów równej 512.



Rysunek 23. Model Hung i in. – schemat działania [12]

Ostatnia warstwa w pełni połączona stanowi zwieńczenie aproksymatora funkcji wartości akcji. Projektując sygnał wzmocnienia, autorzy zaproponowali, że będzie ono wyznaczane przez głęboką sieć w pełni połączoną, której wejście stanowić będą informacje otrzymane w trakcie analizy obrazu pierwszoplanowego oraz obrazu wyni-

kowego. Schemat budowy omówionej sieci został zilustrowany na rysunku 23. Wysunięta przez autorów propozycja to nietypowe podejście, ponieważ w większości przypadków sygnał wzmocnienia określany jest jawnym wzorem. Zaproponowana przez nich metoda działa jednak sprawnie, co potwierdza zestawienie zdjęć na rysunku 24 zawierające (od lewej): obraz pierwszoplanowy, obraz tła oraz cztery różne obrazy wynikowe wraz ze wzmocnieniem im towarzyszącym. Zgodnie z oczekiwaniami można zauważyć, że im większa była wartość wzmocnienia, tym bardziej zadowalający efekt wizualny został otrzymany.



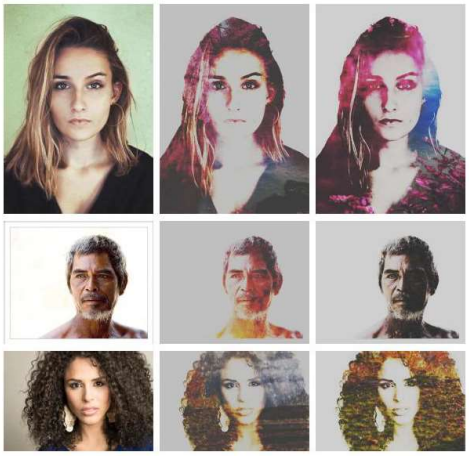
Rysunek 24. Model Hung i in. – przykład par obraz-wzmocnienie [12]

Na podstawie otrzymywanych informacji agent miał za zadanie sterować położeniem oraz rozmiarem ramki, a także wartościami parametrów. Wszystkie dostępne akcje zostały przedstawione w tabeli 6. Należy również zauważyć, że gdy jest mowa o zmianie położenia ramki, chodzi w domyśle o operacje na odpowiednich współrzędnych punktów wyznaczających ramkę.

Tabela 7. Model Hung i in. – opis dostepnych akcji [12]

numer akcji	akcja
1	przesuń ramkę w lewo o 5%
2	przesuń ramkę w prawo o 5%
3	przesuń ramkę w górę o 5%
4	przesuń ramkę w dół o 5%
5	zmniejsz ramkę o 5%
6	zwiększ ramkę o 5%
7	zmniejsz kontrast o 10%
8	zwiększ kontrast o 10%
9	zmniejsz jasność o 10%
10	zwiększ jasność o 10%

Niezmiernie ważnym do odnotowania jest fakt, iż dzięki wykorzystaniu algorytmu uczenia się ze wzmocnieniem, twórcom udało się zaprojektować system automatyzujący w pełni pracę eksperta. Podczas, gdy ekspertowi opracowanie jednego obrazu zajmowało średnio 5 minut z pomocą wcześniej napisanych skryptów. Utworzony system osiągał podobnej jakości efekty w czasie około 5 sekund, wykorzystując do obliczeń kartę graficzną. Może być to zatem niezmiernie duża korzyść oraz pomoc dla profesjonalnego edytora. W celu porównania jakości rezultatów na rysunku 25 przedstawiono (od lewej): obraz pierwszoplanowy, obraz wygenerowany przez utworzony system, obraz opracowany przez eksperta. Na rysunku 26 zostały dodatkowo przedstawione inne przykładowe obrazy wynikowe uzyskane za pomocą omówionego systemu.



Rysunek 25. Model Hung i in. – porównanie działania z ekspertem [12]

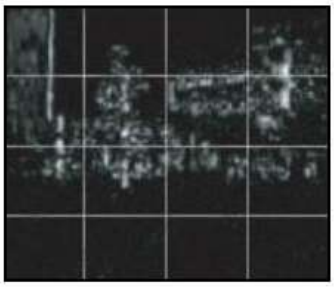
Rysunek 26. Model Hung i in. – przykład działania [12]

4. Algorytmy detekcji i śledzenia obiektów

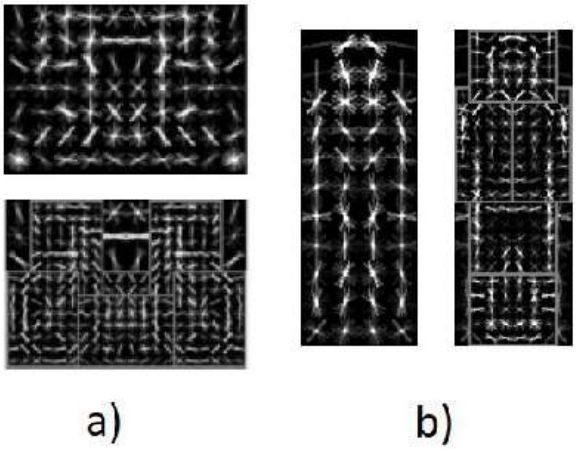
Oprócz korekcji obrazów bardzo ważnym wyzwaniem jest również analiza treści obrazu w celu detekcji oraz śledzenia trajektorii ruchu obiektów. Mając możliwość sprawnego wykrywania obecności obiektów oraz ustalania ich położenia, można tworzyć systemy pozwalające m.in.: badać kontekst obrazu, budować systemy zliczające, budować systemy wykrywające anomalie (np. zagrożenie w postaci pieszego na drodze samochodu). Mając dodatkowo możliwość określania trajektorii ruchu obiektów, można stworzyć systemy badające życie organizmów, analizujące historię ruchu w celu rekonstrukcji przyczyny ich powstania czy też systemy analizujące zachowania społeczne i wykrywające wszelkie anomalie, a tym samym zagrożenia (np. atak serca, zagrożenie terrorystyczne czy panika) lub sytuacje korzystne. W niniejszym rozdziale zostaną omówione różne podejścia aplikacji algorytmu uczenia się ze wzmocnieniem do zadań detekcji oraz określania trajektorii ruchu obiektów.

4.1. Detekcja obiektów

Pierwszym z omawianych zadań drugiej grupy jest detekcja obiektów. W 2012 roku w [13] Karayev i in. zaproponowali metodę mającą na celu zidentyfikować możliwie największą liczbę obiektów, w możliwie najkrótszym czasie. Dodatkowym celem była możliwość przerwania pracy algorytmu w dowolnym momencie wraz z oczekiwaniem optymalnej ilości zidentyfikowanych obiektów dla tego momentu. Zbiór akcji zawierał możliwość uaktualnienia wektora prawdopodobieństwa występowania obiektów, możliwość zbadania kontekstu sceny przedstawionej na obrazie oraz możliwość użycia odpowiednich detektorów wykrywających obecność obiektów danej klasy. Przykładowa reprezentacja kontekstu sceny została przedstawiona na rysunku 27, natomiast reprezentacja przykładowych detektorów została zaprezentowana na rysunku 28.



Rysunek 27. Model Karayev i in. – przykład reprezentacji kontekstu sceny [13]



Rysunek 28. Model Karayev i in. – przykład filtrów wybranych klas obiektów [13]

W celu osiągnięcia zaplanowanych rezultatów wartość wzmocnienia określono wzorem (6).

$$r_t = \Delta_{ap}\left(tim_T^t - \frac{\Delta_{tim}}{2}\right) \quad (6)$$

gdzie:

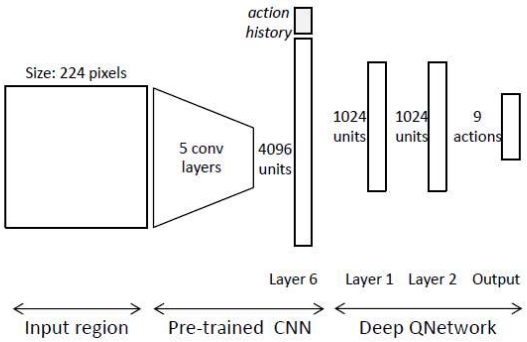
- r_t – wzmocnienie przydzielone agentowi w kroku t
- Δ_{ap} – zmiana metryki określającej średnią precyzję (ang. *Average Precision*)
- Δ_{tim} – czas wykonania podjętej przez agenta akcji
- tim_T – czas pozostały do zakończenia pracy agenta (ang. *time to termination*)

Nieco innym podejściem jest model zaproponowany trzy lata później w [14] przez Caicedo oraz Lazebnik. Zadaniem agenta była odpowiednia modyfikacja ramki okalającej, tak by obiekt zajmował jak największą jej powierzchnię. Agent miał możliwość wykonać jeden z ruchów wymienionych w tabeli 7. Należy również, tak jak wcześniej, zauważyć, że gdy jest mowa o zmianie położenia ramki, chodzi w domyśle o operacje na odpowiednich współrzędnych punktów wyznaczających ramkę.

Tabela 8. Model Caicedo & Lazebnik – opis dostępnych akcji [14]

numer akcji	akcja
1	przesuń ramkę w lewo o 20%
2	przesuń ramkę w prawo o 20%
3	przesuń ramkę w górę o 20%
4	przesuń ramkę w dół o 20%
5	zmniejsz ramkę o 20%
6	zwiększ ramkę o 20%
7	spłaszcz ramkę o 20%
8	ściśnij ramkę o 10%
9	zakończ wyszukiwanie (znaleziono obiekt)

Stanem, w którym znalazł się agent, nazwano krotkę złożoną z wektora cech wybranego regionu oraz wektora zawierającego informacje o 10 ostatnio podjętych przez agenta akcjach. Wektor cech był uzyskiwany przez wycięcie z obrazu aktualnie określonej ramki oraz pola ją otaczającego o grubości 16 pikseli. Było to działanie mające na celu uwzględnianie kontekstu, w którym znajduje się aktualnie określona ramka. Wycinek ten był następnie skalowany do obrazu o wymiarach 224x224 piksele. Ostatnim krokiem w celu uzyskania wektora cech było podanie przeskalowanego obrazu na wcześniej nauczoną konwolucyjną sieć głęboką zakończoną warstwą w pełni połączoną o 4096 neuronach. Aktualny stan, w którym znalazł się agent, był zatem reprezentowany przez wektor o liczbie elementów równej 5006. Rolę aproksymatora funkcji wartości akcji pełniła trójwarstwowa w pełni połączona sieć neuronowa. Na podstawie obliczonych przez nią wartości agent podejmował decyzję o wyborze konkretnej akcji. Schemat opisanego modelu został przedstawiony na rysunku 29.



Rysunek 29. Model Caicedo & Lazebnik – schemat działania [14]

Wzmocnienie zostało określone dwoma wzorami, stosowanymi odpowiednio w sytuacji podjęcia akcji zmieniającej parametry ramki okalającej (7) oraz podjęcia akcji zgłaszającej zlokalizowanie obiektu (8).

$$r_t = sign\big(IoU(f^t, f^*) - IoU(f^{t-1}, f^*)\big) \quad (7)$$

$$r_t = \begin{cases} \eta, IoU(f^{t-1}, f^*) \geq \tau \\ -\eta, IoU(f^{t-1}, f^*) < \tau \end{cases} \quad (8)$$

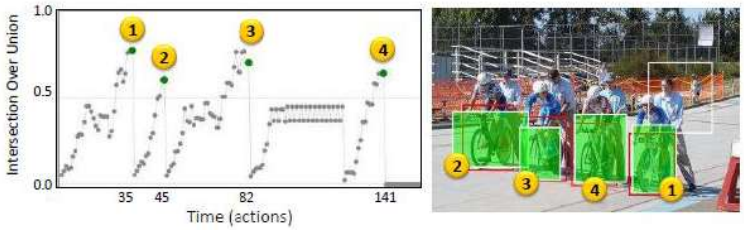
gdzie:

- r_t – wzmocnienie przydzielone agentowi w kroku t
- $IoU(f_1, f_2)$ – metryka określająca stosunek pola powierzchni części wspólnej obszarów f_1 oraz f_2 do pola powierzchni unii tych obszarów
- f^t – obszar określony po wykonaniu przez agenta akcji w kroku t
- f^* – wzorcowy obszar, w którym znajduje się obiekt
- η – stała określająca wartość wzmocnienia
- τ – stała określająca wartość progową dla metryki

Utworzony system potrafi również wykrywać wiele obiektów. Po wykonaniu akcji sygnalizującej zakończenie wyszukiwania pierwszego obiektu na badany obraz nanoszony jest wzór czarnego krzyża, który spowoduje przeszukiwanie przez agenta innych rejonów obrazu. Przykład obrazu przed oraz po zastosowaniu tego kroku został przedstawiony na rysunku 30.

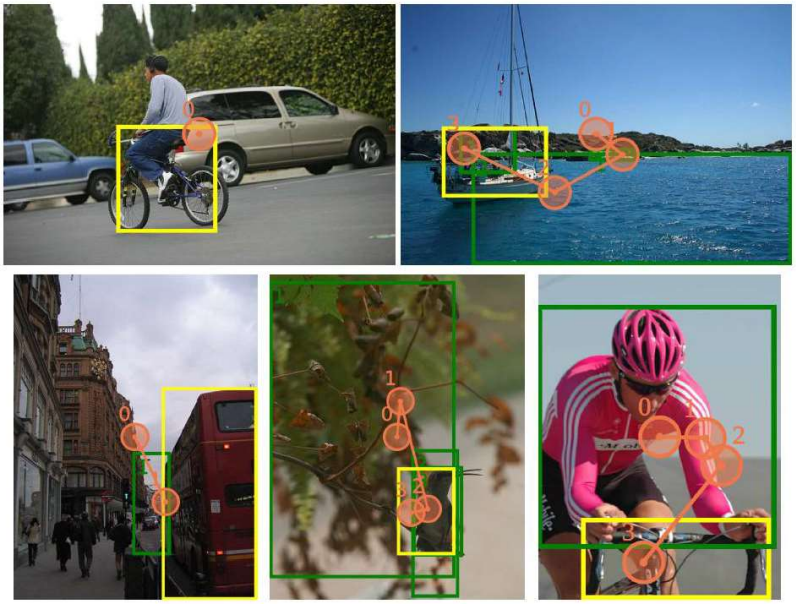
Rysunek 30. Model Caicedo & Lazebnik – przykład działania (1) [14]

Proces wykrywania wielu obiektów (wartości metryki w kolejnych krokach czasowych) został przedstawiony na rysunku 31. Dla lepszej wizualizacji zaznaczono również wzorcowe oraz wyznaczone ramki okalające, a także kolejność detekcji poszczególnych obiektów.



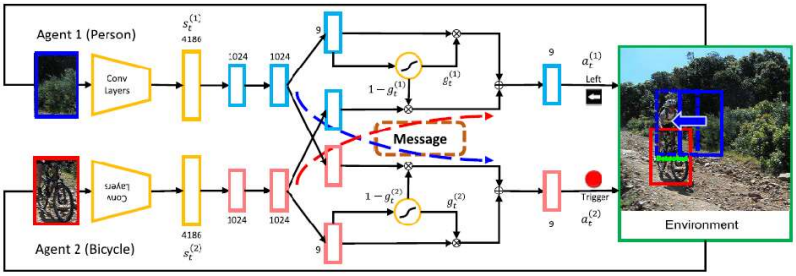
Rysunek 31. Model Caicedo & Lazebnik – przykład działania (2) [14]

Odmiennym podejściem wykazali się zespół Mathe i in., który w [15] w 2016 roku przedstawił implementację algorytmu uczenia się ze wzmocnieniem, w której zadaniem agenta było kierowanie procesem przeszukiwania regionów obrazu w celu minimalizacji ilości kosztownych czasowo operacji. Do ekstrakcji cech regionów obrazu wykorzystano głęboką sieć neuronową. Jako dane wejściowe przyjmowała ona wybrany region obrazu wraz z obramowaniem odpowiedniej grubości w celu zapobiegnięcia utracie kontekstu. Rolę funkcji pewności pełniła sieć LSTM. Stanowiła ona przekonanie agenta o znajdowaniu się szukanego obiektu w danym miejscu. Jako początkowy punkt skupienia wybierany był centralny punkt obrazu, a następnie analizowane były regiony znajdujące się w otoczeniu tego punktu. Mając do dyspozycji listę poprzednich punktów skupienia, listę przeanalizowanych regionów wraz z wartością funkcji pewności, a także informacje na temat nowo przeanalizowanych regionów, agent musiał podjąć decyzję o wyborze nowego punktu skupienia lub zakończeniu wyszukiwania w przypadku, gdy uznał, że jest odpowiednio pewny wykrycia obiektu w jednym z przeanalizowanych regionów. Wzmocnienie, które otrzymywał agent, stanowiło kompromis pomiędzy odpowiednio dużą wartością funkcji pewności, a odpowiednio małą liczbą zmian punktu skupienia, czyli redukcją kosztów poniesionych na rzecz analizy nowych regionów obrazu. Na rysunku 32. przedstawiono przykładowe strategie poszukiwania wypracowane przez agenta. Pomarańczowymi okręgami zaznaczono obszary skupienia, zielonym kolorem oznaczono regiony z największą wartością funkcji pewności w danym kroku czasowym, natomiast kolorem żółtym zaznaczone zostały końcowe regiony oznaczone przez agenta. Na obrazie w lewym górnym rogu agent napotkał sprzyjającą sytuację, ponieważ w jednym z regionów wokół centralnego punktu obrazu odnalazł on obiekt. Mógł on więc od razu wykonać akcję kończącą poszukiwania. Na obrazie w lewym dolnym rogu można zauważyć, że agent wybrał region zawierający ulicę, która to doprowadziła go do odnalezienia autobusu w kolejnym kroku. Podobne zachowanie można dostrzec na obrazach po prawej stronie, tj. agent podążał za regionami, w których niekoniecznie znajduje się obiekt, ale regiony te mogą wskazywać na jego bliską obecność. W przypadku środkowego obrazu na dole widać pewne skonfundowanie agenta, którego w błąd mogła wprowadzić gałąź zajmująca większą część fotografii.



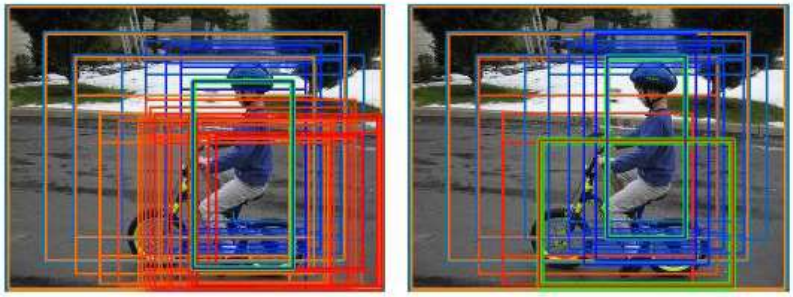
Rysunek 32. Model Mathe i in. – przykład działania [15]

W 2017 roku zainspirowani [14] Kong i in. zaproponowali w [16] rozwiązanie mające wyjść naprzeciw problemowi wyszukiwania obiektów częściowo przesłoniętych lub nachodzących na inne. Utworzyli oni system, w którym dwóch agentów współpracuje w celu odnalezienia obiektów połączonych ze sobą (np. kolarz siedzący na rowerze). Przyjmując zbiór akcji przedstawiony wcześniej w tabeli 7, każdy z agentów miał realizować zadanie lokalizacji jednego z dwóch takich obiektów. Wzmocnienie przydzielane było zgodnie ze wzorami (7) oraz (8). Wymiana informacji pomiędzy agentami realizowana była przez utworzenie współdzielonych fragmentów sieci neuronowych pełniących rolę aproksymatorów funkcji wartości akcji (tuż przed warstwą końcową). Była to dla nich niejako możliwość wglądu w „sposób myślenia" współpracownika. Opisany model został przedstawiony na rysunku 33.



Rysunek 33. Model Kong i in.– schemat działania [16]

Zaproponowana metoda pozwala na lokalizowanie obiektów trudniejszych do wykrycia w mniejszej liczbie kroków niż zajmuje to w standardowym podejściu wykorzystującym algorytm uczenia się ze wzmocnieniem. Przykład porównujący sposób działania obydwu metod został przedstawiony na rysunku 34. Obraz po lewej stronie jest odwzorowaniem działania pojedynczego agenta próbującego zlokalizować obiekty (dziecko oraz rower) sekwencyjnie. Niebieskim kolorem ramki oznaczono poszukiwania dziecka, natomiast czerwony kolor oznacza poszukiwania roweru. Dodatkowo grubą, zieloną linią zaznaczono efekt końcowy poszukiwania. Z łatwością można zauważyć, że pojedynczy agent nie był w stanie odnaleźć drugiego obiektu. Obraz po prawej, posiadający analogiczne oznaczenia, przedstawia proces poszukiwań obydwu obiektów za pomocą metody dwóch agentów. W przedstawionym przykładzie metoda współpracujących agentów odnalazła obydwa obiekty w zaledwie 15 krokach, podczas, gdy metoda pojedynczego agenta nie była w stanie sprostać zadaniu odnalezienia drugiego obiektu nawet po wykonaniu 200 kroków. Dla podniesienia czytelności na rysunku oznaczono wyłącznie 30 pierwszych kroków.



Rysunek 34. Model Kong i in. – przykład działania [16]

4.2. Śledzenie obiektów

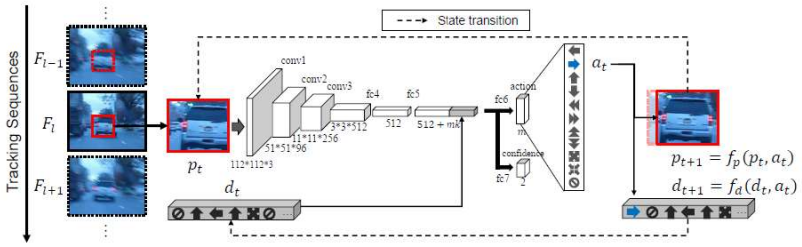
Drugim z omawianych w tej grupie zadań jest zadanie śledzenia ruchu obiektów. W 2017 roku zaprezentowane zostały dwa różne modele podejmujące to wyzwanie. Pierwszy z nich został opisany w [17] przez Yun i in. Zadaniem agenta była odpowiednia modyfikacja ramki okalającej śledzony obiekt, tak, by w każdej klatce był on nią jak najściślej otoczony. Gdy agent stwierdzał, że zakończył niezbędne modyfikacje, podejmował akcję kończącą edycję. Zbiór wszystkich dostępnych akcji został przedstawiony w tabeli 8.

Tabela 9. Model Yun i in. – opis dostępnych akcji [17]

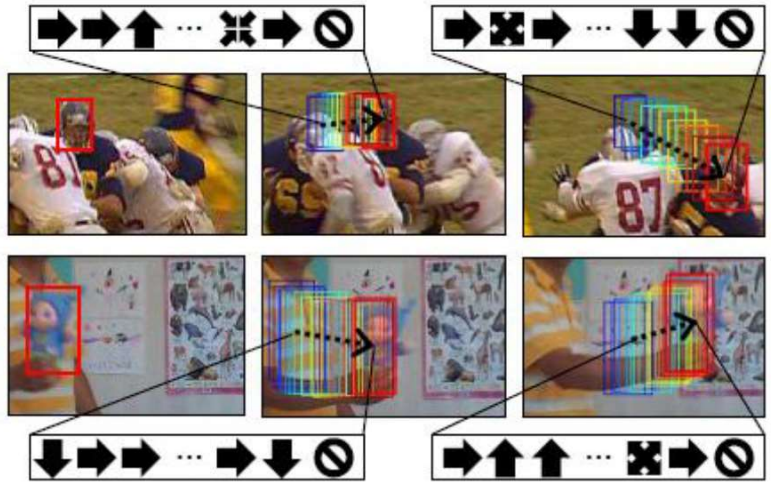
numer akcji	akcja
1	przesuń ramkę w lewo o 2%
2	przesuń ramkę w prawo o 2%
3	przesuń ramkę w górę o 2%
4	przesuń ramkę w dół o 2%
5	przesuń ramkę w lewo o 4%
6	przesuń ramkę w prawo o 4%
7	przesuń ramkę w górę o 4%

8	przesuń ramkę w dół o 4%
9	zmniejsz ramkę o 2%
10	zwiększ ramkę o 2%
11	zakończ wyszukiwanie (znaleziono obiekt)

Stan, w którym znalazł się agent, był określany jako złożenie ograniczonego ramką fragmentu obrazu oraz wektora wartości binarnych zawierającego informację o akcjach podjętych przez agenta w ciągu poprzednich dziesięciu kroków. Wybrany fragment klatki filmu był analizowany przez głęboką sieć konwolucyjną, zwieńczoną dwiema warstwami w pełni połączonymi. Do ostatniej z nich dołączany był wektor zawierający historię podejmowanych wcześniej akcji. Tak przygotowane dane stanowiły wejście dwóch warstw w pełni połączonych. Pierwsza z nich była odpowiedzialna za określenie wartości każdej z możliwych do podjęcia akcji, natomiast druga obliczała tzw. pewność agenta w odniesieniu do aktualnego położenia ramki okalającej. Po wykonaniu akcji kończącej lokalizowanie obiektu agentowi przydzielone zostaje wzmocnienie o wartości 1, gdy wartość metryki IoU jest większa niż 0.7 lub -1 w przeciwnym wypadku. Ilustracja omówionego modelu została przedstawiona na rysunku 35, zaś przykładowe procesy śledzenia obiektu zostały przedstawione na rysunku 36.



Rysunek 35. Model Yun i in.– schemat działania [17]

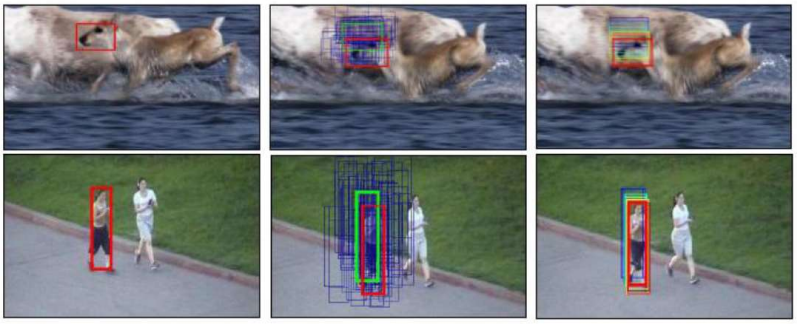


Rysunek 36. Model Yun i in.– przykład działania (1) [17]

Bardzo ważnym aspektem, podnoszącym atrakcyjność zaproponowanego rozwiązania jest możliwość nauki nawet w przypadku, gdy poprawne położenie obiektu określone jest tylko w niskiej liczbie klatek filmu. Przykład takiej sytuacji został przedstawiony na rysunku 37, gdzie kolorem żółtym zostały podane numery klatek, a także wyróżniona została wartość wzmocnienia otrzymanego przez agenta przy przejściu do tych klatek. Czerwonym kolorem oznaczono poprawne położenie obiektu, kolorem niebieskim natomiast położenie ramki okalającej wybrane przez agenta. Agent miał w tym momencie utrudnione zadanie, ponieważ niespodziewanie pojawił się obiekt przypominający obiekt śledzony. Dodatkowym niekorzystnym czynnikiem było przesłonięcie obiektu śledzonego.

Rysunek 37. Model Yun i in.– przykład działania (2) [17]

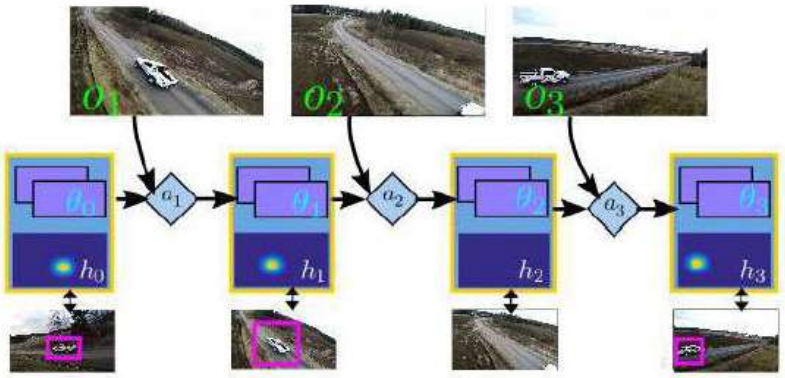
Aby zademonstrować zysk obliczeniowy na rysunku 38, zestawiono ze sobą sposób działania algorytmu „śledzenie przez detekcję" oraz model zaproponowany w [17]. Na rysunku 38 znajdują się kolejno (od lewej): obraz referencyjny, ilustracja działania modelu „śledzenie przez detekcję", ilustracja działania modelu z [17]. Niezmierną korzyścią jest redukcja niezbędnych do wykonania, skomplikowanych obliczeń.



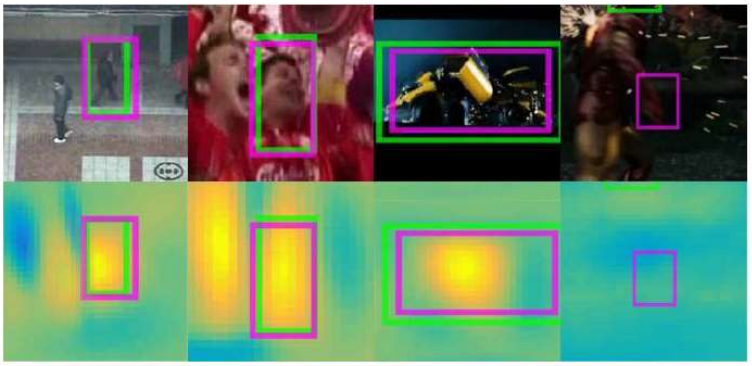
Rysunek 38. Model Yun i in.– przykład działania (3) [17]

Drugim z modeli zaproponowanych w 2017 roku był model autorstwa Suspančič oraz Ramanan [18]. Agent przez nich utworzony miał na celu śledzić obiekt przez aktualizowanie tzw. mapy ciepła (ang. *heatmap*). Miała ona reprezentować wyobrażenie agenta o miejscu przebywania śledzonego obiektu. Stan, w którym znalazł się agent, został zdefiniowany jako ostatnio wyznaczona mapa ciepła. Na podstawie jego analizy agent miał zdecydować o dwóch działaniach. Pierwsze z nich dotyczyło wyboru między kontynuacją śledzenia, a ponowną inicjalizacją mapy ciepła w przypadku, gdy

agent uznał, że zgubił obiekt. Drugie działanie odnosiło się do modyfikacji filtra śledzonego obiektu – agent musiał wybrać, czy należy go zaktualizować, czy też pozostawić bez zmian. Rolę aproksymatora funkcji wartości akcji pełniła sieć głęboka. Na początku mapa ciepła była analizowana przez dwie warstwy konwolucyjne. Następnie otrzymane w ten sposób cechy propagowano do dwóch bliźniaczych, lecz niezależnych gałęzi sieci. Zawierały one odpowiednie warstwy w pełni połączone. Każda z gałęzi odpowiadała za wyznaczenie funkcji wartości akcji dla wspomnianych wcześniej działań. W przypadku, gdy agent podjął decyzję o dalszym śledzeniu obiektu, następuje przycięcie klatki filmu do pewnego obszaru. Jest on wyznaczony przez odpowiednio wysokie wartości mapy ciepła w otoczeniu poprzednio wyznaczonej ramki. Następnie obszar ten jest skalowany do wielkości 224 x 224 piksele oraz analizowany przez konwolucyjną sieć neuronową VGG16 w celu wyprodukowania splotowej mapy cech. Kolejnym krokiem jest nałożenie filtra śledzonego obiektu na wyprodukowaną mapę cech i w efekcie uzyskanie nowej mapy ciepła. W przypadku, gdy agent będzie przeświadczony o utracie obiektu, zostaje wylosowana nowa ramka i powtarza się pozostałe kroki. W przypadku, gdy agent uzna za stosowne, by zaktualizować filtr śledzonego obiektu wybierane są z klatki filmu losowe obszary należące do ramki oraz do niej nienależące, a następnie przeprowadzana jest sama aktualizacja. Polega ona na przedstawianiu dwuwarstwowej sieci konwolucyjnej, pełniącej rolę wspomnianego filtra, wybranych wcześniej fragmentów klatki filmu wraz z informacją wartościującą. Schemat działania opisanej metody dla przykładowego filmu został zilustrowany na rysunku 39, natomiast na rysunku 40. przedstawiono wybrane mapy ciepła oraz lokalizację ramki dla przykładowych klatek. Kolorem fioletowym oznaczono ramkę zdefiniowaną przez agenta, zaś kolorem zielonym ramkę wzorcową. Na ostatnim kadrze (rys. 40.) można zaobserwować sytuację, w której agent zgubił śledzony obiekt i powinien dokonać ponownej inicjalizacji.

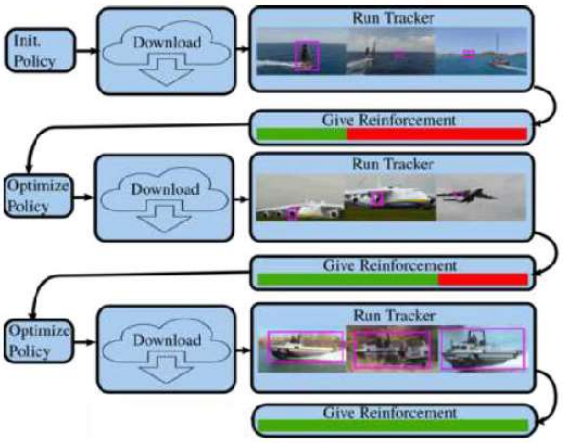


Rysunek 39. Model Suspančič & Ramanan – przykład działania (1) [18]



Rysunek 40. Model Suspančič & Ramanan – przykład działania (2) [18]

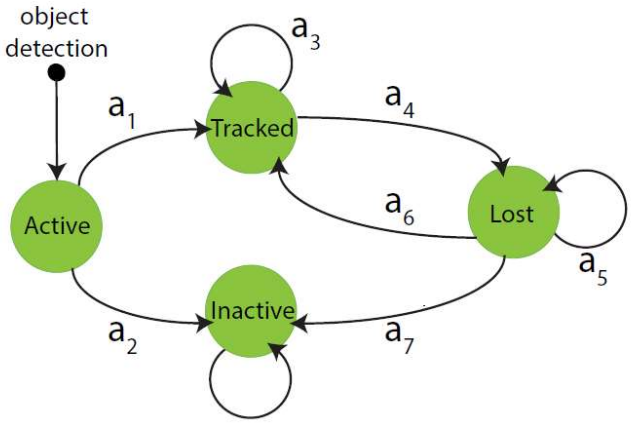
Bardzo ważną zaletą zaproponowanej metody jest możliwość przydzielania wzmocnienia w postaci binarnej po zakończonym procesie śledzenia. Tworzy to warunki, w których nie jest konieczny proces adnotacji każdej klatki filmu, a co za tym idzie, ogromnie zwiększa się ilość dostępnych potencjalnych danych uczących. Dla przykładu można pobierać pliki wideo dostępne w Internecie i pozwolić agentowi na ich analizę oraz ulepszanie strategii śledzenia. Proces ten został zilustrowany na rysunku 41.



Rysunek 41. Model Suspančič & Ramanan – schemat działania [18]

Ważnym zadaniem jest śledzenie wielu obiektów jednocześnie. W 2015 roku w [19] Xiang i in. zaprezentowali rozwiązanie pozwalające zarządzać procesem śledzenia wielu obiektów jednocześnie z uwzględnieniem czasowego znikania ich z kadru. Agent, wykorzystując dostępne metody detekcji obiektów, określał aktualny stan, w którym mógł znaleźć się śledzony obiekt. Wyróżniano więc następujące stany: Aktywny (ang. *Active*), Śledzony (ang. *Tracked*), Utracony (ang. *Lost*) oraz Nieak-

tywny (ang. *Inactive*). Należy w tym miejscu zaznaczyć, że instancji agenta jest tyle samo, co aktualnie rozpoznanych obiektów. W momencie wykrycia nowego obiektu do życia zostaje powołany kolejny agent, który rozpoczyna swoje działanie, znajdując się w stanie *Aktywny*. Gdy okaże się, że sygnał o pojawieniu się nowego obiektu był tylko pomyłką, agent przechodzi do stanu *Nieaktywny* (a_2), w którym pozostaje do zakończenia procesu śledzenia. Jeżeli jednak rzeczywiście wykryto nowy obiekt, agent przechodzi do stanu *Śledzony* (a_1) i pozostanie tam (a_3) dopóki nie zostanie utracony. W przypadku utraty obiektu (np. został on przesłonięty innym obiektem bądź opuścił kadr) agent przejdzie do stanu *Utracony* (a_4). Agent pozostaje w stanie *Utracony* (a_5), jeżeli przed upływem określonego czasu nastąpi ponowne odnalezienie obiektu. Gdy to się stanie, agent wróci do stanu *Śledzony* (a_6). Jeżeli jednak taka sytuacja nie nastąpi (upłynie określony czas), agent przejdzie do stanu *Nieaktywny* (a_7). Tak, jak wcześniej, nie opuści on tego stanu, aż do zakończenia procesu śledzenia. Opisany model działania został przedstawiony w formie grafu na rysunku 42.

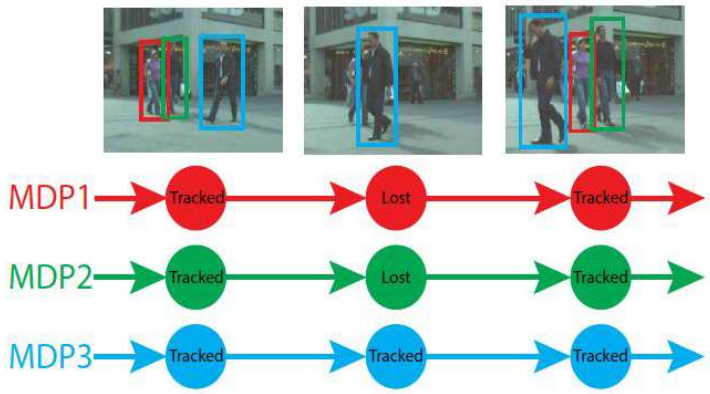


Rysunek 42. Model Xiang i in. – schemat działania [19]

Przykładowe schematy postępowania agenta zostały zaprezentowane na rysunku 43. Dodatkowo na rysunku 44 przedstawiono rezultaty działania opisanego modelu dla wybranych materiałów video. Pod każdym kadrem została podana nazwa sekwencji oraz numer klatki. Kolejnymi kolorami zostały oznaczone ramki okalające wykryte obiekty oraz wyznaczone ich trajektorie.

5. Inne algorytmy

Ostatnią grupą są algorytmy niezaliczające się do dwóch poprzednich, jednak nadal warte uwagi. Są one bowiem wykorzystywane jako narzędzie do rozwiązania problemów klasyfikowania, sterowania lub rekonstrukcji. W niniejszym rozdziale omówione zostaną implementacje algorytmu uczenia się ze wzmocnieniem do rozwiązywania wyżej wymienionych problemów.



Rysunek 43. Model Xiang i in. – przykład działania (1) [19]

Rysunek 44. Model Xiang i in. – przykład działania (2) [19]

5.1. Rozpoznawanie twarzy, zachowań oraz gestów

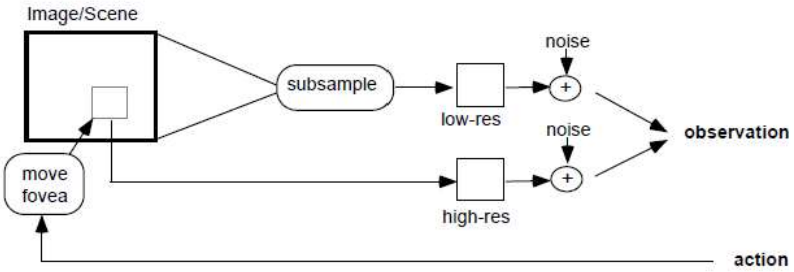
Pierwsza podgrupa zbiera zadania dotyczące rozpoznawania twarzy, zachowań bądź gestów. Już w 1996 roku badacze Darrell oraz Pentland w [20] zaproponowali system, którego zadaniem było rozpoznawanie zachowań obserwowanego człowieka na podstawie jego postawy, mimiki oraz gestów przez niego wykonywanych. Stan, w którym znajdował się agent, był określany na podstawie szeregu zmiennych. Pierwsze trzy z nich były wartościami binarnymi określającymi czy osoba jest obecna, czy została wyciągnięta ręka lewa oraz, czy została wyciągnięta ręka prawa. Wartości te były dostarczane przez pierwszą część systemu wizyjnego, jaką była kamera szerokokątna obserwująca całe pomieszczenie. Trzy kolejne wartości były typu całkowitoliczbowego i określały stan konkretnych partii ciała, tj.: wyraz twarzy (neutralna/-uśmiech/zaskoczenie), ułożenie lewej oraz ułożenie prawej ręki (neutralna/wskazująca/otwarta). Były one dostarczane przez kamerę z możliwością zmiany obserwowanego punktu przestrzeni. Zapewniała ona dokładne ujęcia wybranych partii ciała. Ostatnie

trzy wartości binarne określały miejsce skupienia się drugiej kamery (głowa/lewa dłoń/ prawa dłoń). Zadaniem agenta było wybieranie miejsca skupienia się drugiej kamery, w taki sposób, by zapewnić sobie możliwie najbardziej treściwe informacje. Gdy określone partie ciała nie były obserwowane, zmienne je opisujące były zastępowane wartościami „undefined". Przyjęto, że dany agent uczy się rozpoznawać tylko jeden rodzaj zachowania, co wymusza definicję zbioru akcji takiego, jak ten przedstawiony w tabeli 9. Wzmocnienie przydzielane agentowi miało wartość 1 w przypadku poprawnego rozpoznania zachowania, bądź – 10 w przypadku fałszywego zgłoszenia. W sytuacji, gdy agent odejmował akcję sterującą ustawieniem drugiej kamery, wartość wzmocnienia wynosiła 0.

Rok później (1997) w [21] Darrell nieco zmodyfikował omówiony model. Do zbioru akcji dołączył on „brak działania". Zmienił się również sposób wyznaczania stanu, w którym znajdował się agent. Zmiana ta polegała na sztucznym zaszumianiu obrazów z kamer przed ich przetworzeniem. Wpłynęło to korzystnie na rezultaty działania agenta. Nowe ustawienie systemu ilustruje rysunku 45, natomiast na rysunku 46 przedstawiono przykładowe kadry dla obu kamer.

Tabela 10. Model Darrel & Pentland – opis dostępnych akcji [20]

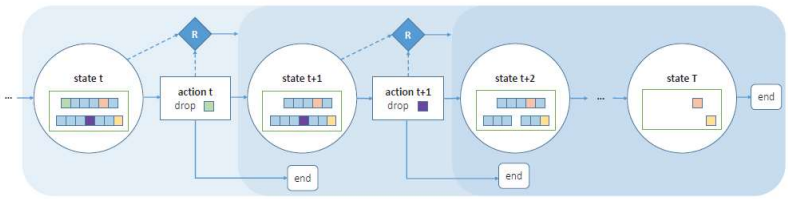
numer akcji	akcja
1	skieruj kamerę na głowę
2	skieruj kamerę na lewą dłoń
3	skieruj kamerę na prawą dłoń
4	skieruj kamerę na całą sylwetkę
5	zgłoś rozpoznanie zachowania



Rysunek 45. Model Darrel – schemat działania [21]

Rysunek 46. Model Darrel – przykład działania [21]

Innym godnym uwagi modelem jest [22] zaproponowany w 2017 roku przez Rao i in. Jego zadaniem było analizowanie dwóch sekwencji wideo w celu porównania twarzy, a w efekcie określenia, czy sekwencje przedstawiały te same osoby. Na początku każda z klatek obu sekwencji była analizowana przez głęboką sieć konwolucyjną, w celu uzyskania wektora cech ją opisującego. Następnie cechy te stały się danymi wejściowymi sieci analizującej kontekst dla każdej klatki. Ostatnim modułem była sieć głęboka będąca aproksymatorem funkcji wartości akcji. Zadaniem agenta jest w tym miejscu wybór klatki, którą należy odrzucić, by ułatwić końcowe porównanie sekwencji. Należy zauważyć, że wraz z upływem czasu liczba klatek pozostałych w sekwencjach spada, co naturalnie implikuje zmniejszenie się ilości możliwych do wykonania akcji. Wzmocnienie mogło przyjmować wartość 1 lub -1. Obliczane było na podstawie metryki uwzględniającej reprezentację ramki i jej przydatność wyznaczone przez odpowiednią sieć głęboką. Proces stopniowej eliminacji kolejnych klatek trwał do momentu, gdy agent uznał (funkcja wartości akcji była niedodatnia dla wszystkich możliwych akcji), że usuwając kolejną klatkę, nie osiągnie poprawy jakości klasyfikacji. Alternatywnym sygnałem przerywającym działanie agenta było osiągnięcie wcześniej ustalonej minimalnej ilości klatek w jednej z sekwencji. Omówiony schemat działania agenta został zilustrowany na rys. 47.

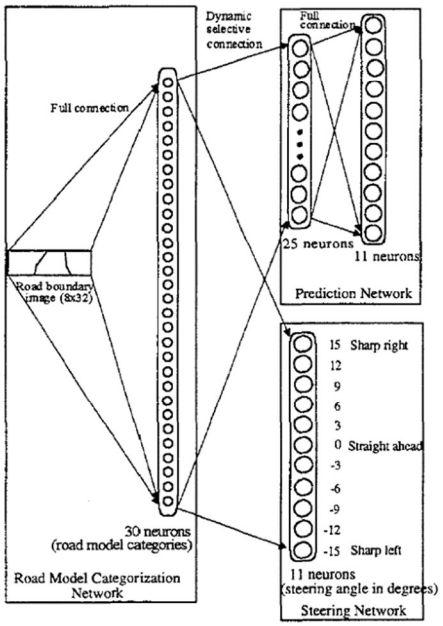


Rysunek 47. Model Rao i in. – schemat działania [22]

5.2. Zadania sterowania

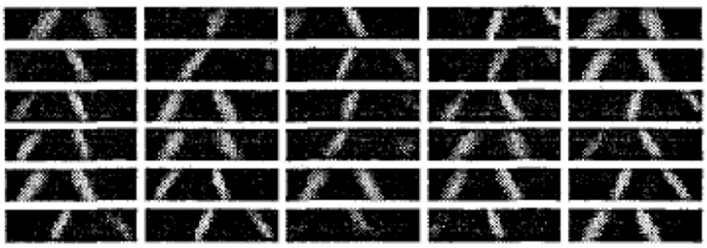
Druga podgrupa to zadania szeroko rozumianego sterowania. Już w 1995 roku w [23] Yu oraz Sethi zaproponowali sposób wykorzystania algorytmu uczenia się ze wzmocnieniem do sterowania pojazdem w taki sposób, by utrzymać pojazd w drodze. Stanem, w którym znalazł się agent, nazwano obraz drogi rejestrowanej z przodu pojazdu. Miał on wymiary 8x32 piksele i stanowił źródło danych przetwarzanych przez sieć będącą samoorganizującą się mapą (ang. *Self Organizing Map*, SOM). Wyjście sieci SOM liczyło 30 neuronów, co pozwalało na rozpoznanie 30 różnych sytuacji. Tak przygotowane dane stanowiły wejście dwóch sieci. Pierwsza z nich, w pełni połączona składała się z warstwy ukrytej o 25 neuronach oraz warstwy wyjściowej o 11 neuronach. Sieć ta obliczała pewność agenta w stosunku do podjęcia określonej akcji. Druga z nich natomiast posiadała jedną warstwę o 11 neuronach i stanowiła aproksymator funkcji wartości akcji. Naturalnym w tej sytuacji jest, że zbiór akcji liczył 11 elementów. Każda z akcji odpowiadała jednej wartości z przedziału [-15; 15] i różniła się o 3 od sąsiednich. Wartości te określały kąt skrętu pojazdu, gdzie wartość -15 to mocny skręt w lewo, a 15 to mocny skręt w prawo. Wartości wzmocnienia wynosiły 0, gdy pojazd poruszał się w granicach drogi. W przeciwnym przy-

padku wzmocnienie przyjmowało wartość -1. Tak dobór popychał agenta do trzymania się w granicach drogi oraz jak najszybszego powrotu na nią w przypadku opuszczenia jej. Schemat omówionego rozwiązania został przedstawiony na rysunku 48.

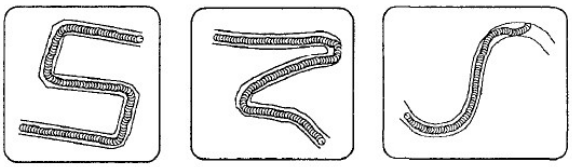


Rysunek 48. Model Yu & Sethi – schemat działania [23]

Ilustracja sytuacji drogowych, wyznaczonych w procesie nauki przez sieć SOM, została przedstawiona na rysunku 49. Dodatkowo na rysunku 50 zaprezentowano przykładowe trasy, służące nauce agenta oraz wizualizację jego podróży po nich.

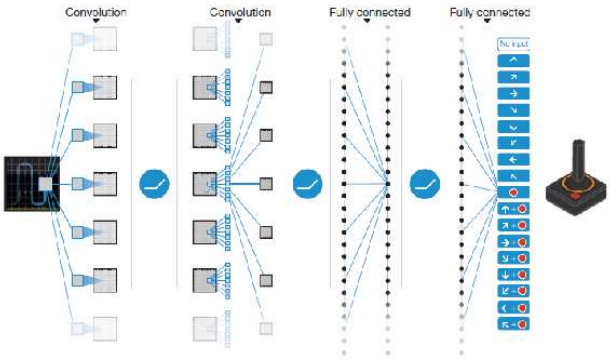


Rysunek 49. Model Yu & Sethi – wizualizacja sytuacji drogowych [23]



Rysunek 50. Model Yu & Sethi – przykłady tras [23]

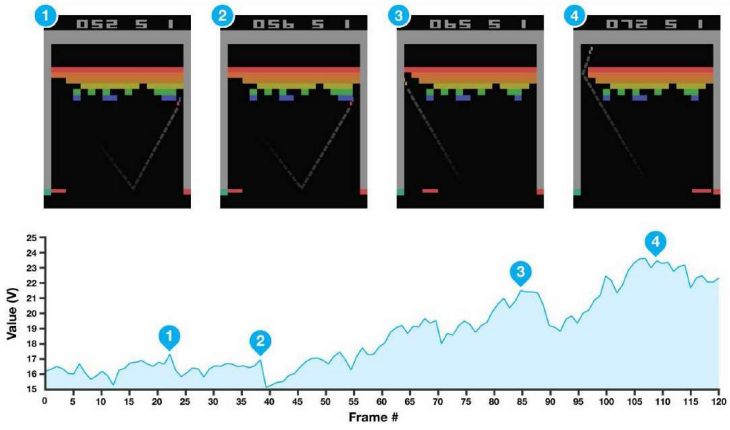
Dwie dekady później Mnih i in. w ramach projektu Google DeepMind opracowali system [24], którego zadaniem było osiągniecie strategii zapewniającej zwycięstwo w 49 grach platformy Atari. Obraz o wymiarach 210x160 pikseli, obserwowany w modelu barw RGB, był poddawany konwersji, tak by w efekcie uzyskać jego luminescencję. Następnie wynik był skalowany do wymiarów 84x84 piksele. Stanem, w którym znalazł się agent, określono macierz o wymiarach 84x84x4 zawierającą dane z 4 ostatnich klatek uzyskane przez wyżej opisany proces obróbki. Wspomniana macierz stanowiła wejście głębokiej sieci konwolucyjnej połączonej z głęboką siecią w pełni połączoną. Ostatnia warstwa liczyła 18 neuronów. Tak skonstruowana sieć stanowiła aproksymator funkcji wartości akcji. W grach platformy Atari sterowanie jest realizowane poprzez wychylenie drążka w jeden z 8 możliwych kierunków, z możliwością jednoczesnego naciśnięcia przycisku. Uwzględniając same wychylenia drążka we wszystkie możliwe strony, wychylenia drążka wraz z naciśnięciem przycisku, naciśnięcie przycisku bez wychylania drążka oraz brak działania, otrzymuje się zbiór 18 akcji. Schemat opisanego modelu wraz z ilustracją zbioru akcji został przedstawiony na rysunku 51.



Rysunek 51. Model Mnih i in. – schemat działania [24]

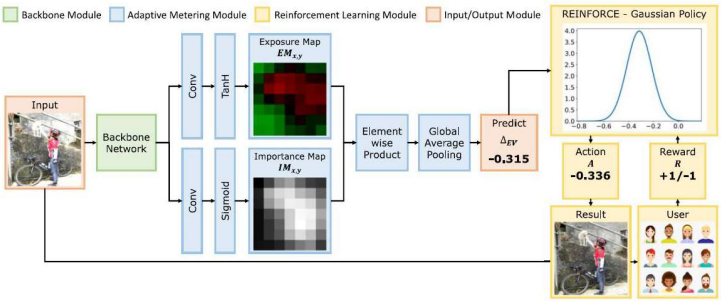
Uwzględniając zróżnicowanie w ilości przydzielanych graczowi punktów dla różnych gier, autorzy zdecydowali się utożsamić wszystkie pozytywne wartości ze wzmocnieniem równym 1, a negatywne ze wzmocnieniem równym -1. Wzmocnienie mogło zatem przyjmować wartości ze zbioru {-1, 0, 1}. Tak zaprojektowany system potrafił poprawnie określić wartość stanu, w którym się znalazł, co zostało przedstawione na rysunku 52. Pozioma oś wykresu definiuje numer klatki, natomiast na pionowej osi przedstawiono sumę zgromadzonych przez agenta wzmocnień. Dodatkowo

zaprezentowano cztery wybrane klatki wraz z oznaczeniem ich na wykresie. Należy również zaznaczyć, że rozwiązanie pozwoliło osiągnąć lepszy wynik od profesjonalnych graczy w 23 z 49 gier.



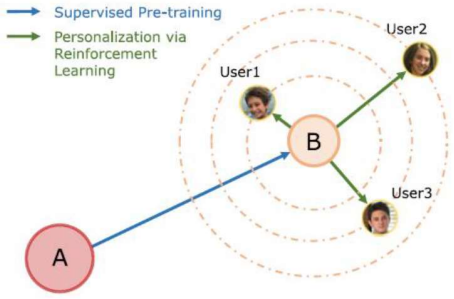
Rysunek 52. Model Mnih i in. – przykład działania [24]

Ostatnim omawianym w tym rozdziale systemem sterowania jest model przedstawiony w [25] w roku 2019 przez Yang i in. Zadaniem agenta było odpowiednie zarządzanie wartością ekspozycji, dzięki której program sterujący aparatu fotograficznego jest w stanie wyznaczyć czas oraz stopień otwarcia przesłony, a także wartość parametru ISO. W każdym kroku czasowym aktualnie widoczna klatka jest analizowana przez wybrany fragment sieci SqueezeNet. Jej wyjście jest z kolei propagowane do dwóch gałęzi, z których każda składa się z sieci konwolucyjnej. Zadaniem pierwszej z nich jest wyznaczenie mapy ekspozycji, natomiast zadaniem drugiej jest wyznaczenie mapy ważności. Rolą mapy ważności jest informacja o miejscach potrzebujących zmiany doświetlenia. Wynik mnożenia elementowego obydwu map jest przekazywany do warstwy uogólniającej, na podstawie której wyliczana jest wartość zmiany ekspozycji. Należy zwrócić uwagę, że opisana sieć pełni rolę aproksymatora funkcji wartości akcji. By przezwyciężyć problem fluktuacji wartości obliczanej przez agenta autorzy wprowadzili dodatkowy moduł wykorzystywany jedynie podczas jego nauki. Jego rolą było drobne odchylanie obliczonej przez agenta wartości zgodnie z rozkładem funkcji Gaussa. Podczas nauki agent otrzymywał wzmocnienie, które przyjmowało wartość -1, gdy wartość ekspozycji była zbyt niska, 0 gdy była odpowiednia oraz 1, gdy wartość ekspozycji była zbyt wysoka. Opisany model został zilustrowany na rysunku 53. Bloki wejścia oraz wyjścia sieci oznaczono jasnym kolorem czerwonym; sieć przetwarzającą wstępnie dane wejściowe kolorem zielonym; bloki wyznaczające mapę ekspozycji, mapę ważności oraz uogólnienie kolorem niebieskim; moduł wykorzystywany tylko podczas nauki kolorem żółtym. Warto również zaznaczyć, że wartości dodatnie w mapie ekspozycji zaznaczono kolorem zielonym, ujemne czerwonym, a zerowe czarnym. Jeżeli zaś chodzi o mapę ważności, kolor biały oznacza regiony ważne, natomiast czarny te mało ważne.



Rysunek 53. Model Yang i in. – schemat działania [25]

Bardzo ważną zaletą omówionego rozwiązania jest możliwość personalizacji zachowania agenta. Rola użytkownika ograniczy się wtedy tylko do udzielania informacji, czy poziom ekspozycji jest zbyt niski, wystarczający czy za wysoki. Naturalnie użytkownik otrzymałby agenta wcześniej przyuczonego w celu tylko jego doskonalenia. Model ten został zilustrowany na rysunku 54. Autorzy prowadzili badania na dwóch urządzeniach mobilnych: iPhone 7 oraz Pixel. Wykonali oni również zestawienie ilustrujące różnice w działaniu natywnej aplikacji (AP) aparatu, w każdym z tych urządzeń oraz rezultaty pracy wyszkolonego agenta. Zestawienie to zostało zaprezentowane na rysunku 55. Kolejne kolumny odpowiadają: AP-iPhone 7, agent- iPhone 7, AP-Pixel, agent-Pixel.

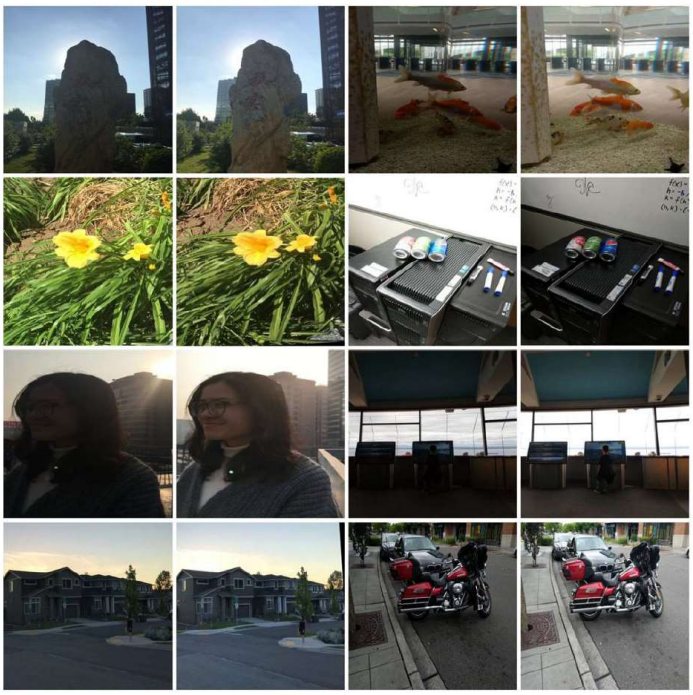


Rysunek 54. Model Yang i in. – schemat modelu personalizacji [25]

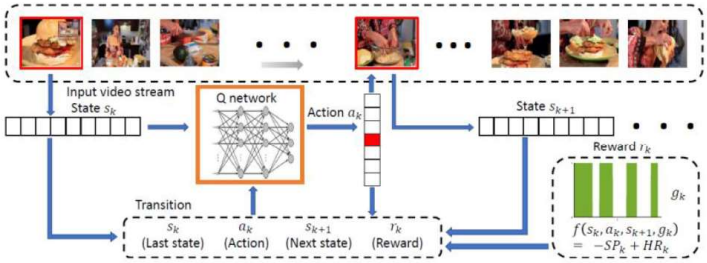
5.3. Ekstrakcja istotnych ramek filmu

Kolejną, godną uwagi implementacją algorytmu uczenia się ze wzmocnieniem, jest wykorzystanie agenta do przyspieszania (przewijania) obserwowanego pliku wideo. Głównym celem agenta jest więc analiza oglądanych klatek filmu oraz podejmowanie decyzji, ile następnych klatek należy pominąć, by użytkownik oglądał możliwie najwięcej treści, którymi jest zainteresowany. W 2018 roku w [26] Lan i in. przedstawili system, który realizował opisane zadanie. Stan, w którym znalazł się agent, określony był przy pomocy wektora cech, wyekstrahowanych z aktualnie oglądanej klatki filmu. Wektor ten był później przekazywany na wejście głębokiej, w pełni połączonej sieci neuronowej. Sieć ta pełniła rolę aproksymatora funkcji wartości akcji. Kolejne, do-

stępne dla agenta akcje odpowiadały pominięciu określonych ilość klatek filmu. Wartość wzmocnienia była określana jako różnica między metryką określającą nagrodę za poprawne trafienia (ang. HR – *Hit Reward*) a metryką określającą karę za pominięte istotne klatki filmu (ang. SP – *Skip Penalty*). Graficznie zilustrowano opisany model na rysunku 56.



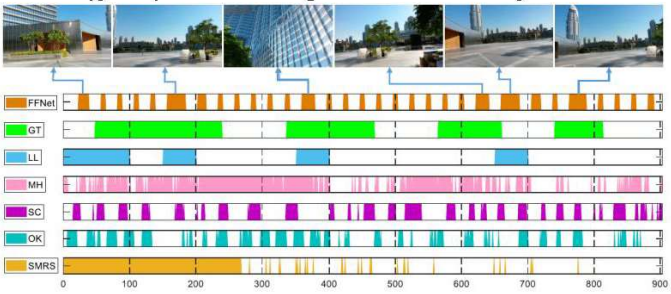
Rysunek 55. Model Yang i in. – przykład działania [25]



Rysunek 56. Model Lan i in. – schemat działania [26]

Zaproponowany model został również porównany z innymi istniejącymi rozwiązaniami. Na rysunku 57. w kolejnych wierszach przedstawiono: rezultat działania zaproponowanej metody (FFNet), wzorcowe wybranie istotnych klatek filmu (GT –

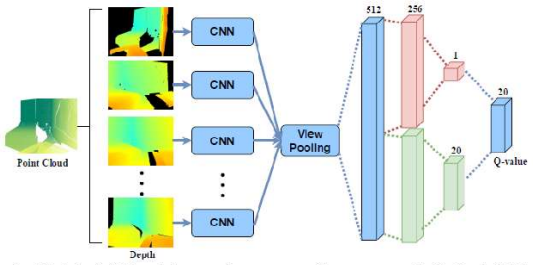
Ground Truth), działanie metody LiveLight (LL), Microsoft Hyperlapse (MH), Spectrual Clustering (SC), Online K-means (OK) oraz Sparse Modeling Representative Selection (SMRS). Można dostrzec, że algorytm wykorzystujący paradygmat uczenia się ze wzmocnieniem osiągał lepsze wyniki od konkurencji. Dodatkowo jego niewątpliwą zaletą jest konieczność przetwarzania tylko części danych, podczas gdy wszystkie algorytmy konkurencyjne są zmuszone do przetworzenia każdej klatki filmu.



Rysunek 57. Model Lan i in. – przykład działania [26]

5.4. Rekonstrukcja sceny 3D

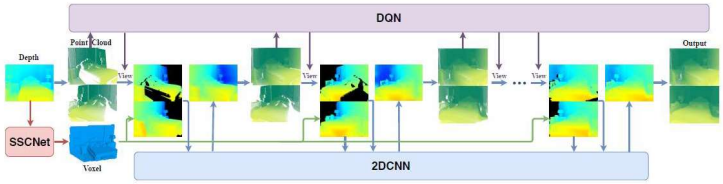
Ostatnim rozważanym zadaniem jest rekonstrukcja sceny 3D. W 2019 roku w [27] Han i in. zaprezentowali system, wykorzystujący paradygmat uczenia się ze wzmocnieniem realizujący wspomniane zadanie. Stanem, w którym znalazł się agent, nazwano chmurę punktów wygenerowaną na podstawie inicjalizującego obrazu głębi. Na podstawie aktualnej chmury punktów, przy pomocy sieci MVCNN generowany był zbiór map głębi z różnych punktów widzenia, a następnie wyznaczane było ujęcie, które zawierało najwięcej braków. Liczba dostępnych ujęć, a tym samym możliwych do podjęcia przez agenta akcji, wynosiła 20. Były one równomiernie rozłożone na dwóch okręgach – jednym poziomym, drugim zaś również poziomym, lecz ustawionym na pewnej wysokości, tak by kamera kierowana na obiekt była ustawiona pod kątem 45°. Schemat budowy opisanego modułu, będącego aproksymatorem funkcji wartości akcji przedstawiono na rysunku 58.



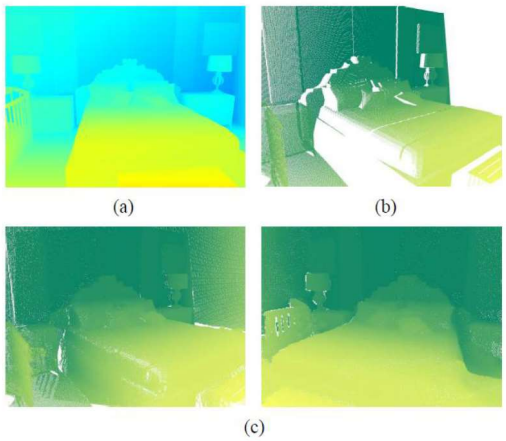
Rysunek 58. Model Han i in. – schemat aproksymatora Q-funkcji [27]

Dodatkowo w procesie rekonstrukcji wykorzystano sieć SSCNet, potrafiącą wygenerować mapę wokseli oraz sieć 2DCNN, której zadaniem było odpowiednie uzupełnianie map głębi. W trakcie procesu nauki agentowi było przydzielane wzmocnienie

uwzględniające ilość brakujących w chmurze punktów oraz ich zmianę w kolejnych krokach czasowych. Dzięki takiemu podejściu agent był nastawiony na wybieranie tych kątów widzenia, które zapewniały rekonstrukcję, jak największej liczby punktów dla każdego kroku czasowego. Ilustracja działania opisanego modelu została przedstawiona na rysunku 59. By natomiast zwizualizować rezultaty działania algorytmu, na rysunku 60. przedstawiono: (a) obraz głębi będący wejściem algorytmu, (b) chmurę punktów wyznaczoną na podstawie wejściowego obrazu głębi, (c) widok z dwóch różnych stron na chmurę punktów po zakończeniu procesu rekonstrukcji.



Rysunek 59. Model Han i in. – schemat działania [27]



Rysunek 60. Model Han i in. – przykład działania [27]

6. Podsumowanie

W opracowaniu wyjaśniono paradygmat uczenia się ze wzmocnieniem oraz omówiono szereg różnego rodzaju implementacji tego algorytmu do zadań szeroko pojętego przetwarzania obrazów. Ponadto wyróżniono podział i wprowadzono pewną systematykę. Wyróżnione grupy oraz konkretne zadania wraz z listą publikacji oraz lat ich wydania zostały przedstawione na rysunku 61.

Utworzone opracowanie pozwala również zauważyć różne metody automatyzacji zadań przetwarzania obrazu, które często są elementem składowym systemów realizujących bardziej skomplikowane wyzwania, z różnych dziedzin nauki. Jest to dowodem interdyscyplinarności aplikacji algorytmu uczenia się ze wzmocnieniem. W treści opracowania uwypuklono również korzyści płynące ze stosowania paradygmatu uczenia się ze wzmocnieniem, takie jak:

- krótszy czas potrzebny na osiągnięcie zamierzonego efektu;
- łatwość wizualizacji oraz interpretacji wypracowanej przez agenta strategii;
- względna łatwość implementacji;
- możliwość stosowania tam, gdzie klasyczne metody uczenia pod nadzorem nie mogą być wykorzystane;
- porównywalna lub wyższa jakość otrzymywanych rezultatów.

Rysunek 61. Klasyfikacja implementacji algorytmu uczenia się ze wzmocnieniem do zadań przetwarzania obrazu [opracowanie własne]

Niezaprzeczalną korzyścią płynąca z opracowanego materiału jest także ułatwienie dostępu do informacji na temat paradygmatu uczenia się ze wzmocnieniem oraz jego praktycznego wykorzystania w różnych sferach technicznych. Brak dostępności podobnego opracowania był dodatkową motywacją. Może ono również stanowić źródło wiedzy i inspiracji dla badaczy chcących budować podobne rozwiązania. Mimo, iż algorytm uczenia się ze wzmocnieniem nie jest tak popularny jak inne metody uczenia maszynowego, to zainteresowanie nim zdecydowanie wzrosło w ostatniej dekadzie. Szczególnie ważne dla badaczy stały się rozwiązania wspomagające medycynę oraz wszelkie zadania pozwalające konstruować autonomiczne roboty lub pojazdy (np. detekcja oraz śledzenie obiektów, czy algorytmy sterowania procesami).

Podziękowania

Szczególne podziękowania składam na ręce dr hab. inż. Romana Zajdla.

Literatura

1. Cichosz P., *Systemy uczące się*, Wydawnictwa Naukowo-Techniczne, Warszawa 2000.
2. Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT press, 2012.
3. Wang, Ye, Mueller, Fessler, *Image reconstruction is a new frontier of machine learning*, IEEE transactions on medical imaging, 37(6), 2018, s. 1289-1296.

4. Xie, Xu, Chen, *Image denoising and inpainting with deep neural networks*, https://openaccess.thecvf.com/content_iccv_2017/html/Supancic_Tracking_as_Online_ICCV_2017_paper.html, 06.04.2021.
5. Furuta, Inoue, Yamasaki, *PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing*, https://ieeexplore.ieee.org/abstract/document/8936404, 06.04.2021.
6. Yu, Dong, Lin, Change Loy, *Crafting a toolchain for image restoration by deep reinforcement learning*, https://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Crafting_a_Toolchain_CVPR_2018_paper.html, 06.04.2021.
7. Park, Lee, Yoo, So Kweon, *Distort-and-recover: Color enhancement using deep reinforcement learning*, https://openaccess.thecvf.com/content_cvpr_2018/html/Park_Distort-and-Recover_Color_Enhancement_CVPR_2018_paper.html, 06.04.2021.
8. Kosugi, Yamasaki, *Unpaired image enhancement featuring reinforcement-learning-controlled image editing software*, https://ojs.aaai.org/index.php/AAAI/article/view/6790, 06.04.2021.
9. Shen, Gonzalez, Chen, Jiang, Jia, *Intelligent parameter tuning in optimization-based iterative CT reconstruction via deep reinforcement learning*, https://ieeexplore.ieee.org/abstract/document/8331966, 06.04.2021.
10. Li, Feng, An, Ng, Zhang, *MRI Reconstruction with Interpretable Pixel-Wise Operations Using Reinforcement Learning*, https://ojs.aaai.org//index.php/AAAI/article/view/5423, 06.04.2021.
11. Vassilo, Heatwole, Taha, Mehmood, *Multi-Step Reinforcement Learning for Single Image Super-Resolution*, https://openaccess.thecvf.com/content_CVPRW_2020/html/w31/Vassilo_Multi-Step_Reinforcement_Learning_for_Single_Image_Super-Resolution_CVPRW_2020_paper.html, 06.04.2021.
12. Hung, Zhang, Shen, Lin, Lee, Yang, *Learning to blend photos*, https://openaccess.thecvf.com/content_ECCV_2018/html/Wei-Chih_Hung_Learning_to_Blend_ECCV_2018_paper.html, 06.04.2021.
13. Karayev, Baumgartner, Fritz, Darrell, *Timely object recognition*, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.359.8630&rep=rep1&type=pdf, 06.04.2021.
14. Caicedo, Lazebnik, *Active object localization with deep reinforcement learning*, https://openaccess.thecvf.com/content_iccv_2015/html/Caicedo_Active_Object_Localization_ICCV_2015_paper.html, 06.04.2021.
15. Mathe, Pirinen, Sminchisescu, *Reinforcement learning for visual object detection*, https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Mathe_Reinforcement_Learning_for_CVPR_2016_paper.html, 06.04.2021.
16. Kong, Xin, Wang, Hua, *Collaborative deep reinforcement learning for joint object search*, https://openaccess.thecvf.com/content_cvpr_2017/html/Kong_Collaborative_Deep_Reinforcement_CVPR_2017_paper.html, 06.04.2021
17. Yun, Choi, Yoo, Yun, Young Choi, *Action-decision networks for visual tracking with deep reinforcement learning*, https://openaccess.thecvf.com/content_cvpr_2017/html/Yun_Action-Decision_Networks_for_CVPR_2017_paper.html, 06.04.2021.
18. Supancic III, Ramanan, *Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning*, https://openaccess.thecvf.com/content_iccv_2017/html/Supancic_Tracking_as_Online_ICCV_2017_paper.html, 06.04.2021.
19. Xiang, Alahi, Savarese, *Learning to track: Online multi-object tracking by decision making*, https://www.cv-foundation.org/openaccess/content_iccv_2015/html/Xiang_Learning_to_Track_ICCV_2015_paper.html, 06.04.2021.
20. Darrell, Pentland, *Active gesture recognition using partially observable Markov decision processes*, https://ieeexplore.ieee.org/abstract/document/547315, 06.04.2021

21. Darrell, *Reinforcement learning of active recognition behaviors*, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.1972&rep=rep1&type=pdf, 06.04.2021.
22. Rao, Lu, Zhou, *Attention-aware deep reinforcement learning for video face recognition*, https://openaccess.thecvf.com/content_iccv_2017/html/Rao_Attention-Aware_Deep_Reinforcement_ICCV_2017_paper.html, 06.04.2021.
23. Yu, Sethi, *Road-following with continuous learning*, https://ieeexplore.ieee.org/abstract/document/528317, 06.04.2021
24. Mnih, Kavukcuoglu, Silver, Rusu, Veness, Bellemare, Graves, Reidmiller, Fidjeland, Ostrovski, Petersen, Brattie, Sadik, Antonoglou, King, Kumaran, Wierstra, Legg, Hassabis, *Human-level control through deep reinforcement learning*, https://www.nature.com/articles/nature14236, 06.04.2021.
25. Yang, Wang, Vesdapunt, Guo, Kang, *Personalized exposure control using adaptive metering and reinforcement learning*, https://ieeexplore.ieee.org/abstract/document/8437183, 06.04.2021.
26. Lan, Panda, Zhu, Roy-Chowdhury, *Ffnet: Video fast-forwarding via reinforcement learning*, https://openaccess.thecvf.com/content_cvpr_2018/html/Lan_FFNet_Video_Fast-Forwarding_CVPR_2018_paper.html, 06.04.2021.
27. Han, Zhang, Du, Yang, Yu, Pan, Yang, Liu, Xiong, Cui, Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image, https://openaccess.thecvf.com/content_CVPR_2019/html/Han_Deep_Reinforcement_Learning_of_Volume-Guided_Progressive_View_Inpainting_for_3D_CVPR_2019_paper.html, 06.04.2021.

Wykorzystanie algorytmów uczenia się ze wzmocnieniem do przetwarzania obrazów

Streszczenie
Opracowanie jest przeglądem istniejących rozwiązań wykorzystujących różne implementacje algorytmu uczenia się ze wzmocnieniem do szeroko pojętych zadań przetwarzania obrazów. Wyróżniono w nim podział na grupy algorytmów: bezpośrednio modyfikujących obrazy, realizujących zadanie wykrywania oraz śledzenia obiektów, a także pozostałe, analizujące treści obrazów w celu realizacji określonych zadań. Wprowadzono również pewną systematykę, nakreślono sposoby automatyzacji procesów obróbki obrazu z wykorzystaniem paradygmatu uczenia się ze wzmocnieniem oraz wyróżniono zalety jego stosowania. Opracowanie wskazuje również kierunki rozwoju badań na przestrzeni ostatniej dekady.
Słowa kluczowe: uczenie się ze wzmocnieniem, przetwarzanie obrazów, praca przeglądowa

Usage of reinforcement learning algorithms in images processing

Abstract
This work is an overview of existing solutions which use different reinforcement learning algorithm implementations for image processing tasks in the broad sense. It distinguishes a division into groups of algorithms: those that directly modify images, those that perform object detection and tracking tasks, and others that analyse image content for specific tasks. It also introduces some systematics, outlines the ways to automate image processing processes using the reinforcement learning paradigm and distinguishes the advantages of its application. The study also indicates the research directions over the last decade.
Keywords: reinforcement learning, image processing, review

Reinforcement learning in car control: A brief survey

Dawid Kalandyk
Doctoral School of Engineering and Technical Sciences
at the Rzeszow University of Technology
Rzeszów, Poland
ORCID 0000-0002-7317-5499

2021 Selected Issues of Electrical Engineering and Electronics (WZEE) | 978-1-6654-2164-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/WZEE54157.2021.9576838

Abstract — **Autonomous Vehicles (AVs) from a far foggy dream and cartoon fiction, are becoming more and more real nowadays. Realizing this dream can have many benefits both in terms of improvement of the traffic flow as well as reduction of pollution and greenhouse gases. Researchers are continuously developing existing technologies and creating new solutions to achieve stable and safe control algorithms for AVs. Encouraged by their work, the author wanted to make his own contribution to build a better transportation system and taking care of the environment. This paper is a first step – a summary of currently published ideas on the application of reinforcement learning algorithms to the problem of AVs. The paper breaks down the tasks to their nature and level of difficulty, also shows various models of the environment, actions and reinforcement signals outlined at just under 50 papers. Additionally, the number of potential development areas has been highlighted.**

Keywords — review, reinforcement learning, autonomous driving, artificial intelligence, machine learning

I. Introduction

The automotive industry is crucial to the vast majority of people in the world. Both directly by making it possible to travel to work ever more quickly and comfortably, and indirectly by reducing noise levels and greenhouse gas emissions. Addressing such problems is a priority for many scientists. The main task is therefore to develop an increasingly higher level of Autonomous Vehicles (AVs). The current division is as follows. Level 0 vehicles are equipped with a different warning systems which can only temporarily take control over particular car mechanisms (e.g. ABS or ESC). Level 1 AVs have systems that share control of the vehicle with the driver. A fine example here is the Adaptive Cruise Control (ACC) which is capable of maintaining a certain speed while the driver keeps control of the steering wheel. This level is also called the "hands on" level. Level 2 AVs are known for their ability to perform simple tasks such as driving on the motorway or parking. They are also called "hands off" vehicles apart from the fact that the driver must monitor their correct functioning and intervene if necessary. Level 3 AVs can handle almost all tasks and emergency situations. They are called "eyes off" vehicles because the driver can do other things, such as chatting with friend or reading a book. He will be asked for taking over a control in minority cases such as stuck in traffic jam. Level 4 AVs are able to handle all situations and manage driver to not care about driving, which led them to be called "mind off" vehicles. They are equipped with all steering devices which may be used at the request of a driver. It is also worth noting that they are limited to certain area or other conditions. In such case they will safely abort the trip by speed reduction and parking in a safe place. At the top there are level 5 AVs which are capable of handling all possible scenarios and weather conditions. They also do not need steering devices for the driver's use because they are fully autonomous.

To achieve this level of autonomy it is necessary to use artificial intelligence methods. Machine Learning (ML) algorithms have developed rapidly over the last 20 years. Many modifications and learning algorithms for Artificial Neural Networks (ANNs) have also been developed. This offers considerable opportunities for researchers working on the problem of AVs. However, it should not be forgotten that ANNs require large amount of training data, and therefore pose difficulties in collecting and managing them.

A. Reinforcement Learning

Instead of dealing with a vast amount of training data one may apply the Reinforcement Learning (RL) paradigm which provides different kinds of solution – learning by experiencing. RL refers to the idea of placing a so-called agent in a specific environment. In each time step (t) the agent is allowed to perform one of the available actions. Based on the environment state (s_t) observation the agent chooses an action (a_t) to perform. This affects the environment and transits the agent and environment to a new state (s_{t+1}). The agent also receives a reward (r_t) which evaluates agents moves. The agent uses a strategy (π) and starts in state (s) is expected to accumulate a total reward expressed by (1), which is called value function. The agent uses a strategy (π), starts in state (s) and chooses there the action (a) is expected to accumulate a total reward expressed by (2), which is called action value function.

$$V^{\pi}(s) = \mathrm{E}_{\pi}[\Sigma_{t=0}(\gamma^t \cdot r^t) \text{ where } s_0 = s] \quad (1)$$

$$Q^{\pi}(s, a) = \mathrm{E}_{\pi}[r_0 + \Sigma_{t=1}(\gamma^t \cdot r^t) \text{ where } s_0 = s,\ a_0 = a] \quad (2)$$

It is also worth noting that for manipulating the agent's so-called foresight discounting factor γ is used. Its value shall be within range (0;1]. In addition, it should also be said that the choice of actions in the next steps is made based on a greedy algorithm that prefers actions with a higher Q-value. Depending on the needs one can use plenty of RL algorithms that were brought together in [1]. By using ANNs as a function approximator it is possible not only to select discrete action but also to determine continuous values. It is called Deep Reinforcement Learning (DRL). Such possibility is a great opportunity to apply RL algorithms in complex control tasks, where both precision and strategy planning are needed to accomplish them.

B. Motivation and Contributions

The development of RL algorithms and other ML methods mentioned at the beginning of this paper is supported by many publications. In 2015, the Google DeepMind team developed [2] – an algorithm achieving results comparable to experts while playing 49 Atari games. Images, displayed on the screen during gameplay, were the only input of the algorithm. That paper encouraged scientists to apply RL algorithms in image processing tasks. In 2018 Hung et al. proposed [3] which was

978-1-6654-2164-5/21/$31.00 ©2021 IEEE

managed to blend photos and create beautiful artworks. In 2019 Furuta et al. introduced PixelRL [4]. By treating each pixel of an image as a separate agent, the algorithm was able to de-noise images, remove captions from them, or even alter images according to a pattern defined during learning. Finally in 2020 Li et al. using a similar method. proposed [5] for reconstructing Magnetic Resonance Images (MRI). Steering tasks that require planning were not left out either and in 2019 Spielberg et al. presented [6] where they consider 3 tasks: paper machine controller, high-purity distillation column controller and Air Conditioning controller. Main advantage of their approach was ability to work and learn at the same time even after deployment. In 2020, [7] was published – the summary of the achievements of Artificial Intelligence (AI) algorithms in determining vehicle trajectories. Due to the lack of a review on RL algorithms for broadly defined driving tasks, the author decided to write this paper as a first step to contribute to creating new solutions in AVs field of study. The main contributions of this work are as follow:

- Systematise the control tasks of AVs and the work devoted to them that uses RL algorithms.
- Identify and characterise different ways of representing the environment, the set of actions and the reinforcement signal in AVs control tasks.
- Identify and highlight areas of potential development of RL algorithms applied in AVs tasks.

The remainder of this paper is structured as follows. Section II presents a breakdown of AV's tasks by level of difficulty and specificity. It also introduces new tasks which have not been considered in summarised papers yet. Section III deals with the implementation specifics of the RL algorithm in AVs tasks. In addition, it summarises the impact of the choice of the action set and the reinforcement function on the quality of the algorithm and its learning process. Furthermore, it also contains a sketch of the author's own idea as a plan for further work. Finally, Section IV concludes the paper.

II. AV'S CONTROL TASKS

It is obvious that driving a car is very complex and requires attention to many aspects. Therefore, it is not trivial to create an algorithm that is up to the challenge of driving AV. To the best of the author's knowledge, none of the automotive companies offers their clients level 3 and higher AVs. This only confirms the difficulty of creating a safe car autopilot. A strategy was therefore adopted to divide the task of controlling an autonomous car into individual subtasks which can be done separately. The following subsections will discuss both the mentioned distribution of AVs control tasks in the summarised articles as well as new tasks proposed by the author.

A. Simple AVs Control Tasks

Consideration should start with basic but not necessarily simple AVs control tasks. The first one is safe parking which was considered in [8] by Maravall et al. in 2003. By design, the vehicle was supposed to move constantly and slowly backwords. The algorithm controlled only steering angle of the front wheels.

The next task is to keep the speed of the car as close as possible to the desired one. In 2018 Buechel et al. presented [9] – a Predictive Reinforcement Learning Controller with Incorporated Knowledge (PRLC-A), that is able to fluently change speed to desire one while driving in car park. The same year Aradi et al. introduced [10] based on Policy Gradient (PG) algorithms. They were able to keep desired speed while driving on a highway. In 2020 Puccetti et al. presented [11] – Adaptive Cruise Control (ACC). Using Actor-Critic (AC) methods and Recursive Neural Network (RNN) they achieved smooth longitudinal control. A very similar task is to keep an appropriate distance between the agent and the following car. In 2019 Lin et al. presented [12] based on Deep Deterministic Policy Gradient (DDPG) algorithm and car model taking into account vehicle response delays. In 2020 Wei et al. introduced [13] which uses Double Deep Q Reinforcement Learning (Double DQRL) algorithm, images from the front camera and current speed info to achieve the goal. In the same year, Cao et al. presented [14] that uses RL based control of Imitative Policies (IP) for safe near-accident driving. One of the tasks was to keep a safe distance while following an unexpectedly halting car.

The last of the first set of tasks is lane-keeping. This is the most popular task that has been considered in 25 papers. First one [15] presented in 1995 by Yu and Sethi that uses Self Organizing Maps (SOM) to extract knowledge from front camera road images. After almost 20 years of stagnation in this field, in 2012 Lange et al. presented [16]. They used birds-eye view images of toy carts track as input and the ClusterRL method for training. For years later, in 2016 Sallab et al. presented [17] which uses the Deep Deterministic Actor Critic (DDAC) algorithm. Also in 2016, Yu et al. introduced [18] using Double DQRL. Year after an innovative method was presented in [19] by Sallab et al. – they used the Attention Glimpse Network (AGN). In year 2018 seven works were published. First one is [20] where Jaritz et al. using Asynchronous Actor-Critic Agent (A3C) algorithm and visual input make it possible to even learn agent how to drift on sharpening bends of the track. Second one was [21] introduced by Chishti et al. who used Deep Q Reinforcement Learning (DQRL) and images from real driver trips as well as the You Only Look Once (YOLO) algorithm to recognize road signs. The third one was [22] presented by Wang et al. They used the DDPG algorithm and a number of metrics for training the agent. The fourth paper was [23] introduced by Wu and Li which uses the Aggregated Multi-deep Deterministic Policy Gradient (AMDDPG) method relying on averaging agents knowledge. The fifth one is [24] presented by Zhang et al. They used images from ordinary drivers trips and the Double DQRL algorithm for training the agent. The penultimate paper [25] written by Feher et al. uses the DQRL method. It also introduces software for generating random paths for training and testing the agent. Last one [26] presented by Kesleman et al. uses a fusion of DQRL and A* algorithm. Input data for the algorithm were birds-eye view of the road with velocity and steering angle incorporated into that image as colourful horizontal stripes. The year 2019 was also full of ideas. The first one is [27] presented by Kendall et al. who successfully attempted to learn the agent how to drive a real car in a time shorter than 24 hours. They used both the DDPG algorithm combined with Variational Autoencoder (VAE) as well as the car's frontal camera image. The next idea was [28] introduced by Liang et al. which uses Federated Transfer Reinforcement Learning (FTRL). The main clue was to gather knowledge from one agent interacting with a simulator and 3 agents driving real small vehicles. Furthermore, Zhu and Zhao presented [29] where they used PILCO algorithm and visual

input data from the driver's perspective to train the agent. Next paper [30] introduced by Zhang et al. uses Double DQRL and a real small car model. In the year 2020 three papers were published. First of them is [31] where Ke et al. compared DDPG and Soft Actor Critic (SAC) algorithms. The second one is [32] presented by Morais et al. who used a fusion of Proximal Policy Optimization (PPO), Evolutionary Algorithm with Numerical Differentiation (EAND) and Robust Recursive Linear Quadratic Regulator (RRLQR). The last one is [33] written by Youssef and Houda who used DDPG and visual input for training the agent. Finally, in 2021, Fuchs et al. presented [34] that uses SAC algorithm and very realistic simulator Gran Turismo Sport (GTS) for training. Agent using only couple of rangefinders has overcome all of the world's best players. The lane-following task is considered by 3 more works [10, 35, 36] but not directly so they will be discussed in subsequent subsections.

B. Advanced AVs Control Tasks

The next group of tasks that should be considered are advanced AVs tasks. The first of them is on-ramp merging. In 2016 Shalev-Shwartz et al. presented [37] that uses Multi-Agent Learning (MAL) for solving merging problems in the "2 – 2" road link. The authors assumed that each agent knows other agents driving intentions. In 2017 Wang and Chang introduced [38] where QRL and Long Short-Term Memory (LSTM) were used for highway on-ramp merging. In the same year Mukadam et al. presented [39] achieving the goal by QRL and Q-masking methods usage. In 2018 Fayjie et al. introduced [40]. They used DQRL for training the agent who moved in an urban environment. The second task which is very similar to the previous one is overtaking the following vehicle. Many of the already mentioned works [10, 18, 21, 23, 24, 26, 39, 40] consider this task. In addition to this in 2011, Ngai and Yung presented [36] where the strategy of Multiple-Goal Reinforcement Learning (MGRL) was used. In 2018 Wang et al. introduced [41] that uses the DQRL algorithm. In 2019 [42] by Chen et al. came up. The authors used Deep Recursive Deterministic Policy Gradient (DRDPG) algorithm supported by LSTM and Convolutional Neural Network (CNN) combination. The same year also [35] by Nageshrao et al. was published. They used Double DQRL algorithm for training the agent who was driving on the highway. Two years after Huynh et al. presented [43] trying to tackle the problem of highway driving by using the PG algorithm.

The next advanced task is the ability to safely going through intersections. It was considered in mentioned [21]. Also in 2019, Bacchiani et al. presented [44] that uses the A3C algorithm. Agent was driving through roundabout using discrete birds-eye view image. In 2020 Wang et al. introduced [45] where the idea of collecting near-intersection sensors info and mapping it into a list of objects was used. The connected task is learning how to pass an area where occlusion occurs. The next three cited papers consider both this task as well as the intersection passing task. The first of them is [46] from 2018 made by Isele et al. where the DQRL algorithm was used. The authors proposed 3 scenarios: a) the agent decided only when to start going through the intersection, b) the agent decides whether to stop or slowly approach the intersection, the third choice is to go all the way through in any time while stopping or approaching, c) the agent has full control over the vehicle acceleration and can control it in real time. The second paper is [47] introduced by Pusse and Klusch in 2019. They used a combination of A3C and heaps making so-called HyLEAP algorithm. They focused on accidents with pedestrians. The last work is [14] presented in 2020 by Cao et al. It uses the PPO algorithm for training the agent whose work is to choose driver type (timid or aggressive). By switching them the agent can ensure safe driving. The last advanced task is to correctly react to unusual and dangerous situations. It is similar to previous one and is also considered by [47] and [14]. Examples of such situations are: unexpected pedestrian crossing road in the wrong place, following vehicle immediately halting with or without reason (e.g. while dear entered the road), a vehicle coming from the opposite direction enters a frontal collision trajectory and other vehicle passing intersection while having no permission. In addition to the mentioned articles, this task is also addressed by [48] presented in 2020 by Kontes et al. Using the DDPG algorithm they approached to avoid an unexpected obstacle while driving through a tunnel at high speed. Apart from driving safely the second goal is to reach the destination point in a reasonably short time. Many works do not consider it because of safety priority. Despite that, the number of papers considering car racing or highway traffic believes that safe driving is as important as to make a trip as quick as possible. Here is a list of these works that have already been mentioned: [16, 20, 22, 23, 28, 29, 31, 34, 39, 42, 43, 45].

C. Other Control Tasks

In addition to simple or advanced vehicle control tasks there are also tasks that include controlling processes or traffic rules. This subsection will distinguish four kinds of such tasks. The first of them is controlling electric engine power in hybrid engine vehicles. Wise use of resources contributes to a reduction in the amount of fuel needed and hence lower greenhouse gases emission. Saved energy can also be used in many situations e. g. while performing overtaking manoeuver where quick acceleration is required. In 2019 Qi et al. introduced [49] where methods of DRL and Duelling Deep Reinforcement Learning (Duelling DRL) were used. A year later Wang et al. presented [50]. The authors used combined DDPG, YOLO and DarkNet methods to properly understand the situation based on front camera image input. The next task appears when a large amount of data from all sensors need to flow through the bottleneck data bus. In 2019 Moon et al. assumed that not every sensor output is crucial every discrete time step when Autonomous Driver (AD) is operating. They developed [51] as a managing system for information flow. To reach the goal authors used the QRL algorithm that decided when to query sensors about their output. The third discussed task is to develop an algorithm capable of driving AVs while recognising road signs and obeying rules indicated by them. Such approach was considered in [21] mentioned before. The last one of such problems category is developing an algorithm for road rules and traffic management. In 2018 Vinitsky et al. proposed [52] where Trust Region Policy Optimization (TRPO) was used to manage maximum speed on 3 out of 5 parts of a highway "4 to 2 to 1" bottleneck. In order to present the distribution of tasks in the discussed works, Table I has been created.

D. New AVs Control Tasks

Despite many works considering RL applied to the field of AVs, some challenging tasks have not yet been addressed. The first of them is to develop a system that is capable of driving safely in different weather conditions and under different lighting conditions. None of mentioned papers addresses the problem of training an agent who can drive not only in daylight. Such an extension of the domain may cause many

systems to struggle. On the one side systems based on LIDAR and other sensors may have problems with correct recognition of the surface of the road and on the other side vision-based systems may have trouble spotting some obstacles. As a solution, one can propose to train number of systems separately and solve the problem by simply switching between them.

TABLE I. TASKS TYPES DISTRIBUTION

Task type	Paper References
parking	[8]
desired speed keeping	[9], [10], [11]
desired distance keeping	[12], [13], [14]
lane-keeping	[10], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]
on-ramp merging	[37], [38], [39], [40]
overtaking	[10], [18], [21], [23], [24], [26], [35], [36], [39], [40], [41], [42], [43]
intersection passing	[14], [21], [44], [45], [46], [47]
driving while occlusion occurs	[14], [46], [47]
unexpected road situations handling	[14], [47], [48]
time trial	[16], [20], [22], [23], [28], [29], [31], [34], [39], [42], [43], [45]
power source management	[49], [50]
data flow management	[51]
obeying road signs	[21]
traffic flow management	[52]

Next task it to learn the ability to overtake while having only one lane for each direction. It is the hardest from described tasks because it consists not only of overtaking manoeuvre but additionally also from opposite lane constant monitoring and aborting the manoeuvre if necessary. It is also the case to react when someone is overtaking and forcing us to move out of the way. Such cases should also be considered. On the other hand AD in the vehicle that is just overtaking should also react in a way minimizing the risk of the accident.

Another difficult task is to recognise emergency vehicles such as ambulances, fire service trucks and police trucks. AD should in such a case slow down and make a space for them to quickly pass traffic. The easiest way to implement it is to use software for wireless communication. Such an approach does, however, carry a potential risk of a hacker attack. Author believes that a better way of solving this problem is to incorporate visual information and image processing algorithms for emergency trucks detection.

III. IMPLEMENTATION ISSUES

This section will briefly discuss summarised papers implementation issues, in turn: the learning environments of the agents and the needs associated with them, the types of input data, the different ways of building up the set of available actions, and the quantities that constitute the elements of the reinforcement signal. At last, the author will outline his idea of AD for AVs.

A. Environment

The first step to develop a properly working solution is to define where it will be working. In summarised articles three main options emerge: a) training and testing an algorithm virtually by simulations, b) training and testing an algorithm on real-life vehicles (cars or smaller models) and c) combining the first and second approaches. As one could expect training and testing algorithms using traffic simulators are the most popular approach. The exact distribution is shown in Table II. So far many different racing and car simulators were developed and are being developed nowadays. These include:

- **SUMO** – free simulator written in Python. It allows users to create almost any desired intersection and manage the traffic flow including vehicles ADs. Users can also use different metrics (even pollution emission level) and birds-eye view images as input for creating an algorithm. More information and code of the simulator can be found under the link: https://www.eclipse.org/sumo/about/.
- **CARLO** – free simulator written in Python based on simplified models (point particles) of vehicles and pedestrians. It also offers birds-eye view image as input. More information and code of simulator can be found in [14] and under the link: https://github.com/Stanford-ILIAD/CARLO.
- **CARLA** – free simulator that offers the possibility to design urban roads environment and quite realistic textures. It also provides vehicle front camera images. More information and code of the simulator can be found under the link: https://carla.org/.
- **TORCS** – free car racing simulator offering a number of different difficulty tracks with different types of road surface and quite realistic physics. It also provides many types of metrics and two view options (from 1st and from 2nd person). More information and code of the simulator can be found under the link: http://torcs.sourceforge.net/.
- **WRC 6** – commercial car racing simulator. There is no open code available but it offers both very realistic vehicle physics and dynamics as well as many tracks with different types of road surface and varying slope. Raw image can be used – view from 1st and from 3rd person. More information about gameplay and the whole series of that video games is available under the link: https://en.wikipedia.org/wiki/World_Rally_Championship_(video_game_series).
- **GTS** – the Gran Turismo Sport commercial, multiplatform and very realistic (both physics and visual) car racing simulator. There is no open code available but it offers both very realistic vehicle physics and dynamics as well as many tracks with different types of road surface and varying slope. Raw image can be used – view from 1st and from 3rd person. More information about gameplay and whole series of that video games is available under the link: https://www.gran-turismo.com/pl/products/gtsport/.
- **Microsoft AirSim** – quite realistic and multiplatform urban traffic simulator made in Unity (https://unity.com/). It offers a vehicle front camera

image and also this image after segmentation. More information and code of simulator can be found under the link: https://microsoft.github.io/AirSim/.

- **Open DS-CTS 1.0** – accidents involving pedestrians the simulator which is limited to the Linux platform only. The authors' consent code could be downloaded from: https://github.com/FlorianPusse/OpenDS-CTS.
- **Unity ML Agents** – helpful toolbox facilitating the work in Unity. Applications done there could be deployed on almost any platform. More information and code of the simulator can be found under the link: https://unity.com/products/machine-learning-agents.

To make it easier for the reader to compare his ideas to other works here are the lists of papers where mentioned simulators were used: Sumo – [39, 46]; CARLO – [14]; CARLA – [14, 19, 33, 48]; TORCS – [17, 22, 23, 29, 31, 42]; WRC 6 – [20]; GTS – [34]; Microsoft AirSim – [28]; Open DS-CTS 1.0 – [47]; Unity ML Agents – [13, 43].

TABLE II. ENVIRONMENT TYPE DISTRIBUTION OVER PAPERS

Environment type distribution	Type of the environment		
	Simulator	***Mixed***	***Real world***
Papers References	[8], [9], [10], [13], [14], [15], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [29], [31], [32], [33], [34], [35], [36], [37], [38], [39], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]	[11], [27], [28], [40]	[16], [30]

A couple of papers [27, 40] were based on transferring knowledge gained by simulations to real vehicles for fine tuning. In [28] authors used real world experience and simulations at once. In [16, 30] agents learned through interacting in the real world. As can be easily seen, many researchers decide to simulate one. The main reason for this may be the lack of high quality recorded trips consisting of camera images, metrics, LIDAR and RADAR data. The second and also very important reason is that working by trials and errors (RL style) is highly risky and costs a vast amount of money. Unfortunately there is no simulator capable of imitating every possible scenario.

The author believes that to improve research quality there is a need for creating a simulator that can offer: very realistic physics engine, very realistic textures (for learning by visual input purpose), changing road and weather conditions (including darkness, harsh light, rain, wet or snowy road surface), managing traffic (implement various ADs to the number of vehicles at one time), many types of vehicles (from bicycles and motorbikes, through cars to lorries and emergency trucks), designing various road scenarios (various intersection types and railway crossings), multi-agent learning support (e. g. as client–server architecture). Such simulators should also be multiplatform.

B. Input data

The next milestone is to decide what kind of data will be the algorithm input. First of all it depends on platform on which the algorithm will be deployed. When working with simulators it can be all of listed: vehicle's velocity; vehicle's acceleration; vehicle's rotation relative to the road direction (yaw angle); vehicle's rotation in space (pitch angle); vehicle's 2D or 3D position vector; vehicle's steering angle; vehicle's battery charge state or fuel level; vehicle's actual fuel consumption rate; distance to destination point; 1st p., 3rd p. or birds-eye view camera image; segmented camera image; list of other vehicles on the road, their relatives velocities' and accelerations; LIDAR or RADAR sensors output. A couple of sensors such as camera and LIDAR produces a lot of data which can be difficult to handle or may cause algorithm long convergence time or even divergence. Other quantities may be hard to obtain in real world applications (e. g. other cars relative velocities and accelerations). There is also a case that LIDAR sensor is struggling when dealing with fog, water, asphalt or tar because of theirs light-absorbing capability. It should also not be overlooked that some people paint their cars with Vantablack which makes the LIDAR sensors blind for them. It is worth noting that camera images can be blurred, overexposed, dimmed or have light flares due to weather conditions. Despite these drawbacks of camera images and LIDAR sensors they cannot be omitted during the creation of AD for AVs. As can be seen from mentioned papers, this task is so complex that other sensors outputs are too simple to be good enaugh domain representation. The author believes that to achieve better performance a night vision systems should be added in future works. Due to the reasons discussed above camera images, LIDAR sensor and other sensors used alone, do not provide a safe and stable solution. To build one it is necessary to combine different kinds of sensors together. This view was shared by only some of the works discussed. The whole action set types distributions over mentioned papers can be seen in Table III.

TABLE III. INPUT DATA TYPE DISTRIBUTION

Data type dstb.	Images			Mixed		Sensors
	1st p.	***3rd p.***	***birds-eye view***	***1st p.***	***birds-eye view***	***metrics, LIDAR, RADAR etc.***
Pep. Refs.	[15], [21], [27], [30], [32], [33], [42]	[18]	[16], [46], [47]	[13], [14], [17], [20], [24], [29], [40], [50]	[26], [44], [45]	[8], [9], [10], [11], [12], [19], [22], [23], [25], [28], [31], [34], [35], [36], [37], [38], [39], [41], [43], [48], [49], [51], [52]

C. Action set

After deciding how input data will look like, the next step is to design the exact form of the action set. Depending on the task under consideration, the set of actions may include: vehicle's acceleration management; vehicle's target velocity; vehicle's steering angle; vehicle's lane change desires; electric engine power share management; AD's type choosing; data bus flow management; maximum allowed speed management; handbrake usage; intersection departure decision undertaking. Some of these actions must, by their nature, be discrete. Considering the remaining cases, defining the set of actions as discrete or continuous actions may be crucial. The characteristics of the action set should be determined by the objective pursued and the expected features of the solution to be prepared. On the one hand, discrete action set benefits the creator for at least two reasons. First of all, in many cases it is far more simple to design it (list possible cases) than to deal with properly bounding continuous values

for all algorithm outputs. The second reason is based on specifying RL algorithm for the learning process – not every the RL algorithm fits well with each type of action set. In other words it can be easier to deal with simple-theory algorithms and tabular representations of value function and action value function than the use of the approximator of these functions which may infer struggling with approximator (in almost all cases some kind of ANNs) training process hyperparameters configuration. In some cases discrete action space can make the learning process converges faster. By turn, using continuous action space can potentially solve more complex control tasks. In the perspective of AVs when dealing with steering control such approach could result in receiving a smoother way of taking bends and reduced fluctuations on straight parts of the road. Overall, the aforementioned issues are only tips because everything depends on the idea of how to solve a specific problem. A good example are papers [48] and [14] where two approaches of dealing with the reaction for unexpected road situations were presented. In the first one it was viewed as a continuous steering problem and in the second one it was switching between ADs types. For a better insight into the distribution of the action set types among mentioned works, Table IV were created.

TABLE IV. ACTION SET TYPE DISTRIBUTION

Action type distb.	Action set type		
	Discrete	*Mixed*	*Continuous*
Pap. Refs.	[14], [15], [16], [17], [18], [20], [21], [24], [26], [33], [35], [36], [37], [39], [40], [43], [44], [45], [46], [47], [51]	[19], [42]	[8], [9], [10], [11], [13], [17], [22], [23], [25], [27], [28], [29], [30], [31], [32], [34], [38], [41], [48], [49], [50], [52]

D. Reinforcement signal

The last but not least element is the reinforcement signal. As it was stated before, its task is to evaluate the agent's actions. The reinforcement signal may consist of many elements due to the necessity of optimising many aspects of the algorithm being developed. Each of the fragments is therefore responsible for dealing with the relevant goal. In summarised papers following quantities were involved: distance travelled by the agent; time consumed for travelling; distance between AD and following driver; distance from the centre of the road; vehicle's velocity; vehicle's yaw angle relevant to the road direction; vehicle's yaw angle changing rate; vehicle's steering angle changing rate; vehicle's acceleration changing rate; the volume of fuel used by the vehicle; the number of the lane in which AD is moving (reward if it's a right lane and punishment when it's one of faster lanes); quality of chosen data packages; intersection flow rate, the kinetic energy of the collision, manoeuver duration time. Depending on the context and aimed goal some of listed quantities may be used in different ways e. g. the travelled distance can be treat as reward when goal for the agent is to travel as far as he can, or can be treated as punishment when one deal with car racing problem. The second example could be the time of the trip. On the one side it should be short if AD's is racing or highway driving, on the other hand it should be as long as possible when considering driving time without collision. It should not be overlooked that some events could also be part of the reinforcement signal e. g.: road accident happened; AD found itself outside of the road boundaries; the AD disregarded the regulations imposed by traffic signs or speed limits; AD succeeded to safely finish travel.

E. Author's approach sketch

Moreover, as one of the presented summary results a sketch of the author's idea will be presented. It considers a highway driving task. An agent will have to control the wheels steering angle, brake and throttle. All outputs will be continuous values ranging from -1 to 1 (break and throttle outputs will be treated as one output because they are mutually exclusive). Doing such should provide smooth driving experience. Furthermore input data will be established by vehicle's metrics, LIDAR sensor data and images from front, rear and side-rear cameras (left and right). Metrics should provide vehicle's state information while LIDAR data ought to be a vehicle on road position reference. Front and rear camera input will extend AD's knowledge for crucial information about distance from others vehicles on the same lane and their desires (spotting direction indicators and sideways creeping). Side-rear cameras address the issue of safety overtaking manoeuvre. The author believes that these camera images will ensure the AD if it is safe to begin overtaking and when it is safe to finish this manoeuvre. It should also be a solution for speeding drivers who approach too fast to be correctly recognised by other systems as hazardous situations. The author leaves the verification of the presented idea for future work.

IV. CONCLUSION

In this paper the summary of currently published ideas on the application of the RL algorithms to the problem of AVs was addressed. Furthermore, the breakdown of AV's tasks was presented, showing various models of the environment, action sets and reinforcement signals outlined in just under fifty papers. Additionally a number of potential development areas were highlighted such as new tasks for AVs and new environment simulator creation necessity. The author also presented the sketch of his own idea for highway driving tasks to which his future work will be devoted.

REFERENCES

[1] D. Mehta, "State-of-the-Art Reinforcement Learning Algorithms", International Journal of Engineering Research & Technology (IJERT), vol. 8, issue 12, December 2019.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, et al., "Human-level control through deep reinforcement learning", Nature, vol. 518(7540), pp. 529-533, February 2015

[3] W. C. Hung, J. Zhang, X. Shen, Z. Lin, J. Y. Lee, M. H. Yang, "Learning to blend photos", Proceedings of the European Conference on Computer Vision (ECCV), pp. 70-86, 2018

[4] R. Furuta, N. Inoue, T. Yamasaki, "PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing", IEEE Transactions on Multimedia, vol. 22, issue 7, December 2019

[5] W. Li, X. Feng, H. An, X. Y. Ng, Y. J. Zhang, "MRI Reconstruction with Interpretable Pixel-Wise Operations Using Reinforcement Learning", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, pp. 792-799, April 2020

[6] S. Spielberg, A. Tulsyan, N. P. Lawrence, P. D. Loewen, R. Bhushan Gopaluni, "Toward self - driving processes: A deep reinforcement learning approach to control", AIChE Journal, vol. 65, issue 10, e16689, October 2019

[7] S. Paravarzar, B. Mohammad, "Motion Prediction on Self-driving Cars: A Review", arXiv preprint arXiv:2011.03635, November 2020

[8] D. Maravall, M. Á. Patricio, J. de Lope, "Automatic car parking: a reinforcement learning approach", International Work-Conference on

Artificial Neural Networks, Springer - Lecture Notes in Computer Science, vol. 2686, pp. 214-221, June 2003

[9] M. Buechel, A. Knoll, "Deep reinforcement learning for predictive longitudinal control of automated vehicles", 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2391-2397, November 2018

[10] S. Aradi, T. Becsi, P. Gaspar, "Policy gradient based reinforcement learning approach for autonomous highway driving", 2018 IEEE Conference on Control Technology and Applications (CCTA), pp. 670-675, August 2018

[11] L. Puccetti, F. Köpf, C. Rathgeber, S. Hohmann, "Speed tracking control using online reinforcement learning in a real car", 2020 6th International Conference on Control, Automation and Robotics (ICCAR), pp. 392-399, April 2020

[12] Y. Lin, J. McPhee, N. L. Azad, "Longitudinal dynamic versus kinematic models for car-following control using deep reinforcement learning", 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 1504-1510, October, 2019

[13] Z. Wei, Y. Jiang, X. Liao, X. Qi, Z. Wang, G. Wu, P. Hao, M. Barth, "End-to-End Vision-Based Adaptive Cruise Control (ACC) Using Deep Reinforcement Learning", arXiv preprint arXiv:2001.09181, January 2020

[14] Z. Cao, E. Biyik, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, D. Sadigh, "Reinforcement learning based control of imitative policies for near-accident driving", Science and Systems (RSS) 2020, arXiv preprint arXiv:2007.00178, 2020

[15] G. Yu, I. K. Sethi, "Road-following with continuous learning", Proceedings of the Intelligent Vehicles' 95. Symposium, pp. 412-417, September, 1995

[16] S. Lange, M. Riedmiller, A. Voigtländer, "Autonomous reinforcement learning on raw visual input data in a real world application", The 2012 international joint conference on neural networks (IJCNN), pp. 1-8, June 2012

[17] G. D. Kontes, D. D. Scherer, T. Nissbeck, J. Fischer, C. Mutschler, "High-Speed Collision Avoidance using Deep Reinforcement Learning and Domain Randomization for Autonomous Vehicles", 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1-8, September 2020

[18] A. Yu, R. Palefsky-Smith, R. Bedi, "Deep reinforcement learning for simulated autonomous vehicle control", Course Project Reports: Winter, 2016

[19] A. E. Sallab, M. Abdou, E. Perot, S. Yogamani, "Deep reinforcement learning framework for autonomous driving", Electronic Imaging, vol. 2017(19), pp. 70-76, January 2017

[20] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, F. Nashashibi, "End-to-end race driving with deep reinforcement learning", 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2070-2075, May 2018

[21] S. O. Chishti, S. Riaz, M. BilalZaib, M. Nauman, "Self-driving cars using CNN and Q-learning", 2018 IEEE 21st International Multi-Topic Conference (INMIC), pp. 1-7, November 2018

[22] S. Wang, D. Jia, X. Weng, "Deep reinforcement learning for autonomous driving", arXiv preprint arXiv:1811.11329, November 2018

[23] J. Wu, H. Li, "Aggregated multi-deep deterministic policy gradient for self-driving policy", International Conference on Internet of Vehicles, Springer, pp. 179-192, November 2018

[24] Y. Zhang, P. Sun, Y. Yin, L. Lin, X. Wang, "Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning", 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1251-1256, June 2018

[25] A. Feher, S. Aradi, T. Becsi, "Q-learning based reinforcement learning approach for lane keeping", 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 000031-000036, November 2018

[26] A. Keselman, S. Ten, A. Ghazali, M. Jubeh, "Reinforcement learning with a* and a deep heuristic", arXiv preprint arXiv:1811.07745, 2018

[27] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.M. Allen, A. Shah, "Learning to drive in a day", 2019 International Conference on Robotics and Automation (ICRA), pp. 8248-8254, May 2019

[28] X. Liang, Y. Liu, T. Chen, M. Liu, Q. Yang, "Federated transfer reinforcement learning for autonomous driving", arXiv preprint arXiv:1910.06001, 2019

[29] Y. Zhu, D. Zhao, "Vision-based control in the open racing car simulator with deep and reinforcement learning", Journal of Ambient Intelligence and Humanized Computing, pp. 1-13, September 2019

[30] Q. Zhang, T. Du, C. Tian, "Self-driving scale car trained by deep reinforcement learning", arXiv preprint arXiv:1909.03467, September 2019

[31] P. Ke, Z. Yanxin, Y. A. Chenkun, "Decision-making Method for Self-driving Based on Deep Reinforcement Learning", Journal of Physics: Conference Series, vol. 1576, no. 1, p. 012025, IOP Publishing, June 2020

[32] G. A. de Morais, L. B. Marcos, J. N. A. Bueno, N. F. de Resende, M. H. Terra, V. Grassi Jr, "Vision-based robust control framework based on deep reinforcement learning applied to autonomous ground vehicles", Control Engineering Practice, vol. 104, p. 104630, November 2020

[33] F. Youssef, B. Houda, "Comparative Study of End-to-end Deep Learning Methods for Self-driving Car", International Journal of Intelligent Systems & Applications, vol. 12, issue 5, pp. 15-27, October 2020

[34] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, P. Dürr, "Super-human performance in gran turismo sport using deep reinforcement learning" IEEE Robotics and Automation Letters, vol. 6 issue 3, pp. 4257-4264, July 2021

[35] S. Nageshrao, H. E. Tseng, D. Filev, "Autonomous highway driving using deep reinforcement learning", 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2326-2331, October 2019

[36] D. C. K Ngai, N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers", IEEE Transactions on Intelligent Transportation Systems, vol. 12, issue 2, pp. 509-522, February 2011

[37] S. Shalev-Shwartz, S. Shammah, A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving", arXiv preprint arXiv:1610.03295, October 2016

[38] P. Wang, C. Y. Chan, "Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge", 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1-6, October 2017

[39] M. Mukadam, A. Cosgun, A. Nakhaei, K. Fujimura, "Tactical decision making for lane changing with deep reinforcement learning", December 2017

[40] A. R. Fayjie, S. Hossain, D. Oualid, D. J. Lee, "Driverless car: Autonomous driving using deep reinforcement learning in urban environment", 2018 15th International Conference on Ubiquitous Robots (UR), pp. 896-901, June 2018

[41] P. Wang, C. Y. Chan, A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers", 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1379-1384, June 2018

[42] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, J. M. Dolan, "Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, June 2019

[43] A. T. Huynh, B. T. Nguyen, H. T. Nguyen, S. Vu, H. D. Nguyen, "A Method of Deep Reinforcement Learning for Simulation of Autonomous Vehicle Control", ENASE, pp. 372-379, 2021

[44] G. Bacchiani, D. Molinari, M. Patander, "Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning", arXiv preprint arXiv:1903.01365, March 2019

[45] Y. Wang, S. Hou, X. Wang, "Reinforcement learning - based bird - view automated vehicle control to avoid crossing traffic", Computer - Aided Civil and Infrastructure Engineering, July 2020

[46] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning", 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2034-2039, May 2018

[47] F. Pusse, M. Klusch, "Hybrid online pomdp planning and deep reinforcement learning for safer self-driving cars", 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 1013-1020, June 2019

[48] G. D. Kontes, D. D. Scherer, T. Nissbeck, J. Fischer, C. Mutschler, "High-Speed Collision Avoidance using Deep Reinforcement Learning and Domain Randomization for Autonomous Vehicles", 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1-8, September 2020

[49] X. Qi, Y. Luo, G. Wu, K. Boriboonsomsin, M. Barth, "Deep reinforcement learning enabled self-learning control for energy efficient driving", Transportation Research Part C: Emerging Technologies, vol. 99, pp. 67-81, February 2019

[50] Y. Wang, H. Tan, Y. Wu, J. Peng, "Hybrid electric vehicle energy management with computer vision and deep reinforcement learning", IEEE Transactions on Industrial Informatics, vol. 17, issue 6, June 2020

[51] J. Moon, M. Cheong, I. Yeom, H. Woo, "Deep Reinforcement Learning Based Sensor Data Management for Vehicles", 2019 International Conference on Information Networking (ICOIN), pp. 345-349, January 2019

[52] E. Vinitsky, K. Parvate, A. Kreidieh, C. Wu, A. Bayen, "Lagrangian control through deep-rl: Applications to bottleneck decongestion", 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 759-765, November 2018

Application of Mamdani Fuzzy Logic Inference System to optimise CNC machine motion dynamics

Dawid Kalandyk
Doctoral School of the Rzeszów University of Technology
Rzeszów, Poland
https://orcid.org/0000-0002-7317-5499

Bogdan Kwiatkowski
Department of Electrical and Computer Engineering Fundamentals
University of Technology
Rzeszów, Poland
b.kwiatkowsk@prz.edu.pl

Damian Mazur
Department of Electrical and Computer Engineering Fundamentals
University of Technology
Rzeszów, Poland
mazur@prz.edu.pl

2023 IEEE International Conference on Fuzzy Systems (FUZZ) | 979-8-3503-3228-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/FUZZ52849.2023.10309802

***Abstract*— The article presents the application of Mamdani Fuzzy Logic Inference System to optimize the operation of a Computerized Numerical Control (CNC) machine for given dynamic parameters. These parameters are maximum speed, acceleration and JERK. The JERK parameter determines the rate of change of acceleration. The parameters are defined for each working axis of the machine. In order to check the correctness of the solution proposed in the paper, the learning and testing process was conducted on specially designed database including different trajectories generated for machining with different machine dynamics parameters. The approach presented in the paper using elements of fuzzy logic to optimize the operation of the CNC machine proved to be very good. The authors, in cooperation with industry, will develop the aforementioned solution leading to implementations and further scientific publications.**

Keywords — CNC machine, Jerk, Reference Points, Fuzzy Inference System, Artificial Intelligence, Particles Swarm Optimization

I. Introduction

The progressive globalisation of the world economy is particularly evident in the field of modern technologies, striving for maximum automation of manufacturing processes. With regard to numerically controlled machines, there is a need to modify existing solutions, linked to the pursuit of more favorable technological and economic effects and a reduction in energy consumption [1] [2]. An increase in machining productivity and the service life of the main components of CNC machines, while meeting the required geometric accuracy and quality of the final product, is possible through the development and improvement of machine tool control algorithms. The specificity of numerical machine tools, related to their design, software and operation, requires the involvement not only of engineers but also of the scientific community from many research fields [3]. The industrial experience of the authors, related to the application of modern computer methods and the operation of CNC machine tools, indicates that the problem requiring the support of the scientific community is the development and implementation of new algorithms and the optimum selection of parameters describing the dynamics of the CNC machine, allowing the exact reflection of geometric points of the performed detail [4] [5]. The main objective of the solution currently used in industry is a systematic approach related to the correlation of the time discretization algorithm and the shape and values of the parameters describing the dynamics of CNC machines. However, the resulting offline solution has two important drawbacks [5] [6]. The first is the need to search through a number of combinations of machine dynamics parameters to find the combination that achieves the targeted accuracy. The second is the way in which the machine's motion is controlled - using the maximum acceleration available (rapid acceleration as well as deceleration), which leads to more wearing of the machine in a shorter period of time and can therefore increase the final cost of the manufacturing process [8] [9]. An additional aspect to consider is the lack of straightforward interpretability of the results obtained, which translates into limited opportunities for the expert to affect the final results of the machining process. To meet such challenges, the authors of this work decided to propose a system capable of similarly dealing with parameter optimization, but allowing its operation to be interpreted and modified by the user. To achieve this, the authors decided to create a control system using Mamdani fuzzy logic trained via a particle swarm optimization algorithm [10]. It has been already applied to handle other problems in CNC machining process [11][12][13].

Contributions of this work are as follows: (i) a new algorithm has been proposed for controlling a CNC machine for a given dynamic parameters using Mamdani Fuzzy Logic Inference System whose rules can be understood and edited by an expert; (ii) a database for training and evaluation of the algorithm has been proposed.

II. Database

A. Used base algorithm

In industry there are various approaches for a solution that allows for the proper selection of the dynamics parameters of a CNC machine [14]. The aforementioned algorithm called Reference Points Realization Optimization (RPRO), which is used nowadays in the technological process, was used for the research. It is based on the given parameters of CNC machine dynamics. The solution is generated offline. The end result is a generated G-code that guarantees the maximum accuracy of hitting the reference points and the fastest execution time. The obtained solution is characterized by the fact that for each reference point the accuracy of hitting can be different. Thus, the main goal is to achieve the desired accuracy for each of the reference points, which is reflected not only in the average accuracy of manufacturing but also in the quality of the detail.

The calculation of the reference time is determined by reaching the first reference point after a certain number of time steps. The number of these steps depends on the position of the reference point. The time step is always the same and is 0.002 s. The accuracy of reaching the reference points is

achieved by correcting the speed and acceleration values accordingly. It should be noted that increasing the accuracy of reaching successive reference points does not change the number of time steps between them, but only the control values are modified for them [15].

A very important parameter that determines the dynamics of a CNC machine is Jerk [7]. The purpose of the Jerk parameter is to optimize the acceleration rate of the spindle, and it is a parameter defined for each axis of the machine. This means that the movement of the tool from a given point to another reference point during acceleration and deceleration is performed according to the dynamics specified in this parameter. Correct selection and determination of this parameter causes the machine tool and all its mechanical components, i.e. servos, spindles, to be operated according to their ratings , which improves the quality of manufactured products and extends the life of the machine [16] [17]. The RPRO algorithm shows what is the maximum possible accuracy of the realization of reference points for selected parameters of the dynamics of the machine while maintaining all factors related to the life of the CNC machine itself and the quality of the manufactured detail. Jerk can be evaluated using formula (1), where acceleration, velocity, time and relocation are denoted by a, v, t and r respectively.

$$\vec{a}(t) = \frac{d\vec{a}}{dt} = \dot{\vec{a}}(t) = \frac{d^2\vec{v}}{dt^2} = \ddot{\vec{v}}(t) = \frac{d^3\vec{r}}{dt^3} = \dddot{\vec{r}}(t) \tag{1}$$

B. *Authors database proposition*

Based on the algorithm (RPRO), a database containing the results of the mentioned algorithm for different combinations of machine dynamics parameters and spindle motion trajectories was created for the experiments. The values of the studied variables are summarized in the Table 1. It is worth noting that for each combination of trajectory length and reference point density, 10 random motion trajectories were generated, which means a final of 9 groups of 10 trajectories. Thus, the total number of analyzed machining processes is $9 \cdot 10 \cdot 4 \cdot 5 \cdot 3 = 5400$ (600 in each group). The database formed from such combinations of dynamics parameters and diversity of trajectories provides the possibility of verifying the correctness of the proposed algorithm and evaluating the level of adaptation to changing conditions of the working environment.

TABLE I. DATABASE PARAMETERS VALUES

Database Parameters Values		
Parameter	***Possible Values***	***Combinations Factor***
Trajectory length	{15, 50, 100}	3
Reference points density	{Low, Medium, *High*}	3
Maximum velocity $\left[\frac{m}{min}\right]$	{2.5, 4.0, 6.0, 8.0}	4
Maximum acceleration $\left[\frac{m}{s^2}\right]$	{1.5, 1.8, 2.0, 2.5, 3.0}	5
Jerk $\left[\frac{m}{s^3}\right]$	{10, 20, 30}	3
Target precision $[mm]$	0.01	1
Time step duration $[s]$	0.002	1

III. PROPOSED FUZZY INFERENCE SYSTEM

With the primary goal of creating a system that the user can easily interpret and modify as needed, the authors decided to use a fuzzy inferencing system according to the Mamdani model. An additional advantage is its relatively simple construction (construction of input signals and covering the fuzzy domain with functions). The study of the Sugeno system, which should get better response times, the authors leave for future work. The trained systems had three inputs. These were, respectively, normalized spindle velocity, normalized spindle acceleration and normalized spindle distance from the next reference point. All inputs were evenly covered with triangular functions in amounts of respectively: 5, 8 and 5. The minimum values of the inputs respectively are 0.0, -1.0 and 0.0, while the maximum value of each input is 1.0. The output, whose values are in the range $[-1.5, 1.5]$, denoting the system's decision at a given time step, is also covered with three functions representing *Deceleration*, *No Action* and *Acceleration* respectively. During calculation and evaluation, the system's response (output value) was rounded to the whole. The diagram of the created system is shown in Figure 2. It is worth noting at this point that the functions implementing the subsequent logical operators are as follows: *AND* (min), *OR* (max), *IMP* (min) and *AGG* (max).

IV. EXPERIMENTS AND RESULTS

To test the proposed solution, experiments were conducted as follows. Wanting to check the correctness of the operation for unobserved data during the learning process, it was always conducted on 2 trajectories from one of the 9 groups. Thus, these were data from 120 machining processes with different machine dynamics parameters. Testing the quality of the obtained system was carried out on the data for the remaining 8 trajectories of a given group. According to the authors' assumptions, a system learned on short trajectories should also be able to handle trajectories of other lengths. So the learning was carried out for each of the groups numbered 1, 2 and 3 containing data for the shortest trajectories for each of the 3 densities of reference points. To confirm the solution's stability, it was decided to implement the learning process for 5 pairs of trajectories in each of the 3 selected groups. During the course of the experiments, however, it turned out that for group number 3 the learning process was significantly longer. In addition, the process of subsequent evaluation of the system was far too long to meet the requirements of implementation in a real solution. It was therefore decided to implement the learning process for group number 3 only for the first pair of trajectories. In addition, it was also necessary to reduce the number of epochs of the learning process for this particular case. The statistics of the learning process are shown in Table II.

Analyzing the results, authors note that as the density of reference points decreases, the number of records of the learning set and the test set (equal to the total number of time steps in all the processing processes that constitute the set) increases drastically. These results explain the noticed problems with the duration of computation. The metric on which the particle swarm optimization algorithm depended was RMSE. For all the learning processes presented, this metric for the test set was comparable or more favorable than for the learning set, which can be defined as success. Confirmation of the successful implementation of the learning process is also provided by the percentage of incorrect decisions. A system response that differed by 0.5 or more from

that contained in the database was considered to be incorrect. Analyzing the results, it can also be noted that as the number of rules of the final system for a given group increases, the quality of this system decreases (visible for group number 2).

TABLE II. TRAINING RESULTS

Reference Points Density	Training Pair Number	Maximum Iterations Number	Training Time	RMSE Training Set [mm]	RMSE Testing Set [mm]	Bad Decisions Fraction	Train Records Number	Test Records Number	Rules Number
High	1	60	1h 8min 43s	0.341	0.338	0.128	13628	53912	52
	2	60	52min 8s	0.355	0.361	0.134	12759	54781	54
	3	60	1h 19min 43s	0.334	0.337	0.116	14552	52988	60
	4	60	1h 14min 15s	0.333	0.337	0.11	12673	54867	81
	5	60	1h 17min 18s	0.325	0.334	0.114	13928	53612	79
Medium	1	60	56min 28s	0.407	0.408	0.093	60623	286040	78
	2	60	1h 36min 9s	0.495	0.441	0.168	77910	268753	145
	3	60	1h 21min 7s	0.501	0.446	0.166	69874	276789	122
	4	60	52min 29s	0.393	0.405	0.089	67603	279060	56
	5	60	1h 28min 33s	0.517	0.473	0.199	70653	276010	136
Low	1	25	4h 50min 18s	0.222	0.154	0.026	693732	2631315	141

TABLE III. ASCSASCD EVALUATION RESULTS

Reference Points Density	Reference Points Number in Trajectory	RPRO	Mamdani				
			1	*2*	*3*	*4*	*5*
High	15	0.0211 ± 0.0087	0.0255 ± 0.0061	0.0255 ± 0.0061	0.0256 ± 0.0062	0.0255 ± 0.0061	0.0232 ± 0.0063
	50	0.0405 ± 0.0172	0.0314 ± 0.0088	0.0314 ± 0.0088	0.0313 ± 0.0088	0.0314 ± 0.0088	0.0315 ± 0.0095
	100	0.0497 ± 0.0222	0.0327 ± 0.0100	0.0327 ± 0.0099	0.0327 ± 0.0099	0.0327 ± 0.0100	0.0332 ± 0.0102
Medium	15	0.0154 ± 0.0158	0.0162 ± 0.0029	0.0166 ± 0.0031	0.0270 ± 0.0109	0.0273 ± 0.0103	0.0162 ± 0.0030
	50	0.0192 ± 0.0169	0.0265 ± 0.0056	0.0269 ± 0.0060	0.0325 ± 0.0109	0.0326 ± 0.0109	0.0269 ± 0.0060
	100	0.0162 ± 0.0143	0.0303 ± 0.0080	0.0311 ± 0.0087	0.0323 ± 0.0101	0.0322 ± 0.0099	0.0305 ± 0.0082

CONCLUSIONS

The authors presented a novel approach to the problem of optimizing the motion dynamics of a CNC machine by using a fuzzy logic inference expert system. The presented solution achieved very good results optimizing the dynamics of the CNC machine. The interaction of the presented RPRO algorithm and the use of fuzzy logic methods results in the smoothing of acceleration and feed rates, which is desirable due to the improvement of the accuracy of the detail as well as the service life of the structural components of the machining center. The presented approach is based on expert knowledge. It can also be customized by both learning or teaching on the appropriate database, as well as by directly editing the rules of the already learned expert system. The presented solution increases the execution time of the machining process, but it can be modified depending on the needs and requirements arising from the technological process specific to certain industries. However, the solution can be applied in industry, several modifications should be made. Therefore, in further work, the authors will undertake the optimization of the system's response time and the learning process itself. The main goal will become the creation of a system that gives satisfactory results for different densities of reference points and for the variable nonlinear value of Jerk. Attention will also be paid to the use of other artificial intelligence methods as well as other fuzzy logic methods including type-2 systems, which can tackle real-world application uncertainties

ACKNOWLEDGMENT

This work was partially supported by Doctoral School of the Rzeszow University of Technology and Department of Electrical and Computer Engineering Fundamentals, University of Technology, Rzeszow, Poland.

REFERENCES

[1] J. M. Langeron, E. Duc, C. Lartigue and P. Bourdet, "A new format for 5-axis tool path computation using Bspline curves". Comput-Aided Design, 36, pp. 1219–1229, 2004.

[2] Q. Bi, N. Huang, C. Sun, Y. Wang, L. Zhu and H. Ding, "Identification and compensation of geometric errors of rotary axes on five-axis machine by on-machine measurement", Int J Machine Tools Manufacture, 89, pp. 182–191, 2015.

[3] H. J. Lee, Y. Liu and S. H. Yang, "Accuracy improvement of miniaturized machine tool: Geometric error modelling and compensation". Int J Machine Tools Manufacture, 46, pp. 1508–1516. 2006.

[4] Y. Sun, S. Sun, J. Xu and D. Guo, "A unified method of generating tool path based on multiple vector fields for CNC machining of compound NURBS surfaces". Comput-Aided Design, 91, pp. 14–26, 2017.

[5] XF. Li, H. Zhao, X. Zhao and H. Ding, "Interpolation-based contour error estimation and component-based contouring control for five-axis CNC machine tools", Sci China Tech Sci, 61, pp. 1666–1678, 2018.

[6] M. Chen and Y. Sun, "A moving knot sequence-based feedrate scheduling method of parametric interpolator for CNC machining with contour error and drive constraints, Int J Adv Manuf Technol, 98, pp. 487–504, 2018.

[7] Barbara Pękala; Ewa Rak; Bogdan Kwiatkowski; Adam Szczur; Rafał Rak, The use of concave and convex functions to optimize the feed-rate of numerically controlled machine tools, 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), DOI: 10.1109/FUZZ48607.2020.9177569

[8] S. Z. Mansour and R. Seethaler, "Feedrate optimization for computer numerically controlled machine tools using modeled and measured process constraints", J Manuf Sci Eng, 139, 9 pages, 2017. https://doi.org/10.1115/1.4033933

[9] M. Rahaman, R. Seethaler and I. Yellowley, "A new approach to contour error control in high speed machining", Int J Machine Tools Manufacture, 88, pp. 42–50, 2015.

[10] J.E. Bobrow, S. Dubowsky and J.S. Gibson, "Time-optimal control of robotic manipulators along specified paths", Int J Robotics Res, 4, pp. 3-17, 1985.

[11] Kar, T., Mandal, N.K. & Singh, N.K. Multi-response Optimization and Surface Texture Characterization for CNC Milling of Inconel 718 Alloy. *Arab J Sci Eng* **45**, 1265–1277 (2020). https://doi.org/10.1007/s13369-019-04324-5

[12] Datta, S., Mahapatra, S.S., Routara, B.C. and Bandyopadhyay, A. October 3, 2011pp 265-282 https://doi.org/10.1504/IJEDPO.2011.042747

[13] Molina, A., Ponce, H., Ponce, P., Tello, G., & Ramírez, M. (2014). Artificial hydrocarbon networks fuzzy inference systems for CNC machines position controller. *International Journal of Advanced Manufacturing Technology*, *72*.

[14] Z. Shiller and H. H. Lu, "Robust computation of path constrained time optimal motion", IEEE Inter. Conf. on Robotics and Automation, Cincinnati, pp. 144-149, 1990.

[15] J. Dong and J. A. Stori, "A generalized time-optimal bi-directional scan algorithm for con-strained feedrate optimization", ASME Journal of Dynamic Systems, Measurement and Control, 128, pp. 379-390, 2006.

[16] S. D. Timar, R. T. Farouki, T. S. Smitha and C. L. Boyadjieff, "Algorithms for time-optimal control of CNC machines along curved tool paths", Robotics and Computer-Integrated Manufacturing, 21, pp. 37-53, 2005.

[17] S. D. Timar and R. T. Farouki, "Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed dependent axis acceleration bounds", Robotics and Computer-Integrated

www.czasopisma.pan.pl PAN www.journals.pan.pl

BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 72(3), 2024, Article number: e148940
DOI: 10.24425/bpasts.2024.148940

CONTROL, INFORMATICS AND ROBOTICS

CNC machine control using deep reinforcement learning

Dawid KALANDYK[1], **Bogdan KWIATKOWSKI**[2], and **Damian MAZUR**[2]*

[1] Doctoral School of the Rzeszów University of Technology, Powstańców Warszawy Ave. 12, 35-959 Rzeszów, Poland
[2] Department of Electrical and Computer Engineering Fundamentals, Rzeszow University of Technology, W. Pola str. 2, 35-959 Rzeszów, Poland

Abstract. Optimization of industrial processes such as manufacturing or processing of specific materials constitutes a point of interest for many researchers, and its application can lead not only to speeding up the processes in question, but also to reducing the energy cost incurred during them. This article presents a novel approach to optimizing the spindle motion of a computer numeric control (CNC) machine. The proposed solution is to use deep learning with reinforcement to map the performance of the reference points realization optimization (RPRO) algorithm used in the industry. A detailed study was conducted to see how well the proposed method performs the targeted task. In addition, the influence of a number of different factors and hyperparameters of the learning process on the performance of the trained agent was investigated. The proposed solution achieved very good results, not only satisfactorily replicating the performance of the benchmark algorithm, but also speeding up the machining process and providing significantly higher accuracy.

Keywords: deep reinforcement learning; CNC machining; machining optimization.

1. INTRODUCTION

The ability to carve out parts of complex shapes with high accuracy not only allows the creation of robots capable of performing many tasks, but also enables the development of technology. For a long time, therefore, computer numeric control (CNC) machines has been the subject of many studies [1, 2]. Fields being developed include general machine control [3], estimation and minimization of machining errors [4, 5]. The topic attracting the most attention is spindle motion path planning and determination of spindle motion control signals in successive time steps [6–15]. Artificial intelligence methods, actively developed recently, are also eagerly used for the previously mentioned tasks [16–20]. A lot of attention has also been paid by researchers to the reinforcement learning (RL) algorithm [21], which, due to its versatility, can be applied to various types of control tasks [22–33]. Due to the operating characteristics of the shop floor, or even the machining process itself (time steps), this algorithm is also widely used for various tasks: manufacturing floor process control [34–38], damage prediction [39], equipment overhaul management [40, 41], selection of equipment settings [42–48] and optimizing the spindle motion path and determining the g-code [49–59] that determines spindle motion. In paper [60], the authors proposed to use fuzzy logic systems taught by the particle swarm method to replicate the operation of the g-code determination algorithm used in the industry [6]. In this manner, they wanted to linearize the computational complexity of this algorithm and make its operation independent of CNC machine dynamics parameters such as maximum spindle velocity or maximum linear acceleration of the spindle. The present work builds on the aforementioned research by using a deep learning algorithm with reinforcement to map the performance of the reference points realization optimization (RPRO) algorithm.

Contributions of this work are as follows:

- a novel approach that involves using a deep reinforcement learning algorithm to mimic the work of the RPRO algorithm,
- study of a number of different configurations of parameters of the learning process,
- testing the accuracy of mimicking the operation of the RPRO algorithm,
- optimization of the machining process

In Section 2 the proposed solution will be characterized. Then, in Section 3 the conducted research will be described and the results of the performed experiments will be presented and analyzed. Section 4 will summarize the paper and indicate the direction of further work.

2. PROPOSED FRAMEWORK

According to the premise, the work involves applying a deep reinforcement learning algorithm to the task of generating the g-code that controls the operation of the CNC machine. Its form influences not only the duration of the machining of the fabricate, but also the accuracy of the production of the final workpiece, as well as the level of tool wear and electricity consumption. The task is to train a neural network to respond to the given input signals in such a way as to mimic the behavior of another algorithm. According to the authors, this allows to

* e-mail: mazur@prz.edu.pl

Manuscript submitted 2023-11-02, revised 2023-12-27, initially accepted for publication 2023-12-30, published in May 2024.

linearize the process of generating the g-code – while the RPRO algorithm for each time step must perform checks for a certain number of forward steps, the proposed solution directly generates the desired output signal. The proposed system of learning and operation of the algorithm is shown in Fig. 1.

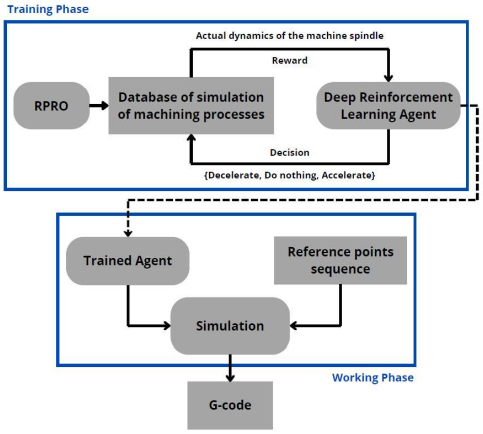


Fig. 1. Proposed framework

2.1. Base algorithm used

The RPRO algorithm is aimed at optimizing the g-code in terms of machining time, bearing in mind, however, to ensure the best possible workpiece manufacturing accuracy at a given time. It is worth noting that the above-mentioned workpiece machining accuracy should be understood as the average accuracy of the spindle reaching each reference point. For each time step of 2 ms, the algorithm decides whether the spindle should accelerate, decelerate, or perhaps move with unchanged dynamics. The calculations are carried out offline, allowing different versions of the g-code to be checked going forward and manipulated accordingly. The disadvantage of this approach is the inability to operate in real time, receiving as input the actual data describing the dynamics of the machine and the spindle, rather than those from the simulation. During the simulation and operation of the algorithm, the current spindle position, spindle velocity and spindle acceleration are calculated based on equation (1). The exact way the algorithm works is described in [6].

$$\vec{J}(t) = \frac{\mathrm{d}a}{\mathrm{d}t} = \dot{\vec{a}}(t) = \frac{\mathrm{d}^2 v}{\mathrm{d}t^2} = \ddot{\vec{v}}(t) = \frac{\mathrm{d}^3 r}{\mathrm{d}t^3} = \dddot{\vec{r}}(t) \tag{1}$$

2.2. Reinforcement learning

The reinforcement learning algorithm is characterized by stepwise action – the agent decides in successive time steps t what action a should be taken to maximize the sum of rewards r granted to it after each choice. Its learning is accomplished through its interaction with the environment in which it moves and updating information about this environment. According to currently accepted methods, this knowledge can be represented in two ways. First, by a value function $V^{\pi}(s)$ (2) specifying the expected total value of the reward to be gained starting from a given state s_0. The second way is by an action value function $Q^{\pi}(s,a)$ (3) specifying the expected total value of the reward to be earned when an agent starts from a given state s_0 and performs an action a_0 in it. The discount factor γ is also an important aspect, controlling the learning process and the final behavior of the agent. It is also referred to as the agent's foresight factor and takes values in the range of (0; 1).

$$V^{\pi}(s) = E_{\pi}\left[\sum_{t=0} \left(\gamma^t \cdot r^t\right)\right] \quad \text{where } s_0 = s, \tag{2}$$

$$Q^{\pi}(s,a) = E_{\pi}\left[r_0 + \sum_{t=0} \left(\gamma^t \cdot r^t\right)\right] \quad \text{where } s_0 = s,\ a_0 = a. \tag{3}$$

For the simplest model of the environment and for many applications, a basic tabular representation of the functions mentioned is sufficient. However, if the problem domain is not finite, then some method of approximating the values of these functions can be used. One such method is to use a neural network for this purpose, the so-called deep learning with reinforcement (DRL). Such an action provides the possibility of training an agent that is likely to be able to perform satisfactorily even in the case of small changes in the environment in which it moves.

2.3. Database

To conduct the experiments, a previously prepared database consisting of stored simulations of machining processes for different reference point paths and a number of combinations of machine dynamics parameters was used. These simulations were prepared using the RPRO algorithm. The values of the various parameters are shown in Table 1. High density of reference points means that the distances between them are less than 1 mm, medium density means that the distance between them is greater than 1 mm but less than 10 mm, while low density means that the distances between successive reference points are between 10 mm and 100 mm.

Table 1
Database of parameters values [60]

Parameter	Possible values	Combinations factor
Trajectory length	{15, 50, 100}	3
Reference points density	{Low, Medium, High}	3
Maximum velocity [m/min]	{2.5, 4.0, 6.0, 8.0}	4
Maximum acceleration [m/s^2]	{1.5, 1.8, 2.0, 2.5, 3.0}	5
Jerk [m/s^3]	{10, 20, 30}	3
Target precision [mm]	0.01	1
Time step duration [s]	0.002	1

3. EXPERIMENTS

The experiments conducted included testing the impact of neural network architectures of different complexity for a number of combinations of hyperparameters of the learning process. The differences in architectures concerned two aspects, namely the number of neurons in successive layers and the type of activation function. The exact set of architectures studied is shown in Table 2, while their general scheme is shown in Fig. 2. It was determined that the input of the network would be the following signals: normalized current spindle velocity, normalized current spindle acceleration and normalized distance to the next reference point. Normalization of the data was performed with respect to, accordingly: the maximum allowed spindle velocity, the maximum allowed spindle acceleration and the largest distance between adjacent reference points. Double deep Q-learning (DDQL) method was selected as the learning algorithm. It is also necessary to specify what signals the agent will receive after performing a particular action. A very simple system of assigning reinforcement was established, namely, when the agent performed an action in accordance with the decision of the RPRO algorithm, the reward was equal to 0, otherwise he received a penalty equal to −0.1. Exact form of reinforcement signal is described by formula (4).

$$r^t = \begin{cases} 0, & \text{if } a_{\text{agent}} = a_{\text{RPRO}}, \\ -0.1 & \text{otherwise.} \end{cases} \tag{4}$$

Table 2
Evaluated architecture parameters

Evaluated architecture number	Layerwise neuron count (N)	Layerwise activation function
1	24	ReLU
2	24	tanh
3	20	ReLU
4	20	tanh
5	16	ReLU
6	16	tanh
7	12	ReLU
8	12	tanh
9	8	ReLU
10	8	tanh
11	6	ReLU
12	6	tanh

At first, it was decided to evaluate the proposed solution depending on the density of the accumulation of reference points. Thus, the study was carried out for 3 subsets of the database corresponding to combinations of other parameters for trajectories of 15 points. Each subset contained 600 recorded machining processes. Learning was repeated on successive 5 pairs from the

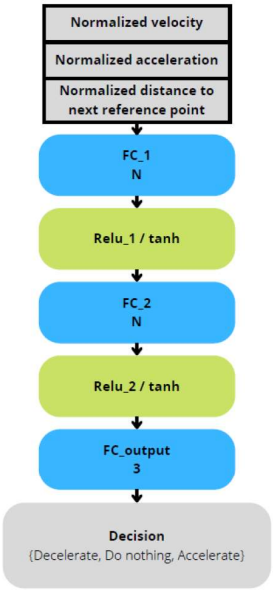


Fig. 2. Diagram of the neural network used

combinations for 10 trajectories, resulting in 120 processes in the training set and 480 in the test set. Each learning process was repeated 5 times to significantly reduce the impact of randomness in the learning process. All experiments were performed using the Matlab 2022a environment extended with the appropriate toolboxes. Subsequent parameters of the learning process were set as follows: *maxEpisodes* (500), *TargetSmoothFactor* (0.05), *TargetUpdateFrequency* (5), *MiniBatchSize* (64), *ExperienceBufferLength* (5000), *MaxStepsPerEpisode* (300). For convenience, the listed parameters are summarized in Table 3.

Table 3
Learning process hyperparameter values

Parameter name	Parameter value
maxEpisodes	500
TargetSmoothFactor	0.05
TargetUpdateFrequency	5
MiniBatchSize	64
ExperienceBufferLength	5000
MaxStepsPerEpisode	300

A measure of the level of replication of the RPRO algorithm's behavior by the proposed solution can be expressed by the Pearson correlation coefficient between the number of steps in suc-

cessive test machining processes. An interesting addition to it is also the average accuracy of the machining process expressed in micrometers. The listed metrics describing the results of the experiment are shown in Table 4 and Table 5. Note that these values are the medians from all repetitions and training pairs for specific combinations of architecture and learning process hyperparameters values. In the situations where the algorithm failed to achieve the intended effect (learning process did not reach convergence), it was not possible to determine the correlation coefficient, which is indicated in the tables by a triple pause (–). Due to the large number of failures for a subset of the base with sparsely distributed reference points, the corresponding part of the results was omitted and will become the focus of future studies.

Table 4

Results of the first experiment – median correlation coefficient between the number of steps in the processing performed by the RPRO algorithm and the trained agent

Steps count correlation		Discount factor					
		0.99			0.999		
Ref. points density	Network arch. number	Learning rate			Learning rate		
		1e-3	1e-4	1e-5	1e-3	1e-4	1e-5
High	1	0.847	0.750	0.848	0.647	0.844	0.844
	2	0.855	0.852	0.855	0.847	0.842	0.846
	3	0.844	0.855	0.848	0.839	0.777	0.847
	4	0.852	0.851	0.849	0.687	0.848	0.849
	5	0.849	0.856	–	0.844	0.853	–
	6	0.849	0.853	–	0.855	0.852	–
	7	0.825	0.857	–	0.847	0.846	–
	8	0.850	0.851	0.849	0.850	0.836	0.848
	9	0.821	0.856	–	0.845	0.850	–
	10	0.852	0.855	0.846	0.846	0.848	0.845
	11	0.842	0.851	–	0.848	0.848	–
	12	0.855	0.849	–	0.844	0.845	–
Medium	1	0.773	0.777	0.774	0.772	–	–
	2	0.777	0.774	0.774	–	0.472	0.538
	3	0.774	0.774	0.774	–	–	–
	4	0.771	0.777	0.775	–	0.773	0.726
	5	0.774	0.774	–	–	–	–
	6	0.771	0.775	0.664	–	0.386	0.237
	7	0.777	0.774	0.578	–	–	0.041
	8	0.775	0.776	0.774	–	0.171	0.172
	9	0.774	0.774	–	–	–	–
	10	0.774	0.775	0.774	–	0.009	0.112
	11	0.772	0.774	0.543	–	–	0.054
	12	0.774	0.775	–	–	–	–

Table 5

Results of the first experiment – median of average error of the machining process

Mean errror		Discount factor					
		0.99			0.999		
Ref. points density	Network arch. number	Learning rate			Learning rate		
		1e-3	1e-4	1e-5	1e-3	1e-4	1e-5
High	1	2.38	2.32	2.56	2.47	2.46	2.55
	2	2.45	2.43	2.51	2.46	2.44	2.63
	3	2.4	2.46	2.49	2.44	2.45	2.58
	4	2.45	2.49	2.51	2.52	2.52	2.46
	5	2.56	2.5	3.11	2.41	2.48	3.29
	6	2.47	2.43	4.69	2.48	2.35	4.81
	7	2.32	2.37	4.45	2.46	2.52	4.57
	8	2.49	2.47	2.56	2.49	2.43	2.57
	9	2.4	2.5	3.16	2.41	2.51	3.38
	10	2.45	2.48	2.54	2.39	2.55	2.54
	11	2.47	2.51	3.86	2.47	2.49	4.48
	12	2.41	2.51	435	2.46	2.48	437
Medium	1	5.53	5.9	5.41	2708	4496	4450
	2	5.79	5.31	5.14	4835	2661	2926
	3	5.54	5.03	1561	4481	4799	3580
	4	4.99	5.47	5.34	4708	2531	816
	5	5.37	6.2	7.08	4653	4742	4599
	6	5.08	5.56	2445	4852	1814	3607
	7	5.02	5.19	2414	3727	4741	3581
	8	5.42	5.14	5.06	4698	3369	2443
	9	4.83	5.14	3344	3568	3742	4455
	10	5.4	5.34	5.13	4677	4286	4366
	11	5.55	5.91	2446	4450	4622	3472
	12	5.21	5.32	4421	4736	2635	4832

The observed results show that for a subset of reference paths with dense reference points, the proposed method achieves a high level of correlation in the number of steps during individual machining processes. It is also interesting to note that the proposed solution achieves significantly better results in terms of machining process accuracy. However, in order to be able to verify whether this is by chance at the expense of the time spent on machining, it is necessary to analyze the median of the steps performed in the individual machining processes. For the RPRO algorithm, this value is 110 for densely spaced reference points and 500 for moderately spaced reference points, respectively. The corresponding medians for the proposed solution are shown in Table 6. It can be observed that for both subsets of the base the median number of steps is slightly smaller than for the RPRO algorithm. Thus, the trained agent does not behave

identically to the benchmark algorithm, but it achieves significantly better processing accuracy than the algorithm in a slightly shorter processing time.

Table 6
Results of the first experiment – median of the number of steps during the machining process

Steps count		Discount factor					
		0.99			0.999		
Ref. points density	Network arch. number	Learning rate			Learning rate		
		1e-3	1e-4	1e-5	1e-3	1e-4	1e-5
High	1	85	88	80	86	85	80
	2	83	83	81	83	82	82
	3	85	83	82	84	85	83
	4	82	82	80	84	80	82
	5	82	81	74	87	82	75
	6	83	84	69	82	84	72
	7	89	82	71	85	86	71
	8	82	83	80	81	88	81
	9	86	80	75	85	80	75
	10	83	81	80	83	83	82
	11	85	80	76	84	81	72
	12	84	80	1	83	80	1
Medium	1	471	469	476	179	71	101
	2	475	482	484	1	293	204
	3	478	482	446	73	1	101
	4	489	467	478	1	322	453
	5	482	465	431	1	1	101
	6	486	480	410	1	263	234
	7	487	482	406	87	1	101
	8	481	485	483	1	290	472
	9	488	486	367	159	82	109
	10	479	481	482	1	280	243
	11	476	471	487	68	1	258
	12	478	481	71	14	297	1

In addition, at this point it is also necessary to analyze the decrease in the quality of the trained agent's work with an increase in the interval between the reference points as seen in Table 4, Table 5 and Table 6. The intention of the first experiment was to test the ability of the studied algorithm to replicate the performance of the RPRO algorithm. Wanting to ensure the constancy of as many parameters as possible while reducing the computation time, the authors assumed that the aforementioned maximum length of the episode would be 300. Less frequently distributed reference points obviously lengthen the machining process, and limiting the episode to the given value meant that for an average distribution of them, the agent was not able to achieve as good results as for densely distributed points, since it did not have the opportunity to experience the end of the machining process during learning. For densely spaced reference points, on the other hand, the agent was unable to achieve satisfactory results at all, having had the opportunity to experience only a small initial portion of the entire machining process (300 out of 4446 steps or less than 7%). The last noteworthy fact is the indication of the presented results that potentially the best values of hyperparameters could be the learning rate equal to 0.001 and discount factor equal to 0.99, respectively. All tested architectures performed very well with a slight advantage for those using the sigmoidal activation function. This is very good news in terms of the development of the idea and the complication of the environment when attempting to achieve control in multiple dimensions of motion.

In the second experiment, the authors decided to investigate whether increasing the diversity of reference point trajectories in the learning dataset has a positive effect on the quality of the trained agent's match with the RPRO algorithm's behavior. Thus, it was decided to gradually increase the number of trajectories in the learning set starting with 2 and ending with 8. This time, the virtual window of the training set moved sequentially through a number of trajectories resulting in 8, 7, 6, . . . , 2 combinations of trajectories, respectively. The results were again combined by calculating the corresponding medians, which are shown in Table 7 and Table 8. In addition, in each row of Table 7, both the largest values of the correlation coefficient of the number of steps, as well as its second best values, are marked in bold. It should also be noted that for this experiment the maximum number of possible episodes was reduced by 20%, resulting in a value of 400. Analyzing the data presented, it can be indicated that the three best performing architectures are those numbered 2, 8 and 10, respectively. In accordance with previous observations, their common feature is the use of sigmoidal activation functions. The results also indicate that increasing the diversity of reference point paths in the learning dataset had no noticeable effect on the quality of the trained agent's performance. Thus, it can be conjectured that as few as two paths are sufficient for an agent to learn decision-making to a satisfactory degree while optimizing the machining process. It can be also observed that the choice of a learning rate coefficient equal to 1e-5 led to a disturbance in the stability of the learning process by, in several cases, failing to successfully complete a single trial. This clearly indicates that such small values of this coefficient should be avoided in the studied case.

At the end of this section we present a comparison of the simulation run of an example processing performed by the RPRO algorithm and in the way proposed by the authors in Fig. 3 and Fig. 4, respectively. The values shown in them are both the normalized spindle velocity and its normalized acceleration in successive time steps. The presented case shows the observations described so far, where the solution proposed by the authors creates a g-code that improves the machining process (realizing it in fewer steps and with higher accuracy). The RPRO algorithm completed the task during 86 steps, achieving an average accuracy of 47.94 micrometers, while the proposed solution

Table 7
Results of the second experiment – median correlation coefficient between the number of steps in the processing performed by the RPRO algorithm and the trained agent

Steps count correlation		Network architecture number											
Learning rate	Train trajectories count	1	2	3	4	5	6	7	8	9	10	11	12
0.001	2	0.827	0.839	**0.853**	–	0.807	0.832	0.828	0.833	0.846	0.835	**0.855**	0.844
	3	**0.858**	0.846	0.851	0.850	0.801	0.838	0.799	**0.856**	**0.858**	**0.858**	0.848	0.840
	4	**0.843**	0.840	0.836	0.827	0.842	0.833	0.798	0.835	0.814	**0.848**	0.831	0.832
	5	0.803	0.816	0.810	0.808	0.805	**0.821**	0.810	0.807	0.807	0.820	**0.822**	0.808
	6	0.819	0.813	0.814	0.812	0.818	–	0.811	**0.824**	0.815	0.821	0.817	**0.825**
	7	0.793	0.834	0.828	0.831	0.796	**0.837**	0.799	0.833	0.835	0.835	0.828	**0.838**
	8	–	**0.855**	0.844	0.848	0.717	0.850	0.842	0.851	**0.851**	0.850	0.844	0.850
0.0001	2	**0.855**	0.837	0.842	0.840	**0.852**	0.843	0.838	0.847	0.840	0.845	0.835	0.829
	3	**0.861**	0.841	0.849	0.843	0.852	0.849	0.846	0.842	0.842	**0.854**	0.839	0.836
	4	0.837	**0.844**	0.838	0.827	**0.839**	0.837	**0.839**	0.824	0.831	0.828	0.832	0.822
	5	0.811	0.815	**0.821**	0.808	0.815	0.808	0.812	**0.819**	**0.819**	0.812	0.815	0.804
	6	0.820	0.819	0.817	0.815	**0.827**	0.823	0.813	**0.826**	0.818	0.812	0.822	0.811
	7	0.837	0.836	**0.838**	0.834	**0.842**	0.835	0.828	0.833	0.831	0.835	0.834	0.828
	8	0.849	**0.854**	0.848	0.847	–	0.845	0.852	0.849	0.845	**0.853**	0.844	0.841
0.00001	2	0.827	**0.832**	–	0.829	–	–	–	**0.834**	–	0.831	–	–
	3	0.837	**0.853**	–	0.842	–	–	–	**0.843**	–	0.840	–	–
	4	0.824	**0.842**	–	**0.835**	–	–	–	0.824	–	0.827	–	–
	5	0.805	**0.827**	0.803	0.806	–	–	–	**0.810**	–	0.808	–	–
	6	**0.818**	**0.824**	0.815	0.812	–	–	–	0.810	–	0.809	–	–
	7	0.833	**0.855**	0.828	**0.840**	–	–	–	0.837	–	**0.840**	–	–
	8	0.848	**0.859**	**0.853**	0.842	–	–	–	0.849	–	0.849	–	–

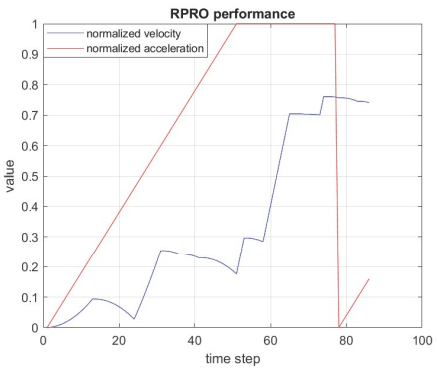


Fig. 3. RPRO algorithm performance on exemplary machining process

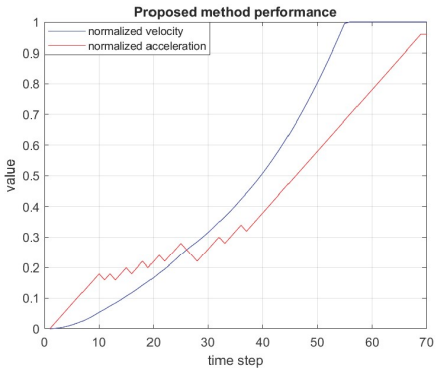


Fig. 4. Proposed method performance on exemplary machining process

Table 8
Results of the second experiment – median of average error of the machining process

Mean error		Network architecture number											
Learning rate	Train trajectories count	1	2	3	4	5	6	7	8	9	10	11	12
0.001	2	2.44	2.04	2.42	2.10	1.81	2.19	1.93	2.44	2.29	2.51	2.43	2.48
	3	2.38	2.46	2.47	2.55	2.47	2.59	2.24	2.48	2.64	2.51	2.58	2.58
	4	2.48	2.54	2.28	2.58	2.47	2.51	2.59	2.50	2.47	2.47	2.59	2.58
	5	2.30	2.55	2.36	2.53	2.38	2.48	2.58	2.48	2.54	2.49	2.48	2.19
	6	2.59	2.59	2.65	2.48	2.48	3.03	2.59	2.58	2.69	2.58	2.59	2.59
	7	2.49	2.59	2.63	2.61	2.05	2.53	2.30	2.59	2.59	2.59	2.59	2.59
	8	2.19	2.36	2.48	2.41	2.27	2.21	2.27	2.24	2.36	2.31	2.39	2.27
0.0001	2	2.47	2.47	2.47	2.48	2.44	2.56	1.80	2.47	2.51	2.58	2.43	2.51
	3	2.47	2.56	2.57	2.49	2.47	2.58	2.13	2.58	2.57	2.45	2.51	2.58
	4	2.58	2.47	2.48	2.59	2.59	2.59	2.40	2.48	2.60	2.58	2.48	2.58
	5	2.59	2.48	2.58	2.57	2.56	2.58	2.50	2.52	2.55	2.54	2.48	2.58
	6	2.56	2.59	2.66	2.60	2.59	2.59	2.56	2.58	2.59	2.59	2.52	2.58
	7	2.59	2.59	2.59	2.59	2.59	2.59	1.68	2.60	2.59	2.59	2.59	2.59
	8	2.19	2.48	2.25	2.32	2.71	2.47	2.16	2.48	2.48	2.43	2.42	2.41
0.00001	2	2.55	2.51	2.28	2.47	3.15	5.15	5.47	2.50	3.21	2.55	4.59	427.33
	3	2.56	2.58	2.21	2.47	3.10	5.10	4.14	2.47	2.86	2.53	5.58	448.67
	4	2.57	2.58	2.47	2.51	2.98	5.20	5.63	2.59	3.09	2.59	4.18	438.00
	5	2.58	2.54	2.62	2.48	3.06	5.25	5.77	2.48	3.06	2.47	3.51	427.33
	6	2.54	2.59	2.59	2.51	3.06	5.13	6.17	2.66	3.06	2.59	3.65	427.33
	7	2.66	2.65	2.59	2.62	2.76	4.92	5.12	2.63	2.76	2.59	3.22	427.33
	8	2.55	2.53	2.48	2.30	2.69	4.28	4.81	2.48	2.69	2.36	2.69	448.67

completed the task during 69 time steps, achieving an average accuracy of 32.98 micrometers. Another interesting aspect is the way in which the dynamics of the spindle's movement change, namely, unlike the RPRO algorithm, which pursued the highest possible acceleration, only to later reduce it just as sharply and reduce the speed as well, the solution obtained by the algorithm proposed by the authors relies on the gradual acceleration of the spindle until it reaches its maximum speed, taking into account minor adjustments that allow it to hit the reference points more accurately.

4. CONCLUSIONS

The authors presented a novel approach to the problem of optimizing the motion dynamics of a CNC machine. It consisted of using a deep learning algorithm with reinforcement to map the operation of the RPRO algorithm used in the industry. The presented solution achieved very good results – it mapped the operation of the RPRO algorithm to a satisfactory degree, and, in addition, it accelerated the machining process and provided distinctly higher accuracy (visibly lower average error). However, in order for the proposed solution to be put into industrial use, some improvements still need to be made and movement in multiple axes needs to be integrated simultaneously, which will be the main focus of the authors' further research. Attention will also be paid to optimizing other parameters of the machining process such as smoothing of acceleration and deceleration cycles, which has a direct impact on extending both machine service intervals and tool life.

REFERENCES

[1] J.E. Bobrow, S. Dubowsky, and J.S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Rob. Res.*, vol. 4, no. 3, pp. 3–17, 1985.

[2] J. H. Lee, Y. Liu, and S.-H. Yang, "Accuracy improvement of miniaturized machine tool: geometric error modeling and compensation," *Int. J. Mach. Tools. Manuf.*, vol. 46, no. 12–13, pp. 1508–1516, 2006.

[3] S.Z. Mansour and R. Seethaler, "Feedrate optimization for computer numerically controlled machine tools using modeled and measured process constraints," *J. Manuf. Sci. Eng.*, vol. 139, no. 1, p. 011012, 2017.

[4] Q. Bi, N. Huang, C. Sun, Y. Wang, L. Zhu, and H. Ding, "Identification and compensation of geometric errors of rotary axes on five-axis machine by on-machine measurement," *Int. J. Mach. Tools. Manuf.*, vol. 89, pp. 182–191, 2015.

[5] X. Li, H. Zhao, X. Zhao, and H. Ding, "Interpolation-based contour error estimation and component-based contouring control for five-axis CNC machine tools," *Sci. China. Technol. Sci.*, vol. 61, pp. 1666–1678, 2018.

[6] B. Kwiatkowski, T. Kwater, D. Mazur, and J. Bartman, "An offline application that determines themaximum accuracy of the realization ofreference points from G-code for givenparameters of CNC machine dynamics," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 72, p. e147345, 2024, doi: 10.24425/bpasts.2023.147345.

[7] J.M. Langeron, E. Duc, C. Lartigue, and P. Bourdet, "A new format for 5-axis tool path computation, using Bspline curves," *Comput.-Aided Des.*, vol. 36, no. 12, pp. 1219–1229, 2004.

[8] Y. Sun, S. Sun, J. Xu, and D. Guo, "A unified method of generating tool path based on multiple vector fields for CNC machining of compound NURBS surfaces," *Comput.-Aided Des.*, vol. 91, pp. 14–26, 2017.

[9] M. Chen and Y. Sun, "A moving knot sequence-based feedrate scheduling method of parametric interpolator for CNC machining with contour error and drive constraints," *Int. J. Adv. Manuf. Technol.*, vol. 98, pp. 487–504, 2018.

[10] B. Pękala, E. Rak, B. Kwiatkowski, A. Szczur, and R. Rak, "The use of concave and convex functions to optimize the feed-rate of numerically controlled machine tools," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2020, pp. 1–8.

[11] M. Rahaman, R. Seethaler, and I. Yellowley, "A new approach to contour error control in high speed machining," *Int. J. Mach. Tools Manuf.*, vol. 88, pp. 42–50, 2015.

[12] S.D. Timar, R.T. Farouki, T.S. Smith, and C.L. Boyadjieff, "Algorithms for time–optimal control of CNC machines along curved tool paths," *Robot Comput. Integr. Manuf.*, vol. 21, no. 1, pp. 37–53, 2005.

[13] J. Dong and J. A. Stori, "A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization," *J. Dyn. Sys., Meas., Control.*, vol. 128, no. 2, pp. 379390, 2006.

[14] Z. Shiller and H.-H. Lu, "Robust computation of path constrained time optimal motions," in *Proc. IEEE International Conference on Robotics and Automation*, IEEE, 1990, pp. 144–149.

[15] S.D. Timar and R.T. Farouki, "Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds," *Robot. Integr. Manuf.*, vol. 24, no. 1, pp. 16–31, 2008.

[16] C. Wang, X.P. Tan, S.B. Tor, and C.S. Lim, "Machine learning in additive manufacturing: State-of-the-art and perspectives," *Addit. Manuf.*, vol. 36, p. 101538, 2020.

[17] A. Molina, H. Ponce, P. Ponce, G. Tello, and M. Ramírez, "Artificial hydrocarbon networks fuzzy inference systems for CNC machines position controller," *Int. J. Adv. Manuf. Technol.*, vol. 72, pp. 1465–1479, 2014.

[18] T. Kar, N.K. Mandal, and N.K. Singh, "Multi-response optimization and surface texture characterization for CNC milling of inconel 718 alloy," *Arab. J. Sci. Eng.*, vol. 45, pp. 1265–1277, 2020.

[19] S. Datta, S.S. Mahapatra, B.C. Routara, and A. Bandyopadhyay, "The fuzzy inference system approach to a multi-performance characteristic index for surface quality improvement in CNC end milling," *Int. J. Exp. Des. Process Optim.*, vol. 2, no. 3, pp. 265–282, 2011.

[20] M.F. Alam, M. Shtein, K. Barton, and D.J. Hoelzle, "Autonomous manufacturing using machine learning: A computational case study with a limited manufacturing budget," in *International Manufacturing Science and Engineering Conference*, 2020, p. V002T07A009.

[21] K. Arulkumaran, M.P. Deisenroth, M. Brundage, and A.A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, 2017.

[22] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[23] G. Lample and D.S. Chaplot, "Playing FPS games with deep reinforcement learning," in *Proc. AAAI Conference on Artificial Intelligence*, 2017.

[24] D. Kalandyk, "Reinforcement learning in car control: A brief survey," in *2021 Selected Issues of Electrical Engineering and Electronics, WZEE 2021*, 2021. doi: 10.1109/WZEE54157.2021.9576838.

[25] J. Marquez, C. Sullivan, R.M. Price, and R.C. Roberts, "Hardware-in-the-Loop Soft Robotic Testing Framework using an Actor-Critic Deep Reinforcement Learning Algorithm," *IEEE Robot. Autom. Lett.*, vol. 8, no. 9, pp. 6076–6082, 2023, doi: 10.1109/LRA.2023.3301215.

[26] Q. Su, B. Li, C. Wang, C. Qin, and W. Wang, "A power allocation scheme based on deep reinforcement learning in HetNets," in *2020 international conference on computing, networking and communications (ICNC)*, IEEE, 2020, pp. 245–250.

[27] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, 2018.

[28] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proc. IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13470–13479.

[29] X. Wang *et al.*, "Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning," *J. Manuf. Syst.*, vol. 65, pp. 130–145, 2022.

[30] G. Velusamy and R. Lent, "Evaluating reinforcement learning methods for bundle routing control," in *2019 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, IEEE, 2019, pp. 1–4.

[31] A.M. Seid, G.O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-UAV-assisted IoT networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12203–12218, 2021.

[32] C. Li, P. Zheng, Y. Yin, B. Wang, and L. Wang, "Deep reinforcement learning in smart manufacturing: A review and prospects," *CIRP J. Manuf. Sci. Technol.*, vol. 40, pp. 75–101, 2023.

[33] B. Fernandez-Gauna, I. Ansoategui, I. Etxeberria-Agiriano, and M. Graña, "Reinforcement learning of ball screw feed drive controllers," *Eng. Appl. Artif. Intell.*, vol. 30, pp. 107–117, 2014.

[34] S. Baer, J. Bakakeu, R. Meyes, and T. Meisen, "Multi-agent reinforcement learning for job shop scheduling in flexible manufacturing systems," in *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, IEEE, 2019, pp. 22–25.

[35] Y.-C. Wang and J. M. Usher, "Application of reinforcement learning for agent-based production scheduling," *Eng. Appl. Artif. Intell.*, vol. 18, no. 1, pp. 73–82, 2005.

[36] T. Zhou, D. Tang, H. Zhu, and Z. Zhang, "Multi-agent reinforcement learning for online scheduling in smart factories," *Robot Comput. Integr. Manuf.*, vol. 72, p. 102202, 2021.

[37] X. Jing, X. Yao, M. Liu, and J. Zhou, "Multi-agent reinforcement learning based on graph convolutional network for flexible job shop scheduling," *J. Intell. Manuf.*, pp. 1–19, 2022.

[38] K. Chang, S.H. Park, and J.-G. Baek, "AGV dispatching algorithm based on deep Q-network in CNC machines environment," *Int. J. Comput. Integr. Manuf.*, vol. 35, no. 6, pp. 662–677, 2022.

[39] J. Yao, B. Lu, and J. Zhang, "Tool remaining useful life prediction using deep transfer reinforcement learning based on long short-term memory networks," *Int. J. Adv. Manuf. Technol.*, vol. 2021, p. 9937846, 2022.

[40] R. Lamprecht, F. Wurst, and M.F. Huber, "Reinforcement learning based condition-oriented maintenance scheduling for flow line systems," in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, IEEE, 2021, pp. 1–7.

[41] M.L.R. Rodríguez, S. Kubler, A. de Giorgio, M. Cordy, J. Robert, and Y. Le Traon, "Multi-agent deep reinforcement learning based Predictive Maintenance on parallel machines," *Robot. Comput. Integr. Manuf.*, vol. 78, p. 102406, 2022.

[42] A. Mishra and V. S. Jatti, "Reinforcement learning based approach for the optimization of mechanical properties of additively manufactured specimens," *Int. J. Interact. Des. Manuf.-IJIDeM*, vol. 17, pp. 2045–2053, 2023.

[43] D. Limoge, A. Sunstrom, V. Pinskiy, and M. Putman, "Defending Industrial Production Using AI Process Control," in *2020 IEEE Systems Security Symposium (SSS)*, IEEE, 2020, pp. 1–4.

[44] Y. Zhang, Y. Li, and K. Xu, "Reinforcement learning based tool orientation optimization for five-axis machining," *Int. J. Adv. Manuf. Technol.*, vol. 119, no. 11–12, pp. 7311–7326, 2022.

[45] Q. Xiao, C. Li, Y. Tang, and L. Li, "Meta-reinforcement learning of machining parameters for energy-efficient process control of flexible turning operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 5–18, 2019.

[46] W. Li *et al.*, "A novel milling parameter optimization method based on improved deep reinforcement learning considering machining cost," *J. Manuf. Process.*, vol. 84, pp. 1362–1375, 2022.

[47] H.A. Taha, S. Yacout, and Y. Shaban, "Deep Reinforcement Learning for autonomous pre-failure tool life improvement," *Int. J. Adv. Manuf. Technol.*, vol. 121, no. 9–10, pp. 6169–6192, 2022.

[48] X. Liu and D. Chang, "An Improved Method for Optimizing CNC Laser Cutting Paths for Ship Hull Components with Thicknesses up to 24 mm," *J. Mar. Sci. Eng.*, vol. 11, no. 3, p. 652, 2023.

[49] M.F. Alam, M. Shtein, K. Barton, and D. Hoelzle, "Reinforcement learning enabled autonomous manufacturing using transfer learning and probabilistic reward modeling," *IEEE Control Syst. Lett.*, vol. 7, pp. 508–513, 2022.

[50] L. Jonath, J. Luderich, J. Brezina, A.M. Gonzalez Degetau, and S. Karaoglu, "Improving the Thermal Behavior of High-Speed Spindles Through the Use of an Active Controlled Heat Pipe System," in *International Conference on Thermal Issues in Machine Tools*, 2023, pp. 203–218.

[51] G. Singh and R. Sharma, "Cnc Machine Handling for Holes Servicing through Programming," in *Proc. International Conference on Innovative Computing & Communication (ICICC) 2022*, 2023.

[52] F. Jaensch, A. Csiszar, J. Sarbandi, and A. Verl, "Reinforcement learning of a robot cell control logic using a software-in-the-loop simulation as environment," in *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, IEEE, 2019, pp. 79–84.

[53] Q. Chen, B. Heydari, and M. Moghaddam, "Leveraging task modularity in reinforcement learning for adaptable industry 4.0 automation," *J. Mech. Des.*, vol. 143, no. 7, p. 071701, 2021. doi: 10.1115/1.4049531.

[54] A.A. Apolinarska *et al.*, "Robotic assembly of timber joints using reinforcement learning," *Autom. Constr.*, vol. 125, p. 103569, 2021.

[55] B. Li, H. Zhang, P. Ye, and J. Wang, "Trajectory smoothing method using reinforcement learning for computer numerical control machine tools," *Robot. Comput. Integr. Manuf.*, vol. 61, p. 101847, 2020.

[56] Y. Jiang, J. Chen, H. Zhou, J. Yang, P. Hu, and J. Wang, "Contour error modeling and compensation of CNC machining based on deep learning and reinforcement learning," *Int. J. Adv. Manuf. Technol.*, pp. 1–20, 2022.

[57] C. Dripke, S. Höhr, A. Csiszar, and A. Verl, "A concept for the application of reinforcement learning in the optimization of CAM-generated tool paths," in *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2016*, Springer, 2017, pp. 1–8.

[58] V. Samsonov, E. Chrismarie, H.-G. Köpken, S. Bär, D. Lütticke, and T. Meisen, "Deep representation learning and reinforcement learning for workpiece setup optimization in CNC milling," *Prod. Eng.*, vol. 17, no. 6, pp. 847–859, 2023.

[59] Z. Lin, T. Chen, Y. Jiang, H. Wang, S. Lin, and M. Zhu, "B-Spline-Based Curve Fitting to Cam Pitch Curve Using Reinforcement Learning," *Intell. Autom. Soft Comput.*, vol. 36, no. 2, p. 2145, 2023.

[60] D. Kalandyk, B. Kwiatkowski, and D. Mazur, "Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics," in *IEEE International Conference on Fuzzy Systems*, 2023. doi: 10.1109/FUZZ52849.2023.10309802.

Engineering Applications of Artificial Intelligence 131 (2024) 107879

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Temporal signed gestures segmentation in an image sequence using deep reinforcement learning

Dawid Kalandyk [a,*], Tomasz Kapuściński [b]

[a] Doctoral School of the Rzeszów University of Technology, al. Powstancow Warszawy 12, 35-959 Rzeszow, Poland
[b] Department of Computer and Control Engineering, Rzeszow University of Technology, Wincentego Pola 2, 35-959 Rzeszow, Poland

ARTICLE INFO

Keywords:
Deep reinforcement learning
Image sequence segmentation
Gesture spotting
Gesture database

ABSTRACT

Continuous sign language recognition is challenging due to coarticulatory distortions, which occur at the beginning and end of each gesture. These distortions depend on the temporal context and introduce additional intraclass variability. To address this issue, a new approach is proposed that extracts segments from the image sequence corresponding to undistorted parts of gestures. This should simplify the task by reducing it to the easier problem of isolated gestures recognition. The proposed approach uses deep reinforcement learning for segmentation and a novel image sequence processing scheme to extract gradient changes over time. A dataset recorded by deaf people and annotated according to the proposed approach, was prepared to evaluate the method. The proposed deep learning architectures achieved leave-one-subject-out recognition accuracies in the range of 0.70 to 0.76. Considering the inability to compare with other works, the authors also proposed other evaluation protocols to thoroughly examine the employed approach. This work will be developed, and the main aspiration of the authors will be to create an integrated framework that converts the raw form of RGB video into a string of words representing the Sign Language user's intentions.

1. Introduction

1.1. Related works

For more than 25 years, the task of automatically interpreting human gestures to develop the integration of humans with a variety of computer systems has challenged many researchers. Some of the earliest works in this area are: spotting gestures from time-varying images (Nishimura and Oka, 1996), spotting dynamic gestures using Hidden Markov Models (HMM) (Morguet and Lang, 1998), Active Gesture Recognition (AGR) using Learned Visual Attention (Darrell and Pentland, 0000), AGR using Partially Observable Markov Decision Processes (POMDP) (Darrell and Pentland, 1996), and using RL to Active Behaviors Recognition (Darrell, 1997). The main branches that developed in the following years were: *(i)* finger-counting using Impulse Radar with Convolutional Neural Network (CNN) (Ahmed et al., 2019); *(ii)* static hand gesture recognition: a real-time approach (Zhu et al., 2002), using HMM (Elmezain et al., 2009), using AutoGesNet (Li et al., 2020); *(iii)* dynamic gesture recognition: for video game control (Kang et al., 2004), real-time approach (Neto et al., 2013), using Long Short-Term Memory (LSTM) Convolutional Recurrent Neural Networks (CRNN) (Tsironi et al., 2017), using Deep Reinforcement Learning (DRL) and IoT sensor device (Seok et al., 2018), using Reinforcement Learning (LR) (Zhang et al., 2019), using RGB-D camera and k-nearest neighbours (k-NN) algorithm with dynaming time warping (DTW) and HMM (Kapuscinski and Wysocki, 2020). Particularly noteworthy are solutions that address both indirectly and directly the tasks of gesture analysis such as face recognition (Rao et al., 2017; Nicholl et al., 2010), face aging (Duong et al., 2019), emotion recognition (Ouyang et al., 2017), masked facial emotion recognition (Thanathamathee et al., 2023), electroencephalogram-based emotion recognition (Wirawan et al., 2022), gesture model learning (Wilson and Bobick, 2000) and even improving robot training process by human gestures support (Cruz et al., 2018; Jevtić et al., 2018).

The complexity of the gesture recognition field has been demonstrated and analyzed in numerous review papers over the years: *(i)* static/hand gesture recognition: (Mitra and Acharya, 2007) in 2007, (Trigueiros et al., 2012) and Hasan and Kareem (2012) in 2012, (Sarkar et al., 2024) in 2013, (Pisharady and Saerbeck, 2015) in 2015, (Sagayam and Hemanth, 2017) in 2017, (Singh et al., 2019) and Anwar et al. (2019) in 2019, finally (Sarma and Bhuyan, 2021) in 2021; *(ii)* dynamic/continuous gesture recognition: (Neiva and Zanchettin, 2018) in 2018, (Aloysius and Geetha, 2020) in 2020, finally (Jain et al., 2022) in 2022; *(iii)* gesture databases/datasets: (Ruffieux et al., 2014)

* Corresponding author.
E-mail addresses: d.kalandyk@prz.edu.pl (D. Kalandyk), tomekkap@prz.edu.pl (T. Kapuściński).

https://doi.org/10.1016/j.engappai.2024.107879
Received 6 September 2023; Received in revised form 9 December 2023; Accepted 8 January 2024
Available online 13 January 2024

in 2014, (Pisharady and Saerbeck, 2015) in 2015, (Sarma and Bhuyan, 2021) in 2021, finally (Jain et al., 2022) in 2022. The issues, despite such developments, are therefore still relevant and constantly being addressed.

Research on automatic sign language recognition aims to bridge communication gap between deaf and hearing communities. The techniques mentioned in the literature can be categorized into two groups: one uses specialized gloves with sensors (Cooper and Bowden, 2010; Pezzuoli et al., 2019), while the other relies on vision-based solutions (Koller et al., 2018, 2019). The former is known for its high precision but lacks flexibility, which results from additional hardware and violates the natural interaction paradigm. Hence, this study aims to explore vision-based solutions. Among the different solutions, two problems can be identified: recognizing isolated and continuous sign language. In the former, signing gestures are shown separately with the beginning and the end of the gesture being marked specifically, for instance by lowering the hand. In the latter, gestures are shown in sequence just as they would be in natural conversation. Since it is of greater practical importance, this work focuses on continuous sign language recognition. The task can typically be divided into two parts: visual representation learning and sequence correspondence learning. In earlier works, hand-crafted features like histogram of oriented gradients (HOG) (Buehler et al., 2009) or scale-invariant feature transform (SIFT) (Pfister et al., 2013) were used. Later, solutions started using deep convolutional networks (DCNN) (Qiu et al., 2019) or their variants (Hu et al., 2021; Pu et al., 2018). Sequence correspondence learning is usually based on hidden Markov models (HMM) (Zhang et al., 2016), encoder–decoder architectures (Guo et al., 2019), recurrent neural networks (RNN) (Cui et al., 2017; Min et al., 2021), or hybrid approaches (Koller et al., 2018). The task is quite difficult. Review studies available in the literature (Bragg et al., 2019) highlight the main challenges faced by researchers, such as the lack of publicly available data sets that are representative, performed by deaf individuals, and properly annotated. This is particularly true for languages other than American Sign Language (ASL). Moreover, during quick and spontaneous conversations, interjected gestures and coarticulatory distortions occur in continuous signed communication. These distortions depend on the temporal context, affect the beginning and end of each gesture, and introduce additional intraclass variability during machine learning. Therefore, this work presents a new dataset and an alternative approach. It involves temporal segmentation of sign sequences to extract undistorted gesture fragments, which can then be classified using one of the established methods for isolated sign language recognition.

The reinforcement learning algorithm, and in particular its deep versions, is also booming. Its versatility combined with the capabilities of deep artificial neural networks allow it to be applied to a wide range of image processing tasks such as: image blending (Hung et al., 2018), video fast-forwarding (Lan et al., 2018), object finding and tracking (Minut and Mahadevan, 2001; Yun et al., 2017; Supancic, III and Ramanan, 2017), frame iterpolation (Bao et al., 2019), game playing (Mnih et al., 0000, 2015; Lample and Chaplot, 2017), robot and vehicle control tasks (Cruz et al., 2016; Kim et al., 2017) wider described in Kalandyk (2021). Many existing algorithms including reinforcement learning algorithms are also used for semantic segmentation of images to facilitate processing. Examples include: liver tumor detection (Durrani et al., 2022), multimodal medical images (Ye et al., 2021) and histopathology images (Park et al., 2022) analysis, or even steel surface defects detection (Bi et al., 2022).

1.2. Proposed solution

The authors of this study decided to attempt an automatic temporal segmentation of sign language gestures, in order to isolate the individual gestures and clear them of the noise caused by additional hand movements both before the first gesture, between successive gestures and after the last gesture in the sequence. A deep reinforcement learning algorithm was used for this purpose. To the best of the authors' knowledge, this is a novel solution, the application of which should allow better recognition results for dynamic sign language gestures. In addition, the algorithm has the potential to be used for other tasks with a similar purpose, such as the detection of pedestrian behavior anomalies. Proposed segmentation task requires specially prepared database, the labels of which enable the proposed algorithm to be trained. The structure of prepared database, as well as the database itself and its creation process, are described in this work. Algorithm 1 presents briefly main parts of proposed framework and on Fig. 1 it is illustrated in the context of the whole continuous sign language gestures recognition task.

Algorithm 1: Proposed framework

Data: X; %Input RGB image sequence
Result: Y; %Segmentation results (labels)
1 $X_{processed} \leftarrow DataPreprocessing(X)$; %See Section 3.1
2 $Y_{preliminary} \leftarrow Segmentation(X_{processed})$; %See Section 3.2
3 $Y \leftarrow ResultsSmoothing(X)$; %See Section 3.3

Contributions of this work are as follows:

- A deep reinforcement learning algorithm was used for signed gestures extraction by temporal segmentation of image sequence to enable transition from a continuous to isolated gesture recognition task.
- A novel image sequence processing scheme to extract temporal changes allowing agent to focus on distinctive part of information which better represents subject's movement was introduced.
- A dedicated database for evaluation of the proposed system has been introduced along with appropriate labeling. The database was created with the help of deaf people, who are potential users of the future system.
- A suitable method for adjustment of the proposed system response which providing a beneficial effects on its performance has been proposed.

The rest of the paper is organized as follows. In Section 2, new database is introduced. Section 3 describes details of the proposed segmentation method including data preprocessing, reinforcement learning implementation and results post-processing. Section 4 presents results and analysis of conducted experiments. Finally, Section 5 concludes the paper and indicates directions of future research.

2. Proposed database

To the best of the authors' knowledge, the available databases are oriented towards the task of gesture recognition. By this statement, it shall be understood that the labels consist of blocks of consecutive numbers denoting specific gestures. This way of assigning labels is based on the assumption that a person's movement is a representation of successive gestures, each of which can be slightly deformed. Such assumptions do not allow the extraction of potentially superfluous parts of the video that could negatively affect the performance of the gesture recognition algorithm, as stated in Section 1. Therefore, there is a need for a database which structure makes it possible to distinguish both the gestures themselves, the transitions between them and hand moves before and after gesture sequence. According to the thought of the authors of this work, the database should allow to distinguish the mentioned fragments of a person's movement. This approach to the problem will make it possible to extract key chunks of the video unambiguously representing particular gestures.

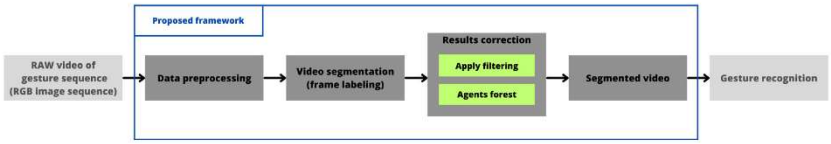


Fig. 1. Proposed framework.

2.1. The source database creation and upgrade process

Guided by these reasons, the authors decided to create a database suitable for new approach needs by upgrading database introduced in Kapuscinski and Wysocki (2020) using yet unpublished annotation files and modify it, which will be discussed further in this section. Original database was made in association with the Subcarpathian Association of the Deaf. The (Kapuscinski and Wysocki, 2020) authors made recordings of a number of expressions (gesture sequences) used during a conversation with an administrative officer. In addition to deaf people (subject presenting gestures), professional sign language interpreters were also present during the recordings. In order to organize the base in the subsequent study, it was decided to keep a fixed distance between the person being recorded and the camera lens. The person was additionally seated on a chair, behind which a solid blue background was set. A Microsoft Kinect Xbox One 2.0 camera with a frame rate of 29.98 fps was used for the recordings. The obtained images were RGB images with 1920×1080 dimensions saved as files with the JPEG extension. As stated before to address the challenge, novel annotations were created using ELAN[1] software (Crasborn and Sloetjes, 2008). Each video formed from the previously obtained images was annotated given a series of start- and end-gesture markers. The annotations thus prepared were saved in files with the .*eaf extension (one file for each video).

2.2. Modified database structure analysis

To ensure uniformity and consistency of the data, the expressions for which there are **5 repetitions of each expression for each person** were selected. An additional condition was the existence of the corresponding annotation file for each repetition. All requirements were met by **33 expressions, which were demonstrated by 3 subjects**. Thus, after the selection process, **495 videos with a total of 40716 frames** were obtained with **average length of 82 frames**. Brought together they build up a database of which diagram is shown in Fig. 2, while a list of the individual expressions qualified for it are presented in Table 1.

It should be, taken into consideration at this point, that Polish Sign Language has different syntax and grammar from spoken Polish language, which means that the expressions presented are not correct sentences in Polish. In order to better understand the content that is expressed by the individual expressions, an explanatory column including their meanings in spoken English language has been added to Table 1. It should also be mentioned that one given word in Polish may be represented by more than one gesture. For convenience, such gestures have been numbered consecutively (e.g. *dowod1* and *dowod2*). In addition, many different expressions can convey the same intention, which is also quite natural in spoken languages. It is also important to be aware that, in different regions of a country, the same words are naturally expressed by different gestures. The database was recorded for the Subcarpathian region.

[1] Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands (https://archive.mpi.nl/tla/elan).

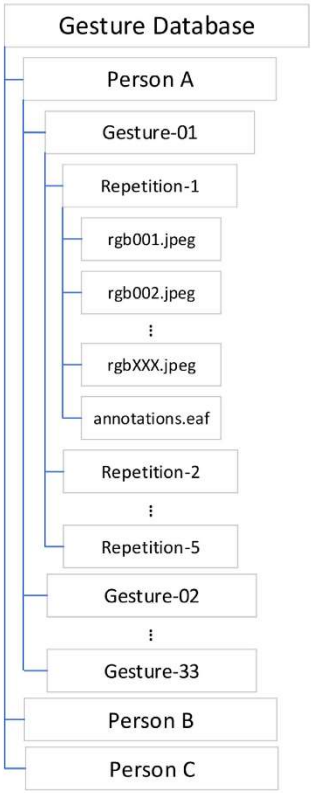


Fig. 2. Database structure.

To better understand characteristics of the database which allows to analyze the results of the study in more detail,

Table 2, Table 3, Table 4, and Table 5 present the distributions of the gestures and the existing combinations of two gestures in the database. The statistics show the over-representation of the gestures: *dowod1*, *dowod2*, *ile*, *ja*, *mozna*, *nowy* and *wczesniej*, as well as the gesture pair *nowy_dowod2*. It can therefore be expected that their segmentation (and recognition) will be more effective than others. The frequency of each gesture as the first and last gesture in an expression also varies. Counter-intuitive, the unbalanced nature of the

Table 1
Used Polish Sign Language Expressions and their meaning in English. The underscore sign separates the successive labels of each gesture in the expression.

Lp.	Gesture sequence	Gesture sequence meaning in spoken language
1.	adres1_zmiana1	I have changed my address.
2.	adres1_zmiana2	I have changed my address.
3.	adres2_zmiana2	I have changed my address.
4.	byc_moj_nowy_dowod2	Can i receive my identity card?
5.	chciec_nowy_dowod2	I would like to apply for an identity card.
6.	co_pisac_miec	I don't know how to fill in an application.
7.	czy_juz_nowy_dowod1	Can i receive my identity card?
8.	czy_juz_nowy_dowod2	Can i receive my identity card?
9.	dlaczego	Why?
10.	dom_przeniesc_sie	I have changed my address.
11.	dowod1_koniec_wazny	The expiry date of the identity card has passed.
12.	dowod1_odbior_wczesniej_mozna	Is it possible to come earlier to collect an identity card?
13.	dowod1_wczesniej_gotowy_mozna	Can the identity card manufacturing be faster?
14.	dowod2_gotowy	Can i receive my identity card?
15.	dowod2_koniec_wazny	The expiry date of the identity card has passed.
16.	dowod2_niewazny	The expiry date of the identity card has passed.
17.	dowod2_odbior_wczesniej_mozna	Is it possible to come earlier to collect an identity card?
18.	dowod2_wczesniej_gotowy_mozna	Can the identity card manufacturing be faster?
19.	dowod2_zgubic	I have lost my identity card.
20.	dziekowac1	Thank you.
21.	dziekowac2	Thank you.
22.	dzien_dobry	Good morning.
23.	ile_dni_czekac	How long do I have to wait?
24.	ile_koszt_nowy_dowod1	How much do I pay to obtain an identity card?
25.	ile_koszt_nowy_dowod2	How much do I pay to obtain an identity card?
26.	ile_placic_dowod1	How much do I pay to obtain an identity card?
27.	ile_sztuk_zdjecie_potrzeba	How many photos are required?
28.	ja_dowod1_moja_zona_chciec	I want to apply for an identity card for my wife.
29.	ja_dowod1_zgubic	I want to report the loss of my identity card.
30.	ja_dowod2_moje_dziecko_chciec	I want to apply for an identity card for my child.
31.	ja_dowod2_zgubic	I want to report the loss of my identity card.
32.	ja_malzenstwo	I got married.
33.	ja_nie_rozumiec_co_pisac	I don't know how to fill in an application.

Table 2
The number of occurrences of individual gestures in the considered expressions.

Number of occurrences	Gestures
1	adres2; byc; czekac; dlaczego; dni; dobry; dom; dziecko; dziekowac1; dziekowac2; dzien; malzenstwo; miec; moj; moja; moje; nie; niewazny; placic; potrzeba; przeniesc; rozumiec; sie; sztuk; zdjecie; zmiana1; zona
2	adres1; co; czy; juz; koniec; koszt; odbior; pisac; wazny; zmiana2
3	chciec; gotowy; zgubic
4	mozna; wczesniej
5	ile
6	ja; nowy
8	dowod1
12	dowod2

Table 3
The number of occurrences of individual gestures at the beginning of the considered expressions.

Number of occurrences	Gestures
1	adres2; byc; chciec; co; dlaczego; dom; dziekowac1; dziekowac2; dzien
2	adres1; czy
3	dowod1
5	ile
6	dowod2; ja

database can be treated as an advantage, as it presents data taken from the real world rather than artificially created. The obtained database supports the creation and testing of solutions for both segmentation and recognition of continuous sign language gestures. It was named GEST. Its original form and processed form are available at http://vision.kia.prz.edu.pl/.

Table 4
The number of occurrences of individual gestures at the end of the considered expressions.

Number of occurrences	Gestures
1	czekac; dlaczego; dobry; dziekowac1; dziekowac2; gotowy; malzenstwo; miec; niewazny; pisac; potrzeba; sie; zmiana1
2	chciec; wazny; zmiana2
3	dowod1; zgubic
4	dowd2; mozna

3. Proposed method

The gesture recognition problem can be solved by dividing it into two main tasks, namely the segmentation of the continuous data stream and the classification of the gestures presented within each segment. To facilitate the second task, an initial categorization of the conversation topic can additionally be performed, and only then, delegated for gesture classification by a subject-specialized classifier. This issue will be considered in a future work, but the initial focus should be on how to perform segmentation of the continuous data stream. The solution proposed by the authors includes the following parts: data pre-processing, reinforcement learning and post-processing of the results.

3.1. Data preprocessing

There is a number of factors that can potentially confuse the algorithm's segmentation task or unnecessarily expand the domain in which it operates. These can include skin color, clothing color, hair color, subject image occupancy factor, background shadows, background motion, etc. If the color of the various elements just mentioned is similar, the algorithm may have difficulty recognizing the correct hand movement and interpreting it correctly. An example of this situation can be

(a) Data stream at Stage 2.

(b) Data stream at Stage 8.

Fig. 3. GSADGM processing results.

observed in Fig. 3(a), where the color of the trouser fragment is similar to the color of the hand.

To address those issues, the authors decided to focus the algorithm's attention on body movement, which should decrease above-mentioned factors influence. Consequently, each video was processed through a series of operations in accordance with the procedure provided in the Algorithm 2 which, along with additional and detailed information, has been shown in Fig. 4. Initially, each person's videos were cropped using bounding box tailored respectively (stg. 1 → stg. 2). The main purpose of such action is to reduce unnecessary background scenery around the person showing the gestures. This cropping reduced the amount of data to an average of 77% of the original size. Subsequently, a conversion of each frame from RGB color space to gray scale (GS) is performed (stg. 2 → stg. 3). Such an action reduced the amount of data by three more times - the data that had been processed in this way already took up less than 26% of the original size. The Sobel filter is used to calculate images gradients (*imgradient* Matlab function). Only information about the magnitudes of the calculated gradients is left for further transformations, which did not change the dimensionality of the data (stg. 3 → stg. 4). In the following step differential images between successive frames of the film are calculated, which effects in reducing the number of frames in each video by one (stg. 4 → stg. 5). Such operation makes it possible to extract the contours of the subject carrying information about its movement. During the course of the work, it became apparent that some differential images were completely black, which meant that the two consecutive frames of the film were identical and therefore did not represent the movement of the subject. Difference images that do not carry the information (the number of which is on average 20%), had to be eliminated from the sequence (stg. 5→ stg. 6). At the end of the pre-processing, the images were scaled down to 224 × 224, which made them uniform to fit the network input, and they were normalized using min–max formula form range 0–255 to floating point numbers within range 0–1 (stg. 6 → stg. 7). The described processing method was named Gray Scale Absolute Difference Gradient Method (GSADGM).

Algorithm 2: Proposed data preprocessing approach

Data: X; %Input RGB image sequence
Result: Y; %GSADGM results
1 $Y_{Stg1-2} \leftarrow Crop(X)$; %according to the subject
2 $Y_{Stg2-3} \leftarrow RGB2GS(Y_{Stg1-2})$
3 $Y_{Stg3-4} \leftarrow imgradient(Y_{Stg2-3}, 'Sobel')$; %Only magnitudes
4 $Y_{Stg4-5} \leftarrow ConsecutiveFramesAbsoluteDifference(Y_{Stg3-4})$;
5 $Y_{Stg5-6} \leftarrow RemoveEmptyFrames(Y_{Stg4-5})$;
6 $Y_{Stg6-7} \leftarrow ScaleDown(Y_{Stg5-6}, 224, 224)$;
7 $Y \leftarrow Normalize(Y_{Stg6-7}, 0, 1)$;

In parallel to the processing of the films, processing of the annotation files (*.eaf) began. Based on the read tags, each video was annotated by assigning it a vector composed of '0' and '1', denoting a gesture break and an ongoing gesture, respectively. During testing, however, it became apparent that some pauses between gestures were too short, which led to their elimination during differential imaging. It was therefore decided to adopt vectors with extended gaps for the experiments. Empirically, gaps of length 1 and 2 frames were selected for the expansion, and symmetrically stretched to last 3 and 4 frames respectively. This action did not significantly affect the content of the gestures shown ($2\ frames \approx 0.067s$) and avoided manual correction of the *.eaf files, which saved a tremendous amount of time. After the application of all transformations, the database contains a total of 31741 frames which is about 78% of the initial value, and the average length of the video is 64.12 frames.

Table 5
The number of occurrences of pairs of gestures in the considered expressions.

Number of occurrences	Gesture pairs
1	adres1_zmiana1; adres1_zmiana2; adres2_zmiana2; byc_moj; chciec_nowy; dni_czekac; dom_przeniesc; dowod1_koniec; dowod1_moja; dowod1_odbior; dowod1_wczesniej; dowod1_zgubic; dowod2_gotowy; dowod2_koniec; dowod2_moje; dowod2_niewazny; dowod2_odbior; dowod2_wczesniej; dziecko_chciec; dzien_dobry; ile_dni; ile_placic; ile_sztuk; ja_malzenstwo; ja_nie; moj_nowy; moja_zona; moje_dziecko; nie_rozumiec; pisac_miec; placic_dowod1; przeniesc_sie; rozumiec_co; sztuk_zdjecie; zdjecie_potrzeba; zona_chciec
2	co_pisac; czy_juz; dowod2_zgubic; gotowy_mozna; ile_koszt; ja_dowod1; ja_dowod2; juz_nowy; koniec_wazny; koszt_nowy; nowy_dowod1; odbior_wczesniej; wczesniej_gotowy; wczesniej_mozna
4	nowy_dowod2

3.2. Deep reinforcement learning for segmentation task

Reinforcement Learning (RL) refers to the idea of placing a so-called agent in a specific environment (Kaelbling et al., 1996). In each time step t the agent is allowed to perform one of the available actions. Based on the environment state $s^{(t)}$ observation the agent chooses an action $a^{(t)}$ to perform. This affects the environment, as well as, transits the agent and environment to a new state $s^{(t+1)}$. The agent also receives a reward $r^{(t)}$ which evaluates agents moves. The agent uses a strategy π, starts in state s and chooses there the action a is expected to accumulate a total reward expressed by (1), which is called action value function Q. It is also worth noting that for manipulating the agent's so-called foresight discounting factor γ is used. Its value shall be within range (0;1]. In addition, the choice of actions in the next steps is made based on a greedy algorithm that prefers actions with a higher Q-value. Depending on the needs, plenty of RL algorithms that were brought together in Arulkumaran et al. (2017), can be used. By using ANNs as a function approximator it is possible not only to

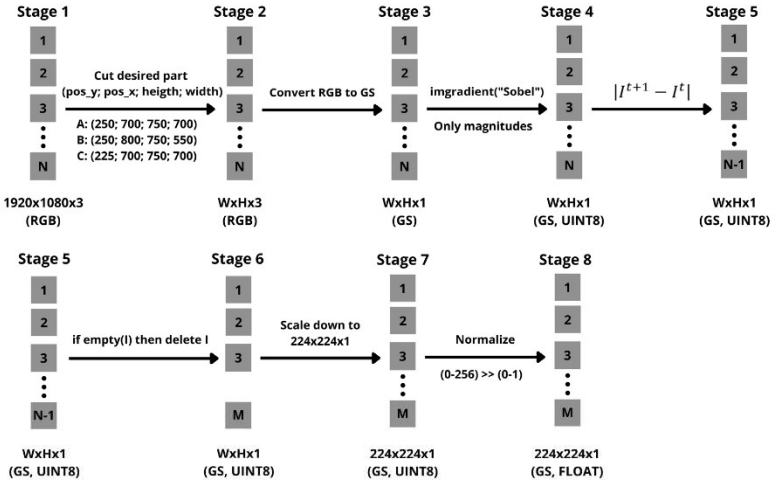


Fig. 4. Pipeline of the proposed data preprocessing approach.

select discrete action, but also to determine continuous values. It is called Deep Reinforcement Learning (DRL). Such possibility is a great opportunity to apply RL algorithms in complex decision-making tasks, where both environment knowledge and ability to predicting its future behavior are needed to accomplish them.

$$Q_\pi(s,a) = E_\pi[r_0 + \sum_{t=0} \gamma^t r^t \text{ where } s_0 = s, a_0 = a] \quad (1)$$

One such task might be deciding whether to split a video stream into frames sequences, containing gestures and frames sequences, containing pauses. The authors therefore assumed that the episode for the agent would be the evaluation of the individual frames of the video. The agent should have information about the current state of the environment at its disposal to respond correctly. This information is contained in the appropriately processed images described earlier. To broaden the agent's perspective, it was decided that instead of the currently classified frame of the film, frames within a radius (R) of it should also be available for the agent to inspect. For this reason, the state was defined as a given set of pre-processed film frames, respectively. To better adapt the algorithm to the complex task, the approximation of the action value function was delegated to a deep convolutional neural network. In order to ensure fast convergence and a stable learning process as well, the Double Deep Q-Network algorithm (Van Hasselt et al., 2016) was used. The exact architecture of the network (being the representation of both Policy Network and Value Network) have been presented in the form of a diagram in Fig. 7 and Fig. 9 as well as described in Section 4. It should be noted that the network is an extremely important part of the learning process, as it is what defines the limits of an agent's perceptual capabilities.

It is also necessary to specify the reinforcement signal, which boiled down to assigning a fixed value of 0.0 in the case of a correct frame classification and -0.2 in the case of a mistake. The corresponding equation (2) describes the reinforcement signal, where l_i is the label designated by the agent while l_i^* is the label as determined by the database. This way of shaping the reinforcement signal ensures that the agent is motivated to make as few mistakes as possible, while at the same time there is no limit to the length of the episode. By episode length, we mean the number of frames of the video minus twice the radius of the agent's field of view. The study of other complex reinforcement signals is left for future work.

$$r^l = \begin{cases} -0.2, & \text{if } l_i \neq l_i^* \\ 0, & \text{if } l_i = l_i^* \end{cases} \quad (2)$$

The entire segmentation process for an example agent field-of-view radius of 2 was schematically shown in Fig. 5. Red, green and blue colors have been used to indicate the successive sets of data given at the input to calculate the respective first second and third labels. Fields marked with a cross symbolize frames for which the algorithm does not determine labels, and their number is always equal to twice the agent's viewing range ($2R$), which in the example considered is equal to 4. The remaining frames are assigned a label equal to 0 (break from showing the gesture) or 1 (showing the gesture) depending on the agent's decision.

3.3. Results post-processing

The result of the agent's work is a vector of labels assigned to each block of the input data (preprocessed film). It has length of $M - 2R$. To support the analysis of the obtained results, a comparison between the proper labels (ground truth) and those determined by the agent (segmentation results) was presented also in the form of graph. Exemplary comparisons are illustrated in Fig. 6. Black circles indicate labels that are in accordance with the database, while blue circles indicate labels that are the result of the agent's work. In addition, in between, green and red asterisks mark the correctness of the decision at each time step. The areas marked with a red box are places where the agent for a short moment (few frames) recognizes that the state of the environment has changed to the opposite (*break* → *gesture* or *gesture* → *break*). While the pauses or transitions between gestures can last one or two frames, the gestures cannot be that short. The authors therefore proposed that the vector obtained by the agent describing the labels of the video should be subjected to filtering. The following filters have been proposed: (i) to remove hills and valleys of length 1 (Cutting Peaks - CP_1); to remove only hills of length 1 (Cutting Peaks Hills Only - CP_HO_1); to remove only hills of length 1 and 2 (Cutting Peaks Hills Only - CP_HO_2).

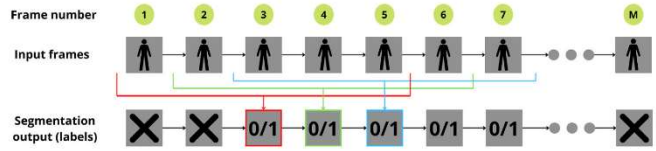
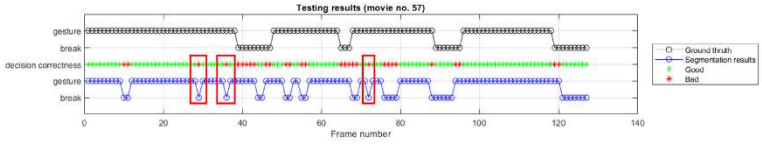
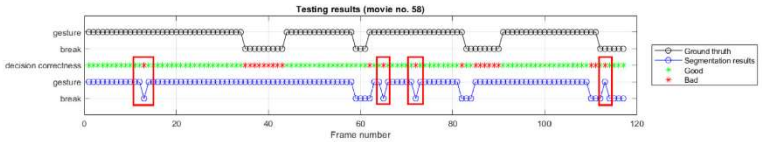


Fig. 5. Agents working process scheme for view range of 2 frames ($R = 2$). Red, green and blue colors indicate the successive sets of data given at the input to calculate the respective first second and third labels. Fields marked with a 0/1 symbolize frames for which the algorithm determine labels. Fields marked with a cross symbolize frames for which the algorithm does not determine labels by design.



(a) Example segmentation results (movie no. 57)



(b) Example segmentation results (movie no. 58)

Fig. 6. Example segmentation results check. Ground truth frame labels sequence are marked by black circled line while segmentation results frame labels sequence are marked by blue circled line. Label validity tags for each frame separately were added for ease of sequence comparison. Thus, a green asterisk indicates a correct label, while a red asterisk indicates a wrong label. In addition, red rectangles mark situations of potential use of one of the proposed filters.

To demonstrate the stability of the solution, the learning process for each combination of parameter was repeated 15 times. Inspired by the tree forest method, the authors decided to test whether the result of voting on the labels by multiple independently learned agents, would represent a good segmentation of the input data. Voting results were prepared for the usual test results and for the results after application of each of the three filters. The decision to label a given video frame is made by a simple majority vote.

4. Experiments

At the beginning of this section, the evaluation protocols will be defined. Next, the research methodology will be presented, including the metrics used to describe the quality of the proposed solution's performance. The research environment will also be described and the hyperparameters of the learning process will be specified. This will be followed by a presentation of the results from the two successive phases of the conducted research and their analysis with a conclusion.

4.1. Evaluation protocols

Due to the fact that there is a lack of work with which to compare the results achieved by the proposed system, the authors of this paper, wanting to study it as carefully as possible, proposed a series of experiment protocols. Each protocol should be understood as a selection of an appropriate subset of the available database. Each successive protocol was increasingly diverse in terms of the subjects showing the gestures. In addition, the last protocol (LOSO) also introduces an additional difficulty, which is the segmentation of recordings of a subject whose records have not been previously observed by the algorithm. Such approach will allow the quality of the proposed segmentation algorithm to be analyzed. In total, the following four protocols were introduced:

- **SS (Single Subject)** – dividing the data into training and test sets of 80% and 20% respectively. The protocol considers five-fold cross-validation respective to repetitions of each subject. This protocol represents the lowest level of difficulty of the segmentation task and allows to test whether the proposed algorithm is able to tackle this type of task.
- **DS (Double Subject)** – dividing the data into training and test sets of 80% and 20% respectively. The protocol considers five-fold cross-validation respective to repetitions of each subject tuple. This protocol represents the average difficulty level of the segmentation task. Differences in the physical appearance of the subjects displaying the gestures and the accuracy of the gesture display can be here an additional difficulty.
- **TS (Triple Subject)** – dividing the data into training and test sets of 80% and 20% respectively. The protocol considers five-fold cross-validation respective to repetitions of all subjects. This protocol represents a high level of difficulty of the segmentation task. Differences in the size of the subjects showing the gestures and the accuracy of the gesture display can be here an additional difficulty. This protocol also provides the greatest variety of subjects.
- **LOSO (Leave One Subject Out)** – dividing the data into testing data, which include all the films of the selected subject, and training data, which are the films of the other subjects. This protocol assumes that experiments are conducted for the selection of each of the available subjects. It represents by far the highest level of difficulty of the segmentation task. An additional difficulty may be the need to segment the videos of a person whose pattern has never been observed by the algorithm and diversity which cannot be well represented by limited dataset used. This is also the situation that is closest to the real

environment conditions of the subsequent operation of the proposed algorithm.

The second aspect of experiments was the choice of an appropriate Deep Convolutional Neural Network (DCNN) architecture. This is an extremely important element, as it determines the agent's potential perception of information about the environment, i.e. its perceptual abilities. Guided by the practical application of the introduced solution, the authors decided to study architectures similar to existing ones presented in Simonyan and Zisserman (2015), but much more shallow than them. In addition, less classical approach was tested, that prioritizes information reduction by using a MaxPool layer at the beginning of the network. All architectures examined in specific experiments phase are shown in further subsections.

A third important aspect was the choice of the radius of the agent's viewing window. What is worth considering, the selection of this parameter can potentially bring positive, as well as, negative results. The expected benefit may be an improvement in the quality of the system's performance, which will have access to additional information describing the current state of the environment. The negative aspect, on the other hand, will be an increase in the complexity of the aforementioned DCNN and a possible lengthening of its learning process. In the successive phases of the experiments, the impact of different values of this parameter was investigated.

4.2. Methodology

Three metrics were adopted to evaluate the results of the algorithm. The first is Accuracy (Performance - Perf) (3) which defines the ratio of the number of correctly annotated frames to the number of total frames in a given video. The second (Gesture Difference - GD) (4) and the third (Breaks Difference - BD) (5) are calculated based on the absolute value from the difference of the number of gestures recognized by the agent and the correct number of gestures (and similarly for breaks). In (3), (4) and (5) symbol L represents labels vector for specific frames sequence and L^* represents proper labels vector.

$$M_{Perf} = \frac{\sum_{l=1} \zeta(L_l)}{len(L)} \text{ where } \zeta(L_l) = \begin{cases} 0 \text{ for } L_l \neq L_l^* \\ 1 \text{ for } L_l = L_l^* \end{cases} \tag{3}$$

$$M_{GD} = \frac{\zeta_G(L)}{\zeta_G(L^*)} \text{ where } \zeta_G(L) = \{\text{'1' segments count }\} \tag{4}$$

$$M_{BD} = \frac{\zeta_B(L)}{\zeta_B(L^*)} \text{ where } \zeta_B(L) = \{\text{'0' segments count }\} \tag{5}$$

4.3. Evaluation environment

The calculations were carried out using a computer architecture based on an AMD EPYC 7742 64-Core processor, two Nvidia A100 40 GB cards and access to 1TB RAM. To exclude randomness and to demonstrate the stability of the solution, the learning process for each of the later described parameter combinations was repeated 15 times. In order to speed up the computation, parallelization was applied eventually training 5, 8 or 15 agents simultaneously depending on the complexity (learnable variables number) of the deep network. The total number of trained networks was 3180.

The learning process was limited to 500 episodes ensuring its convergence. The discount factor was set to 0.02 since, in the adopted environment and with the chosen reward system, choosing a suboptimal action in one move has no effect on choosing an optimal action in the next move. In order to ensure fast convergence and a stable learning process, the Double Deep Q-Network algorithm (Van Hasselt et al., 2016) was used by adopting Matlab framework parameters values accordingly: *TargetSmoothFactor* (0.05), *TargetUpdateFrequency* (5) and *NumStepsToLookAhead* (1). The *EpsilonGreedyExploration* algorithm was chosen as the exploration method with the parameters: *EpsilonInitial* (1.0), *EpsilonMin* (0.01) and *EpsilonDecay* (0.01). This provided a trade-off between exploration and exploitation. ADAM was adopted as the optimization algorithm, setting the learning rate to 0.0001, the maximum buffer capacity to 5000 and the *MiniBatchSize* to 128. Implementation of the solution and experiments were conducted using Matlab 2022a software. All the hyperparameters discussed, along with their values, have been gathered in Table 6.

Table 6
Learning hyperparameters values.

Name	Value
discountFactor	0.02
maxEpisodes	500
TargetSmoothFactor	0.05
TargetUpdateFrequency	5
NumStepsToLookAhead	1
EpsilonInitial	1.0
EpsilonMin	0.01
EpsilonDecay	0.01
LearningRate	0.0001
maxBufferCapacity	5000
MiniBatchSize	128

4.4. Phase I

In the first phase, four deep convolutional neural network architectures (illustrated in Fig. 7) and three different agents viewing window radii (equal respectively to 1, 2 and 3) were examined. The main objective of the first round of testing was to reduce the number of experiments by discovering a potential trend for the size of the agent's viewing window radius or deep convolutional network architecture. To achieve this goal, a study was performed according to the SS protocol. It results taking into account the filters discussed earlier and the voting result for each filter are presented in Table 7. The studied architectures differ in the arrangement of layers - two are similar to resnet and another two have a maxpool layer in front, which at the very beginning leads to a significant reduction in the data processed by these networks. In addition, the architectures in the two pairs differ in the size and number of filters in the subsequent convolution layers.

Thanks to the appropriate selection of hyperparameters, all the realized learning processes reached convergence within the assumed time. To better understand the course of the learning process, the record of one of them was analyzed. Fig. 8 shows the values of the metrics obtained by the agent during each of the 500 episodes - respectively, in the top graph there is a record of the classification accuracy of successive blocks of input data, while below in yellow and cyan color there are discrepancies in the number of gestures and gaps between gestures, as determined by the agent. Each metric was also supplemented with a moving average calculated on the basis of the last 50 episodes. Bearing in mind that in each episode one of 132 films is drawn and shown to the agent (33 expressions times 4 repetitions), it can be estimated that during the learning process each film was viewed by the agent an average of 4 times. The very high score of the performance metric translates directly into a high percentage of correctly classified blocks. The low average values of the other two metrics (below 1.0 at the end of the learning process) should also be taken into account. Combined with the high classification rate, this is a very good indication of the correct performance of the algorithm.

The analysis should also include the results obtained when testing the system on repetitions that the agent has not seen during the learning process. These test results, have been collected in Table 7. All metrics shown are averages and include all videos from the test set for each agent trained for a given configuration. Bold indicates the best score for a given metric among the scores for a given architecture and filter. The best and second-best score of the metric for a given filter among the results marked in bold are highlight with gray background. It is worth noting that M_{Perf} should be close to 1.0 and both M_{GD} and M_{BD} should be close to 0.0.

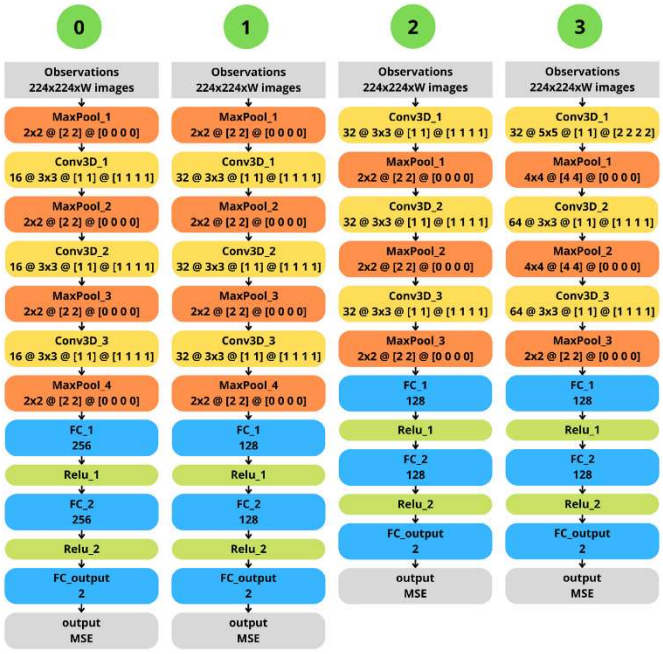


Fig. 7. Phase I examined architectures.

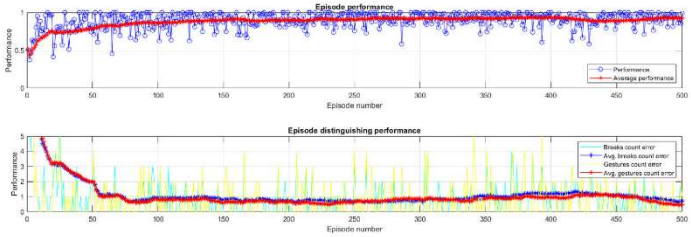


Fig. 8. Exemplary agent learning process.

Analyzing the data presented in Table 7, one can easily notice a positive correlation between the radius of the agent's range of view and the quality of the segmentation achieved. This relationship is preserved both for the results of individual agents and for the forests of agents. It is also noteworthy that it has been a definite improvement in system performance when using the agents forest method - on average about 0.02 to 0.03 in accuracy which is around 3% increase and in some cases almost double decrease in Gesture Difference and Breaks Difference. The best architecture turned out to be the '0' architecture achieving almost always first or second place. This is an interesting phenomenon because it is not a standard architecture starting from the convolution layer, but a MaxPool. The two classical-like architectures ('2' and '3') achieved comparable results, which were also satisfactory. When analyzing the effect of the applied filters on the results, it can be seen that the CP_1 filter was the most effective for individual agents and CP_HO_1 filter was the most effective for agents forest. As intended by the authors, the proposed method is able to correctly segment videos containing continuous sign language gestures. It has also been possible to successfully study the impact of using post-processing methods and preliminarily assess the capabilities of different types of architectures.

4.5. Phase II

After the initial diagnosis in phase one, the authors moved on to the second phase of the study. Based on the conclusions drawn in the first phase, they decided that the second phase would cover:

- test of the best classical-like phase I architecture and three new architectures (two classical-like ones of different complexity and one similar to the phase I best with higher complexity). All phase II architectures with layers characteristics were shown in Fig. 9;

Fig. 9. Phase II examined architectures.

Table 7
Phase I mean results.

Results Type	Filter Type	View Window Radius	Performance Network Architecture 0	1	2	3	Gesture Difference Network Architecture 0	1	2	3	Breaks Difference Network Architecture 0	1	2	3
Agents	none	1	0.851	0.851	0.856	0.853	0.719	0.736	0.800	0.763	0.835	0.865	0.891	0.888
		2	0.862	0.863	**0.859**	0.864	0.650	0.643	0.775	0.594	0.775	0.755	0.858	0.724
		3	**0.864**	**0.864**	0.858	**0.868**	**0.570**	**0.610**	**0.734**	**0.543**	**0.701**	**0.751**	**0.858**	**0.705**
	CP_1	1	0.850	0.850	0.857	0.853	0.711	0.740	0.739	**0.756**	0.850	0.864	0.848	0.879
		2	0.862	0.862	**0.861**	0.863	0.661	0.688	**0.706**	0.669	0.789	0.810	**0.833**	0.803
		3	**0.863**	**0.863**	0.858	**0.865**	**0.638**	**0.671**	0.725	0.659	**0.765**	**0.800**	0.845	**0.790**
	CP_HO_1	1	0.852	0.851	0.856	0.853	0.560	0.575	**0.590**	0.593	0.703	0.723	0.708	0.743
		2	0.862	0.863	**0.860**	0.865	0.521	0.519	0.605	0.481	0.660	**0.648**	0.739	**0.629**
		3	**0.864**	**0.864**	0.857	**0.867**	**0.474**	**0.513**	0.594	**0.470**	**0.613**	0.669	**0.738**	0.686
	CP_HO_2	1	0.852	0.852	0.856	0.853	0.629	0.648	0.698	0.674	0.758	0.785	0.804	0.813
		2	0.863	0.863	**0.860**	0.865	0.579	0.575	0.678	0.541	0.708	**0.698**	0.804	0.676
		3	**0.865**	**0.864**	0.858	**0.867**	**0.516**	**0.556**	**0.651**	**0.505**	**0.650**	0.700	**0.781**	**0.665**
Agents forests	none	1	0.881	0.878	0.887	0.880	0.432	0.424	0.472	0.455	0.557	0.568	0.561	0.587
		2	0.893	0.890	0.891	0.888	0.337	0.360	**0.356**	0.345	0.472	0.534	0.511	0.461
		3	**0.897**	**0.892**	**0.895**	**0.891**	**0.258**	**0.288**	0.360	**0.303**	**0.432**	**0.439**	0.462	**0.435**
	CP_1	1	0.876	0.875	0.885	0.876	0.633	0.606	0.610	0.689	0.780	0.750	0.705	0.811
		2	0.890	**0.887**	0.886	0.885	0.553	0.617	**0.572**	0.591	0.648	0.742	**0.678**	**0.716**
		3	**0.891**	0.886	**0.889**	**0.886**	**0.508**	**0.595**	0.625	**0.591**	**0.640**	**0.689**	0.727	0.723
	CP_HO_1	1	0.881	0.879	0.888	0.880	0.360	0.379	0.367	0.409	0.530	0.545	0.508	0.553
		2	0.893	0.890	0.891	0.888	0.299	0.311	**0.318**	0.345	**0.417**	0.489	**0.477**	0.470
		3	**0.897**	**0.892**	**0.894**	**0.892**	**0.258**	**0.273**	0.326	**0.295**	0.420	**0.405**	0.481	**0.443**
	CP_HO_2	1	0.881	0.879	0.887	0.881	0.379	0.394	0.375	0.436	0.538	0.553	0.511	0.568
		2	0.894	0.890	0.891	0.888	0.311	0.326	**0.333**	0.345	**0.424**	0.500	0.489	0.473
		3	**0.897**	**0.892**	**0.895**	**0.891**	**0.254**	**0.284**	**0.333**	**0.311**	0.428	**0.428**	**0.466**	**0.458**

Table 8
Phase II mean results.

Results Type	Protocol Name	Filter Type	Performance Network Architecture 2	4	5	6	Gesture Difference Network Architecture 2	4	5	6	Breaks Difference Network Architecture 2	4	5	6
Agents	SS	none	0.850	0.857	0.850	0.852	0.888	0.691	1.012	0.881	0.999	0.827	1.107	0.993
		CP_1	0.852	0.858	0.853	0.855	0.661	0.576	0.675	0.646	0.817	0.753	0.815	0.787
		CP_HO_1	0.850	0.857	0.850	0.852	0.713	0.557	0.765	0.687	0.855	0.723	0.907	0.833
		CP_HO_2	0.850	0.857	0.850	0.853	0.785	0.605	0.878	0.767	0.914	0.761	0.996	0.899
	DS	none	0.840	0.846	0.836	0.840	0.997	0.817	1.212	1.013	1.101	0.945	1.301	1.117
		CP_1	0.842	0.847	0.840	0.843	0.742	0.685	0.793	0.741	0.883	0.845	0.926	0.889
		CP_HO_1	0.840	0.846	0.836	0.840	0.801	0.673	0.945	0.813	0.933	0.828	1.076	0.949
		CP_HO_2	0.840	0.847	0.836	0.840	0.880	0.730	1.061	0.896	1.000	0.871	1.172	1.016
	TS	none	0.830	0.836	0.824	0.829	1.068	0.914	1.286	1.088	1.162	1.026	1.364	1.186
		CP_1	0.832	0.837	0.829	0.833	0.810	0.764	0.860	0.808	0.940	0.904	0.988	0.938
		CP_HO_1	0.831	0.836	0.825	0.830	0.878	0.774	1.020	0.892	1.004	0.910	1.136	1.016
		CP_HO_2	0.830	0.836	0.824	0.830	0.954	0.826	1.136	0.970	1.066	0.952	1.232	1.082
	LOSO	none	0.714	0.720	0.705	0.714	1.567	1.403	2.150	1.650	1.623	1.497	2.233	1.713
		CP_1	0.720	0.724	0.713	0.720	1.140	1.130	1.247	1.163	1.267	1.293	1.387	1.290
		CP_HO_1	0.713	0.718	0.702	0.713	1.387	1.267	1.817	1.443	1.473	1.387	1.940	1.533
		CP_HO_2	0.714	0.719	0.704	0.714	1.467	1.320	1.970	1.537	1.547	1.430	2.080	1.617
Agents forests	SS	none	0.885	0.888	0.886	0.889	0.392	0.313	0.406	0.362	0.564	0.507	0.560	0.539
		CP_1	0.882	0.885	0.885	0.886	0.509	0.465	0.483	0.459	0.651	0.630	0.622	0.610
		CP_HO_1	0.885	0.888	0.887	0.889	0.362	0.293	0.354	0.317	0.539	0.475	0.521	0.489
		CP_HO_2	0.885	0.888	0.887	0.889	0.368	0.289	0.370	0.325	0.549	0.483	0.527	0.503
	DS	none	0.880	0.882	0.881	0.880	0.501	0.462	0.525	0.499	0.665	0.645	0.679	0.671
		CP_1	0.878	0.879	0.878	0.879	0.591	0.554	0.594	0.583	0.752	0.722	0.735	0.736
		CP_HO_1	0.880	0.882	0.882	0.881	0.468	0.426	0.476	0.453	0.642	0.611	0.639	0.614
		CP_HO_2	0.880	0.882	0.882	0.881	0.479	0.433	0.490	0.465	0.645	0.622	0.655	0.628
	TS	none	0.870	0.874	0.869	0.873	0.584	0.566	0.655	0.582	0.733	0.723	0.780	0.743
		CP_1	0.868	0.871	0.868	0.871	0.703	0.697	0.719	0.681	0.834	0.816	0.861	0.830
		CP_HO_1	0.871	0.874	0.869	0.874	0.558	0.549	0.612	0.539	0.705	0.681	0.754	0.697
		CP_HO_2	0.871	0.874	0.870	0.874	0.564	0.560	0.626	0.539	0.709	0.701	0.766	0.705
	LOSO	none	0.751	0.759	0.756	0.757	1.119	1.079	1.166	1.073	1.317	1.275	1.349	1.313
		CP_1	0.752	0.760	0.757	0.758	1.271	1.240	1.149	1.192	1.412	1.410	1.313	1.404
		CP_HO_1	0.749	0.758	0.754	0.755	1.115	1.081	1.121	1.081	1.289	1.255	1.297	1.317
		CP_HO_2	0.750	0.758	0.755	0.756	1.123	1.077	1.133	1.093	1.297	1.263	1.327	1.319

Table 9
Total learnables of proposed architectures (in thousands).

Network architecture	View window radius 1	2	3
0	874	875	875
1	839	840	840
2	3 248	3 248	3 249
3	476	478	479
4	–	–	1 693
5	–	–	6 752
6	–	–	6 641

- conducting the test only for an agent's window of view with a radius equal to 3, because this value turned out to be unquestionably the best of all those studied so far;
- perform the test for all four proposed experiments protocols (SS, DS, TS and LOSO), which should provide insight into the proposed system and allow for a very thorough evaluation of it under different conditions;
- re-examination the impact of filtering and the voting results (so far made only in the SS model).

The proposed Architecture 4 has doubled the number of neurons in the fully connected layers compared to Architecture 1. Architecture 6 has been modified similarly compared to Architecture 2, and in addition to this, the number of filters in all convolutional layers has also been increased, the number of which has been increased by 1. Architecture 5, on the other hand, has an additional 3 convolutional layers compared to Architecture 6.

Table 8 shows the metrics resulting from the work of the obtained agents on the respective test data. The data illustration style used matches that proposed in the section discussing the first phase of the experiments. As expected, a slight decrease in performance can be observed in association with an increase in task difficulty. This is particularly noticeable for the LOSO protocol, which is caused by the lack of diversity in representation of the test set in the training set. The authors believe that expanding the proposed database to include more subjects should fill this gap and at the same time provide a much better learning environment for the system under study. The results presented also confirm previous observations regarding the positive impact of the filters used, and the introduced voting method. The combination of the two methods provides, as before, about a 3%-3.5% increase in the classification accuracy of the data blocks received at the input, as well as, in many cases, works very well by strongly reducing the values of the other two metrics (GD and BD). The most effective architectures turned out to be those numbered 4 and 6. These are architectures representing the classical-like approach (6) and the approach proposed by the authors of the paper (4). At this point, it is worth noting that the proposed architecture number 4 has a considerably lower complexity (number of learnable parameters), which makes it more attractive for commercial applications.

For a better analysis of the complexity of the investigated deep convolutional neural networks, the number of learnable parameters for each architecture considering different widths of the agent's view window is presented in Table 9. It is also worth noting that the complexity of the proposed architectures, which can be expressed in terms of learnable parameters, is a few or even up to a tens of times lower than for state-of-the-art architectures such as resnet18 (11.6 mln), resnet50 (25.5 mln) or vgg16 (138.3 mln).

4.6. Trained filters

In order to better understand how the information provided in the input data block is analyzed by a trained agent, it was decided to analyze the performance of learned filters. Thus, Fig. 10 shows an example of a block of input data for the agent's field-of-view radius $R = 2$, while Fig. 11 graphically shows the excitation of neurons for the selected 7 of 32 filters. In order to improve the readability of the graphics, the colors have been inverted. It was observed that some filters focused on the analysis of specific channels (frames from the input block) others taking into account only to a small extent - filter 3

Fig. 10. Exemplary deep network input block ($R = 2$).

Fig. 11. Selected 7 of 32 first convolutional layer filters activations.

Table 10
Comparison of best's agents metrics.

Results Type	Protocol Name	Performance Network Architecture				Gesture Difference Network Architecture				Breaks Difference Network Architecture			
		2	4	5	6	2	4	5	6	2	4	5	6
Agents	SS	0.852	0.858	0.853	0.855	0.661	0.576	0.675	0.646	0.817	0.753	0.815	0.787
	DS	0.842	0.847	0.840	0.843	0.742	0.685	0.793	0.741	0.883	0.845	0.926	0.889
	TS	0.832	0.837	0.829	0.833	0.810	0.764	0.860	0.808	0.940	0.904	0.988	0.938
	LOSO	0.720	0.724	0.713	0.720	1.140	1.130	1.247	1.163	1.267	1.293	1.387	1.290
Agents Forests	SS	0.885	0.888	0.887	0.889	0.362	0.293	0.354	0.317	0.539	0.475	0.521	0.489
	DS	0.880	0.882	0.882	0.881	0.468	0.426	0.476	0.453	0.642	0.611	0.639	0.614
	TS	0.871	0.874	0.869	0.874	0.558	0.549	0.612	0.539	0.705	0.681	0.754	0.697
	LOSO	0.752	0.760	0.757	0.758	1.271	1.240	1.149	1.192	1.412	1.410	1.313	1.404

and 5 from the left. In the same time, others considered a larger number of frames while differing in their subset. Such a combination not only provides multi-modal analysis of the received block of data, but also creates an extensive set of features on the basis of which, the agent is later able to make accurate recognition of the state of the person showing the gesture by appropriately classifying the received blocks of data.

4.7. Summary

Presented results build on existing evidence of that it is possible to transit from the very difficult task of continuous sign language gesture recognition to the far simpler task of isolated gesture recognition through temporal video segmentation using a deep learning algorithm with reinforcement. At the same time, an additional achievement of this work is the proposal of an algorithm for preprocessing the video stream, which enabled not only the reduction of information wich may be potentially confusing for the system, but also the extraction of valuable information about the movement of the subject's hand. In addition, this algorithm also provided up to almost 75% reduction in data volume. In addition, the authors of the paper also presented not only a database adapted to the new type of task, but also methods to improve the performance of the system (obtained labels) providing a significant increase in classification precision.

Table 10 shows the best results for each of the experiment protocols and tested deep neural network architectures as a trade-off between the values of each metric. It can easily be seen that the voting method achieves significantly better results despite the more difficult task (TS vs. SS).

The proposed system, despite achieving high quality results, will be further developed. The authors, wanting to optimize its performance, will undergo a detailed evaluation:

- larger values of the radius of the agent's field of view in order to find a compromise between the size of the block forming the input and the computation time focusing on ensuring real-time operation of the system;
- deep network architectures with a different complexity than currently studied - deep analysis will be carried out on currently learned filters in order to design an architecture that will allow to determine a much wider representation of the input received without significantly increasing the number of learnable parameters;
- new input data formats - the authors plan to allow the agent to have direct insight not only into the hand movement itself but also its dynamics.

In addition, efforts will be made to expand the established database to include more subjects. On the one hand, this will not only allow the proposed system to be tested more precisely, but will also make it possible to verify the hypothesis claiming that an appropriate increase in the learning data set should unambiguously and significantly improve its performance. As a counter to this hypothesis, one can cite the work of Bolon-Canedo and Remeseiro (2020), whose authors, referring to the Kolmogorov complexity, show that this is not necessarily obvious in their case. This thesis is worth testing because it may affect the need to build more complex networks wanting to acquire interesting features or even the opposite (Kabir and Garg, 2023).

5. Conclusions

The authors proposed a novel solution involving Deep reinforcement learning for signed gestures extraction by temporal segmentation of image sequence to enable transition from a continuous gesture recognition to isolated gesture recognition. The first aspect of the development is the introduction of a GEST database of a new nature enabling not only the gesture recognition task, but also a new segmentation task. In addition, the authors also proposed a novel method for preprocessing the RGB image stream (GSADGM) allowing for the representation of the motion of the recorded subject and a significant reduction in unwanted noise and the size of the processed data. Furthermore, the authors proposed several protocols for evaluating the quality of the segmentation algorithm's performance and two methods for correcting

(filtering and a voting system) the algorithm's response, allowing to considerably improve the results achieved by the proposed algorithm (GSADGM-DRLFS). A future work plan includes further investigation of potential applications of the proposed GSADGM-DRLFS in different fields of study. In addition, it is planned to subsequently develop not only the database but also the presented solution itself to enable the construction of an application that translates sign language gestures in real time.

CRediT authorship contribution statement

Dawid Kalandyk: Conceptualization, Methodology, Data curation, Software, Validation, Writing – original draft, Visualization, Investigation. **Tomasz Kapuściński:** Conceptualization, Resources, Supervision, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Funding

This project is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900PLN.

References

Ahmed, S., Khan, F., Ghaffar, A., Hussain, F., Cho, S.H., 2019. Finger-counting-based gesture recognition within cars using impulse radar with convolutional neural network. Sensors 19 (6), 1429.

Aloysius, N., Geetha, M., 2020. Understanding vision-based continuous sign language recognition. Multimedia Tools Appl. 79 (31), 22177–22209.

Anwar, S., Sinha, S.K., Vivek, S., Ashank, V., 2019. Hand gesture recognition: A survey. In: Nanoelectronics, Circuits and Communication Systems. Springer, pp. 365–371.

Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A., 2017. Deep reinforcement learning: A brief survey. IEEE Signal Process. Mag. 34 (6), 26–38.

Bao, W., Lai, W.-S., Ma, C., Zhang, X., Gao, Z., Yang, M.-H., 2019. Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3703–3712.

Bi, Z., Wu, Q., Shan, M., Zhong, W., 2022. Segmentation-based decision networks for steel surface defect detection. J. Internet Technol. 23, 1405–1416.

Bolon-Canedo, V., Remeseiro, B., 2020. Feature selection in image analysis: A survey. Artif. Intell. Rev., Springer 53 (4), 2905–2931.

Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., 2019. & Others sign language recognition, generation, and translation: An interdisciplinary perspective. In: Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility. pp. 16–31.

Buehler, P., Zisserman, A., Everingham, M., 2009. Learning sign language by watching TV (using weakly aligned subtitles). In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2961–2968.

Cooper, H., Bowden, R., 2010. Sign language recognition using linguistically derived sub-units. In: Proceedings of 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies. pp. 57–61.

Crasborn, O., Sloetjes, H., 2008. Enhanced ELAN functionality for sign language corpora. In: 6th International Conference on Language Resources and Evaluation (LREC 2008)/3rd Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora. pp. 39–43.

Cruz, F., Parisi, G.I., Twiefel, J., Wermter, S., 2016. Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS), IEEE, pp. 759–766.

Cruz, F., Parisi, G.I., Wermter, S., 2018. Multi-modal feedback for affordance-driven interactive reinforcement learning. In: 2018 International Joint Conference on Neural Networks. (IJCNN), IEEE, pp. 1–8.

Cui, R., Liu, H., Zhang, C., 2017. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7361–7369.

Darrell, T., 1997. Reinforcement learning of active recognition behaviors. In: Portions of this paper previously appeared in Advances in Neural Information Processing Systems (NIPS 1995), 8(1997). pp. 73–80.

Darrell, T., Pentland, A., 0000. Active gesture recognition using learned visual attention, Advances in Neural Information Processing Systems 8.

Darrell, T., Pentland, A., 1996. Active gesture recognition using partially observable Markov decision processes. In: Proceedings of 13th International Conference on Pattern Recognition, Vol. 3. IEEE, pp. 984–988.

Duong, C.N., Luu, K., Quach, K.G., Nguyen, N., Patterson, E., Bui, T.D., Le, N., 2019. Automatic face aging in videos via deep reinforcement learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10013–10022.

Durrani, M., Yasmin, S., Rho, S., 2022. An internet of medical things based liver tumor detection system using semantic segmentation. J. Internet Technol. 23, 363–375.

Elmezain, M., Al-Hamadi, A., Michaelis, B., 2009. Hand trajectory-based gesture spotting and recognition using HMM. In: 2009 16th IEEE International Conference on Image Processing. (ICIP), pp. 3577–3580. http://dx.doi.org/10.1109/ICIP.2009.5414322.

Guo, D., Zhou, W., Li, A., Li, H., Wang, M., 2019. Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation. IEEE Trans. Image Process. 29, 1575–1590.

Hasan, H.S., Kareem, S.A., 2012. Human computer interaction for vision based hand gesture recognition: A survey. In: 2012 International Conference on Advanced Computer Science Applications and Technologies. (ACSAT), IEEE, pp. 55–60.

Hu, H., Zhou, W., Pu, J., Li, H., 2021. Global-local enhancement network for NMF-aware sign language recognition. ACM Trans. Multimed. Comput., Commun., Appl. (TOMM) 17, 1–19.

Hung, W.-C., Zhang, J., Shen, X., Lin, Z., Lee, J.-Y., Yang, M.-H., 2018. Learning to blend photos. In: Proceedings of the European Conference on Computer Vision. (ECCV), pp. 70–86.

Jain, R., Karsh, R.K., Barbhuiya, A.A., 2022. Literature review of vision-based dynamic gesture recognition using deep learning techniques. Concurr. Comput.: Pract. Exper. 34 (22), e7159.

Jevtić, A., Colomé, A., Alenya, G., Torras, C., 2018. Robot motion adaptation through user intervention and reinforcement learning. Pattern Recognit. Lett. 105, 67–75.

Kabir, H., Garg, N., 2023. Machine learning enabled orthogonal camera goniometry for accurate and robust contact angle measurements. Sci. Rep., Nature Publishing Group UK London 13 (1), 1497.

Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: A survey. J. Artif. Intell. Res. 4, 237–285.

Kalandyk, D., 2021. Reinforcement learning in car control: A brief survey. In: 2021 Selected Issues of Electrical Engineering and Electronics. (WZEE), pp. 1–8.

Kang, H., Lee, C.W., Jung, K., 2004. Recognition-based gesture spotting in video games. Pattern Recognit. Lett. 25 (15), 1701–1714.

Kapuscinski, T., Wysocki, M., 2020. Recognition of signed expressions in an experimental system supporting deaf clients in the city office. Sensors 20 (8), 2190.

Kim, S.K., Kirchner, E.A., Stefes, A., Kirchner, F., 2017. Intrinsic interactive reinforcement learning–using error-related potentials for real world human–robot interaction. Sci. Rep. 7 (1), 1–16.

Koller, O., Camgoz, N., Ney, H., Bowden, R., 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. IEEE Trans. Pattern Anal. Mach. Intell. 42, 2306–2320.

Koller, O., Zargaran, S., Ney, H., Bowden, R., 2018. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. Int. J. Comput. Vis. 126, 1311–1325.

Lample, G., Chaplot, D.S., 2017. Playing FPS games with deep reinforcement learning. In: Thirty-First AAAI Conference on Artificial Intelligence.

Lan, S., Panda, R., Zhu, Q., Roy-Chowdhury, A.K., 2018. Ffnet: Video fast-forwarding via reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6771–6780.

Li, Y., Xu, L., Shu, W., Mei, K., et al., 2020. AutoGesNet: Auto gesture recognition network based on neural architecture search. In: 2020 12th International Conference on Advanced Computational Intelligence. (ICACI), IEEE, pp. 257–262.

Min, Y., Hao, A., Chai, X., Chen, X., 2021. Visual alignment constraint for continuous sign language recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11542–11551.

Minut, S., Mahadevan, S., 2001. A reinforcement learning model of selective visual attention. In: Proceedings of the Fifth International Conference on Autonomous Agents. pp. 457–464.

Mitra, S., Acharya, T., 2007. Gesture recognition: A survey. IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) 37 (3), 311–324.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 0000. Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Morguet, P., Lang, M., 1998. Spotting dynamic hand gestures in video image sequences using hidden Markov models. In: Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269). IEEE, pp. 193–197.

Neiva, D.H., Zanchettin, C., 2018. Gesture recognition: A review focusing on sign language in a mobile context. Expert Syst. Appl. 103, 159–183.

Neto, P., Pereira, D., Pires, J.N., Moreira, A.P., 2013. Real-time and continuous hand gesture spotting: An approach based on artificial neural networks. In: 2013 IEEE International Conference on Robotics and Automation. IEEE, pp. 178–183.

Nicholl, P., Ahmad, A., Amira, A., 2010. Optimal discrete wavelet transform (DWT) features for face recognition. In: 2010 IEEE Asia Pacific Conference on Circuits and Systems. IEEE, pp. 132–135.

Nishimura, T., Oka, R., 1996. Spotting recognition of human gestures from time-varying images. In: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition. IEEE, pp. 318–322.

Ouyang, X., Kawaai, S., Goh, E.G.H., Shen, S., Ding, W., Ming, H., Huang, D.-Y., 2017. Audio-visual emotion recognition using deep transfer learning and multiple temporal models. In: Proceedings of the 19th ACM International Conference on Multimodal Interaction. pp. 577–582.

Park, H., Ghimire, R., Poudel, S., Lee, S., 2022. Deep learning for joint classification and segmentation of histopathology image. J. Internet Technol. 23, 903–910.

Pezzuoli, F., Corona, D., Corradini, M., Cristofaro, A., 2019. Development of a wearable device for sign language translation. In: Human Friendly Robotics: 10th International Workshop. pp. 115–126.

Pfister, T., Charles, J., Zisserman, A., 2013. Large-scale learning of sign language by watching TV (using co-occurrences). In: BMVC.

Pisharady, P.K., Saerbeck, M., 2015. Recent methods and databases in vision-based hand gesture recognition: A review. Comput. Vis. Image Underst. 141, 152–165.

Pu, J., Zhou, W., Li, H., 2018. Dilated convolutional network with iterative optimization for continuous sign language recognition. In: IJCAI, Vol. 3. p. 7.

Qiu, Z., Yao, T., Ngo, C., Tian, X., Mei, T., 2019. Learning spatio-temporal representation with local and global diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12056–12065.

Rao, Y., Lu, J., Zhou, J., 2017. Attention-aware deep reinforcement learning for video face recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3931–3940.

Ruffieux, S., Lalanne, D., Mugellini, E., Abou Khaled, O., 2014. A survey of datasets for human gesture recognition. In: International Conference on Human–Computer Interaction. Springer, pp. 337–348.

Sagayam, K.M., Hemanth, D.J., 2017. Hand posture and gesture recognition techniques for virtual reality applications: A survey. Virtual Real. 21 (2), 91–107.

Sarkar, A.R., Sanyal, G., Majumder, S., 2024. Hand gesture recognition systems: A survey. Int. J. Comput. Appl. 71 (15).

Sarma, D., Bhuyan, M.K., 2021. Methods, databases and recent advancement of vision-based hand gesture recognition for hci systems: A review. SN Comput. Sci. 2 (6), 1–40.

Seok, W., Kim, Y., Park, C., 2018. Pattern recognition of human arm movement using deep reinforcement learning. In: 2018 International Conference on Information Networking. (ICOIN), IEEE, pp. 917–919.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition.

Singh, S., Gupta, A.K., Singh, T., 2019. Computer vision based hand gesture recognition: A survey. Int. J. Comput. Sci. Eng 7 (5), 548–556.

Supancic, III, J., Ramanan, D., 2017. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 322–331.

Thanathamathee, P., Sawangarreerak, S., Kongkla, P., Nizam, D., 2023. An optimized machine learning and deep learning framework for facial and masked facial recognition. Emerg. Sci. J. 7, 1173–1187.

Trigueiros, P., Ribeiro, F., Reis, L.P., 2012. A comparison of machine learning algorithms applied to hand gesture recognition. In: 7th Iberian Conference on Information Systems and Technologies. (CISTI 2012), IEEE, pp. 1–6.

Tsironi, E., Barros, P., Weber, C., Wermter, S., 2017. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. Neurocomputing 268, 76–86.

Van Hasselt, H., Guez, A., Silver, D., 2016. Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30.

Wilson, A.D., Bobick, A.F., 2000. Realtime online adaptive gesture recognition. In: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, Vol. 1. IEEE, pp. 270–275.

Wirawan, I., Wardoyo, R., Lelono, D., Kusrohmaniah, S., 2022. Continuous capsule network method for improving electroencephalogram-based emotion recognition. Emerg. Sci. J. 7, 116–134.

Ye, E., Ye, E., Bouthillier, M., Ye, R., 2021. Deepimagetranslator v2: analysis of multimodal medical images using semantic segmentation maps generated through deep learning. BioRxiv. 2021–10.

Yun, S., Choi, J., Yoo, Y., Yun, K., Young Choi, J., 2017. Action-decision networks for visual tracking with deep reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2711–2720.

Zhang, Z., Pu, J., Zhuang, L., Zhou, W., Li, H., 2019. Continuous sign language recognition via reinforcement learning. In: 2019 IEEE International Conference on Image Processing. (ICIP), IEEE, pp. 285–289.

Zhang, J., Zhou, W., Xie, C., Pu, J., Li, H., 2016. Chinese sign language recognition with adaptive HMM. In: 2016 IEEE International Conference on Multimedia and Expo. (ICME), pp. 1–6.

Zhu, Y., Xu, G., Kriegman, D.J., 2002. A real-time approach to the spotting, representation, and recognition of hand gestures for human–computer interaction. Comput. Vis. Image Underst. 85 (3), 189–208.

10.24425/acs.2024.xxxxxx

Archives of Control Sciences
Volume xx(xxx), 2024
No. x, *pages xxx–xxx*

Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System

Dawid Kalandyk[1]*, Bogdan Kwiatkowski[2], Damian Mazur[2]

The widespread desire to automate the CNC machine control process and optimize it is leading to the development of new algorithms. The article presents both a novel approach to this task based on a fuzzy decision-making system as well as an evaluation of the proposed solution on a large database containing data from multiple machining processes and a comparison with the Reference Points Realization Optimization (RPRO) algorithm used in industry. In addition to achieving the intended accuracy of the machining process, the presented system is also easily interpretable for the expert operating the machine. It is also possible to manipulate the presented system easily and shape it according to specific needs.

Key words: CNC machine; G-Code calculation; Fuzzy Logic

[1] Doctoral School of the Rzeszow University of Technology, Rzeszow 35-959, Poland; d.kalandyk@prz.edu.pl
[2] Department of Electrical Engineering and Fundamentals of Computer Science, Rzeszow University of Technology, Rzeszow 35-959, Poland; b.kwiatkowsk@prz.edu.pl, mazur@prz.edu.pl,
* Correspondence: d.kalandyk@prz.edu.pl

Received xx.xx.2023.

1. Introduction

Automation of technological processes used in many branches of the global economy requires solutions that will optimize aspects of product quality and energy intensity of production lines and manufacturing time [1-2]. A special and widely used solution is the use of CNC machines in the manufacturing process. Optimizing the operation of CNC machines requires the prior preparation of a data set that unambiguously describes all aspects related to the dynamics of the machine, requirements arising from the technological processes for the workpiece to be fabricated, and economic aspects. These factors arise from the guidelines set for modern technical systems that are part of Industry 4.0. This requires the acquisition of a large number of geometric data describing the parameters of the manufactured parts and electrical data describing the energy state of the machine. Optimizing the operation of such the machine requires the cooperation of scientists from different fields [3]. The industrial experience of the authors, related to the application of modern computer methods and the operation of CNC machine tools, indicates that the development and implementation of new algorithms and the optimum selection of parameters describing the dynamics of the CNC machine, allowing the exact reflection of geometric points of the performed detail are still needed [4] [5].

One of the approaches available and operational in the industry for automatic G-CODE calculation is the Reference Points Realization Optimization (RPRO) algorithm proposed in [6]. This algorithm has as its main goal the fastest possible execution of the machining process, using the full capabilities of the dynamics of the CNC machine, while maintaining the intended average accuracy of the machining process as much as possible. However, this and similar solutions have some drawbacks [7-8]. The first is high computational complexity: the need to check many combinations of machine dynamics parameters in order to find the one that achieves the targeted accuracy. The second one results from the way the machine is controlled based on the generated g-code, specifically the use of full spindle acceleration and deceleration (rapid movements). Such action leads to vibrations, which can result in loss of machining accuracy, and can also lead to faster wear of machine components. Machining in this way can, instead of the intended decrease in the price of the part's products, have the opposite effect of increasing the price [9-10]. A final aspect worth mentioning is the fact that the operation of the available tools is not interpretable to the expert controlling the CNC machine. To answer the challenge posed, the authors, using the aforementioned RPRO algorithm, established a database of many different processing operations and proposed their own g-code determination system based on fuzzy logic. Fuzzy logic has been used before to solve other cnc machine tasks [11-13].

The remaining part of the article is organized as follows. Section 2 describes the proposed alternative motion planning system for a CNC machine. Section 3 describes the proposed versions of the system and the tests that were conducted, and considers the results. Finally, Section 4 summarizes the achievements and provides a plan for the future work.

2. Proposed method

The authors aimed to create a system that allows the determination of the G-CODE for given machine dynamics parameters and a specified sequence of reference points. This system should work in the offline mode and be easily understood and intuitive for the expert optimizing the machining process. Therefore, the proposed solution uses a fuzzy logic expert system based on the Mamdani first-type model. The authors considered the process of generating the G-CODE as a series of decisions concerning movement of the spindle in successive time steps. Between consecutive steps of performance quality evaluation, the expert can freely modify the set of rules representing the system's operating strategy. The proposed model requires the following information:

- parameters of machine dynamics,
- the actual spindle dynamics and its position referred to as the spindle state,
- the target reference point sequence.

Based on the above information, a simulation of spindle movement is executed, where in subsequent steps one of three discrete values is selected: Decelerate, DoNothing and Accelerate, which constitute the considered set of decisions. The simulation terminates when the spindle reaches the last of the reference points. The investigation allows to take a decision using Fuzzy Inference System (FIS) to focus on the movement of the spindle along a single axis without the possibility of reversing. A diagram demonstrating the proposed framework is shown in Figure 1.

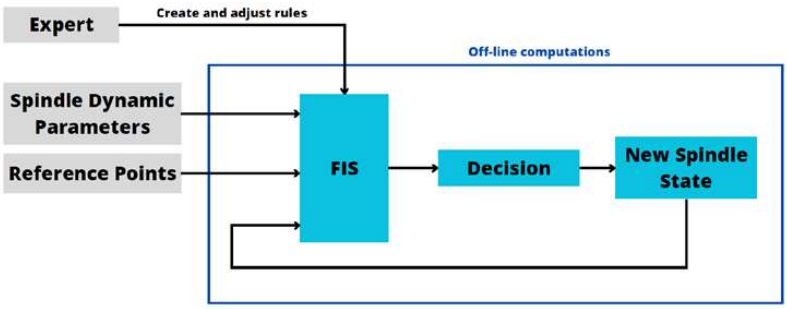


Figure 1. Proposed framework scheme

The authors proposed two different models shown in Figure 2. The first model has two inputs: normalized spindle speed and normalized spindle distance to the next reference point. Each of the inputs of this model received values in the range [-1; 1] and were covered with evenly distributed five features corresponding to the labels: "Very Small", "Small", "Medium", "Large", and "Very Large". These signals are analyzed according to 25 rules determined by the expert, each of which refers to every possible combination of input signals. The output of the system is a number in the range [-1; 1] covered by 3 functions corresponding to possible decisions. The second model is an attempt to extend the perceptual abilities of the first model by supplementing its knowledge with a normalized spindle speed. This input is a number in the range [-1.5; 2.5]. It is covered by 8 functions corresponding in consecutive order to the labels: "Large backward", "Medium backward", "Small backward", "Very small", "Small", "Medium", "Large", "Very large". Consequently, the number of rules proposed for this model is 200.

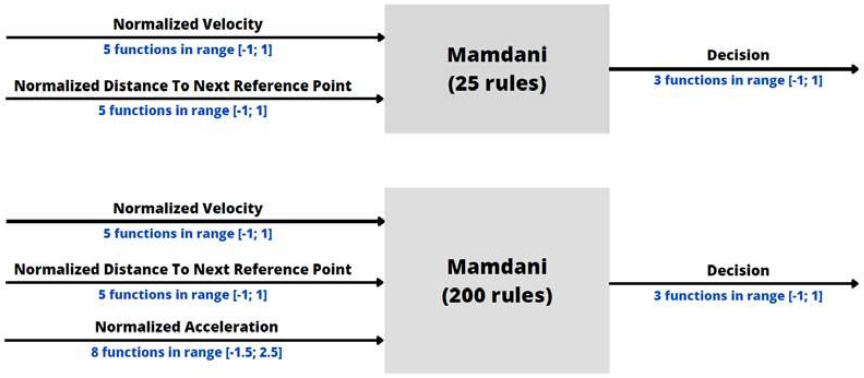


Figure 2. Proposed Fuzzy Expert Systems Models Diagram

Two shapes of the coverage function, namely the triangular function and the bell curve, were also considered in the conducted research. The proposed rule sets in the form of graphs are shown in Figure 3. and Figure 4. Individual symbols denote specific decisions of the system:

- *Decelerate* - red triangle,
- *DoNothing* - blue dot
- *Accelerate* - green triangle.

Each point on the graph denotes one rule determined for a specific combination of input signals and indicating the corresponding decision of the fuzzy system. Subsequent numbers on the axes denote individual discrete values of the input signals in the same order in which they were given earlier in this paper.

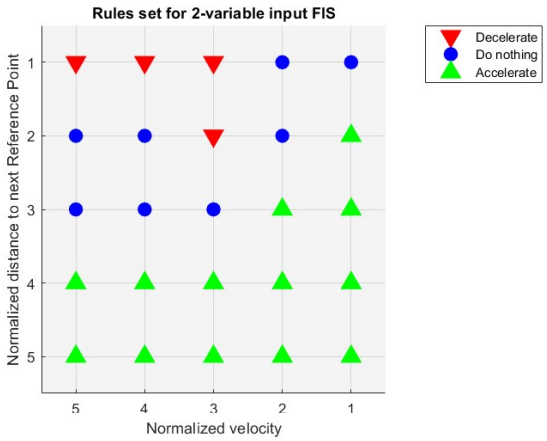


Figure 3. Rule set for 2-input model

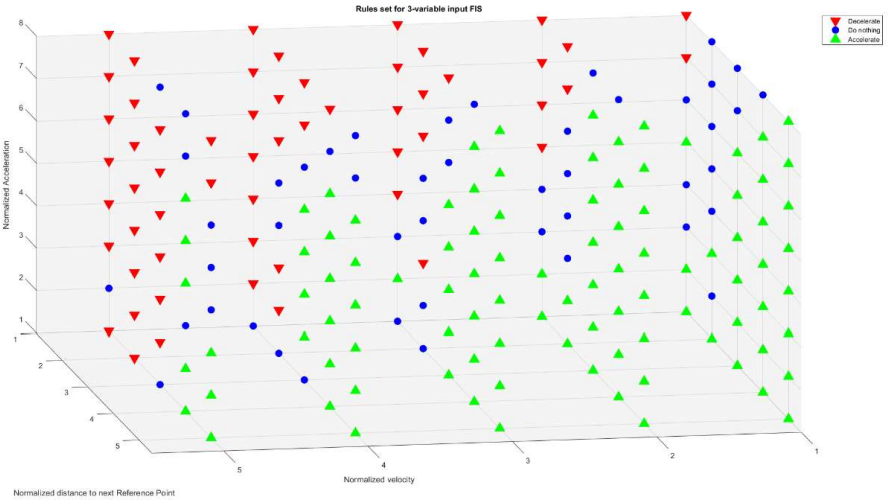


Figure 4. Rule Set for 3-input model

As one can easily notice, the number of proposed rules is very large, especially for a model with three inputs. Therefore, the authors decided to reduce the number of rules using the mentioned graphs as an assist in finding dependencies. For example, eight rules containing three predecessors of the implication, represented by the column in the closer right corner of the graph, can be replaced by a rule with two predecessors "If Normalized velocity is Very Small and Normalized distance to next Reference Point is Very Big then Decission is Accelerate". Making equivalent transformations, one alternative set of rules was proposed for each model. The number of rules with individual successors and the total number of rules for each version of the system are shown in Table 1.

Table 1. Distribution of proposed fuzzy systems rules

Rules Type	System type	Decelerate	Do nothing	Accelerate	Total rules number
full cover	**2-variable**	4	8	13	25
	3-variable	56	47	97	200
reduced	**2-variable**	4	8	4	16
	3-variable	21	23	32	76

3. Experiments

To check the quality of the 8 proposed fuzzy expert systems, extensive study was conducted using a set of processing proposed in [14]. The achieved results were compared with results obtained using the RPRO algorithm, with target accuracy set to 10μm. This parameter describes maximum value of mean accuracy among all reference points. The database considered includes machining processes with different issues: reference path lengths (3), densities (3), trajectories (10), maximum spindle speeds (4), maximum spindle accelerations (5) and Jerk values (3). For convenience of analysis, they are assigned to 9 groups (combinations of length and density of reference paths) with 600 processes in each. A detailed description of the parameters can be found in [14].

Table 2. Test results - machining process accuracy as average error [μm]

Accuracy			Testing trajectory group number								
Number of input variables	Functions type	Rules type	1	2	3	4	5	6	7	8	9
2	bell	full cover	15.72 ± 4.85	28.87 ± 10.50	29.72 ± 11.37	28.24 ± 8.32	33.90 ± 1.73	32.93 ± 11.58	32.50 ± 10.45	32.97 ± 10.76	33.17 ± 11.16
		reduced	15.92 ± 5.19	28.90 ± 10.53	29.70 ± 11.21	28.38 ± 8.68	33.80 ± 1.61	32.93 ± 11.58	**32.13 ± 10.03**	32.99 ± 10.78	33.14 ± 11.31
	triangle	full cover	16.30 ± 4.35	29.35 ± 10.34	29.69 ± 11.23	28.90 ± 8.34	34.19 ± 1.96	32.93 ± 11.58	32.53 ± 10.27	33.02 ± 10.84	33.16 ± 11.53
		reduced	16.21 ± 4.46	29.45 ± 10.42	29.65 ± 11.23	29.00 ± 8.53	33.91 ± 1.71	32.93 ± 11.58	32.53 ± 10.45	33.16 ± 11.02	33.16 ± 11.40
3	bell	full cover	15.33 ± 4.81	28.32 ± 10.66	**25.29 ± 11.16**	27.50 ± 8.16	**33.69 ± 12.06**	**28.12 ± 11.99**	32.27 ± 10.17	**32.96 ± 11.02**	**28.68 ± 11.74**
		reduced	**15.31 ± 4.96**	28.74 ± 10.37	30.46 ± 11.82	27.60 ± 8.08	33.99 ± 1.86	33.13 ± 11.46	32.16 ± 10.07	33.22 ± 10.89	33.79 ± 11.74
	triangle	full cover	**15.28 ± 4.34**	**28.54 ± 10.39**	28.32 ± 11.44	**27.39 ± 8.24**	33.97 ± 2.02	33.27 ± 11.87	**32.07 ± 9.94**	33.13 ± 10.97	33.91 ± 12.26
		reduced	15.41 ± 4.74	28.73 ± 10.21	30.40 ± 11.55	**27.29 ± 8.15**	34.21 ± 1.94	33.28 ± 11.51	32.14 ± 10.05	33.10 ± 10.82	33.91 ± 11.84
RFRO			21.36 ± 8.74	**15.39 ± 1.585**	**3.67 ± 1.47**	40.54 ± 17.17	**19.21 ± 16.88**	**3.88 ± 1.27**	49.72 ± 22.17	**16.23 ± 14.32**	**3.95 ± 1.29**

Table 3. Test results - number of processing steps

Time steps count			Testing trajectory group number								
Number of input variables	Functions type	Rules type	1	2	3	4	5	6	7	8	9
2	bell	full cover	201 ± 29	1126 ± 47	11484 ± 627	1:700 ± 0	11700 ± 0	37120 ± 1681	40110 ± 0	40110 ± 0	73753 ± 2559
		reduced	201 ± 29	1126 ± 47	11484 ± 627	1:700 ± 0	11700 ± 0	37120 ± 1681	40110 ± 0	40110 ± 0	73753 ± 2559
	triangle	full cover	201 ± 29	1126 ± 47	11484 ± 627	1:700 ± 0	11700 ± 0	37120 ± 1681	40110 ± 0	40110 ± 0	73753 ± 2559
		reduced	201 ± 29	1126 ± 47	11484 ± 627	1:700 ± 0	11700 ± 0	37120 ± 1681	40110 ± 0	40110 ± 0	73753 ± 2559
3	bell	full cover	201 ± 29	1126 ± 47	11485 ± 625	1:700 ± 0	11700 ± 0	37131 ± 1673	40110 ± 0	40110 ± 0	73770 ± 2534
		reduced	201 ± 29	1126 ± 47	11485 ± 625	1:700 ± 0	11700 ± 0	37131 ± 1673	40110 ± 0	40110 ± 0	73770 ± 2534
	triangle	full cover	201 ± 29	1126 ± 47	11486 ± 622	1:700 ± 0	11700 ± 0	37132 ± 1674	40110 ± 0	40110 ± 0	73770 ± 2524
		reduced	201 ± 29	1126 ± 47	11486 ± 622	1:700 ± 0	11700 ± 0	37132 ± 1674	40110 ± 0	40110 ± 0	73770 ± 2524
RPRO			113 ± 21	578 ± 236	5542 ± 2529	273 ± 80	1934 ± 809	19456 ± 8856	470 ± 169	4052 ± 1717	37977 ± 17048

Table 2. and Table 3. show the test results of the proposed fuzzy expert systems. The accuracy metric is measured as the average error of each of the 600 machining processes of a given group of trajectories and expressed in micrometers. In addition, the standard deviation is also given for each measurement, and the best and second best results are shown in bold. A metric indicating the number of time steps, each of which lasted 2ms, was constructed similarly. In this case, only the best results were marked in bold for clarity. As expected, the RPRO algorithm provided shorter calculation time however, only for 3 of the 9 groups of trajectories it was able to achieve the intended accuracy. The proposed expert systems were successful in achieving satisfactory accuracy between 15μm and 34μm. The results of the number of steps indicate that the small machining error was achieved at the expense of not using the full potential of the machine dynamics. However, it should be noted that this behavior is fully dependent on the set of rules established by the expert that constitute the strategy of the fuzzy system controlling the machining process. Depending on the needs, the expert may propose a set of rules prioritizing the processing time. An interesting effect visible during the analysis of the results is the preservation of the stability of the system operation after minimizing the size of the rule set. This is a very advantageous effect that allows to significantly reduce the calculation time. An additional advantage of the proposed system is the smoothness of acceleration and deceleration of the spindle movement. This phenomenon makes it possible to definitely reduce the level of vibration of the working CNC machine, and consequently extend its lifetime and definitely reduce operating costs.

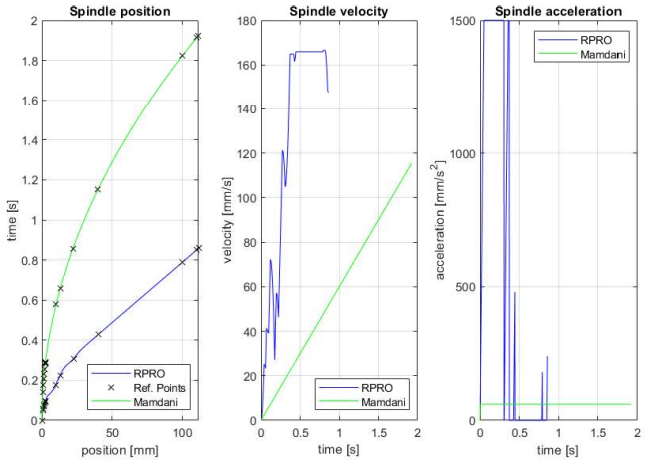


Figure 5. Comparative analysis for mixed density trajectory

To better observe the described behavior of the algorithm and to confirm that it also tackles mixed-density trajectories, a comparative simulation was conducted for trajectory [0.0, 0.6, 0.9, 1.3, 1.6, 1.97, 2.38, 2.45, 2.51, 10.0, 13.0, 22.0, 40.0, 100.0, 110.0, 111.0]. The RPRO algorithm achieved an accuracy of 26.83μm during 431 steps, while the proposed expert system achieved an accuracy of 19.51μm during 964 steps. As expected, the RPRO algorithm determined a significantly shorter G-CODE. As for the vibration level, on the other hand, the proposed solution provides an incomparably lower vibration level during the machining process. This can be observed by comparing the level of fluctuation of the velocity graph and spindle acceleration, which are shown in Figure 5.

4. Conclusions

The paper proposes a novel, off-line, Mamdani I-type algorithm that allows the generation of a G-CODE for given machine dynamics parameters and a specified sequence of reference points. Its capabilities and various versions have been tested during extensive experiments and compared with the results achieved by the RPRO algorithm used in industry. The proposed solution provides the possibility of personalizing the way of operation by appropriate construction of rules for the fuzzy system by an expert. In addition, by constructing the rules in a manner similar to that presented in the paper, it is possible to ensure an effective reduction in the level of machine vibration during the machining process. The authors also proposed a method for optimizing the set of rules while maintaining the stability of the algorithm's operation. Future work will focus on further optimizing the speed of the algorithm by using the Sugeno system, as well as other artificial intelligence methods.

References

[1] J. M. Langeron, E. Duc, C. Lartigue and P. Bourdet, ”A new format for 5-axis tool path computation using Bspline curves”, Comput-Aided Design, 36, pp. 1219–1229, 2004.. DOI: https://doi.org/10.1016/j.cad.2003.12.002

[2] Q. Bi, N. Huang, C. Sun, Y. Wang, L. Zhu and H. Ding, ”Identification and compensation of geometric errors of rotary axes on five-axis machine by on-machine measurement”, Int J Machine Tools Manufacture, 89, pp. 182–191, 2015. DOI: https://doi.org/10.1016/j.ijmachtools.2014.11.008

[3] H. J. Lee, Y. Liu and S. H. Yang, ”Accuracy improvement of miniaturized machine tool: Geometric error modelling and compensation”, Int J Machine Tools Manufacture, 46, pp. 1508–1516, 2006. DOI: https://doi.org/10.1016/j.ijmachtools.2005.09.004

[4] Y. Sun, S. Sun, J. Xu and D. Guo, ”A unified method of generating tool path based on multiple vector fields for CNC machining of compound NURBS surfaces”, Comput-Aided Design, 91, pp. 14–26, 2017. DOI: https://doi.org/10.1016/j.cad.2017.04.003

[5] XF. Li, H. Zhao, X. Zhao and H. Ding, ”Interpolation-based contour error estimation and component-based contouring control for five-axis CNC machine tools”, Sci China Tech Sci, 61, pp. 1666–1678, 2018. DOI: https://doi.org/10.1007/s11431-017-9204-y

[6] Kwiatkowski, B., Kwater, T., Mazur, D., & Bartman, J. (2024). An off-line application that determines the maximum accuracy of the realization of reference points from G-code for given parameters of CNC machine dynamics. Bulletin of the Polish Academy of Sciences. Technical Sciences, 72(1). http://dx.doi.org/10.24425/bpasts.2023.147345

[7] M. Chen and Y. Sun, ”A moving knot sequence-based feedrate scheduling method of parametric interpolator for CNC machining with contour error and drive constraints, Int J Adv Manuf Technol, 98, pp. 487–504, 2018. DOI: https://doi.org/10.1007/s00170-018-2279-0

[8] Barbara Pękala; Ewa Rak; Bogdan Kwiatkowski; Adam Szczur; Rafał Rak, The use of concave and convex functions to optimize the feed-rate of numerically controlled machine tools, 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). DOI: https://doi.org/10.1109/FUZZ48607.2020.9177569

[9] S. Z. Mansour and R. Seethaler, ”Feedrate optimization for computer numerically controlled machine tools using modeled and measured process constraints”, J Manuf Sci Eng, 139, 9 pages, 2017. DOI: https://doi.org/10.1115/1.4033933

[10] M. Rahaman, R. Seethaler and I. Yellowley, ”A new approach to contour error control in high speed machining”, Int J Machine Tools Manufacture, 88, pp. 42–50, 2015. DOI: https://doi.org/10.1016/j.ijmachtools.2014.09.002

[11] Kar, T., Mandal, N.K. & Singh, N.K. Multi-response Optimization and Surface Texture Characterization for CNC Milling of Inconel 718 Alloy. Arab J Sci Eng 45, 1265–1277 (2020). DOI: https://doi.org/10.1007/s13369-019-04324-5

[12] Datta, S., Mahapatra, S. S., Routara, B. C., & Bandyopadhyay, A. (2011). The fuzzy inference system approach to a multi-performance characteristic index for surface quality improvement in CNC end milling. International Journal of Experimental Design and Process Optimisation, 2(3), 265-282. DOI: https://doi.org/10.1504/IJEDPO.2011.042747

[13] Molina, A., Ponce, H., Ponce, P., Tello, G., & Ramírez, M. (2014).

Artificial hydrocarbon networks fuzzy inference systems for CNC machines position controller. International Journal of Advanced Manufacturing Technology, 72. DOI: https://doi.org/10.1007/s00170-014-5676-z

[14] Kalandyk, D., Kwiatkowski, B., Mazur, D. (2023) Application of Mamdani Fuzzy Logic Inference System to optimise CNC machine motion dynamics. 2023 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). DOI: https://doi.org/10.1109/FUZZ52849.2023.10309802

Subject: Paper to Archives of Control Sciences
From: Zbigniew Ogonowski <Zbigniew.Ogonowski@polsl.pl>
Date: 24.05.2024, 12:48
To: "d.kalandyk@prz.edu.pl" <d.kalandyk@prz.edu.pl>

Dear Dawid Kalandyk,

It is my pleasure to inform you that your paper entitled "Calculating G-code for CNC machine using the Mamdani fuzzy logic inference system" is accepted for publication in Archives of Control Sciences – the quarterly of the Polish Academy of Science.

The paper is going to be published in the currently prepared issue i.e. ACS vol 34, no 3, 2024.

Sincerely yours,

Dr Zbigniew Ogonowski
Polish Academy of Sciences
ACS Managing Editor

Streszczenie w języku polskim

Niniejsza rozprawa doktorska stanowi monotematyczny cykl publikacji naukowych dotyczących aplikacji metod sztucznej inteligencji ze szczególnym uwzględnieniem algorytmu uczenia się ze wzmocnieniem. Praca ma charakter interdyscyplinarny i rozważa dwie główne gałęzie tematyczne: zadania przetwarzania obrazu oraz zadania sterowania. W pierwszej z nich poruszony został temat wykrywania gestów poprzez czasową segmentację strumienia wideo. W drugiej gałęzi skupiono się na sterowaniu dynamiką wrzeciona maszyny sterowanej numerycznie (ang. Computer Numeric Control – CNC). Duża część badań została zaprezentowana w trakcie międzynarodowych konferencji naukowych. Uwzględniając powyższe w pracy sformułowano następującą hipotezę badawczą.

> *Możliwa jest aplikacja różnych metod sztucznej inteligencji, a w szczególności algorytmu uczenia się ze wzmocnieniem, zarówno do zadań przetwarzania obrazu jak i do zadań sterowania, celem uzyskania rezultatów nie gorszych niż przy pomocy innych metod znanych z literatury.*

Hipoteza została uprawdopodobniona poprzez realizację następujących zadań:

1) Studia literaturowe dotyczące wykorzystania algorytmu uczenia się ze wzmocnieniem do:
    a. rozwiązywania zadań przetwarzania obrazów,
    b. rozwiązywania zadań sterowania.

2) Zebranie niezbędnych danych oraz utworzenie zbioru pozwalającego na trenowanie oraz weryfikowanie poprawności działania badanych metod:
    a. do zadania wykrywania gestów,
    b. do zadania sterowania dynamiką ruchu wrzeciona maszyny CNC.

3) Zaproponowanie metody pozwalającej na:
    a. czasową segmentację ciągłego strumienia gestów,
    b. optymalizację sterowania dynamiką ruchu wrzeciona maszyny CNC z wykorzystaniem logiki rozmytej,
    c. optymalizację sterowania dynamiką ruchu wrzeciona maszyny CNC z wykorzystaniem paradygmatu uczenia się ze wzmocnieniem

Zadanie 1. Studia literaturowe dotyczące wykorzystania algorytmu uczenia się ze wzmocnieniem

a) Zadanie zostało zrealizowanie poprzez przygotowanie przeglądu literatury, który został zaprezentowany podczas konferencji krajowej oraz przedstawiony w formie publikacji. Zgromadzona wiedza stanowiła inspirację podczas dalszych prac.

b) Zadanie zostało zrealizowanie poprzez przygotowanie przeglądu literatury, który został zaprezentowany podczas konferencji międzynarodowej oraz przedstawiony w formie publikacji. Zgromadzona wiedza stanowiła inspirację podczas dalszych prac.

Zadanie 2. Zebranie niezbędnych danych oraz utworzenie zbioru pozwalającego na trenowanie oraz weryfikowanie poprawności działania badanych metod

a) Zadanie zostało wykonane poprzez utworzenie dedykowanej bazy danych wykorzystanej podczas badań nad propozycją algorytmu, którego zadaniem było wykrywanie gestów w ciągłym strumieniu wideo poprzez jego czasową segmentację.

b) Zadanie zostało wykonane poprzez utworzenie dedykowanej bazy danych wykorzystanej podczas badań nad propozycją algorytmu, którego zadaniem była optymalizacja pracy maszyny CNC poprzez odpowiednie sterowanie dynamiką ruchu wrzeciona.

Zadanie 3. Zaproponowanie autorskiej metody

a) Zadanie zostało wykonane poprzez publikację artykułu zawierającego opis proponowanej metody czasowej segmentacji strumienia gestów. W pracy zaproponowano również autorski sposób przetwarzania wstępnego klipów wideo w celu minimalizacji niekorzystnego wpływu szeregu czynników. W pracy wykorzystano zarówno głębokie sieci neuronowe jak również paradygmat uczenia się ze wzmocnieniem.

b) Zadanie zostało wykonane poprzez publikację dwóch artykułów naukowych, w których prezentowane są proponowane rozwiązania oparte na systemach eksperckich logiki rozmytej. W trakcie badań do nauki zbioru reguł wykorzystano też algorytm optymalizacji rojem cząstek oraz

algorytm genetyczny. Wyniki badań zostały przedstawione w ramach konferencji.

c) Zadanie zostało wykonane poprzez publikację artykułu naukowego, w którym prezentowane jest proponowane rozwiązanie oparte na sieci neuronowej oraz na paradygmacie uczenia się ze wzmocnieniem. Wyniki badań zostały przedstawione w ramach konferencji.

Słowa kluczowe: sztuczna inteligencja, uczenie się ze wzmocnieniem, przetwarzanie obrazu, zadania sterowania, maszyny CNC, optymalizacja

Streszczenie w języku angielskim

This dissertation is a monothematic series of scientific publications on the application of artificial intelligence methods with a particular focus on the reinforcement learning algorithm. The work is interdisciplinary in nature and considers two main main branches: image processing tasks and control tasks. In the first, the topic of gesture detection through temporal segmentation of the video stream is addressed. The second branch focuses on controlling the spindle dynamics of a Computer Numeric Control (CNC) machine. Much of the research has been presented at international scientific conferences. Taking the above into account, the paper formulates the following research hypothesis.

> *It is possible to apply various artificial intelligence methods, especially the reinforcement learning algorithm, to both image processing and control tasks, with the goal of obtaining results no worse than with other methods known from the literature.*

The hypothesis was made probable by performing the following tasks:

1) Literature studies on the use of the reinforcement learning algorithm to:
    a) solving image processing tasks,
    b) solving control tasks.
2) Collecting the necessary data and creating a set that allows training and verifying the correctness of the tested methods:
    a) for the task of gesture detection,
    b) for the task of controlling the dynamics of movement of the CNC machine spindle.
3) To propose a method that allows:
    a) temporal segmentation of a continuous stream of gestures,
    b) optimization of the control of the dynamics of the CNC machine spindle movement based on fuzzy logic,
    c) optimize the control of the spindle motion dynamics of the CNC machine based on the reinforcement learning paradigm

Task 1. Literature studies on the use of the reinforcement learning algorithm

a) The task was accomplished by preparing a literature review, which was presented at a national conference and collected in the form of a publication. The accumulated knowledge provided inspiration during further work.

b) The task was accomplished by preparing a literature review, which was presented at an international conference and collected in the form of a publication. The accumulated knowledge provided inspiration during further work.

Task 2. Collect the necessary data and create a collection that allows to train and verify the correctness of the methods studied

a) The task was carried out by creating a specialized database used during the research on the proposal of an algorithm whose task was to detect gestures in a continuous video stream through its temporal segmentation.

b) The task was performed by creating a specialized database used during the research on the proposal of an algorithm whose task was to optimize the operation of a CNC machine through appropriate control of the dynamics of spindle movement.

Task 3. Propose the author's method

a) The task was accomplished by publishing a paper describing a proposed method for temporal segmentation of a gesture stream. The paper also proposed an author's method of preprocessing video clips to minimize the adverse effects of a number of factors. The paper uses both deep neural networks and a reinforcement learning paradigm.

b) The task was accomplished through the publication of two scientific articles that present proposed solutions based on fuzzy logic expert systems. In the course of the research, a particle swarm optimization algorithm and a genetic algorithm were also used to learn a set of rules. The results of the research were presented at the conference.

c) The task was accomplished through the publication of a scientific article, in which the proposed solution based on neural network and reinforcement learning paradigm is presented. The results of the research were presented in the framework of the conference.

Keywords: artificial intelligence, reinforcement learning, image processing, control tasks, CNC machines, optimization

Oświadczenia współautorów

Niniejszy rozdział zawiera oświadczenia dotyczące indywidualnego wkładu merytorycznego autora rozprawy oraz współautorów w przygotowanie, przeprowadzenie i opracowanie wyników badań oraz przedstawienie prac w formie publikacji, a także informacje o procentowym składzie autorskim. Oświadczenia dotyczą kolejno następujących artykułów:

[A-3] Kalandyk, D., Kwiatkowski, B., & Mazur, D. (2023, August). Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics. In 2023 IEEE International Conference on Fuzzy Systems (FUZZ) (pp. 1-4). IEEE. https://doi.org/10.1109/FUZZ52849.2023.10309802; wkład 70%; liczba punktów wcześniej: 140; liczba punktów teraz: 70; (autor korespondencyjny)

[A-4] Kalandyk, D., Kwiatkowski, B., & Mazur, D. CNC Machine Control Using Deep Reinforcement Learning. Bulletin of the Polish Academy of Sciences Technical Sciences, e148940-e148940. https://doi.org/10.24425/bpasts.2024.148940; wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.8

[A-5] Kalandyk, D., & Kapuściński, T. (2024). Temporal signed gestures segmentation in an image sequence using deep reinforcement learning. Engineering Applications of Artificial Intelligence, 131, 107879. https://doi.org/10.1016/j.engappai.2024.107879; wkład 90%; liczba punktów: 140; IF: 8.0; CS: 12.3, (autor korespondencyjny)

[A-6] Kalandyk, D., Kwiatkowski, B., & Mazur, D. Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System. Archives of Control Sciences, artykuł przyjęty do publikacji, wkład 33.3%; liczba punktów: 100; IF: 1.2; CS: 2.7; (autor korespondencyjny)

Oświadczenie współautorów publikacji

Niniejszym potwierdzam wkład autorski w publikację pt.: „Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics". Wkład Dawida Kalandyka w czasie powstania publikacji obejmuje:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Imię i nazwisko współautora	**Procentowy wkład autorski**	**Data i podpis współautora**
Dawid Kalandyk	70%	17.06.2024 Dawid Kalandyk
Bogdan Kwiatkowski	15%	17.06.2024 Bogdan Kwiatkowski
Damian Mazur	15%	17.06.2024 Damian Mazur

Boguchwała 11.04.2025

miejscowość, data

dr inż. Bogdan Kwiatkowski

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 15%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Bogdan Kwiatkowski

podpis

Rzeszów 17.06.24

miejscowość, data

dr hab. inż. Damian Mazur

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „Application of Mamdani Fuzzy Logic Inference System to Optimise CNC Machine Motion Dynamics" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 15%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Damian Mazur

podpis

Oświadczenie współautorów publikacji

Niniejszym potwierdzam wkład autorski w publikację pt.: „CNC Machine Control Using Deep Reinforcement Learning". Wkład Dawida Kalandyka w czasie powstania publikacji obejmuje:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Imię i nazwisko współautora	**Procentowy wkład autorski**	**Data i podpis współautora**
Dawid Kalandyk	33.33%	19.06.2024 Dawid Kalandyk
Bogdan Kwiatkowski	33.33%	17-06-2024 Bogdan Kwiatkowski
Damian Mazur	33.33%	17.06.24 Damian Mazur

Rzeszów, dn. 17.06.2021

miejscowość, data

dr inż. Bogdan Kwiatkowski

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „CNC Machine Control Using Deep Reinforcement Learning" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33.33%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Bogdan Kwiatkowski

podpis

Rzeszów 14.06.24

miejscowość, data

dr hab. inż. Damian Mazur

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „CNC Machine Control Using Deep Reinforcement Learning" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33.33%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Dawida Mazur

podpis

Oświadczenie współautorów publikacji

Niniejszym potwierdzam wkład autorski w publikację pt.: „Temporal signed gestures segmentation in an image sequence using deep reinforcement learning". Wkład Dawida Kalandyka w czasie powstania publikacji obejmuje:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Imię i nazwisko współautora	**Procentowy wkład autorski**	**Data i podpis współautora**
Dawid Kalandyk	90%	17.06.2024 Dawid Kalandyk
Tomasz Kapuściński	10%	18.06.2024r. Tomasz Kapuściński

Rzeszów, 18.06.2024 r.

miejscowość, data

dr hab. inż. Tomasz Kapuściński

Katedra Informatyki i Automatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „Temporal signed gestures segmentation in an image sequence using deep reinforcement learning" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie i analizę wyników,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 10%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Tomasz Kapuściński

podpis

Oświadczenie współautorów publikacji

Niniejszym potwierdzam wkład autorski w publikację pt.: „Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System". Wkład Dawida Kalandyka w czasie powstania publikacji obejmuje:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Imię i nazwisko współautora	**Procentowy wkład autorski**	**Data i podpis współautora**
Dawid Kalandyk	33.33%	19.06.2024 Dawid Kalandyk
Bogdan Kwiatkowski	33.33%	11-06-2024 Bogdan Kwiatkowski
Damian Mazur	33.33%	17.06.24 Damian Mazur

Rzeszów dn. 12.06.2023

miejscowość, data

dr inż. Bogdan Kwiatkowski

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33.33%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Bogdan Kwiatkowski

podpis

Rzeszów 17.06.24

miejscowość, data

dr hab. inż. Damian Mazur

Katedra Elektrotechniki i Podstaw Informatyki

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Oświadczenie współautora publikacji

Jako współautor pracy pt. „Calculating G-Code for CNC machine using the Mamdani Fuzzy Logic Inference System" oświadczam, iż mój własny wkład merytoryczny w przygotowanie przeprowadzenie i opracowanie badań oraz przedstawienie pracy w formie publikacji stanowi:

- wspólne opracowanie metodologii zaproponowanej w publikacji,
- wspólne opracowanie kodów oraz przeprowadzenie eksperyemntów,
- wspólne opracowanie i analizę wyników,
- wspólne przygotowanie rysunków oraz tabel,
- współredakcję pracy.

Mój udział procentowy w przygotowaniu publikacji określam jako 33.33%. Jednocześnie wyrażam zgodę na wykorzystanie w/w pracy jako części rozprawy doktorskiej mgr inż. Dawida Kalandyka.

Damian Mazur

podpis