Wojskowa Akademia Techniczna

im. Jarosława Dąbrowskiego

ROZPRAWA DOKTORSKA



ZASTOSOWANIE KONWOLUCYJNYCH SIECI NEURONOWYCH DO ROZPOZNAWANIA OBIEKTÓW W RADIOLOKACJI

kpt. mgr inż. Błażej Ślesicki

Promotor: prof. dr hab. inż. Adam Kawalec

Promotor pomocniczy: kpt. dr inż. Anna Ślesicka

Dyscyplina: Automatyka, Elektronika, Elektrotechnika i Technologie Kosmiczne

Warszawa 2025

Pragnę serdecznie podziękować mojemu Promotorowi prof. dr hab. inż. Adamowi Kawalcowi za opiekę, cierpliwość, nieocenioną pomoc merytoryczną, stworzenie warunków do rozwoju naukowego oraz poświęcony czas.

Szczególne słowa podziękowania kieruje do całej mojej rodziny, w szczególności dziękuje mojej kochanej żonie Annie, na którą zawsze mogę liczyć, która mnie wspiera i docenia oraz ukochanym dzieciom Antosi i Nikosiowi. To Wam dedykuję tę pracę.

Streszczenie

W rozprawie przedstawiono zagadnienie zastosowania konwolucyjnych sieci neuronowych w radiolokacji. Główną ideą jest pokazanie możliwości zastosowania metod sztucznej inteligencji w obszarach radiolokacji, gdzie do tej pory wykorzystywane były klasyczne podejścia. Celem pracy jest opracowanie metody detekcji i rozpoznawania wybranych obiektów za pomocą radaru z falą ciągłą 0 modulowanej częstotliwości, krótkoczasowej transformacji Fouriera, a także dedykowanej struktury konwolucyjnej sieci neuronowej. W pracy zawarto opis teoretyczny budowy struktury konwolucyjnej sieci neuronowej, właściwości radaru z falą ciągłą o modulowanej częstotliwości, krótkookresowej transformacji Fouriera oraz opis szeregu metod prowadzących do stworzenia kompleksowej metody klasyfikacji obiektów. Ponadto, zwrócono szczególną uwagę na twórcze wykorzystanie zjawiska mikro-Dopplera. Zaproponowane rozwiązanie bazuje na przetworzeniu echa sygnału emitowanego przez radar z falą ciągłą o modulowanej częstotliwości i odbitego od obiektu za pomocą krótkoczasowej transformacji Fouriera celem uzyskania zobrazowań będących danymi wejściowymi konwolucyjnej sieci neuronowej. Wydajność zaproponowanego rozwiązania została zweryfikowana szeregiem symulacji oraz badaniami praktycznymi dla jednakowych zbiorów danych wejściowych. Praktycznym zagadnieniem analizowanym w pracy jest ważne społecznie zagadnienie poprawy bezpieczeństwa uczestników ruchu drogowego poruszających się po drogach, parkingach, lotniskach lub innych ważnych miejscach użyteczności publicznej, gdzie należy zadbać o bezpieczeństwo lub identyfikację pojawiających się obiektów. Przeprowadzone eksperymenty, a także przedstawione wyniki oraz analizy skutecznie dowiodły o możliwości zastosowania konwolucyjnych sieci neuronowych do rozpoznawania obiektów w radiolokacji.

Słowa kluczowe: konwolucyjne sieci neuronowe, radar z falą ciągłą o modulowanej częstotliwości, krótkoczasowa transformacja Fouriera, zjawisko mikro-Dopplera.

Abstract

The dissertation presents the application of convolutional neural networks in radar technology. The primary aim is to demonstrate the potential of artificial intelligence methods in radar domains where traditional approaches have previously been employed. The aim of the study is to develop a method for detecting and recognizing selected objects using a frequency modulated continuous-wave radar, short-time Fourier transform, and a dedicated convolutional neural network structure. The dissertation includes a theoretical description of the convolutional neural network structure, the properties of frequency modulated continuous-wave radar, short-time Fourier transform, and a range of methods leading to a comprehensive object detection approach. Special attention is given to the creative use of the micro-Doppler phenomenon. The proposed solution is based on processing the echo signal emitted by the frequency modulated continuous-wave radar and reflected from an object using the short-time Fourier transform to generate representations that serve as input data for the convolutional neural network. The effectiveness of the proposed solution has been validated through a series of simulations and practical studies on identical input data sets. The practical aspect analyzed in the study addresses the socially significant issue of improving the safety of road users moving through roads, parking lots, airports, and other critical public spaces, where ensuring the security and identification of detected objects is essential. The conducted experiments, along with the presented results and analyses, have effectively demonstrated the feasibility of using convolutional neural networks for object recognition in radar applications.

Keywords: convolutional neural networks, frequency modulated continuous-wave radar, short-time Fourier transform, micro-Doppler phenomenon.

Spis tr Lista sl	eści krótów	11
Lista s	ymboli	13
Wstęp	·	15
Wpro	wadzenie	15
Moty	wacje	16
Cel i	teza pracy	17
Wkła	d własny	18
Ukłac	I pracy	18
1. SZ	TUCZNA INTELIGENCJA	21
1.1.	Rys historyczny rozwoju sztucznej inteligencji	21
1.2.	Sieci neuronowe	23
1.3.	Konwolucyjne sieci neuronowe	26
1.4.	Miary oceny jakości klasyfikacji	32
2. KO	NWOLUCYJNE SIECI NEURONOWE W RADIOLOKACJI	35
2.1.	Radar jako sensor	35
2.2.	Reprezentacje danych radarowych	36
2.3.	Przykłady zastosowań konwolucyjnych sieci neuronowych w radiolokacji	39
3. AN	ALIZA RADARU FMCW ORAZ TRANSFORMACJI STFT	49
3.1.	Radar FMCW	49
3.2.	Radar uRAD USB v1.2	53
3.3.	Krótkoczasowa transformacja Fouriera STFT	55
4. AN	ALIZA OPRACOWANYCH METOD	59
4.1.	Metoda symulowania radaru, środowiska oraz obiektów badań	59
4.2.	Metoda generowania spektrogramów za pomocą radaru uRAD v1.2	63
4.3.	Metoda doboru struktury konwolucyjnej sieci neuronowej	67
4.4.	Metoda rozpoznawania obiektów za pomocą radaru FMCW	72
5. WA	LIDACJA EKSPERYMENTALNA	75
5.1.	Baza danych zastosowana w eksperymentach numerycznych	75
5.2.	Analiza symulacyjna wyników eksperymentów numerycznych	77
5.2.1	Radar FMCW 77 GHz	77
5.2.2	Radar FMCW 24 GHz	80
5.3.	Baza danych zastosowana w eksperymentach praktycznych	81
5.4.	Wyniki eksperymentów praktycznych	84
5.5.	Podsumowanie wyników eksperymentów	85
5.6.	Porównanie otrzymanych wyników badań z danymi literaturowymi	87
6. PO	DSUMOWANIE I KIERUNKI DALSZYCH BADAŃ	91
6.1.	Wnioski	91

6.2.	Proponowane kierunki dalszych badań	
Bibliog	grafia	95
Spis ry	/sunków	101
Spis tabel		103
Wykaz	najważniejszych publikacji autora rozprawy	
Kody ź	ródłowe zastosowanych programów i funkcji	107

Lista skrótów

AI	 – sztuczna inteligencja (ang. Artificial Intelligence);
ATR	 automatyczne rozpoznawanie obrazów (ang. Automatic Target Recognition);
CFAR	 algorytm stabilizacji poziomu fałszywego alarmu (ang. Constant False Alarm Rate);
CHIRP	 – skompresowany impulsu radarowy o wysokiej intensywności (ang. Compressed High Intensity Radar Pulse);
CNN	 konwolucyjna sieć neuronowa (ang. Convolutional Neural Network);
DARTS	 algorytmy automatycznego doboru struktury konwolucyjnej sieci neuronowej oparty na gradientach (ang. Differentiable Architecture Search);
DFT	 dyskretna transformacja Fouriera (ang. Discrete Fourier Transform);
EA	 algorytmy ewolucyjne (ang. Evolutionary Algorithms);
FFT	 – szybka transformacja Fouriera (ang. Fast Fourier Transform);
FP	 wynik klasyfikacji fałszywie pozytywny (ang. False Positive);
FMCW	 radar z falą ciągłą o modulowanej częstotliwości (ang. Frequency-Modulated Continuous Wave);
FN	 wynik klasyfikacji fałszywie negatywny (ang. False Negative);
GIDAS	 system zbierania danych o wypadkach w Niemczech (ang. German Incident Data Acquisition System);
LIDAR	 urządzenie służące do wykrywania obiektów i pomiaru odległości od nich za pomocą impulsów świetlnych (ang. Light Detection and Ranging);
LSTM	 odmiana sieci rekurencyjnej zdolnej do pamiętania krótkich wzorów (ang. Long Short-Term Memory);
MTI	 – filtracja ech stałych (ang. Moving Target Indication);
MTD	 wykrywanie ech ruchomych (ang. Moving Target Detecyion);
NAS	 algorytmy automatycznego doboru struktury głębokiej sieci neuronowej (ang. Neural Architecture Search);

ROC	 krzywa przedstawiająca zależność czułości i fałszywego alarmu w pełnym zakresie pracy klasyfikatora (ang. Receiver Operating Characteristic); 			
RBM	 – ograniczona maszyna Boltzmanna (ang. Restricted Boltzmann Machine); 			
RL	 – uczenie przez wzmocnienie (ang. Reinforcement Learning); 			
RNN	 – rekurencyjna sieć neuronowa (ang. Recurrent Neural Networks); 			
SAR	– radar z syntetyczną aperturą (ang. Synthetic Aperture Radar);			
SGDM	 stochastyczny spadek gradientu z momentum (ang. Stochastic Gradient Descent with Momentum); 			
STFT	 krótkoczasowa transformacja Fouriera (ang. Short Time Fourier Transform); 			
SVM	 technika wektorów podtrzymujących (ang. Support Vector Machine); 			
ТР	 wynik klasyfikacji prawdziwie pozytywny (ang. True Positive); 			
TN	– wynik klasyfikacji prawdziwie negatywny (ang. True Negative).			

Lista symboli

- acc dokładność klasyfikacji (ang. accuracy);
- b-próg (ang. bias);
- B-pasmo przenoszenia;
- c prędkość światła;
- f funkcja aktywacji neuronu;
- f_B różnica częstotliwości sygnału nadawanego i odebranego;
- fc częstotliwość pracy radaru;
- f_D częstotliwość Dopplera;
- fmax częstotliwość maksymalna;
- fmin częstotliwość minimalna;
- I dwuwymiarowa macierz danych wejściowych;
- K dwuwymiarowa macierz jądra (operacja konwolucji);
- n położenie ramki sygnału poddawanego STFT;
- N liczba wejść neuronu lub długość ramki sygnału poddawanego STFT;
- M całkowita liczba klas w klasyfikacji wieloklasowej;
- prc precyzja;
- R-odległość do obiektu;
- spc swoistość;
- sst czułość (ang. sensitivity);
- t-czas;
- $S_x^{SPECTOGRAM}(t, f)$ spektrogram STFT;
- T- okres czasu;
- V prędkość obiektu;
- w wektor wag;
- w(t) funkcja okna w STFT;
- x wejście neuronu;
- X(t, f) ciągła krótkoczasowa transformacja Fouriera STFT;

X(n,m) – dyskretna krótkoczasowa transformacja Fouriera STFT;

y – wyjście neuronu;

 β – współczynnik będący ilorazem pasma sygnału chirp do jego okresu nadawania.

Wstęp

Wprowadzenie

Od ponad 10 lat obserwowany jest dynamiczny rozwój sztucznej inteligencji, w tym uczenia maszynowego, a dokładniej algorytmów głębokiego uczenia. Do przełomowego osiągnięcia należy zaliczyć rok 2012, kiedy to w konkursie projektowania i oceny algorytmów rozpoznawania obrazów "ImageNet" zwyciężyło rozwiązanie oparte o głębokie sieci neuronowe autorstwa A. Krizhevsky, I. Sutskever oraz G. E. Hinton. Ich rozwiązanie uzyskało dokładność wyższą o prawie 10 punktów procentowych od kolejnego [1].

W 2016 roku doszło do następnego przełomu, kiedy model AlphaGo wygrał z człowiekiem w grę Go, która jest zdecydowanie bardziej złożona niż szachy [2]. W 2020 roku świat ujrzały pierwsze artykuły napisane przez modele GPT3, które poddano weryfikacji osobom nieświadomym tego, że są dziełem maszyny [3]. Nadszedł wreszcie 2022 rok i pojawił się ChatGPT, który jak się później okazało na stałe wkroczył w życie wielu osób [4]. W 2023 roku stworzono jeden z najbardziej zaawansowanych algorytmów, który jest zdolny rekonstruować myśli człowieka na podstawie wyników badania funkcjonalnego rezonansu magnetycznego [5]. Właściciel ChatGPT, czyli firma Open Al planuje niedługo udostępnić komercyjnie model SORA, którego zadaniem będzie generacja wysokiej jakości filmów na podstawie tekstu podanego przez użytkownika [6].

To tylko niektóre osiągnięcia sztucznej inteligencji do których doprowadził człowiek. Warto w tym miejscu jeszcze zauważyć, że sztuczna inteligencja jest wykorzystywana z sukcesem w takich dziedzinach jak medycyna, robotyka, chemia, rolnictwo czy pojazdy autonomiczne [7]. Jednakże z czasem zdarzały się przypadki katastrof czy nawet dochodziło do śmierci niewinnych osób w wyniku kraksy autonomicznych pojazdów bądź pomyłek sztucznej inteligencji [8]. Rozpoczęły się dyskusje nad dalszym rozwojem i ingerencją sztucznej inteligencji w normalne funkcjonowanie ludzkości. Wniosek nasuwa się oczywisty, że należy się do wykorzystania sztucznej inteligencji w wielu obszarach naszego życia przyzwyczaić, a nawet przygotować.

Zastosowanie sztucznej inteligencji zauważalne jest również w radiolokacji, co jest przedmiotem niniejszej rozprawy. Szczególnie duże nadzieje wiąże się

z zastosowaniem sztucznej inteligencji w wojskowych systemach radarowych. Omawiana technologia ma doprowadzić do stworzenia nowej generacji zautomatyzowanych systemów dowodzenia i narzędzi do wypracowywania przewagi informacyjnej i decyzyjnej.

Motywacje

Czynników motywacyjnych które skłoniły autora do przedstawienia swoich dotychczasowych badań nad zastosowaniem głębokich sieci neuronowych w radiolokacji jest kilka.

Główna idea jaka przyświecała autorowi polegała na tym, aby pokazać możliwość zastosowania metod sztucznej inteligencji w obszarach radiolokacji, gdzie do tej pory wykorzystywane były klasyczne podejścia. Stąd, praca ta stanowi swoistą próbę zmierzenia się z problematyką zastosowania sztucznej inteligencji właśnie w radiolokacji. W szczególności, w pracy podjęto badania nad zastosowaniem głębokich sieci neuronowych do rozpoznawania wybranych obiektów. Praktycznym zagadnieniem analizowanym w pracy jest ważne społecznie zagadnienie poprawy bezpieczeństwa uczestników ruchu drogowego poruszających się po drogach, parkingach, lotniskach lub innych ważnych miejscach użyteczności publicznej, gdzie należy zadbać o bezpieczeństwo lub identyfikacje pojawiających się obiektów.

Według najlepszej wiedzy autora, bardzo trudne bądź praktycznie niemożliwe jest doszukanie się opracowania w języku polskim traktującym o radiolokacji w aspekcie zastosowania w niej metod sztucznej inteligencji. Stąd, niniejsza rozprawa może stanowić źródło wiedzy dla studentów, przyszłych inżynierów-radiotechników oraz wszystkich zainteresowanych tą dziedziną nauki.

Autorowi zależało także na pokazaniu potencjału metod sztucznej inteligencji z wykorzystaniem radaru z falą ciągłą z modulacją częstotliwości FMCW (ang. Frequency-Modulated Continuous Wave). Biorąc pod uwagę korzyści jakie już wniosła sztuczna inteligencja w rozwój różnych technologii, celowym jest dalsze inwestowanie w nią, szczególnie w obszarze radiolokacji, w tym w aspekcie wojskowym. Takie działanie pozwoli podnieść potencjał bojowy wojsk własnych i jednocześnie zwiększyć odporność na negatywne oddziaływanie przeciwnika.

Powyższe jest zawarte w dokumencie "Priorytetowe kierunki badań w resorcie obrony narodowej w latach 2021-2035", gdzie jednoznacznie wskazuje się kierunki

badań naukowych i rozwój technologii zgodne z perspektywicznymi potrzebami Sił Zbrojnych RP. Stąd, podjęta praca jest interdyscyplinarna, ponieważ wpisuje się w 5 z 15 obszarów technologicznych opisanych w wyżej wymienionym dokumencie, którymi są: sztuczna inteligencja, autonomia i autonomizacja, Big Data, sensory oraz systemy symulacyjne.

Cel i teza pracy

Celem pracy jest opracowanie metody detekcji i rozpoznawania obiektów za pomocą radaru z falą ciągłą o modulowanej częstotliwości (ang. Frequency-Modulated Continuous Wave, FMCW), krótkoczasowej transformacji Fouriera (ang. Short Time Fourier Transform STFT) a takze dedykowanej struktury konwolucyjnej sieci neuronowej (ang. Convolutional Neural Network CNN). Ponadto, za cel autor postawił sobie również stworzenie kompleksowego i uniwersalnego kompendium wiedzy dotyczącego zastosowania sztucznej inteligencji w radiolokacji. Oprócz szczegółowego przeglądu literatury w tym zakresie, tzn. opisania podstaw budowy i działania konwolucyjnych sieci neuronowych a także radaru FMCW przedstawiono w szczególności w rozprawie metodę detekcji i rozpoznania obiektów za pomocą konwolucyjnych sieci neuronowych. Metoda bazuje na przetworzeniu echa sygnału emitowanego przez radar FMCW i odbitego od obiektu za pomocą krótkoczasowej transformacji Fouriera celem uzyskania zobrazowań będących danymi wejściowymi konwolucyjnej sieci neuronowej. Wydajność zaproponowanego rozwiązania została zweryfikowana szeregiem symulacji oraz badaniami praktycznymi dla jednakowych zbiorów danych wejściowych.

Biorąc powyższe pod uwagę, a także analizę dostępnej literatury oraz szereg wykonanych eksperymentów autor niniejszej rozprawy postawił tezę:

Możliwe jest stworzenie zautomatyzowanej metody pozwalającej na detekcję i jednoznaczne rozpoznanie wybranych obiektów ruchu drogowego za pomocą radaru FMCW stosując krótkoczasową transformację Fouriera echa odebranego sygnału oraz dedykowaną strukturę konwolucyjnej sieci neuronowej jako klasyfikatora obiektów.

Wkład własny

Na wkład własny pracy składa się:

- opis rozwoju sztucznej inteligencji oraz charakterystyka konwolucyjnych sieci neuronowych;
- przegląd literatury w zakresie wykorzystania sztucznej inteligencji w radiolokacji;
- opracowanie metody pozwalającej na detekcję i jednoznaczne rozpoznanie obiektów w postaci uczestników ruchu drogowego za pomocą radaru FMCW z wykorzystaniem krótkoczasowej transformacji Fouriera echa odebranego sygnału oraz dedykowaną strukturę konwolucyjnej sieci neuronowej jako klasyfikatora obiektów;
- opracowanie metody generowania wejściowej bazy danych symulowanych obiektów takich jak piesi, rowerzyści oraz samochody o różnych właściwościach;
- przygotowanie rzeczywistej bazy danych obrazów treningowych, testowych oraz walidacyjnych dla proponowanej metody stosując sztuczne metody zwiększania zbioru danych;
- projekt opracowanej metody opartej na rzeczywistych sygnałach pochodzących z radaru FMCW;
- wykonanie badań numerycznych opracowanej metody wykorzystującej radar FMCW pracujący w dwóch pasmach częstotliwości celem weryfikacji przyjętej tezy;
- wykonanie badań praktycznych opracowanej metody wykorzystującej radar FMCW pracujący na częstotliwości 24 GHz celem weryfikacji przyjętej tezy;
- analiza uzyskanych wyników w celu wykorzystania zaproponowanej metody do ważnych społecznie zagadnień jak chociażby poprawa bezpieczeństwa uczestników ruchu drogowego.

Układ pracy

Praca została podzielona na 6 rozdziałów. W pierwszym rozdziale przybliżono rys historyczny oraz opis konwolucyjnych sieci neuronowych. Drugi rozdział poświecono problematyce obecnego stanu badań nad zastosowaniem metod sztucznej inteligencji w radiolokacji. Rozdział trzeci zawiera opis radaru z falą ciągłą o modulowanej częstotliwości oraz opis krótkoczasowej transformacji Fouriera. W rozdziale czwartym opisano opracowane metody niezbędne do zbadania możliwości rozpoznawania obiektów za pomocą radaru FMCW i konwolucyjnej sieci neuronowej. Natomiast w rozdziale piątym opisane zostały przedstawione wyniki badań eksperymentalnych oraz symulacyjnych. W ostatnim, szóstym rozdziale zawarto podsumowanie oraz wnioski płynące z przeprowadzonych badań.

Rozdział 1

1. SZTUCZNA INTELIGENCJA

rozdziale pierwszym przybliżono etapy rozwoju sztucznej inteligencji W w aspekcie dziedziny nauk technicznych. Przedstawiono budowę, zasadę działania zastosowania konwolucyjnych sieci neuronowych jako narzędzia oraz klasyfikacji obiektów, do rozpoznawania oraz а także analizy obrazów i ich przetwarzania.

1.1. Rys historyczny rozwoju sztucznej inteligencji

Sztuczna inteligencja (ang. Artificial Intelligence - AI) to w ogólności zbiór bardzo różnych metod, które zaimplementowane w urządzeniach i aplikacjach mają zaspokajać ludzkie potrzeby, a jednocześnie sposób ich zachowania wykazuje zdolności ludzi to znaczy rozumowanie, uczenie się oraz kreatywne myślenie. W literaturze stosuje się ogólny podział metod sztucznej inteligencji na metody całościowe i metody symboliczne [10]. Metody symboliczne oparte na wykorzystaniu logiki i operacji na symbolach nie są zdolne do rozwiązywania bardzo złożonych problemów. Do takich zadań zdolne są metody całościowe oparte na algorytmach heurystycznych. Do metod całościowych zalicza się uczenie maszynowe, w tym sieci neuronowe, algorytmy genetyczne czy logikę rozmytą. Kompleksowy podział metod całościowych sztucznej inteligencji wraz z obszarami wspólnych zastosowań bardzo dobrze oddaje rysunek 1.1. zaczerpnięty z jednej z bardzo wielu publikacji prof. dr hab. inż. Ryszarda Tadeusiewicza, wybitnego polskiego uczonego w zakresie sztucznej inteligencji ale przede wszystkim wspaniałego popularyzatora nauki [11].

Wiedza o sieciach neuronowych jest ściśle związana z ludzkim mózgiem. To od fascynacji możliwości tego narządu rozpoczęły się prace naukowe związane z opracowaniem sztucznych sieci neuronowych. Pierwsze próby matematycznego opisu funkcjonowania pojedynczych komórek nerwowych czyli neuronów notuje się na lata 40-te XX wieku [12]. Znacznie bardziej rozbudowany opis prac nad sieciami neuronowymi zawarto w tamtym czasie w publikacji [13].





W 1958 roku w Cornell Aeronautical Laboratory za sprawą Franka Rosenblata oraz Charlesa Wightmana powstał "perceptron" czyli pierwsza działająca sztuczna sieć neuronowa [14]. Równolegle dwoje naukowców David Hubel oraz Torsten Wiesel prowadzili eksperymenty mające na celu zbadanie reakcji kota na pokazywane obrazki. Do mózgu kota przypięto elektrody i po serii badań wymienieni naukowcy doszli do wniosku, iż na każdy obrazek reagowała tylko wybrana grupa neuronów mózgu kota. Kilkanaście lat później, w roku 1981 obaj naukowcy otrzymali nagrodę Nobla w dziedzinie fizjologii lub medycyny za odkrycie mechanizmu przetwarzania informacji wzrokowych [15]. Niestety, ogromny entuzjazm i zapał wielu badaczy pracujących nad rozwojem sztucznych sieci neuronowych został gwałtownie zahamowany przez publikację z 1969 roku Panów Marvina Minsky'ego oraz Seymoura Papert'a [16]. Autorzy tej pracy jednoznacznie dowiedli tezę o bardzo ograniczonym zakresie zastosowania opracowywanych dotychczas jednowarstwowych sieciach neuronowych.

Trwający do początku lat 80-tych marazm w rodzącej się w bólach dziedzinie nauki przerwał postęp technologiczny w zakresie produkcji układów scalonych mogących bardzo dobrze modelować sieci neuronowe, a także wzrost nakładów finansowych na rozwój tej dziedziny nauki [10]. Stąd, na początku lat 90-tych za sprawą Kunihiko Fukushimy został zdefiniowany wielowarstwowy "neocognitron" dający początek technologii głębokich sieci neuronowych [17]. Następnie doszło do zdefiniowania specjalizowanej sieci wielowarstwowej czyli przedmiotu badań niniejszej rozprawy a mianowicie konwolucyjnej sieci neuronowej [18]. Połączenie opracowanej sieci przez profesora Yanna LeCun z postępem technologii informatycznych spowodowało, że realnym stało się projektowanie pojazdów autonomicznych, rozpoznawania obrazów w czasie rzeczywistym, generacja tekstu na podstawie emitowanego głosu czy rozwiązywanie szeregu innych problemów jak prognozowanie zapotrzebowania na energie elektryczną lub prognozowanie rozwoju przedsiębiorstw.

Kolejnym punktem zwrotnym w historii rozwoju konwolucyjnych sieci neuronowych był wspomniany we wstępie rok 2012 i efektowne zwycięstwo A. Krizhevsky, I. Sutskever oraz G. E. Hinton w konkursie "ImageNet". Dalsze sukcesy w tej dziedzinie nauki pokazujące wzrost dokładności opracowywanych sieci głębokiego uczenia przedstawiono na rysunku 1.2.



Rys. 1.2. Porównanie dokładności najlepszych modeli sieci CNN na przestrzeni lat 2016-2024 w konkursie ImageNet [19].

1.2. Sieci neuronowe

Sieci neuronowe z definicji są modelami uczenia maszynowego, które zbudowane są z neuronów odpowiedzialnych za wykonywanie prostych działań matematycznych. W literaturze można znaleźć wiele różnych modeli neuronów i metod ich uczenia [7]. Najprostszy, matematyczny model neuronu przedstawiono na rysunku 1.3.



Wagi

Rys. 1.3. Matematyczny model neuronu [10].

Działanie takiego neuronu można opisać zależnością [10]:

$$y = f\left(\sum_{i=0}^{N} \boldsymbol{w}_{i}\boldsymbol{x}_{i} + b\right)$$
(1.1)

gdzie:

y-wyjście neuronu,

x – wejście neuronu,

N – liczba wejść,

f – funkcja aktywacji,

w-wektor wag,

Funkcja aktywacji, wektor wag, a także liczba i rodzaj warstw sieci neuronowej, współczynnik uczenia czy szereg innych ustawień modelu, o których będzie mowa później składają się na pojęcie określane hiperparametrami i w sposób bezpośredni określają działanie neuronu. Wartości hiperparametrów są dobierane przed każdym treningiem sieci. Seria treningów sieci jest określana mianem procesu uczenia sieci. Stąd proces uczenia polega na dostrajaniu hieperparametrów modelu sieci w celu uzyskania optymalnej skuteczności realizacji założonego zadania.

Spośród wyżej wymienionych hiperparametrów bardzo ważne na etapie projektowania sieci neuronowej jest dobranie odpowiedniej funkcji aktywacji.

Do najbardziej popularnych funkcji aktywacji zalicza się funkcję liniową, sigmoidalną, tangesoidalną czy najczęściej stosowaną funkcję ReLU. Na rysunku 1.4. pokazano przebiegi wymienionych funkcji aktywacji.



Rys. 1.4. Przebiegi wymienionych funkcji aktywacji.

Na uwagę zasługuje funkcja ReLU lub jej warianty, ponieważ jest szeroko stosowana w większości struktur głębokich sieci neuronowych. Ponadto jej używanie usprawnia proces uczenia, co jest kluczowe w głębokich sieciach neuronowych. Po za tym jest prosta w implementacji i przeciwdziała problemowi zanikającego gradientu [7].

W związku z tym, że neurony komunikują się ze sobą poprzez przesyłanie sygnałów, a ich organizacja przypomina sieć, stąd ich nazwa – sieci neuronowe. Organizacja w postaci sieci pozwala rozwiązywać skomplikowane problemy. Najczęściej stosowanym typem organizacji sieci jest organizacja warstwowa. W literaturze wyróżnia się warstwy wejściową, wyjściową oraz warstwy ukryte. Stąd, wyjście jednej warstwy stanowi wejście kolejnej. Na rysunku 1.5. przedstawiono prosty model sieci neuronowej zbudowanej z omówionych warstw.





Podsumowując powyższe należy zauważyć, że działanie każdej sieci neuronowej zależy wprost od przyjętego modelu neuronu, przyjętej struktury sieci, a także od wartości hiperparametrów ustalanych w wyniku uczenia.

Jednokierunkowe sieci neuronowe o warstwowej strukturze były w stanie rozwiązywać różne problemy, ale wymagało to dodatkowych działań, takich jak wcześniejsze przygotowanie danych wejściowych lub wyodrębnianie ważnych cech z tych danych. Gdy dane wejściowe nie były przetwarzane, osiągane wyniki były bardzo słabe. Aby temu zaradzić, zaczęto zwiększać liczbę danych lub rozmiar samej sieci. Niestety, mimo tych zabiegów, skuteczność takich sieci pozostawała niezadowalająca. Dopiero wprowadzenie warstw konwolucyjnych pozwoliło na znaczną poprawę wyników, a dane wejściowe nie musiały już być wcześniej przetwarzane czy analizowane pod kątem istotnych cech.

1.3. Konwolucyjne sieci neuronowe

Konwolucyjne sieci neuronowe są jedną z wielu odmian sieci głębokiego uczenia. Po za nimi należy wymienić autoenkoder, sieci rekurencyjne LSTM Long Short-Term Memory) oraz ograniczoną maszynę Boltzmana (ang. (ang. Restricted Boltzmann Machine RBM) [7]. Autoenkoder to sieć neuronowa zdolna do uczenia się reprezentacji danych wejściowych. Sieć LSTM to rekurencyjna sieć neuronowa (ang. Recurrent Neural Networks RNN), w której sygnał z wyjścia sieci trafia ponownie na jej wejście. Powoduje to, że sieci rekurencyjne mają zdolność długiego zapamiętywania krótkich wzorców. Ograniczona maszyna Boltzmanna to sieć

neuronowa, która jest zdolna nauczyć się rozkładu prawdopodobieństwa danych wejściowych [10-11].

Warstwy konwolucyjne potrafią stopniowo wyodrębniać cechy z obrazu poprzez jego wielokrotne filtrowanie. Wynika to z właściwości jaką są obdarzone neurony w warstwie konwolucyjnej. Każdy neuron w warstwie konwolucyjnej połączony jest tylko i wyłącznie z kilkoma sąsiadującymi neuronami z wcześniejszej warstwy. Powoduje to wprost, że neurony w warstwie konwolucyjnej analizują bardzo mały wycinek analizowanego problemu np. małą część wybranego obrazu stąd są w stanie skutecznie i bardzo precyzyjnie wykryć na tym obrazie jego charakterystyczne cechy jak chociażby krawędzie. Warstwa konwolucyjna zasadniczo realizuje operację splotu dyskretnego (często też nazywaną konwolucją), która jest operacją liniową [7]:

$$\mathbf{Y}(i,j) = \mathbf{I}(i,j) * \mathbf{K}(i,j)$$
(1.2)

$$\boldsymbol{Y}(i,j) = \sum_{m} \sum_{n} \boldsymbol{I}(m,n) \boldsymbol{K}(i-m,j-n)$$
(1.3)

gdzie dwuwymiarowa macierz *I* reprezentuje tablice danych wejściowych, w przypadku obrazów elementy tej macierzy odpowiadają stopniu jasności pikseli *I(m, n)*, zaś macierz dwuwymiarowa *K* oznacza jądro.

Skrótowy opis działania powyższego splotu na przykładzie rozpoznawania cech na obrazach jest następujący. Na wejście podawany jest obraz wejściowy – to tzw. warstwa wejściowa (ang. input layer). Obraz wejściowy charakteryzuje się zawsze trzema parametrami: wysokością, szerokością oraz liczbą kanałów. Jeżeli obraz jest kolorowy to liczba kanałów jest równa 3 (RGB). Jeżeli obraz jest w skali szarości to liczba kanałów jest równa 1. Następnie filtr (jądro) przesuwa się w kierunku pionowym i poziomym po obrazie wejściowym o stałą wartość kroku (ang. stride) i dokonuje operacji iloczynu skalarnego Hadamarda. Rozmiar filtru (ang. kernel size) określony jest przez wymiar przesuwanego okna nad obrazem wejściowym. Następnie uzyskane iloczyny są sumowane, generując tzw. mapę cech czyli dochodzi do wyodrębnienia cech odróżniających od siebie obrazy. Mapa cech to obraz wygenerowany po operacji splotu i co ważne obraz ten w niczym nie przypomina oryginału chociaż reprezentuje jego cechy charakterystyczne. Aby uzyskać mapę cech o takiej samej wysokości i szerokości jak obraz wejściowy stosuje się sztuczne wypełnianie zerami (ang. padding) dodatkowych, zewnętrznych kolumn i wierszy

obrazu wejściowego. Przykład realizacji splotu (konwolucji) na niebieskim obrazie wejściowym o szerokości 5 pikseli i wysokości 5 pikseli poprzez filtr (jądro) o wymiarze 3 x 3 (szary kwadrat) z krokiem 2 (parametr stride) i wypełnieniem zerami równym 1 (parametr padding) przedstawiono na poniższym rysunku 1.6. W efekcie tego działania otrzymano zielony obraz (mapę cech) w wymiarach 3 x 3.



Rys. 1.6. Przykładowa operacja splotu w CNN [7].

W architekturze konwolucyjnych sieci neuronowych, oprócz warstw konwolucyjnych stosuje się również warstwy grupujące, normalizujące, spłaszczające oraz warstwy w pełni połączone.

Najczęściej stosowanymi warstwami grupującymi są warstwy "MaxPooling" oraz "AveragePooling". W ogólności operacja grupowania ma na celu stopniowe zmniejszanie rozmiaru obrazu, co automatycznie zmniejsza liczbę hiperparametrów do wytrenowania [10]. Tego typu upraszczanie sieci przekłada się wprost na czas działania sieci, skracając go. Działanie obu warstw grupujących jest wstępnie bardzo Polega na podziale całego obrazu wejściowego (mapy cech) podobne. na kwadratowe, rozłączne obszary a następnie wyborze wartości maksymalnej "MaxPooling" wyborze wartości średniej w przypadku lub przypadku w "AveragePooling". Oba rodzaje warstw grupujących wraz z przykładem liczbowym przedstawiono na rysunku 1.7.



Rys. 1.7. Warstwy grupujące w CNN.

Zastosowanie warstw normalizujących (ang. batch normalization layer) wynika z faktu występowania zjawiska polegającego na niezamierzonej zmianie rozkładu danych wejściowych do poszczególnych warstw w trakcie treningu. Przeciwdziałanie temu zjawisku następuję właśnie w warstwach normalizujących, gdzie następuje sprowadzenie średniej wartości aktywacji warstwy do zera i jednocześnie odchylenia standardowego do wartości równej jeden.

Warstwa spłaszczająca (ang. flatten layer) jest kluczowym elementem w konwolucyjnych sieciach neuronowych, służącym do zamiany wielowymiarowych danych wyjściowych na jednowymiarowy wektor. Dzięki temu dane mogą być wprowadzone do klasyfikatora, takiego jak warstwa w pełni połączona (ang. fully connected layer). Proces ten jest niezbędny, ponieważ klasyfikatory działają na wektorach o stałej liczbie wymiarów. Warto również zwrócić uwagę, że warstwa spłaszczająca nie dodaje nowych parametrów do modelu, co sprawia, że jest lekka i nie wpływa na złożoność obliczeniową sieci.

Warstwa w pełni połączona to najczęściej przedostatnia warstwa w większości struktur konwolucyjnych sieci neuronowych. Jej zadanie polega na identyfikacji do jakiej klasy należy obiekt wejściowy poprzez zwracanie wartości

prawdopodobieństwa, że obiekt na obrazie jest konkretnego rodzaju. W warstwie pełni połączonej, jak z resztą nazwa wskazuje, każdy neuron jest połączony z każdym neuronem z poprzedniej warstwy, a każde połączenie ma swoją własną wagę wynikającą z procesu uczenia.

Ostatnią warstwą najczęściej jest funkcja aktywacji "softmax", która zapewnia, że wyniki prawdopodobieństwa przynależności obiektu do poszczególnych klas sumują się do 1. Liczba neuronów w tej warstwie odpowiada liczbie klas, do których sieć może zaklasyfikować obiekt. Funkcja aktywacji "softmax" jest szczególnie użyteczna w zadaniach klasyfikacji wieloklasowej.

Proces uczenia zaprojektowanej sieci neuronowej polega na prezentowaniu kolejnych przykładów uczących na wejście sieci, generowaniu odpowiedzi i aktualizacji hiperparametrów tak aby za każdą taką iteracją minimalizować różnice pomiędzy odpowiedzią modelu a odpowiedzią oczekiwaną. W związku z tym, aby osiągnąć zamierzoną skuteczność i jednocześnie uczynić sieć zdolną do generowania poprawnych odpowiedzi dla przykładów, które nie były użyte w trakcie uczenia stosuje się szereg zabiegów i technik takich jak rozszerzanie danych, metoda "transfer learning" czy metoda "dropout" [7].

Rozszerzanie danych to generowanie dodatkowych, sztucznych danych wejściowych na zasadzie modyfikacji istniejących danych np. przesunięcie obrazu o określoną liczbę pikseli, obroty, zmiana kontrastu czy skalowanie obrazu [7]. Często w literaturze można spotkać się z określeniem "augmentacja" zbioru danych, gdzie słowo augment jest zapożyczone z języka angielskiego i oznacza zwiększanie.

Metoda "transfer learning" to technika, która polega na wykorzystaniu wiedzy zdobytej podczas trenowania jednego modelu sieci CNN do trenowania innego modelu. Jest szczególnie użyteczna, gdy ograniczone są zasoby danych wejściowych bądź mocy obliczeniowej do trenowania nowego modelu od podstaw. Technika posiada wiele zalet. Po pierwsze, nie ma potrzeby trenowania całego modelu od podstaw, co znacząco skraca czas trenowania. Po drugie, można uzyskać bardzo dobre wyniki przy użyciu mniejszych zbiorów danych. Po trzecie, wykorzystanie wstępnie przeszkolonych modeli często prowadzi do lepszej dokładności w porównaniu do modeli trenowanych od podstaw na małych zbiorach danych. Na szeroką skale wykorzystywane są wstępnie wytrenowane i dostosowane

do techniki "transfer learning" struktury sieci CNN takie jak AlexNet, ZFNet, VGGNet czy ResNet [7].

Metoda "dropout" stosowana jest w praktyce jako oddzielna warstwa w strukturze sieci CNN, której zadaniem w procesie uczenia sieci jest usuwanie w sposób losowy neuronów z poprzedniej warstwy na zasadzie zerowania ich parametrów. Prowadzi to do zwiększenia zdolności generalizacji sieci neuronowej [7].

przedstawiono Podsumowując, rysunku 1.8. przykładową strukture na konwolucyjnej sieci neuronowej z wyraźnym zaznaczeniem jej dwóch zasadniczych funkcji czyli ekstrakcji cech oraz klasyfikacji. To czyni sieci CNN jako doskonałe narzedzie do rozwiazywania różnego rodzaju zadań eksploracji danych jak klasyfikacja, segmentacja, czy rekonstrukcja. Stąd, biorąc pod uwagę powyższe właściwości sieci CNN, w niniejszej rozprawie specjalizowana struktura konwolucyjnej sieci neuronowej posłużyła jak ostateczny klasyfikator rodzaju obiektu, który znalazł się w polu wiązki radaru FMCW.



Rys. 1.8. Przykładowa struktura konwolucyjnej sieci neuronowej.

Warto również zwrócić uwagę na zupełnie pomijany przez wielu badaczy aspekt stosowania głębokich sieci neuronowych. Pomimo bardzo imponujących osiągnięć i możliwości zastosowania konwolucyjnych sieci neuronowych, praktyczna realizacja i implementacja wiąże się z uwzględnieniem i rozwiązaniem szeregu wyzwań. Do najważniejszych należy zaliczyć właściwy dobór architektury i hiperparametrów sieci, dostęp do poprawnie oznaczonego zbioru danych, niejasna interpretacja sposobu działania podejmowanego przez sieci neuronowe oraz pomyłki sieci wynikające z nieprawidłowej interpretacji danych wejściowych.

1.4. Miary oceny jakości klasyfikacji

Miary oceny jakości klasyfikacji pozwalają na ocenę wiarygodności wyników uzyskanych przez model sieci neuronowej. Poniżej opisano stosowane w ramach prowadzonych badań miary oceny jakości klasyfikacji.

Kiedy model sieci poprawnie przypisuje obserwację do jej rzeczywistej klasy, określa się taki wynik prawdziwie pozytywnym (ang. True Positive TP). Oprócz prawdziwie pozytywnych wyników, są też trzy inne kategorie. Fałszywie pozytywne (ang. False Positive FP) to sytuacja, gdy model błędnie przypisuje obserwację do danej klasy, mimo że należy ona do innej. Prawdziwie negatywne (ang. True Negative TN) to wynik, w którym model słusznie klasyfikuje obserwację jako nieprzynależną do danej klasy. Fałszywie negatywne (ang. False Negative FN) sytuacja, gdy model nie zauważa pozytywnego to przypadku. W tabeli 1.1. przedstawiono występowanie wszystkich czterech wyników klasyfikacji. Utworzona w ten sposób macierz nazywana jest w literaturze macierzą pomyłek (ang. confusion matrix).

Tabela 1.1. Macierz pomyłek stosowana do klasyfikacji dwuklasowej.

		Klasa predykowana	
		Negatywna	Pozytywna
asa wista	Negatywna	TN	FP
Kla rzeczy	Pozytywna	FN	ТР

Macierz pomyłek służy do zobrazowania liczebności występujących przypadków TP, FP, TN oraz FN. Przekątna macierzy pomyłek zawiera liczbę przypadków poprawnie sklasyfikowanych do swoich klas, natomiast wszystkie pozostałe wyniki oznaczają liczbę fałszywie sklasyfikowanych [7].

Ponadto, na bazie czterech wyżej opisanych wyników klasyfikacji definiuje się wskaźniki służące ocenie jakości modelu klasyfikacyjnego. Najbardziej znanym

i najczęściej stosowanym jest dokładność (ang. accuracy). Jest ona definiowana jako [7]:

$$acc = \frac{TP + TN}{TP + TN + FP + FN}$$
(1.4)

Dokładność jest jedną z najprostszych miar stosowanych w zadaniu klasyfikacji zarówno dwu oraz wieloklasowej. Dokładność może osiągać wartości od 0 do 1, co oznacza odpowiednio wszystkie przykłady sklasyfikowane niepoprawnie oraz wszystkie przykłady sklasyfikowane poprawnie.

Kolejnym stosowanym wskaźnikiem jest czułość (ang. sensitivity), czyli stosunek przykładów prawdziwie pozytywnych do wszystkich przykładów w badanej klasie. Jest ona definiowana jako [7]:

$$sst = \frac{TP}{TP + FN} \tag{1.5}$$

Swoistość określa zdolność wykrycia przykładów negatywnych wśród wszystkich przykładów negatywnych. [7]:

$$spc = \frac{TN}{TN + FP}$$
 (1.6)

Precyzja z kolei określa stosunek poprawnie przewidzianych pozytywnych przypadków, do liczby przypadków, które zostały określone przez model jako pozytywne [7]:

$$prc = \frac{TP}{TP + FP} \tag{1.7}$$

W niniejszej pracy rozpatrywany jest przypadek klasyfikacji wieloklasowej. Zatem macierz pomyłek, a także powyższe wskaźniki należy wyznaczyć dla poszczególnych klas oddzielnie. Stąd, macierz pomyłek będzie tabelą o wymiarze *M* x *M*, gdzie *M* oznacza liczbę klas. W przypadku opisanych w kolejnych podpunktach wyników badań, macierz pomyłek będzie miała wymiar 10 x 10 dla badań numerycznych (10 klas obiektów) oraz wymiar 6 x 6 dla badań praktycznych (6 klas obiektów). Macierz pomyłek dla klasyfikacji wieloklasowej przedstawiona została w tabeli 1.2.

	Klasa predykowana					
		1	2		М	
	1	<i>m</i> _{1,1}	<i>m</i> _{1,2}		<i>m</i> _{1,M}	$\sum_{i=1}^{M} m_{1,i}$
czywista	2	<i>m</i> _{1,1}	<i>m</i> _{1,1}		$m_{1,1}$	$\sum_{i=1}^{M} m_{2,i}$
Klasa rze	:	:	:	:		:
-	М	<i>m</i> _{1,1}	•••		<i>m</i> _{1,1}	$\sum_{i=1}^{M} m_{M,i}$
		$\sum_{i=1}^{M} m_{i,1}$	$\sum_{i=1}^{M} m_{i,2}$		$\sum_{i=1}^{M} m_{i,M}$	m

Tabela 1.2. Macierz pomyłek stosowana do klasyfikacji wieloklasowej.

W literaturze przedmiotu stosuję się również inne wskaźniki oceny miary jakości klasyfikacji. Szczegółowy opis takich wskaźników jak miara F1 bądź analiza krzywej ROC (ang. Receiver Operating Characteristic) tzw. charakterystyki operacyjnej odbiornika łączącej w sobie precyzję i czułość, można znaleźć w [7].

Przedstawione miary pozwalają na dokładną ocenę działania modelu, jednakże wymagają właściwego zastosowania. W uczeniu maszynowym zbiór danych dzieli się na trzy różne zestawy: zbiór uczący, zbiór walidacyjny i zbiór testowy. Każdy z tych zestawów ma swoją rolę. Zbór uczący służy do trenowania modelu i dostosowywania jego parametrów przy użyciu algorytmu optymalizacji. Jednorazowe użycie w procesie uczenia wszystkich przypadków uczących zawartych w zbiorze uczącym nazywane jest epoką. Zbór walidacyjny natomiast jest używany do monitorowania procesu uczenia, doboru hiperparametrów i struktury modelu. Po zakończeniu treningu ocenia się skuteczność modelu na podstawie zbioru testowego.

Rozdział 2

2. KONWOLUCYJNE SIECI NEURONOWE

W RADIOLOKACJI

W rozdziale drugim przedstawiono bardzo szeroki przekrój prac naukowych traktujących o zastosowaniu metod sztucznej inteligencji w radiolokacji. Skategoryzowano obszary zastosowania metod sztucznej inteligencji w radiolokacji, zwracając szczególną uwagę na wykorzystanie konwolucyjnych sieci neuronowych.

2.1. Radar jako sensor

Radar w swojej istocie z historycznego punktu widzenia miał za zadanie wspomagać funkcjonowanie wojska, szczególnie lotnictwa i marynarki wojennej. Najlepszym przykładem jest zespół radarów Chain Home zbudowany w Wielkiej Brytanii podczas II wojny światowej, który odegrał ważną rolę podczas bitwy o Anglię [20]. Jednakże na przestrzeni lat radar znalazł swoje zastosowanie również na rynku cywilnym jak chociażby kontrola ruchu lotniczego, lotnictwo, meteorologia, inżynieria lądowa czy medycyna.

Obecnie można zaobserwować, że radar będący drogim i bardzo zaawansowanym technologicznie urządzeniem produkowanym głównie na potrzeby wojska staje się stosunkowo tanim i dostępnym produktem penetrującym szereg gałęzi przemysłu, rynku motoryzacyjnego czy konsumenckiego. Ta zmiana oczywiście odbyła się dzięki zastosowaniu najnowszych osiągnięć technologii półprzewodnikowych ale również w dużej mierze dzięki implementacji w systemach radarowych metod sztucznej inteligencji. Zasadniczo wykorzystuje się sieci neuronowe głębokiego uczenia jak chociażby wcześniej opisane konwolucyjne sieci neuronowe.

Warto w tym miejscu wskazać zalety oraz wady stosowania radaru jako sensora. Radar jest urządzeniem emitującym fale elektromagnetyczne w zakresie mikrofal. Dlatego posiada istotną zaletę w stosunku do kamer czy LIDARu, że prowadzi obserwacje w każdych warunkach atmosferycznych zarówno w dzień jak i w nocy. Radar jest też urządzeniem trudniej rozpoznawalnym niż kamera, w przypadku gdy ma zastosowanie w bliskim kontakcie z ludźmi. Radar ma zdecydowanie większy zasięg niż wspomniane sensory optoelektroniczne. Stąd możliwe jest wykorzystanie

chociażby radaru z syntetyczną aperturą (ang. Synthetic Aperture Radar SAR) do zobrazowania terenu zarówno w dzień jak i w nocy, a także w przypadku występowania chmur, mgły i dymu. Ponadto radar ma możliwość detekcji obiektów ruchomych, a nawet jest w stanie wykrywać obiekty zamaskowane np. roślinnością lub liśćmi. Z drugiej strony echo odebrane przez radar zawiera zakłócenia utrudniające prawidłową detekcje. Występuje również zjawisko wielodrogowości oraz fałszywe alarmy. Z wojskowego punktu widzenia radar jest łatwy do wykrycia przez przeciwnika ze względu na emitowanie fali elektromagnetycznej. Zaś zobrazowania radarowe np. z radaru SAR są trudne do interpretacji, wymagają wykwalifikowanych analityków [21].

2.2. Reprezentacje danych radarowych

Jednym z najtrudniejszych zadań przy wykorzystywaniu sygnałów radarowych z modelami głębokiego uczenia jest przedstawienie sygnałów radarowych w formie odpowiadającej jako dane wejściowe do różnych algorytmów głębokiego uczenia. W tym celu przez lata zaproponowano wiele reprezentacji danych radarowych. Do najczęściej stosowanych należy zaliczyć:

- mapę zajętości siatki radarowej;
- trójwymiarową reprezentację sygnałów radarowych;
- radarową chmurę punktów;
- sygnaturę mikro-Dopplera.

Mapa zajętości siatki radarowej (ang. radar occupancy grid maps) to technika używana w systemach autonomicznych, zwłaszcza pojazdach autonomicznych, do modelowania otoczenia na podstawie sygnałów radarowych. Polega ona na podzieleniu przestrzeni otaczającej radar na regularną siatkę komórek, w której każda komórka reprezentuje pewien obszar przestrzeni. Na podstawie odbitych sygnałów radarowych, komórki siatki są klasyfikowane jako "zajęte", "wolne" lub "nieznane" w zależności od tego, czy wykryto w nich obiekt, czy nie [22].

Trójwymiarowa reprezentacja sygnałów radarowych (ang. range-doppler-azimuth tensor) to trójwymiarowa struktura danych do reprezentowania sygnałów odbitych od obiektów. Pierwszy wymiar czyli zasięg oznacza odległość między radarem a obiektem. Drugi – Doppler reprezentuje prędkość względną obiektu względem radaru. Ostatni, trzeci oznacza kierunek, w którym znajduje się obiekt względem
radaru, mierzony w płaszczyźnie horyzontalnej [22]. Na rysunku 2.1. przedstawiono widok wszystkich wspomnianych reprezentacji.

Mapa zajętości siatki radarowej



Trójwymiarowa reprezentacja danych radarowych



Radarowa chmura punktów





Rys. 2.1. Stosowane reprezentacje danych radarowych [22].

Radarowa chmury punktów (ang. radar point clouds) to chmury punktów generowane na podstawie sygnałów radarowych, które reprezentują obiekty

w otoczeniu radaru w postaci trójwymiarowych punktów. Każdy punkt w tej chmurze odpowiada odbiciu sygnału radarowego od powierzchni obiektu, co pozwala na stworzenie przestrzennej reprezentacji otoczenia. W odróżnieniu od LIDARów, które również generują chmury punktów, radary są, jak już wcześniej wspomniano, mniej podatne na warunki atmosferyczne, takie jak mgła, deszcz czy śnieg [22].

Niniejsza rozprawa w głównej mierze poświęcona jest ostatniej reprezentacji danych radarowych – sygnaturom mikro-Dopplera, którą można uzyskać dzięki występowaniu efektu mikro-Dopplera. Podstawowa różnica między efektem mikro-Dopplera, a klasycznym efektem Dopplera polega na tym, że efekt Dopplera mierzy zmianę częstotliwości fali związanej z ruchem obiektu, podczas gdy efekt mikro-Dopplera analizuje dodatkowo zmiany częstotliwości wynikajace z wewnętrznych ruchów strukturalnych obiektu. Innymi słowy, podczas gdy efekt Dopplera zajmuje się zwykłymi zmianami częstotliwości spowodowanymi ruchem obiektu jako całości, efekt mikro-Dopplera jest wykorzystywany do badania dodatkowo mikroskopijnych ruchów elementów obiektu, które również wpływają na wielkości fizyczne opisujące falę [23]. Przykład tego zjawiska pokazano na rysunku 2.2. na podstawie pomiaru zmian częstotliwości w czasie jakie wnoszą poruszające się kończyny pływaka [23].



Rys. 2.2. Efekt zjawiska mikro-Dopplera na przykładzie pływaka [23].

W przypadku radaru zjawisko mikro-Dopplera ma miejsce gdy opromieniowany przez radar obiekt zainteresowania wykonuje obrót, rotację bądź wibracje mechaniczne. Wtedy w odebranym sygnale opisane wibracje bądź ruchy w osi ciała obiektu mogą indukować dodatkowe składowe częstotliwościowe w sygnale echa. Przykładem ruchów generujących zjawisko mikro-Dopplera, są wirniki śmigłowców czy kończyny człowieka podczas ruchu [23]. Podsumowując, dynamika różnych części obiektu umieszczonego przed radarem wytwarzają sygnatury mikro-Dopplera, które można wykorzystać do identyfikacji obiektu.

Każda z wyżej wymienionych reprezentacji danych radarowych ma swoje wady i zalety, których szczegółowy opis można znaleźć w [22].

2.3. Przykłady zastosowań konwolucyjnych sieci neuronowych w radiolokacji

Poziom zainteresowania i liczbę prowadzonych badań w zakresie stosowania metod uczenia maszynowego, w tym głębokich sieci neuronowych w radiolokacji najlepiej oddaje poniższy wykres. Na rysunku 2.3. przedstawiono liczbowe zestawienie publikacji dostępnych w bazie IEEE Xplore dotyczących właśnie przedmiotu niniejszej rozprawy.



Rys. 2.3. Zestawienie publikacji dotyczących zastosowania głębokich sieci neuronowych w radiolokacji w latach 2014-2024.

Jak można zauważyć poruszana tematyka jest bardzo aktualna, a jej popularność stale rośnie. Opisywany trend jest również bardzo widoczny podczas konferencji naukowych organizowanych w naszym kraju o tematyce związanej z radiolokacją. Chociaż nie jest to przedmiotem niniejszej rozprawy to warto w tym miejscu zwrócić jeszcze uwagę na często poruszane zagadnienie podczas konferencji naukowych tj. radar kognitywny. To radar, który wykorzystuje metody sztucznej inteligencji do uczenia się otaczającego go środowiska. Powoduje to, że odbiornik i nadajnik radaru kognitywnego potrafi w czasie rzeczywistym zaadaptować się do zmian środowiska pracy [24-26].

Zastosowań metod sztucznej inteligencji w połączeniu z radarami jest bardzo wiele. Według najlepszej wiedzy autora stosuję się ją w takich obszarach jak:

- autonomiczne pojazdy [27-30],
- automatyczne wykrywanie obecności w domach lub budynkach inteligentnych [31],
- rozpoznawanie rodzaju aktywności fizycznej ludzi [32-34],
- wykrywanie, rozpoznawanie i kontrola ruchu dłoni i palców (gestów) celem zdalnego sterowania urządzeniami [35-36],
- wykrywanie upadków [37-38],
- identyfikacja i redukcja wpływu zakłóceń w odebranym sygnale echa [39-43],
- rozpoznawanie typów sygnałów docierających do radaru oraz identyfikacja jego parametrów [44-47],
- automatyczne rozpoznawanie i klasyfikacja obiektów (ang. Automatic Target Recognition ATR):
 - o za pomocą zobrazowań radaru SAR [48-51],
 - o za pomocą sygnatur mikro-Dopplera [52-58].

W związku z faktem, że niniejsza rozprawa dotyczy rozpoznawania obiektów biorących udział w ruchu drogowym tj. pieszych, rowerzystów bądź samochodów za pomocą sensorów radiolokacyjnych poprzez sygnatury mikro-Dopplera i konwolucyjną sieć neuronową, to dalsza część analizy literaturowej poświęcona została ściśle tej tematyce.

Wykrywanie i klasyfikacja celów oparta na głębokim uczeniu na podstawie sygnatur mikro-dopplerowskich zvskuje coraz większą popularność dziedzinie W rozpoznawania ruchomych obiektów automatycznego poruszających się po powierzchni ziemi, takich jak ludzie, zwierzęta czy pojazdy. Dzięki głębokiemu uczeniu, modele mogą być szkolone na analizę tych sygnatur, co umożliwia automatyczne rozpoznawanie i klasyfikację różnych typów obiektów na podstawie ich ruchu.

W artykule [52] przedstawiono modułowe podejście do przetwarzania danych radarowych, skupiając się na śledzeniu wielu obiektów oraz ich klasyfikacji. Zastosowano trzy architektury sieci neuronowych przedstawione na rysunku 2.4.

40



Warstwa w pełni połączona (1 neuron)

Rys. 2.4. Struktury sieci neuronowych: I - uproszczona sieć VGG16, II - sieć ResNet-50 oraz III - połączenie CNN z LSTM [52].

Zastosowane trzy architektury sieci neuronowych to: uproszczona sieć VGG16, sieć ResNet-50 oraz nowatorskie połączenie CNN z LSTM, które umożliwia przetwarzanie danych radarowych jako sekwencji czasowych.

W eksperymentach zarejestrowano ruchy różnych obiektów, takich jak pojedynczy pieszy, pojedynczy samochód, pojedynczy rowerzysta oraz dwie osoby idące obok siebie. Surowe dane zebrane za pomocą radaru TEF810X firmy NXP pracującego na częstotliwości 77 GHz podzielono na bloki, poddano przetwarzaniu przy użyciu szybkiej transformaty Fouriera (ang. Fast Fourier Transform FFT) oraz filtru górnoprzepustowego, co umożliwiło usunięcie obiektów stacjonarnych. Najlepsze wyniki klasyfikacji (88,6%) osiągnięto w przypadku wykrywania pojedynczych obiektów takich jak pieszy, samochód czy rowerzysta za pomocą sieci VGG-16. Autorzy jako wniosek wskazują na niewielki zbiór danych wejściowych, który przyczynił się do słabego i czasochłonnego trenowania sieci, sięgającej aż 200 epok.

W pracy [53] zaproponowano system rozpoznawania pieszych przy użyciu radaru samochodowego o częstotliwości 79 GHz. Głównym celem opracowanego systemu jest wczesne wykrywanie pieszych w sytuacjach bliskich wypadkowi (0-15 m). Jako klasyfikator zastosowano technikę wektorów podtrzymujących (ang. Support Vector Machine SVM) do rozróżniania pieszych i obiektów niebędących pieszymi. W pracy rozważono cztery różne modele oparte na SVM, mające na celu poprawę zarówno wydajności klasyfikacji, jak i szybkości, porównując trzy różne funkcje jądra. Do jednoznacznego przypisania danych radarowych do dwóch różnych klas wykorzystano obraz z kamery, która była tak samo skierowana na cel co radar. Stąd, za pomocą algorytmów etykietowania sąsiednich komórek na zobrazowaniu radarowym i weryfikacji obrazu z kamery utworzono zbiór danych [54-55]. Przeprowadzono badania dla sześciu różnych scenariuszy ruchu pieszego na podstawie analizy systemu zbierania danych o wypadkach w Niemczech GIDAS (ang. German Incident Data Acquisition System). We wszystkich sześciu scenariuszach przeprowadzono łącznie 220 różnych testów. Autorzy podsumowując zauważyli bardzo słuszny wniosek, który został zobrazowany na rysunku 2.5. We wspomnianych badaniach dysponowano ograniczonymi zasobami obliczeniowymi i małym zbiorem danych w porównaniu do problemów na dużą skalę, co uzasadnia wybór metody SVM. Finalnie, osiągnięto dokładność klasyfikacji na poziomie 99,5%.



Rys. 2.5. Uzasadnienie wyboru metody SVM w badaniach [53].

Autorzy artykułu [56] zaprezentowali po raz pierwszy w literaturze przedmiotu możliwość wykorzystania techniki *"transfer learning*", czyli wstępnie wytrenowanej konwolucyjnej sieci neuronowej VGG-16 jako ekstraktora cech. Uzyskane cechy posłużyły do trenowania klasyfikatora SVM dla problemu wieloklasowego.

Struktura zastosowanej sieci neuronowej była identyczna jak w [53]. Jest to model sieci CNN, który osiągnął najlepsze wyniki dokładności w wyzwaniu "ImageNet 2014" w kategoriach lokalizacji i klasyfikacji. Do walidacji zaproponowanego podejścia wykorzystano publicznie dostępną bazę danych o nazwie RadEch. Baza danych RadEch została opublikowana przez Akademię Wojskową w Belgradzie w oparciu o radar pracujący na częstotliwości 16,8 GHz. Baza danych zawierała zarejestrowane w dziedzinie czasu sygnały o maksymalnym czasie trwania 4 sekundy. Surowe sygnały przekształcono w spektrogramy, czyli w graficzną reprezentację sygnału, która pokazuje, jak zmienia się jego widmo częstotliwościowe w czasie. Ponadto, w celu powiększenia bazy danych spektrogramów zastosowano szereg technik augmentacji danych tj. metod sztucznego zwiększania liczby danych treningowych. W ten sposób poszerzono bazę danych aż siedemnastokrotnie. Finalnie, osiągnięto dokładność klasyfikacji na poziomie 96,56%. W eksperymencie wykorzystano wszystkie dostępne 8 klas obiektów z bazy RadEch:

- 1. jedna osoba idąca;
- 2. jedna osoba biegająca;

- 3. jedna osoba czołgająca się;
- 4. grupa maszerujących osób;
- 5. grupa biegnących osób;
- 6. pojazd kołowy;
- 7. ciężarówka;
- 8. zakłócenia pasywne (ang. clutter).

Autorzy artykułu [57] zaprezentowali sposób klasyfikacji ludzi i zwierząt, opierając się na sygnaturach mikro-Dopplera uzyskanych za pomocą transformacji STFT, a także pokazali sposób sztucznego zwiększania danych treningowych. Proponowane podejście wykorzystuje konwolucyjne sieci neuronowe o strukturze przedstawionej na rysunku 2.6.



Rys. 2.6. Struktura sieci CNN w artykule [57].

Zbadano wpływ architektury konwolucyjnych sieci neuronowych oraz metody zwiększania liczby danych treningowych (tzw. augmentacja danych) na efektywność klasyfikacji. Wykorzystano zbiór gotowych danych MAFAT.

Zidentyfikowano zjawisko przeuczania, zwłaszcza w bardziej złożonym modelu ośmiowarstwowym. Jak się okazało wprowadzenie techniki "dropout" znacząco zredukowały przeuczanie. Analizowano także różne głębokości architektur konwolucyjnych sieci neuronowych oraz ich wpływ na wyniki. Zbadano także skuteczność różnych metod augmentacji, takich jak odwracanie obrazów, wprowadzanie szumów i losowe przesunięcia czasowe. Przykład operacji zwiększania zbioru metodami danych różnymi przedstawiono na rysunku 2.7. Dzięki tym metodą, zbiór danych MAFAT mający wstępnie 6500 obrazów został rozszerzony do ponad 15 000 obrazów. Finalnie, osiągnięto dokładność klasyfikacji na poziomie 93,62%.



Rys. 2.7. Metody zwiększania zbioru danych treningowych w artykule [57]. A – obraz oryginalny, B – losowe przesunięcie częstotliwości, C - losowe przesunięcie w czasie, D – odwrócenie w poziomie, E – odwrócenie w pionie, F – dodanie szumów.

W pracy [58] również przedstawiono badania nad wykorzystaniem techniki radarowej do rozpoznawania pieszych i zwierząt, skupiając się na analizie sygnatur mikro-Dopplera. Autorzy zastosowali specjalizowaną strukturę konwolucyjnej sieci neuronowej (CNN) w celu klasyfikacji obiektów, osiągając wysoką dokładność w ramach wyzwania *MAFAT Radar Challenge*. Strukturę tej sieci przedstawiono na rysunku 2.8.

Dane w wyzwaniu MAFAT są danymi rzeczywistymi zebranymi z różnych lokalizacji geograficznych, w różnych porach dnia i roku, przy użyciu różnych radarów oraz o różnej jakości sygnału (np. stosunek sygnału do szumu).



Rys. 2.8. Struktura sieci CNN w artykule [58].

W badaniach zaprezentowano nową architekturę sieci głębokiego uczenia. Proces klasyfikacji obejmował wiele kroków wstępnego przetwarzania surowych danych radarowych, w tym transformację FFT i normalizację. Ostateczne podejście do klasyfikacji wykorzystuje różnorodne techniki augmentacji zbioru danych oraz wiele warstw sieci CNN, co pozwoliło na osiągnięcie wysokich wyników, mimo problemów

z przeuczeniem i niezbilansowaniem zbioru danych. Praca [58] potwierdza, że odpowiednie techniki augmentacji są kluczowe w kontekście głębokiego uczenia, zaś procedura optymalizacji hiperparametrów ma znaczenie dla osiągnięcia wysokiej dokładności klasyfikacji obiektów. Finalnie, osiągnięto dokładność klasyfikacji na poziomie 94,04%.

Podsumowując powyższy przegląd literatury zaproponowano brakujące podejście w zakresie wykrywania obiektów takich jak piesi, rowerzyści, samochody za pomocą radaru o ciągłej fali modulowanej częstotliwością z wykorzystaniem danych radarowych w postaci spektrogramów uzyskanych poprzez krótkoczasową transformację Fouriera. Najważniejsze właściwości nowego podejścia to:

- wykrywanie większej klasy obiektów takich jak chociażby dwoje maszerujących pieszych lub przypadek jednoczesnego ruchu pieszego i rowerzysty w warunkach otwartej przestrzeni;
- ekstrakcja cech oraz klasyfikacja obrazów będących spektrogramami wielu klas obiektów poprzez specjalizowaną, konwolucyjną sieć neuronową CNN;
- specjalizowana sieć CNN zdolna do uzyskania bardzo wysokiej dokładności klasyfikacji obiektu (> 90 %);
- sieć CNN odporna na przeuczenie czyli przypadek gdy model sieci CNN zbyt dobrze dopasowuje się do danych treningowych, ale traci zdolność do generalizowania na nowe, niewidziane wcześniej dane;
- rozwiązanie oparte o symulacje numeryczne na symulowanym zbiorze danych wejściowych w postaci spektrogramów (dla radaru pracującego na częstotliwości 24 GHz oraz 77 GHz);
- praktyczna weryfikacja rozwiązania numerycznego oparta na własnym zbiorze danych z dostępnego radaru FMCW pracującego na częstotliwości 24 GHz;
- zastosowanie techniki sztucznego zwiększania (tzw. augmentacji) zbioru danych celem przezwyciężenia trudnego i czasochłonnego przetwarzania danych z pojedynczego pomiaru radarem FMCW.

W poniższej tabeli zebrano omówiony przegląd literaturowy, który pozwolił sformułować powyższe wnioski oraz zaproponować brakujące rozwiązanie w temacie niniejszej rozprawy.

Publikacja	Sensor lub zbiór danych	Przetwarzanie wstępne	Klasyfikator	Rozpoznawane obiekty (% dokładności)
Angelov [52] 2018	Radar TEF810X 77 GHz	STFT	I sieć - uproszczona VGG-16; II sieć – ResNet-50; III sieć - połączenie CNN i LSTM	Piesi, rowerzysta, samochód 88,6 %
Joao [53] 2019	Radarlog 79 GHz	kamera + algorytmy [53-54]	SVM	Piesi 99,5 %
Hadhrami [56] 2020	Zbiór danych RadEch 16,8 GHz	Brak informacji o konkretnej metodzie	Metoda t <i>ransfer learning</i> VGG16 + SVM	Pojedynczy pieszy, piesi, osoby czołgające się, biegacze, pojazd kołowy, ciężarówka 96,56 %
Dadon [57] 2021	Zbiór danych MAFAT	STFT	CNN	Piesi, zwierzęta 93,62 %
Buchman [58] 2022	Zbiór danych MAFAT	FFT	CNN	Piesi, zwierzęta 94, 04%

Tabela 2.1. Zbiorcze zestawienie zastosowania metod sztucznej inteligencji do wykrywania obiektów w radiolokacji.

Rozdział 3

3.ANALIZA RADARU FMCW ORAZ TRANSFORMACJI STFT

Zasadniczym celem niniejszej rozprawy jest opracowanie metody detekcji i rozpoznawania wybranych obiektów za pomocą radaru z falą ciągłą o modulowanej częstotliwości i krótkoczasowej transformacji Fouriera. Stąd, rozdział trzeci został poświęcony budowie i zasadzie działania radaru FMCW, a także przetwarzania surowych, odebranych sygnałów echa od obiektów za pomocą transformacji STFT. Dodatkowo przedstawiono również specyfikacje zastosowanego do pomiarów eksperymentalnych radaru uRAD USB v1.2.

3.1. Radar FMCW

W ogólności radar wykorzystuje zjawisko odbicia fal elektromagnetycznych w celu detekcji obiektów. Studiując szczegółowo literaturę traktującą o radarach aktywnych można je podzielić na różne klasy w zależności od przyjętych kryteriów. Z perspektywy niniejszej pracy intersującym jest podział typów radarów ze względu na rodzaj emitowanej fali elektromagnetycznej. W ogólności należy wymienić tutaj radar impulsowy, radar z falą ciągłą, a także radar z falą ciągłą o modulowanej częstotliwości [59-60].

Radar impulsowy, najbardziej popularny w technice radiolokacyjnej, wysyła ciąg impulsów sondujących w określonych odstępach będących wycinkami fali elektromagnetycznej o bardzo wielkiej częstotliwości i o określonym czasie trwania. Radary tego typu bardzo dobrze radzą sobie z precyzyjnym pomiarem odległości [61].

Radar z falą ciągłą nieprzerwanie emituje sygnał sondujący poprzez antenę nadawczą i jednocześnie odbiera echo za pomocą anteny odbiorczej. W związku z tym, radar z falą ciągłą dzięki informacjom o przesunięciu fazowym i częstotliwości Dopplera zawartych w sygnale echa bardzo dobrze mierzy prędkość radialną oraz położenie kątowe obiektu [61].

Z kolei radar z falą ciągłą o modulowanej częstotliwości łączy w sobie zalety radaru impulsowego oraz radaru z falą ciągłą. Oprócz możliwości pomiaru prędkości radialnej oraz położenia kątowego, tego typu radar dzięki modulacji nadawanego sygnału sondującego jest w stanie zmierzyć jednocześnie odległość do obiektu [62].

49

Zaletami radaru FMCW jest prosta konstrukcja, mała moc sygnału sondującego, a tym samym co ważne z wojskowego punktu widzenia - mała podatność na wykrycie przez przeciwnika. Nieskomplikowana konstrukcja radaru FMCW wynika z pominięcia elementów przełączających i synchronizujących, które występują w radarach impulsowych. Brak tych elementów związany jest z ciągłą transmisją i odbieraniem fal radiowych przez radar FMCW. W literaturze stosuje się schemat blokowy radaru FMCW, który przedstawiono na rysunku 3.1.



Rys. 3.1. Schemat blokowy radaru FMCW [62].

Zgodnie z powyższym schematem część sygnału generowanego w torze nadawczym poprzez sprzęgacz kierunkowy trafia do toru odbiorczego (sygnał LO). Pozostała część jest emitowana przez anteny nadawcze. Sygnał po odbiciu od obiektu trafia z powrotem do anten odbiorczych i następuje jego przetwarzanie w torze odbiorczym. Po wzmocnieniu sygnał odebrany (RF) zostaje wymnożony w mieszaczu z sygnałem referencyjnym (LO). W wyniku mnożenia powstają dwa sygnały: sygnał o częstotliwości sumacyjnej oraz różnicowej. Umieszczony na wyjściu mieszacza filtr dolnoprzepustowy odpowiada za odfiltrowanie składowej sumacyjnej [63].

Ponadto, postać wyjściowego sygnał IF podlega demodulacji kwadraturowej (inaczej demodulacji IQ) oraz zamianie na postać cyfrową w przetworniku analogowo-cyfrowym. Operacja demodulacji polega na przekształceniu jednowymiarowego sygnału IF w zespolony sygnał kwadraturowy IQ. Składowa

synfazowa I (ang. In-Phase) reprezentuję część rzeczywistą sygnału, zaś składowa kwadraturowa Q (ang. Quadrature) reprezentuje część urojoną. Finalnie, dyskretny zespolony sygnał IF zawiera informacje o amplitudzie sygnału dzięki składowej I oraz informacje o jego fazie poprzez składową Q [64].

W zależności od zastosowania, otrzymana wcześniej postać sygnału podlega dalszemu przetwarzaniu za pomocą jednej z dostępnych metod analizy zmian widma częstotliwości w czasie tj. transformacji Gabora, transformacji falkowej, transformacji Wignera-Ville'a czy krótkoczasowej transformacji Fouriera. Do jednoznacznego wykrycia obiektu stosuje się ostatecznie jeden z dostępnych algorytmów detekcji np. algorytm detekcji progowej bądź algorytm stabilizacji poziomu fałszywego alarmu CFAR (ang. Constant False Alarm Rate) [59].

Radar FMCW emituje falę elektromagnetyczną, której częstotliwość jest zmienna w czasie. Natomiast do najczęściej stosowanych przebiegów sygnałów zalicza się przebieg piłokształtny, trójkątny oraz sinusoidalny. W ramach niniejszej rozprawy przyjęto do rozważań, jak również do symulacji oraz pomiarów, sygnał z modulacją częstotliwości o kształcie piłokształtnym. Przykład takiego sygnału przedstawiono na rysunku 3.2.



Rys. 3.2. Modulacja częstotliwości na przykładzie przebiegu piłokształtnego [62].

Częstotliwość sygnału transmitowanego przez radar rośnie w sposób liniowy przez okres czasu *T* od wartości częstotliwości minimalnej *f_{min}* do wartości maksymalnej *f_{max}*. W czasie jednego okresu emitowany sygnał określany jest w literaturze jako skompresowany impulsu radarowy o wysokiej intensywności (ang. Compressed High Intensity Radar Pulse CHIRP). Przyjęto, że radar wysyła skończoną liczbę *K* impulsów CHIRP, każdy z liniowo narastającą częstotliwością w funkcji czasu.

Zależność częstotliwości emitowanego sygnału od czasu wyrażona została jako zależność [62]:

$$f(t) = \beta t \tag{3.1}$$

przy czym czas $t \in [0, T]$, zaś współczynnik β dany jest zależnością:

$$\beta = \frac{B}{T} \tag{3.2}$$

B oznacza szerokość pasma jako różnica pomiędzy wartością częstotliwości maksymalnej f_{max} a minimalnej f_{min} :

$$B = f_{max} - f_{min} \tag{3.3}$$

W związku z powyższym, sygnał piłokształtny z modulacją częstotliwości transmitowany przez radar FMCW można zamodelować jako [62]:

$$s(t) = \cos(2\pi f_C t + \varphi(t))$$
(3.4)

gdzie f_c oznacza częstotliwość pracy radaru, zaś fazę $\varphi(t)$ można wyrazić jako [62]:

$$\varphi(t) = \pi k \beta T^2 + \pi \beta t^2 \tag{3.5}$$

gdzie k = 0, ..., K-1. W przypadku gdy transmitowany sygnał napotka na swojej drodze obiekt, część energii fali elektromagnetycznej powróci z powrotem do radaru. Po przejściu sygnału przez tor odbiorczy sygnał przyjmuje postać [62]:

$$y(t) \approx \varepsilon e^{-j2\pi f_B t} e^{-j2\pi f_D kT}$$
(3.6)

gdzie ε oznacza zespolony współczynnik zależny od zysku antenowego, f_B oznacza różnicę częstotliwości sygnału nadawanego i echa, zaś f_D oznacza częstotliwość Dopplera. Jeżeli obiekt, od którego odbija się sygnał, jest nieruchomy, fala odbita będzie miała taką samą częstotliwość co sygnału nadawanego. Jednak gdy obiekt się porusza, częstotliwość odbitej fali zmienia się o wartość f_D , która zależy od prędkości

ruchu obiektu w kierunku radaru. Ta prędkość nazywa się prędkością radialną i jest to ta składowa prędkości obiektu, która jest skierowana wprost na radar lub od niego oddala. Znając wartości f_B oraz f_D można wyznaczyć odpowiednio prędkość i odległość do obiektu [59-61]:

$$V = \frac{f_D c}{2f_c} \tag{3.7}$$

$$R = \frac{f_B c}{2\beta} \tag{3.8}$$

gdzie c oznacza prędkość światła.

3.2. Radar uRAD USB v1.2.

Do eksperymentów fizycznych wykorzystano radar uRAD USB v1.2 pracujący na częstotliwości 24 GHz, a także dedykowane oprogramowanie do obsługi radaru. Widok radaru przedstawiono na rysunku 3.3.



Rys. 3.3. Radar uRad USB v1.2 wykorzystany do badań [65].

Moduł uRAD USB v1.2 jest w pełni funkcjonalnym mikrofalowym radarem FMCW, działającym w paśmie 24 GHz (24,005 – 24,245 GHz). Wersja USB pozwala na połączenie go do dowolnego urządzenia posiadającego złącze USB. Dzięki zastosowanemu szykowi anten pole widzenia radaru wynosi 30° w płaszczyźnie horyzontalnej oraz wertykalnej. Zasilanie radaru to 5 V (USB). Radar jest w stanie pracować w zakresie temperatur od – 20 do + 65 °C. Wymiary radaru to 56 mm x 76 mm x 5 mm, zaś jego waga to 8 gram [65]. Pozostałe właściwości urządzenia zestawiono w tabeli 3.1.

Tryb pracy	1	2	3	4		
Sygnał	Fala ciągła	Piłokształtny	Trójkątny	Dualny sygnał trójkątny		
Przebieg	Częstotliwość	Częstotliwość	Czes	Częstotliwość		
Mierzony parametr	Prędkość	Odległość	Prędkość & Odległość	Prędkość & Odległość		
Zasięg (m)	0,45-60	0,45-100	0,45-100	0,45-75		
Rozdzielczość zasięgu (m)	-	max. ±0,04 m lub ±0,3%	max. ±0,04 m lub ±0,3%	max. ±0,04 m lub ±0,3%		
Prędkość (m/s)	±0,7 to ±75	-	±0,2 to ±75	±0,2 to ±75		
Rozdzielczość prędkości (m/s)	3	-	3	3		
Maksymalna częstotliwość próbkowania (próbek/s)	103	69	39	20		
Częstotliwość operacyjna (MHz)		5-2	45			

Tabela 3.1. Zestawienie właściwości radaru uRAD USB v1.2.

Obsługę radaru oraz możliwość zapisu danych pomiarowych zapewnia dostarczone przez producenta radaru dedykowane oprogramowanie. Ponadto, producent zapewnia wizualizację danych w czasie rzeczywistym. Graficzny interfejs użytkownika oprogramowania radaru uRAD USB v1.2. przedstawiono na rysunku 3.4.



Rys. 3.4. Widok graficznego interfejsu użytkownika oprogramowania radaru uRAD USB v1.2. [65]

Oprogramowanie umożliwia zmianę parametrów pracy radaru takich jak:

- tryb pracy radaru (cztery tryby zgodnie z tabelą 3.1.);
- wybór portu komputera połączonego z radarem;
- szerokość pasma przemiatania ("BW");
- częstotliwość początkowa przemiatania ("fo");
- liczbę próbek, które radar pobiera z echa w celu wyznaczenia odległości lub prędkości ("Ns"). Możliwe jest ustawienie tego parametru w zakresie od 50 do 200 próbek;
- maksymalną liczbę wykrywanych obiektów ("NTAR") w zakresie od 1 do 5;
- aktywacje trybu MTI (ang. Moving Target Indication) czyli wykrywania tylko tych obiektów, które poruszają się. Obiekty nieruchome są pomijane;
- czułość urządzenia ("M_{th}") w zakresie od 0 do 4.

3.3. Krótkoczasowa transformacja Fouriera STFT

Aby przetworzyć surowe sygnały zebrane przez radar w trakcie pomiarów posłużono się krótkoczasową transformacją Fouriera. W ten sposób wygenerowano zbiór danych w postaci obrazów dla konwolucyjnej sieci neuronowej. Obrazem będzie kwadrat modułu krótkoczasowej transformaty Fouriera, który nazwano spektrogramem. Ta sama metoda generacji spektrogramów będzie wykorzystana

w trakcie eksperymentów symulacyjnych. Zastosowanie właśnie tej transformacji wynika z faktu, że jest ona sprawdzona i najczęściej stosowana w rozwiązaniach omówionych w analizie literatury niniejszej rozprawy.

Otrzymany sygnał od obiektu wynikający z występowania efektu mikro-Dopplera można traktować jako sygnał niestacjonarny. Sygnał niestacjonarny to taki, którego widmo częstotliwościowe zmienia się w funkcji czasu. Stąd, celem zaobserwowania różnić w widmie sygnałów odebranych od różnych obiektów, a nawet ich wspólnego występowania, stosuje się w literaturze szereg różnych transformacji [66]. Jedną z bardziej podstawowych ale jednocześnie popularnych jest właśnie krótkoczasowa transformacja Fouriera STFT.

W przeciwieństwie do standardowej transformaty Fouriera, która daje całkowitą informację o widmie sygnału, STFT pozwala na uzyskanie informacji o widmie w krótkich fragmentach sygnału. W związku z tym, STFT dzieli sygnał na krótkie, nakładające się na siebie okna czasowe, często też zwane ramkami. Po określeniu długości ramki następuje mnożenie sygnałów z tych ramek przez funkcję okna *w(t)*. Z kolei wynik mnożenia poddawany jest finalnie transformacji DFT, która najczęściej jest realizowana za pomocą konkretnego algorytmu FFT. Zabieg pozwala na uzyskanie informacji o częstotliwościach obecnych w danym momencie. Wynikiem STFT jest spektrogram – dwuwymiarowe zobrazowanie, w którym oś czasu pokazuje, jak zmienia się sygnał, a oś częstotliwości przedstawia rozkład częstotliwości dla każdego okna czasowego. Kolory lub intensywność na wykresie reprezentują amplitudy sygnału [66].

Przykład zastosowania transformacji STFT przedstawiono na rysunku 3.5. Transformacji STFT poddano sygnał z liniową modulacją częstotliwości, która rośnie liniowo od 100 Hz do 300 Hz. Wygenerowano cztery spektrogramy dla okien Hamminga, Hanna, Blackmana oraz Kaisera. W każdym przypadku długość ramki wynosiła 128 próbek.

56



Rys. 3.5. Przykład zastosowania transformacji STFT.

Istotną wadą transformacji STFT jest zależność szerokości ramki od rozdzielczości w osi częstotliwości i w osi czasu. Wąska ramka powoduje dużą rozdzielczość w osi czasu, a jednocześnie mniejszą w osi częstotliwości. Szeroka ramka skutkuje odwrotnym efektem. Powyższe zjawisko przedstawiono na rysunku 3.6.



Rys. 3.6. Przykład zastosowania transformacji STFT dla różnej długości ramki.

Definicja ciągłej krótkoczasowej transformacji Fouriera STFT wyrażona jest zależnością [65]:

$$X(t,f) = \int_{-\infty}^{\infty} x(\tau)w(\tau-t)e^{-j2\pi ft}d\tau$$
(3.9)

gdzie $x(\tau)$ oznacza analizowany sygnał w dziedzinie czasu, $w(\tau - t)$ oznacza okno czasowe, które przesuwa się wzdłuż sygnału $x(\tau)$. Najczęściej stosowane okna to okno Hanninga, Hamminga, Kaisera oraz Blackmana.

Analogicznie, definicja dyskretnej krótkoczasowej transformacji Fouriera STFT wyrażona jest zależnością [67]:

$$X(n,m) = \frac{1}{N} \sum_{k=-\infty}^{\infty} x(k) w(k-n) e^{-j2\pi \frac{mk}{N}}$$
(3.10)

gdzie x(k) oznacza ciąg próbek sygnału dyskretnego, *N* oznacza długość ramki, *n* oznacza położenie ramki wzdłuż analizowanego sygnału oraz przesunięcie czasowe m = 0, 1, 2, ..., N-1.

Dla krótkoczasowej transformacji Fouriera definiuje się tzw. spektrogram jako kwadrat jej modułu [67]:

$$S_x^{SPECTOGRAM}(t, f) = |X(t, f)|^2$$
 (3.11)

Rozdział 4

4. ANALIZA OPRACOWANYCH METOD

Rozdział zawiera opis wypracowanych w ramach prowadzonych badań metod, które pozwalają na weryfikację numeryczną oraz praktyczną postawionej we wstępie tezy. Opisane metody symulacji radaru FMCW, a także obiektów (podrozdział 4.1.) zostały już wcześniej opublikowane w [68]. To samo dotyczy opisu metody doboru struktury konwolucyjnej sieci neuronowej (podrozdział 4.3.) jak również opisu całościowej metody rozpoznawania obiektów w postaci uczestników ruchu drogowego za pomocą radaru FMCW, krótkoczasowej transformacji STFT oraz dedykowanej konwolucyjnej sieci neuronowej (podrozdziały 4.2. oraz 4.4). Dopełnienie metod symulacyjnych badaniami praktycznymi opublikowano częściowo w [69].

4.1. Metoda symulowania radaru, środowiska oraz obiektów badań

W podrozdziale opisano przyjęte w artykule modele radaru, pieszego, rowerzysty oraz samochodu. Na rysunku 4.1. przedstawiono widok symulowanej sytuacji, gdzie radar opromieniowuję określony obszar, w którym mogą występować poruszający się piesi, rowerzyści oraz samochody. Sytuacja taka może mieć miejsce na drogach, parkingach, lotniskach lub innych ważnych miejscach użyteczności publicznej, gdzie należy zadbać o bezpieczeństwo lub identyfikacje pojawiających się obiektów. Symulacje wykonano w środowisku Matlab.



Rys. 4.1. Symulowany model radaru oraz obiektów w środowisku Matlab [68].

Istotny jest przy tym sposób modelowania wybranego obiektu. Model poruszającego się człowieka opisany jest poprzez 16 segmentów ciała, tj. szyi i głowy, lewego i prawego barku, lewej i prawej kończyny górnej, lewego i prawego ramienia dolnego, lewego i prawego ramienia górnego, lewego i prawego biodra, lewej i prawej kończyny dolnej, lewej i prawej stopy, przy czym echo od pieszego zawiera składowe od każdego segmentu. Do parametrów ruchu pieszego przyjęto trójwymiarową lokalizację początkową, losowy wzrost pieszego, losową stałą prędkość i przemieszczania oraz zwrot kierunek przemieszczania się pieszego. Rysunek 4.2. przedstawia trójwymiarowy model pieszego.



Rys. 4.2. Symulowany model pieszego [68].

Model rowerzysty składa się z kierowcy i roweru. Stąd model rowerzysty zbudowany jest z 7 segmentów to jest z ramy roweru i rowerzysty, pedałów roweru, górnej i dolnej części nóg rowerzysty, tylnego i przedniego koło rowerzysty. Do parametrów rowerzysty przyjęto trójwymiarową lokalizację początkową, losową stałą prędkość przemieszczania, zwrot i kierunek przemieszczania się, liczbę szprych w kole roweru, stosunek obrotów koła do obrotów pedału oraz informacje o tym, czy rowerzysta pedałuje czy też nie pedałuje w czasie trwania symulacji. Rysunek 4.3. przedstawia trójwymiarowy model rowerzysty.



Rys. 4.3. Symulowany model rowerzysty [68].

Model samochodu został opisany jako obiekt będący określoną platformą w środowisku Matlab. Stąd do parametrów samochodu przyjęto trójwymiarową lokalizację początkową, losową stałą prędkość przemieszczania, zwrot i kierunek przemieszczania się oraz określoną wartość skutecznej powierzchni odbicia.

Do symulacji przyjęto radar FMCW pracujący na dwóch częstotliwościach tj. 24 GHz oraz 77 GHz. Radar nie zmieniał swojej lokalizacji, tak samo jak podczas eksperymentów praktycznych. Parametry poszczególnych modeli zostały zbiorczo przedstawione w tabeli 4.1.

Radar									
Częstotliwość pracy	24 GHz	77 GHz							
Maksymalny wykrywalny zasięg	100 m	100 m							
Rozdzielczość w odległości	0,04 m	0,5 m							
Maksymalna wykrywalna prędkość	11,11 m/s	11,11 m/s							
Czas przemiatania	3,667 ms	3,667 ms							
Maksymalna różnica częstotliwości	0,682 GHz	0,682 GHz							
Maksymalna częstotliwość Dopplera	1,778 kHz	5,703 kHz							
Pie	szy								
Początkowa lokalizacja	xε[0;50] yε[-1	0;10] z=[0, 0]							
Wzrost	1,5 – 2,0 m								
Prędkość przemieszczania się	0,1 – 1,0 m/s								
Kierunek przemieszczania się	0-360 stopni								
Rowerzysta									
Początkowa lokalizacja	xє[0;50] yє[-1	0;10] z=[0, 0]							
Liczba szprych w kole	1	8							
Prędkość przemieszczania się	1 – 10 m/s								
Kierunek przemieszczania się	0-360 stopni								
Stosunek obrotów koła do obrotów	0,5 – 6								
pedału									
Rowerzysta pedałuje czy nie	0 lub 1 (fałsz lub prawda)								
Samochód									
Początkowa lokalizacja	xε[0;50] yε[-1	0;10] z=[0, 0]							
Model ruchu	Ze stałą p	rędkością							
Prędkość przemieszczania się	1 – 1	0 m/s							
Skuteczna powierzchnia odbicia	10 m ²								

Tabela 4.1. Zestawienie parametrów symulowanych modeli [68].

W środowisku Matlab zostały utworzone skrypty odpowiadające za generację poszczególnych modeli. Otrzymane sygnały dla każdej modelowanej sytuacji w dziedzinie czasu przetransformowano w dziedzinę częstotliwości za pomocą STFT z odpowiednio długim oknem czasowym Kaisera. Diagram przepływu generowanych danych przedstawiono na poniższym rysunku 4.4.



Rys. 4.4. Diagram przepływu symulowanych danych [68].

4.2. Metoda generowania spektrogramów za pomocą radaru uRAD v1.2

Metoda generowania spektrogramów za pomocą radaru uRAD v1.2 została przedstawiona na rysunku 4.5. Radar FMCW opromieniowuję pojawiający się obiekt (lub obiekty), po czym odbity sygnał echa zostaje zapisany w pamięci komputera podłączonego do radaru za pomocą złącza USB. Następnie poprzez przetwarzanie

surowych danych (składowa I oraz Q) w sygnały zdolne do przetransformowania za pomocą STFT uzyskuje się spektrogram dla danego obiektu (obiektów).



Rys. 4.5. Schemat pozyskiwania danych z radaru i generowania spektrogramów [69]. Przedstawione na rysunku 4.5. pliki tekstowe I.txt oraz Q.txt zawierają surowe dane. Plik I.txt zawiera składową synfazowa I reprezentującą część rzeczywistą sygnału, zaś plik Q.txt zawiera składową kwadraturową Q reprezentującą część urojoną. Na rysunku 4.6. przedstawiono strukturę pliku I.txt.

Każdy wiersz oznacza echo pojedynczego CHIRP-u, czyli każdy wiersz oznacza jeden pełny okres emitowanego sygnału. Liczba wierszy zależy od czasu trwania pomiaru. Liczba kolumn oznacza liczbę próbek, które radar pobiera z echa w celu wyznaczenia odległości lub prędkości. Dwie ostatnie kolumny w każdym wierszu zawierają informacje o dacie i czasie odbioru sygnału. Do dalszego przetwarzania należało pozbyć się dwóch ostatnich kolumn z każdego pliku tekstowego. Wartość pojedynczej próbki oznacza wartość napięcia jakie zostało odebrane przez radar po przetworzeniu przez 12 bitowy przetwornik analogowo-cyfrowy. Przyjmując jeden

bit na określenie znaku napięcia, pozostałe 11 bitów pozwala określić wartość napięcia zapisanego w pliku dla każdej próbki w zakresie -1 [V] i 1 [V].

2165	2179	2185	2177	2189	2199	2174	2166	2199	2200	2176	2186	2169	.2182	2179	2024-07-10	10:12:37.695
2193	2200	2196	2184	2205	2206	2174	2170	2199	2199	2186	2191	2176	.2171	2184	2024-07-10	10:12:37.731
2177	2187	2193	2187	2193	2189	2172	2166	2189	2183	2172	2186	2170	.2179	2120	2024-07-10	10:12:37.761
2178	2198	2198	2180	2193	2199	2173	2172	2199	2200	2181	2185	2174	.2184	2178	2024-07-10	10:12:37.794
2181	2192	2194	2184	2198	2198	2176	2170	2200	2188	2168	2174	2159	.2173	2176	2024-07-10	10:12:37.821
2179	2188	2187	2180	2192	2184	2164	2170	2203	2196	2179	2186	2177	.2184	2192	2024-07-10	10:12:37.849
2177	2193	2197	2186	2204	2198	2171	2178	2200	2188	2174	2184	2176	.2180	2182	2024-07-10	10:12:37.875
2172	2190	2185	2171	2190	2186	2150	2152	2183	2181	2157	2168	2168	.2176	2179	2024-07-10	10:12:37.902
2179	2199	2200	2181	2195	2197	2164	2161	2189	2190	2168	2176	2166	.2169	2176	2024-07-10	10:12:37.928
2182	2192	2194	2187	2205	2194	2168	2167	2192	2188	2179	2181	2168	.2180	2190	2024-07-10	10:12:37.955
2182	2195	2190	2178	2202	2196	2162	2168	2200	2198	2181	2189	2182	.2184	2128	2024-07-10	10:12:37.982
2179	2190	2187	2181	2207	2203	2173	2175	2208	2202	2180	2188	2182	.2182	2184	2024-07-10	10:12:38.007
2179	2197	2191	2178	2197	2199	2166	2166	2197	2195	2176	2186	2186	.2194	2188	2024-07-10	10:12:38.035
2189	2199	2198	2186	2203	2198	2171	2172	2200	2199	2181	2187	2178	.2190	2182	2024-07-10	10:12:38.062
2194	2200	2197	2185	2196	2192	2170	2173	2203	2198	2183	2191	2174	.2187	2180	2024-07-10	10:12:38.090
2183	2189	2196	2186	2194	2192	2171	2170	2199	2197	2183	2193	2176	.2185	2180	2024-07-10	10:12:38.116
2167	2183	2189	2181	2187	2193	2168	2174	2204	2201	2176	2188	2180	.2183	2173	2024-07-10	10:12:38.142
2167	2179	2189	2176	2187	2195	2172	2172	2201	2201	2174	2181	2178	.2174	2167	2024-07-10	10:12:38.168
2173	2184	2191	2173	2195	2196	2173	2173	2207	2201	2183	2192	2182	.2176	2178	2024-07-10	10:12:38.196
2178	2182	2188	2181	2198	2196	2184	2180	2206	2205	2189	2188	2176	.2183	2190	2024-07-10	10:12:38.223
2169	2181	2191	2172	2186	2195	2174	2169	2192	2192	2170	2172	2166	.2181	2180	2024-07-10	10:12:38.250
2178	2184	2188	2172	2189	2192	2170	2159	2192	2192	2171	2179	2172	.2169	2177	2024-07-10	10:12:38.276
2177	2189	2194	2172	2188	2198	2169	2162	2193	2200	2178	2181	2173	.2179	2174	2024-07-10	10:12:38.303
2171	2191	2197	2173	2185	2194	2169	2169	2202	2201	2178	2186	2186	.2185	2173	2024-07-10	10:12:38.331
2175	2194	2199	2178	2191	2202	2185	2179	2213	2210	2190	2204	2193	.2188	2180	2024-07-10	10:12:38.359
2175	2185	2192	2174	2194	2194	2174	2170	2200	2194	2178	2186	2171	.2167	2169	2024-07-10	10:12:38.384

Rys. 4.6. Struktura pliku I.txt.

Dane zaimportowane do środowiska Matlab przyjmują postać macierzy o liczbie wierszy równej czasie trwania pomiaru oraz liczbie kolumn równej liczbę próbek, które radar pobiera z echa. Po usunięciu zbędnych kolumn z datą i czasem, normalizacji danych, połączono składową I oraz składową Q w zespolony sygnał. Następnie sygnał zespolony przetransformowano z dziedziny czasu w dziedzinę częstotliwości za pomocą STFT z oknem czasowym Kaisera dostosowanym do długości trwania jednego CHIRP-u. Przekształcenie sygnału do dziedziny częstotliwości wykonano oddzielnie dla każdego wiersza macierzy danych IQ poprzez wbudowane funkcję w Matlabie. W efekcie otrzymano spektrogram jak na rysunku 4.7.

Otrzymany spektrogram w dalszym kroku należy poddać analizie przez dedykowaną sieć neuronową. Utworzona wcześniej i wytrenowana sieć neuronowa CNN zdolna jest do rozpoznawania podanych na jej wejście spektrogramów. Utworzenie wcześniej wymaganych baz spektrogramów dla poszczególnych klas obiektów celem trenowania sieci zostanie omówione w kolejnym rozdziale. W wyniku działania sieci spektrogram zostaje sklasyfikowany do jednej z wielu klas. Struktura zastosowanej sieci CNN została przedstawiona w kolejnym podpunkcie.

65



Rys. 4.7. Zamiana surowych sygnałów I oraz Q na spektrogram za pomocą STFT dla pojedynczego chirp-u.

4.3. Metoda doboru struktury konwolucyjnej sieci neuronowej

Jednym z głównych zadań stanowiących wkład własny niniejszej rozprawy był wybór odpowiedniej struktury sieci neuronowej, a także jej hiperparametrów. W literaturze przedmiotu można znaleźć rozwiazania oparte o stosowanie gotowych sieci, takich jak chociażby te stworzone na potrzeby konkursu "ImageNet". Jednak te rozwiązania charakteryzują się tym, że są zoptymalizowane pod kątem dużych zadań. Zastosowanie zbyt skomplikowanej sieci do mniejszego zadania prowadzi do zwiększenia wymagań obliczeniowych, wydłużenia czasu treningu oraz wzrostu zapotrzebowania na pamięć, co może ograniczyć ich użycie na mniej wydajnych systemach. Ponadto, nadmiernie rozbudowana sieć może prowadzić do przetrenowania i obniżenia zdolności do uogólniania wyników. W związku z powyższym, sięgnięto po sprawdzone wcześniej rozwiązanie będące metodą doboru struktury i jej hiperparametrów w sposób ręczny czyli ewolucji jej struktury na podstawie kolejnych eksperymentów. Wadą takiego rozwiązania jest jego czasochłonność, ponieważ po każdej korekcie początkowych wartości hiperparametrów (np. współczynnik uczenia) należy przeprowadzić proces uczenia sieci od nowa. Jeżeli danymi wejściowymi są obrazy o dużych rozmiarach, to trening sieci może trwać od kilku godzin do nawet kilku dni.

W pracy został wykorzystany model CNN, który charakteryzuje się wielowarstwową (głęboką) strukturą. Model ten zawiera międzyobszarowe połączenia konwolucyjne w pierwszych warstwach, co pozwala na efektywną ekstrakcję cech z danych wejściowych. Pierwsza warstwa konwolucyjnej sieci neuronowej, zwana warstwą wejściową, jest początkowym elementem sieci, który otrzymuje dane wejściowe. Nie wykonuje ona żadnych operacji związanych z konwolucją ani aktywacją. Jej główną rolą jest przyjmowanie danych wejściowych i przekazywanie ich do kolejnych warstw sieci, zachowując ich oryginalne wymiary i cechy. To punkt startowy, od którego rozpoczyna się proces analizy danych przez sieć neuronową. W trakcie procesu uczenia zastosowano również warstwy normalizacji wsadowej, które mają na celu przyspieszenie procesu uczenia oraz stabilizację sieci [29]. Te elementy składają się na kompleksową architekturę modelu CNN, która umożliwia skuteczną klasyfikację danych wejściowych.

Opracowana głęboka sieć neuronowa w efekcie uczenia na dużym odpowiednio przygotowanym zbiorze danych wydobywa z nich odpowiednie cechy, poprzez kolejne

67

warstwy konwolucyjne, tworząc reprezentacje danych o coraz większej złożoności. Pierwsze warstwy mają za zadanie wykrywać bezpośrednio na obrazie prymitywne cechy takie jak elementarne kształty czy kolory. Kolejna warstwa konwolucyjna bazujac na tej reprezentacji ma za zadanie wykrywać bardziej złożone elementy. Natomiast ostatnie warstwy na podstawie reprezentacji generowanych przez warstwy poprzednie, mają za zadanie wykrywać cechy o najwyższym poziomie. Opisana realizacja ekstrakcji cech przez sieć neuronową warunkowana jest przez odpowiednio dobrane wartości jej hiperparametrów. Część hiperparametrów takich jak liczba neuronów, funkcje aktywacji, współczynnik uczenia czy liczba epok są stałe w całym procesie uczenia. Natomiast inne hiperparametry jak wartości wag są dostosowywane w procesie uczenia przy użyciu algorytmu SGDM (ang. Stochastic Gradient Descent with Momentum). Jest to algorytm, który wprowadza dodatkowy mechanizm o nazwie *momentum*, który pomaga przyspieszyć proces uczenia i sprawia, że zmiany wag są bardziej stabilne. Przyjęty model sieci potrafi dzięki temu analizować słabo przetworzone dane bez konieczności projektowania algorytmów ekstrakcji cech. Wadą rozwiązania jest zgromadzenie dużego zbioru uczącego.

Sieć CNN tworzona była od podstaw z zastosowaniem odpowiednich, wbudowanych funkcji w środowisku Matlab (tzw. toolboxy). Finalna struktura sieci, zdefiniowana po wielu eksperymentach wstępnych, zawierała cztery warstwy konwolucyjne:

- pierwsza warstwa 16 neuronów, filtr o polu recepcyjnym 5×5, przesunięcie (stride) 2×2, funkcja aktywacji ReLU, max pooling, warstwa normalizacji wsadowej (batch normalization);
- druga warstwa 32 neurony, pole recepcyjne filtru 5×5, przesunięcie (stride)
 2×2, funkcja aktywacji ReLU, max pooling, warstwa normalizacji wsadowej (batch normalization);
- trzecia warstwa 64 neuronów, pole recepcyjne filtru 5×5, przesunięcie (stride)
 2×2, funkcja aktywacji ReLU, max pooling, warstwa normalizacji wsadowej (batch normalization);
- czwarta warstwa 128 neuronów, pole recepcyjne filtru 5×5, przesunięcie (stride) 2×2, funkcja aktywacji ReLU, max pooling, warstwa normalizacji wsadowej (batch normalization).

Dodatkowo, po 4 warstwach konwolucyjnych dobrano warstwę spłaszczającą (flatten), jedną warstwę w pełni połączoną zawierająca 10 neuronów reprezentujących 10 rozpoznawanych klas obrazów (przypadek badań symulacyjnych), zaś dla badań praktycznych 6 neuronów reprezentujących 6 rozpoznawanych klas obrazów.

Podsumowując, w opracowanych modelach sieci konwolucyjnej występuję określona liczba parametrów:

• pierwsza warstwa konwolucyjna:

warstwa zawiera 16 filtrów o rozmiarze 5x5 i liczbie kanałów wejściowych równiej 3 (odpowiadająca liczbie kanałów w obrazach RGB). Liczba parametrów w tej warstwie jest obliczana jako:

$$5 \times 5 \times 3 \times 16 + 16 = 1216 \tag{4.1}$$

gdzie $10 \times 10 \times 3 \times 16$ to liczba wag filtrów, a 16 to liczba przesunięć (bias), które są dodawane do wyników konwolucji.

druga warstwa konwolucyjna:
 w tej warstwie zastosowano 32 filtry o rozmiarze 5x5 oraz 16 kanałów
 wejściowych. Liczba parametrów wynosi:

$$5 \times 5 \times 16 \times 32 + 32 = 12832 \tag{4.2}$$

• trzecia warstwa konwolucyjna:

w tej warstwie znajdują się 64 filtry o rozmiarze 5x5 oraz 32 kanały wejściowe. Liczba parametrów w tej warstwie jest obliczana jako:

$$5 \times 5 \times 32 \times 64 + 64 = 51264 \tag{4.3}$$

czwarta warstwa konwolucyjna:
 w tej warstwie zastosowano 128 filtrów o rozmiarze 5x5 oraz 64 kanały
 wejściowe. Liczba parametrów w tej warstwie jest obliczana jako:

$$5 \times 5 \times 64 \times 128 + 128 = 204928 \tag{4.4}$$

• warstwa w pełni połączona:

łączy ostatnią warstwę konwolucyjną z wyjściem klasyfikacyjnym, liczba neuronów wynosi 128, a liczba wyjść to 10 (badania symulacyjne) lub 6 (badania praktyczne). Liczba parametrów w tej warstwie wynosi:

$$128 \times 10 \times +128 = 1290 \tag{4.5}$$

Suma wszystkich parametrów wyniosła 270240. Są one aktualizowane podczas procesu uczenia sieci w celu zoptymalizowania jej wydajności w zadaniu klasyfikacji obrazów. Architektury zastosowanych w badaniach sieci zostały przedstawione na rysunkach 4.8. oraz 4.9.



Rys. 4.8. Struktura zaproponowanej głębokiej sieci neuronowej do badań symulacyjnych.



Rys. 4.9. Struktura zaproponowanej głębokiej sieci neuronowej do badań praktycznych.

4.4. Metoda rozpoznawania obiektów za pomocą radaru FMCW

Całościowa metoda, która realizuje najważniejszy cel niniejszej pracy czyli pokazanie nowego podejścia do detekcji i rozpoznawania obiektów za pomocą metod sztucznej inteligencji w radiolokacji została przedstawiona na rysunku 4.10.



Rys. 4.10. Schemat opracowanej metody klasyfikacji obiektów.
Radar FMCW opromieniowuje pojawiający się obiekt (lub obiekty), po czym sygnał echa zostaje zapisany w pamięci komputera podłączonego do radaru za pomocą złącza USB. Następnie poprzez obróbkę surowych danych (składowa I oraz Q) za pomocą transformacji STFT uzyskuje się spektrogram dla danego obiektu (obiektów). Po analizie obrazu jakim jest spektrogram przez dedykowaną sieć CNN następuje jego jednoznaczna klasyfikacja do właściwej klasy obiektów.

Rozdział 5

5. WALIDACJA EKSPERYMENTALNA

W poprzednich rozdziałach przedstawiono opis poszczególnych metod oraz wykorzystanych narzędzi do realizacji zasadniczego celu niniejszej rozprawy. Stąd, w tym rozdziale przedstawiono wyniki eksperymentów numerycznych oraz badań praktycznych, które mają dowieść tezy postawionej we wstępie tj. możliwe jest stworzenie zautomatyzowanej metody pozwalającej na detekcje i jednoznaczne rozpoznanie obiektów w postaci uczestników ruchu drogowego za pomocą radaru FMCW stosując krótkoczasową transformację Fouriera echa odebranego sygnału oraz dedykowaną strukturę konwolucyjnej sieci neuronowej jako klasyfikatora obiektów.

W rozdziale tym oprócz wyników klasyfikacji poszczególnych obiektów przedstawiono opis sposobów oceny działania sieci CNN, a także proces gromadzenia oraz etykietowania spektrogramów składających się na poszczególne bazy danych obrazów.

5.1. Baza danych zastosowana w eksperymentach numerycznych

Gromadzenie bazy danych do aplikacji opartych na metodach głębokiego uczenia jest procesem czasochłonnym. Jest to wniosek jaki płynie zarówno z analizy literatury dotyczącej sieci neuronowych ale przede wszystkim z przeglądu literatury zawartego w rozdziale 2. Stąd, podrozdział ten poświęcono omówieniu sposobu generowania spektrogramów tworzących 10 klas obiektów będących bazą danych wejściowych dla opracowanej struktury konwolucyjnej sieci neuronowej. W symulacyjnych eksperymentach numerycznych zastosowano 10 klas obiektów:

- 1. pojedynczy pieszy;
- 2. dwoje pieszych;
- 3. troje pieszych;
- 4. pojedynczy rowerzysta;
- 5. pojedynczy samochód;
- 6. pojedynczy pieszy oraz pojedynczy samochód;
- 7. pojedynczy pieszy oraz pojedynczy rowerzysta;
- 8. pojedynczy samochód oraz pojedynczy rowerzysta;
- 9. dwoje pieszych oraz pojedynczy samochód;
- 10. dwoje pieszych oraz pojedynczy rowerzysta.



Przykłady spektrogramów dla poszczególnych klas przedstawiono na rysunku 5.1.

Rys. 5.1. Zbiór klas zastosowanych w eksperymentach numerycznych.

Powyższe oznaczenia klas będą stosowane przy prezentacji wyników działania modelu klasyfikatora jakim jest dedykowana sieć CNN.

Spektrogramy wygenerowano poprzez metodę opisaną w podrozdziale 4.1. Implementacja powyższej metody w środowisku Matlab była czasochłonna, wykonywana etapami i finalnie sięgnęła po 1000 spektrogramów dla każdej klasy. Każda symulacja w każdej klasie obiektów trwała 10 sekund – generacja ruchu obiektu bądź zespołu obiektów. Natomiast proces tworzenia spektrogramu na podstawie 10 sekundowego ruchu trwał zdecydowanie dłużej. Pozostałe parametry symulacji są tożsame z tabelą 4.1. Ostateczna liczba spektrogramów okazała się niezbędna aby osiągnąć wysoką dokładność klasyfikacji. Podsumowując, baza danych obrazów zawierała po 1000 obrazów na każdą z 10 klas, każdy obraz o wymiarze 875 x 656 pikseli.

5.2. Analiza symulacyjna wyników eksperymentów numerycznych

Do eksperymentów numerycznych mających na celu określić za pomocą wskaźników omówionych w rozdziale pierwszym jakość klasyfikacji poszczególnych obiektów przyjęto następujący podział zbioru danych wejściowych: zbiór uczący stanowił 70 % obrazów, zbiór walidacyjny 15 % obrazów, zaś zbiór testowy również stanowił 15 % wszystkich obrazów, co jest zgodne z kształtowaniem liczności zbiorów opisanym w [70]. Generacja bazy danych spektrogramów jak i kolejne próby budowania ostatecznej struktury sieci dającej najwyższą dokładność klasyfikacji wykonano w środowisku Matlab (wersja R2023b, 23.2.0.2365128) oraz na laptopie Dell 13-sta generacja Intel(R) Core(TM) i5-1345U 1,60 GHz, 16 GB RAM, Windows 11 Pro.

5.2.1. Radar FMCW 77 GHz

W pierwszej kolejności na rysunku 5.2. przedstawiono przebieg krzywej uczenia dla najlepszego, uzyskanego przypadku. W symulacji przyjęto dla każdej klasy po 1000 obrazów (spektrogramów) oraz 10 epok.. Krzywa uczenia wyraźnie pokazuje, że w każdej epoce losowo zmieniano kolejność prezentowania obrazów z zestawu uczącego. Taka strategia umożliwiała lepsze dostosowanie wag modelu, co sprzyjało efektywniejszemu zbliżaniu się do optymalnego zestawu wag, minimalizującego błąd klasyfikacji. Ostateczna liczba epok została wybrana, gdy model osiągnął bardzo wysoką dokładność klasyfikacji dla każdej z klas.

77



Rys. 5.2. Krzywa uczenia sieci dla 10 epok, radar FMCW 77 GHz.

Następnie, w tabeli 5.1. dla tego samego przypadku została przedstawiona macierz pomyłek celem pokazania jak przyjęty model sieci CNN przewiduje klasy w stosunku do rzeczywistych wartości, pozwalając na bardziej szczegółową analizę wyników klasyfikacji.

Na przekątnej zawarte się liczby odpowiadające prawdziwie pozytywnym wynikom klasyfikacji. Zbiór testowy zawierał po 150 obrazów, stąd pojawienie się liczby 150 na przekątnej oznacza stu procentową dokładność klasyfikacji dla danej klasy obiektów. Po za przekątną znajdują się przypadki błędnego przypisania do klas. Najwięcej błędów model sieci popełniał myląc:

- pojedynczego pieszego z samochodem;
- dwoje pieszych i samochód z samymi pieszymi;
- pojedynczego pieszego z samochodem i rowerzystą jadącymi jednocześnie.

Zestawienie wszystkich wykonanych symulacji dla radaru FMCW pracującego na częstotliwości 77 GHz przedstawiono zbiorczo w tabeli 5.2. Przedstawione wyniki są najlepszymi otrzymanymi wartościami z wielu uzyskanych rezultatów. Są one efektem czasochłonnych symulacji, które różniły się wieloma parametrami takimi jak liczbą obrazów dla poszczególnych klas, liczbą epok czy różnymi wartościami hiperparamterów sieci neuronowej.

		Klasa predykowana									
		1	2	3	4	5	6	7	8	9	10
	1	150				26			12		
	2		131								
sta	3			146							
wi.	4				150						
(z)	5					124					
rze	6						143				
Isa	7							150			
Kla	8								138		
	9		19	4			7			150	
	10										150
Dokład	dność	0,976	0,978	0,967	1,000	0,973	0,975	1,000	0,962	0,980	1,000
Czułość		0,798	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,833	1,000
Swoistość		1,000	0,987	0,997	1,000	0,981	0,995	1,000	0,992	1,000	1,000
Precyzja		1,000	0,873	0,973	1,000	0,827	0,953	1,000	0,920	1,000	1,000

Tabela 5.1. Macierz pomyłek dla najlepszego przypadku– radar FMCW 77 GHz.

Tabela 5.2. Zestawienie wyników symulacji – radar FMCW 77 GHz.

	Liczba obrazów w klasie	Dokładność	
	50	33,23 %	
Symulacia 1	100	56,67 %	
Cymaiacja i	400	71,08 %	
	700	88,54 %	
	1000	94,22 %	
	Liczba epok	Dokładność	
Symulacja 2	3	88,54 %	
(10 klas obiektów	5	92,12 %	
po 1000 obrazów)	7	94,04 %	
	10	98,11 %	
Symulacia 2	Współczynnik uczenia	Czas uczenia	
(10 klas objektów	0,00001	2560 min	
po 1000 obrazów, 10 epok)	0,0001	1140 min	
· · · · · ·	0,001	450 min	

5.2.2. Radar FMCW 24 GHz

Dla radaru pracującego na częstotliwości 24 GHz do symulacji przyjęto 6 klas obiektów:

- 1. pojedynczy pieszy,
- 2. dwoje pieszych,
- 3. pojedynczy rowerzysta,
- 4. pojedynczy samochód,
- 5. pojedynczy pieszy oraz pojedynczy samochód,
- 6. pojedynczy pieszy oraz pojedynczy rowerzysta.

Przyczyną tego jest fakt, że badania praktyczne radarem uRAD 24 GHz również wykonano dla 6 klas. Zaś sama liczba 6 klas została przyjęta ze względu na najczęstsze występowanie akurat tych uczestników ruchu drogowego w wypadkach komunikacyjnych [71-73].

Podobnie jak we wcześniejszym podpunkcie w tabeli 5.3. pokazano macierz pomyłek dla najlepszego wariantu, czyli dla przypadku gdzie dla każdej z 6 klas wykorzystano po 1000 obrazów (spektrogramów), zaś proces uczenia obejmował 10 epok. Ponadto, w tabeli 5.4. zestawiono wyniki cząstkowe wykonanych symulacji prowadzących do otrzymania optymalnego rozwiązania będącego finalną strukturą sieci CNN przedstawioną podpunkcie 4.3.

				Klasa pre	dykowana		
		1	2	3	4	5	6
a	1	150					
wist	2		150				
czy	3			150			
ı rze	4				150		5
lase	5					131	
×	6					19	145
Dokła	dność	1,000	1,000	1,000	0,994	0,979	0,973
Czułość		1,000	1,000	1,000	0,967	1,000	0,884
Swoistość		1,000	1,000	1,000	1,000	0,975	0,993
Precyzja		1,000	1,000	1,000	1,000	0,873	0,967

Tabela 5.3. Macierz	pomyłek dla naj	jlepszego przy	/padku– radar F	FMCW 24 GHz.

Największą liczbę błędów model sieci popełniał myląc:

- pojedynczego pieszego z samochodem a pojedynczego pieszego z rowerzystą;
- pojedynczy samochód a pojedynczego pieszego z rowerzystą.

Tabela 5.4. Zestawienie wyników symulacji- radar FMCW 24 GHz.

	Liczba obrazów w klasie	Dokładność	
	50	35,17 %	
Symulacia 1	100	61,28 %	
• ,	400	72,76 %	
	700	89,36 %	
	1000	93,89 %	
	Liczba epok	Dokładność	
Symulacja 2	3	89,03 %	
(10 klas obiektów	5	93,01 %	
po 1000 obrazów)	7	94,35 %	
	10	98,46 %	
Symulacia 3	Współczynnik uczenia	Czas uczenia	
(10 klas obiektów	0,00001	2119 min	
po 1000 obrazów, 10 epok)	0,0001	1045 min	
• , • • •	0,001	398 min	

5.3. Baza danych zastosowana w eksperymentach praktycznych

Utworzenie bazy danych spektrogramów zrealizowano zgodnie z metodą opisaną w podrozdziale 4.2. W tym celu zestawiono stanowisko pomiarowe przedstawione na rysunku 5.3. Składało się ona oczywiście z radaru uRAD v1.2. zamontowanego na statywie i jednocześnie połączonego z laptopem poprzez złącze USB z dedykowanym oprogramowaniem. Obiektami pomiarowymi byli ludzie (dwoje mężczyzn o różnej budowie ciała), rowerzysta oraz samochód osobowy widoczny na rysunku 5.3. Obiekty poruszały się z różnymi prędkościami prostopadle do zbudowanego stanowiska pomiarowego w obu kierunkach. Badania wykonywano z wykorzystaniem drugiego trybu pracy radaru, czyli wykrywania odległości do

obiektów. W tym przypadku sygnałem sondującym był przebieg piłokształtny (Sawtooth).



Rys. 5.3. Stanowisko pomiarowe wykorzystane do zbadania opracowanej metody klasyfikacji obiektów.

W eksperymentach praktycznych zastosowano 6 klas obiektów:

- 1. pojedynczy pieszy;
- 2. dwoje pieszych;
- 3. pojedynczy rowerzysta;
- 4. pojedynczy samochód;
- 5. pojedynczy pieszy oraz pojedynczy samochód;
- 6. pojedynczy pieszy oraz pojedynczy rowerzysta;

Przykłady spektrogramów dla poszczególnych klas przedstawiono na rysunku 5.4.

Podczas realizacji badań wartość częstotliwości początkowej sygnału wynosiła 24,005 GHz. Szerokość pasma przemiatania ustawiono na 240 MHz. Liczba próbek pobieranych z sygnału echa ustawiono na 200 próbek. Radar miał możliwość wykrywania maksymalnie do 5 obiektów z odległości 100 metrów. Starano się aby czas trwania pomiaru w każdym przypadku oscylował w granicach 10 sekund. Dane otrzymane w czasie trwania pomiarów zapisano do plików tekstowych. Na podstawie tak zapisanych danych wykonano dalszą analizę sygnału oraz klasyfikację obiektów zgodnie z wspomnianą metodą zawartą w podrozdziale 4.4.





Dla każdego z sześciu przypadków wykonano serię kilkudziesięciu pomiarów. Aby uzyskać dużą liczbę obrazów wejściowych dla sieci neuronowej, a także biorąc pod uwagę czasochłonny proces pozyskiwania i etykietowania danych, zastosowano generowanie sztucznych, dodatkowych przykładów uczących. W tym celu zastosowano w środowisku Matlab transformacje geometryczne jak przesunięcia obrazu o określoną liczbę pikseli oraz skalowanie. Oprócz transformacji geometrycznych zastosowano również operacje na wartościach pikseli np. dodanie szumu, zmiana jasności i kontrastu. Powyższe zabiegi okazały się skutecznymi metodami zwiększenia zbioru uczącego Finalnie baza danych obrazów zawierała po 1000 obrazów dla każdej z 6 klas, każdy obraz o wymiarze 875 x 656 pikseli.

5.4. Wyniki eksperymentów praktycznych

Do eksperymentów praktycznych przyjęto następujący podział zbioru danych wejściowych: zbiór uczący stanowił 70 % obrazów, zbiór walidacyjny 15 % obrazów, 15 % zbiór testowy również stanowił wszystkich obrazów. zaś Obrazy do poszczególnych zbiór dobrano losowo. Augmentację bazy danych spektrogramów jak i kolejne próby budowania ostatecznej struktury sieci neuronowej dającej najwyższą dokładność klasyfikacji wykonano tak samo jak eksperymenty 23.2.0.2365128) numeryczne w środowisku Matlab (wersja R2023b, oraz z wykorzystaniem laptopa Dell 13-sta generacja Intel(R) Core(TM) i5-1345U 1,60 GHz, 16 GB RAM, Windows 11 Pro.

Podobnie jak wcześniej, w pierwszej kolejności w tabeli 5.5 pokazano macierz pomyłek dla najlepszego wariantu, czyli dla przypadku gdzie dla każdej z 6 klas wykorzystano po 1000 obrazów (spektrogramów), zaś proces uczenia obejmował 10 epok. Ponadto, w tabeli 5.6. zestawiono wyniki cząstkowe wykonanych symulacji prowadzących do otrzymania optymalnego rozwiązania będącego finalną strukturą sieci CNN przedstawioną podpunkcie 4.3.

				Klasa pre	dykowana		
		1	2	3	4	5	6
ŋ	1	139					
wist	2	11	150				
czy	3			150			
ı rze	4				150		6
Klasa	5					128	
	6					22	144
Dokła	dność	1,000	0,932	1,000	0,962	1,000	0,867
Czułość		0,988	0,987	1,000	0,993	0,976	0,969
Swoistość		0,985	1,000	1,000	1,000	0,972	0,992
Precyzja		0,927	1,000	1,000	1,000	0,853	0,960

Tabela 5.5. Macierz pomyłek dla najlepszego przypadku badań.

Największą liczbę błędów model sieci popełniał myląc:

• pojedynczego pieszego z dwojgiem pieszych;

- pojedynczy samochód z pojedynczym pieszym oraz pojedynczym rowerzystą;
- pojedynczego pieszego oraz pojedynczego rowerzystę z pojedynczym pieszym oraz pojedynczym samochodem.

Tabela 5.6. Zestawienie	wyników	badań	[69].
-------------------------	---------	-------	-------

	Liczba obrazów w klasie	Dokładność	
	50	31,76 %	
Symulacia 1	100	53,27 %	
ey maraoja 1	400	69,90 %	
	700	87,34 %	
	1000	94,12 %	
	Liczba epok	Dokładność	
Symulacja 2	3	87,24 %	
(10 klas obiektów	5	91,82 %	
po 1000 obrazów)	7	93,64 %	
	10	96,00 %	
Sumulacia 3	Współczynnik uczenia	Czas uczenia	
(10 klas objektów	0,00001	2378 min	
po 1000 obrazów, 10 epok)	0,0001	1096 min	
· , , , , , , , , , , , , , , , , , , ,	0,001	421 min	

5.5. Podsumowanie wyników eksperymentów

Wnioski płynące z otrzymanych wyników dotyczą przede wszystkim budowy struktury konwolucyjnej sieci neuronowej oraz jakości przetwarzanych spektrogramów. Stąd analizując wyniki przedstawione w tabelach 5.2., 5.4. oraz 5.6. w pierwszej kolejności omówiono przyczyny zastosowanej struktury konwolucyjnej sieci neuronowej, na którą składają się zasadniczo trzech kluczowe problemy:

- dobór właściwego ze względu na jakość i liczność zbioru danych;
- doboru właściwej liczby epok;
- doboru odpowiedniego współczynnika uczenia.

Rozmiar bazy danych, a przede wszystkim zbioru uczącego ma znaczenie. Im większy zbiór danych, tym bardziej zróżnicowane i reprezentatywne są obrazy, które model przetwarza podczas uczenia się. To pozwala na lepszą generalizację oraz mniejszy błąd predykcji. Większa liczba obrazów w każdej klasie oznacza większą różnorodność danych treningowych, co sprawia, że model ma większą szanse na naukę charakterystycznych cech różnych klas. Dzięki temu zmniejsza się ryzyko nadmiernego dopasowania modelu do specyficznych danych treningowych. Większy zbiór danych to przede wszystkim więcej informacji, na których model może bazować, co pozwala na bardziej precyzyjne przewidywanie wyników [74].

Badając wpływ liczby epok należy zauważyć, że dla małej liczby epok model nie przeanalizuje (przefiltruje) danych uczących wystarczająco wiele razy, a tym samym może nie nauczyć się odpowiednio rozpoznawać wzorców. Model pozostanie niedouczony, a jego dokładność będzie niska, gdyż nie zdążył optymalnie zaktualizować wag. Z drugiej strony, jeśli liczba epok jest zbyt duża, model może zacząć przeuczać się na dane treningowe, co oznacza, że zacznie zapamiętywać specyficzne przypadki i przestanie dobrze generalizować na nowe dane. W efekcie model osiąga świetne wyniki na danych treningowych, ale jego wydajność na zbiorze testowym (czy rzeczywistych danych) spada [75].

Podsumowując relację rozmiaru zbioru danych z liczbą epok należy stwierdzić, rozmiar zbioru danych i liczba epok są ściśle związane z jakością i dokładnością klasyfikacji w głębokich sieciach neuronowych. Duży i dobrze zróżnicowany zbiór danych w połączeniu z odpowiednią liczbą epok pozwala modelowi na naukę kluczowych wzorców i optymalizację swoich wag, co skutkuje wyższą dokładnością klasyfikacji oraz lepszą generalizacją wyników.

Współczynnik uczenia ma kluczowe znaczenie dla skuteczności trenowania sieci neuronowych. Gdy współczynnik uczenia przyjmuje wstępnie bardzo niską wartość, kroki w kierunku minimum funkcji straty są bardzo małe. W takim przypadku model potrzebuje znacznie więcej epok, aby dotrzeć do odpowiedniego minimum, co oznacza wydłużony czas treningu. Po za tym, jak wskazuje literatura, niski współczynnik uczenia może sprawić, że model utknie w jednym z lokalnych minimów funkcji straty, zamiast znaleźć globalne minimum. Powolne zmiany wag mogą być niewystarczające, aby wyjść z niekorzystnej lokalnej wartości. Z kolei zbyt duża wartość współczynnika uczenia powoduje krótszy czas uczenia. Po za tym model może nie być w stanie zbiec

86

do optymalnej wartości funkcji straty. Zamiast stopniowo minimalizować stratę, może oscylować wokół niej lub nawet powodować wzrost błędu, co prowadzi do niestabilności [7].

Bardzo istotnym aspektem jest jakość danych wejściowych użytych w modelu. Analizując wyniki symulacji, należy zauważyć, że model sieci CNN opierał się na spektrogramach wygenerowanych w niemal idealnych warunkach, gdzie wpływ szumów, niedoskonałości symulowanego radaru, modeli obiektów oraz torów propagacji fali elektromagnetycznej został zredukowany do minimum. Dzięki temu uzyskane spektrogramy charakteryzują się wysoką czytelnością i rozdzielczością, co sprawia, że różnice między poszczególnymi klasami są łatwo dostrzegalne gołym okiem.

Porównując zbiór obrazów wykorzystany w eksperymentach praktycznych ze zbiorem obrazów w symulacyjnych eksperymentach numerycznych, należy jednoznacznie stwierdzić, iż obrazy uzyskane na bazie realnych pomiarów za pomocą radaru uRAD są zdecydowanie mniej czytelne. Być może wynika to z budowy radaru i jego jakości na tle innych tego typu rozwiązań dostępnych na rynku. Albo z drugiej strony warto by było w przyszłości głębiej pochylić się na sposobem przetwarzania surowych sygnałów celem otrzymania przejrzystszych spektrogramów. Mimo to doprowadzenie do sytuacji, w której każda klasa zawierała po 1000 sztucznie uzyskanych obrazów pozwoliło na osiągnięcie wysokiej dokładności klasyfikacji.

5.6. Porównanie otrzymanych wyników badań z danymi literaturowymi

We wcześniejszych podpunktach pokazano różne warianty doboru hiparametrów konwolucyjnej sieci neuronowej prowadzących do uzyskania w efekcie czasochłonnych eksperymentów bardzo dobrych wyników klasyfikacji uczestników ruchu drogowego, które okazały się lepsze lub porównywalne z najlepszymi rezultatami prezentowanymi w literaturze światowej. Szczegółowe porównanie otrzymanych w rozprawie wyników na tle osiągnięć spotykanych w literaturze przedstawiono w tabeli 5.7.

87

Publikacja	Rozpoznawane obiekty	Dokładność	
Angelov [52] 2018	 piesi, rowerzysta, samochód 	88,60 %	
Joao [53] 2019	1. piesi	99,50 %	
Hadhrami [56] 2020	 pojedynczy pieszy, piesi, osoby czołgające się, biegacze, pojazd kołowy, ciężarówka 		96,56 %
Dadon [57] 2021	 piesi, zwierzęta 	93,62 %	
Buchman [58] 2022	1. piesi 2. zwierzęta	94,04%	
Rozprawa doktorska 2025	 Badania numeryczne: 1. pieszy; 2. dwoje pieszych; 3. troje pieszych; 4. rowerzysta; 5. samochód; 6. pieszy + samochód; 7. pieszy + rowerzysta; 8. samochód + rowerzysta; 9. dwoje pieszych + samochód; 10. dwoje pieszych + rowerzysta. 	 Badania praktyczne: 1. pieszy; 2. dwoje pieszych; 3. rowerzysta; 4. samochód; 5. pieszy + samochód; 6. pieszy + rowerzysta; 	96,00 %

Tabela 5.7. Porównanie dokładności różnych metod klasyfikacji obiektów.

Podobne wyniki do tych przedstawionych w niniejszej rozprawie osiągnęła metoda rozpoznawania aż 6 klas obiektów, opisana w pracy [56]. Niestety autorzy tej publikacji nie podali szczegółów dotyczących przetwarzania danych z dostępnej bazy RadEch. Niemniej jednak zastosowali zupełnie inne podejście niż w tej rozprawie, korzystając z metody transfer learning oraz klasyfikatora SVM.

Rozdział 6

6. PODSUMOWANIE I KIERUNKI DALSZYCH BADAŃ

Celem pracy było opracowanie metody detekcji i rozpoznawania obiektów za pomocą radaru z falą ciągłą o modulowanej częstotliwości, krótkoczasowej transformacji Fouriera a także dedykowanej struktury konwolucyjnej sieci neuronowej.

Oprócz realizacji powyższego celu omówiono zagadnienia dotyczące zastosowania sztucznej inteligencji w radiolokacji, twórczego wykorzystania zjawiska mikro-Dopplera wzbogacone przeglądem literatury oraz opisem zasad działania radaru FMCW oraz krótkoczasowej transformacji Fouriera. Przedstawiono również opis wypracowanych w ramach prowadzonych badań metod, które pozwoliły na weryfikację numeryczną oraz praktyczną postawionej we wstępie tezy. Finalnie, przedstawiono otrzymane wyniki eksperymentów numerycznych oraz badań praktycznych wykorzystując w tym celu radar FMCW uRAD pracujący na częstotliwości 24 GHz.

6.1. Wnioski

Na podstawie przeprowadzonych eksperymentów, a także szczegółowej analizy uzyskanych wyników zawartych w powyższym podsumowaniu można stwierdzić, że założony cel rozprawy doktorskiej został osiągnięty. Stąd, możliwe jest sformułowanie następujących wniosków:

- przedstawiona wiedza oraz charakterystyka dotycząca budowy i właściwości konwolucyjnych sieci neuronowych pozwala na jej twórcze zastosowanie jako klasyfikatora odpowiedzialnego za rozpoznanie obiektów w radiolokacji;
- 2. opracowana metoda detekcji i rozpoznawania obiektów za pomocą radaru z falą ciągłą o modulowanej częstotliwości oraz krótkoczasowej transformacji Fouriera, a także dedykowanej struktury konwolucyjnej sieci neuronowej pozwala na skuteczną klasyfikacje uczestników ruchu drogowego w postaci pieszych, rowerzystów oraz samochodów z dokładnością sięgająca 96 %;
- przeprowadzone badania przyczyniły się do pogłębienia zrozumienia zagadnień związanych ze sztuczną inteligencją, konwolucyjnymi sieciami neuronowymi, ze szczególnym uwzględnieniem ich zastosowania w obszarze radiolokacji. Uzyskane wyniki stanowią tym samym istotny wkład merytoryczny w rozwój tej dziedziny nauki;

- 4. w pracy przebadane zostały różne warianty doboru hiparametrów sieci neuronowych prowadzących do uzyskania w efekcie czasochłonnych eksperymentów bardzo dobrych wyników klasyfikacji uczestników ruchu drogowego, które okazały się lepsze lub porównywalne z najlepszymi rezultatami prezentowanymi w literaturze światowej;
- 5. niniejsza rozprawa wpisuje się w najnowsze trendy badań naukowych, ponieważ porusza pięć z piętnastu obszarów technologicznych określonych w "priorytetowych kierunkach badań naukowych w resorcie obrony narodowej w latach 2021-2035" sztuczną inteligencję, autonomię i autonomizację, analizę i przetwarzanie dużych zbiorów danych i zarządzanie nimi, sensory oraz systemy symulacyjne [9].

Podsumowując, za najważniejsze osiągnięcia rozprawy autor uważa:

 dokonanie szczegółowej charakterystyki konwolucyjnych sieci neuronowych, obejmującej architekturę, warstwy, metody wspomagające proces uczenia, a także miary jakości klasyfikacji;

 dokonanie przeglądu obszarów oraz metod zastosowania sztucznej inteligencji w radiolokacji, w tym głównego przeglądu oraz analizy porównawczej w zakresie metod rozpoznawania obiektów za pomocą konwolucyjnych sieci neuronowych;

zaproponowanie metody detekcji i rozpoznawania obiektów za pomocą radaru z falą
 ciągłą o modulowanej częstotliwości oraz krótkoczasowej transformacji Fouriera,
 a także dedykowanej struktury konwolucyjnej sieci neuronowej;

przeprowadzenie szeregu eksperymentów numerycznych dla sprawdzenia skuteczności zaproponowanej metody;

 przeprowadzenie badań praktycznych za pomocą radaru FMCW pracującego na częstotliwości 24 GHz celem weryfikacji skuteczności zaproponowanej metody.

6.2. Proponowane kierunki dalszych badań

Przedstawione w pracy zagadnienia są jedynie zalążkiem bardzo szerokiego spektrum zastosowania metod sztucznej inteligencji w radiolokacji, a w szczególności konwolucyjnych sieci neuronowych. Najlepszym dowodem świadczącym o aktualności podjętej tematyki jest chociażby lawinowy wzrost prac naukowych przedstawiony na wykresie zawartym w drugim rozdziale niniejszej rozprawy pokazujący zestawienie

92

publikacji dotyczących zastosowania głębokich sieci neuronowych w radiolokacji w latach 2016-2023.

Pomimo znaczących sukcesów głębokiego uczenia, wdrażanie systemów opartych na tych technikach, ich efektywne trenowanie oraz kompleksowa ocena jakości klasyfikacji wiążą się z wieloma wyzwaniami. Wśród najistotniejszych trudności można wymienić: wybór właściwego typu i struktury sieci neuronowej, ustalenie odpowiednich wartości hiperparametrów, ograniczony dostęp do precyzyjnie oznaczonych danych, złożoność oraz niejednoznaczność interpretacji działań wykonywanych przez sieci neuronowe, a także potencjalne błędy wynikające z niepoprawnej analizy danych wejściowych.

Są to problemy aktualne, których podjęcie może być dalszym kierunkiem badań autora. Niemniej jednak najbardziej palącym problemem czekającym na weryfikację autor uważa konieczność zweryfikowania zaproponowanej metody za pomocą radaru pracującego na częstotliwości 77 GHz.

Kolejnym zagadnieniem do rozwiązania pozostaje opracowanie aplikacji gotowej w czasie rzeczywistym przetwarzać surowe sygnały radarowe i rozpoznawać obiekty za pomocą przedstawionej metody. Optymalizacji na pewno podlegać powinien czas uczenia sieci, sięgający w niektórych przypadkach kilka dni. Warto w tym kierunku spróbować podejścia równoległego bądź najnowszych rozwiązań sprzętowych.

Bardzo ciekawym zagadnieniem i jednocześnie mającym szerokie pespektywy jest rozszerzenie opracowanej metody klasyfikacji obiektów w radiolokacji o zastosowanie algorytmów automatycznego doboru struktury głębokiej sieci neuronowej, w zależności od typu rozwiązywanego zadania, a także dostępnej bazy danych wejściowych. Algorytmy automatycznego doboru struktury głębokiej sieci neuronowej (ang. Neural Architecture Search NAS) to bardzo obiecujący kierunek. NAS jest procesem, który automatycznie znajduje optymalną architekturę sieci neuronowej dla konkretnego zadania, co może znacząco poprawić wydajność modelu bez potrzeby ręcznego dostrajania przez eksperta [76].

Jednym z popularnych rozwiązań typu NAS jest wykorzystanie uczenia przez wzmacnianie (ang. Reinforcement Learning RL), które traktuje architekturę sieci jako działanie, a jej wydajność jako nagrodę. Przykładem jest NASNet, który używa RL do optymalizacji struktury. Kolejnym podejściem są algorytmy ewolucyjne

93

(ang. Evolutionary Algorithms EA), które naśladują procesy biologicznej ewolucji, selekcjonując najlepsze architektury poprzez mechanizmy mutacji i krzyżowania. Z kolei metody oparte na gradientach, takie jak DARTS (ang. Differentiable Architecture Search), umożliwiają optymalizację architektur poprzez gradientową optymalizację, co pozwala na szybsze przeszukiwanie przestrzeni architektur [77].

W opinii autora, przeprowadzone eksperymenty, a także przedstawione wyniki oraz analizy dostarczyły przekonujących dowodów na pozytywną weryfikację postawionej tezy, iż "możliwe jest stworzenie zautomatyzowanej metody pozwalającej na detekcję i jednoznaczne rozpoznanie obiektów w postaci uczestników ruchu drogowego za pomocą radaru FMCW stosując krótkoczasową transformację Fouriera echa odebranego sygnału oraz dedykowaną strukturę konwolucyjnej sieci neuronowej jako klasyfikatora obiektów".

Bibliografia

- [1] Krizhevsky A., Sutskever I., Hinton G., *ImageNet Classification with Deep Convolutional,* Communications of the ACM vol. 60, s. 84 90, 2012.
- [2] Silver D. et. al., *Mastering the game of Go with deep neural networks and tree search*, Nature vol. 529, s. 484 489, 2016.
- [3] Dale R., *GPT-3: What's it good for?*, Natural Language Engineering, vol. 27, s. 113–118, 2021.
- [4] De Silva D., Mills N., El-Ayoubi M., Manic M., Alahakoon D., ChatGPT and Generative AI Guidelines for Addressing Academic Integrity and Augmenting Pre-Existing Chatbots, 2023 IEEE International Conference on Industrial Technology (ICIT), 2023.
- [5] Rupni U., Priya A., Hybrid Deep Learning Classification Model for Attention-Deficit-Hyperactivity Disorder using functional Magnetic Resonance Imaging, 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), 2023.
- [6] Yu H., Liu X., Tian Y., Wang Y., Gou C., Wang F., Sora-Based Parallel Vision for Smart Sensing of Intelligent Vehicles: From Foundation Models to Foundation Intelligence, IEEE Transactions on Intelligent Vehicles, vol. 9, s. 3123-3126, 2024.
- [7] Osowski S., *Sieci neuronowe do przetwarzania informacji*, Warszawa: OWPW, 2021.
- [8] Prakash S., Mali S., Ali F. A., Mohapatra S. K., Mohanty J., *A Self-Driving Vehicle Navigation System*, 2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT), 2023.
- [9] *Priorytetowe kierunki badań naukowych w resorcie obrony narodowej w latach 2021–2035*, Ministerstwo Obrony Narodowej, Warszawa 2023.
- [10] Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1993.
- [11] Tadeusiewicz R., Archipelag sztucznej inteligencji, Wydawnictwo Exit, 2021.
- [12] McCulloch W., Pitts W., A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics, vol. 5, s.115–133, 1943.
- [13] Neumann J., *First Draft of a Report on the EDVAC*, University of Pennsylvania, 1945.
- [14] Rosenblatt F., *The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain.* Psychological Review, vol. 65, s. 386–408, 1958.
- [15] Grzybowski A, Pietrzak K. David Hubel (1926-2013): the man who developed our understanding of vision. Neurol Sci., vol. 35, s. 919-921, 2014.
- [16] Minsky M., S. Papert., *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, USA, 1969.

- [17] Fukushima K., Neocognitron a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, vol. 36, s. 193–202. 1980.
- [18] LeCun Y., Bengio Y., Convolutional networks for images, speech, and timeseries, in Arbib M. A. (editor), The Handbook of Brain Theory and Neural Networks. MIT Press, Massachusetts, 1995.
- [19] *Image Classification on ImageNet*, https://paperswithcode.com/sota/imageclassification-on-imagenet.
- [20] Czekała Z., *Parada radarów*, Bellona, 2014.
- [21] Rypulak A., *Sensory obrazowe bezzałogowych statków powietrznych*, Lotnicza Akademia Wojskowa, 2023.
- [22] Abdu, F.J., Zhang, Y., Fu, M., Li, Y., Deng, Z., *Application of Deep Learning on Millimeter-Wave Radar Signals: A Review. Sensors*, *21*, 1951, 2021.
- [23] Chen V., *The Micro-Doppler Effect in Radar*, Artech House, 2019.
- [24] Metcalf J., Blunt S. D., Himed B., A machine learning approach to cognitive radar detection. In 2015 IEEE Radar Conference (RadarCon), s. 1405–1411, 2015.
- [25] Goodman N. A., Chapter 3 foundations of cognitive radar for next-generation radar systems. In R. Chellappa and S. Theodoridis, editors, Academic Press Library in Signal Processing, vol. 7, s. 153 – 195, 2018.
- [26] Abad R., Ierkic M., Ortiz-Rivera E., *Basic understanding of cognitive radar*, s.1–4, 2016.
- [27] Patel, K., Rambach K., Visentin T., Rusev D., Pfeiffer M., Yang B., Deep learning-based object classification on automotive radar spectra. In Proceedings of the 2019 IEEE Radar Conference (RadarConf), Boston, MA, USA, 2019.
- [28] Major B., Fontijne D., Ansari A., Teja R.; Gowaikar R.; Hamilton M.; Lee S., Grzechnik, S., Subramanian, S., Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seul, Korea, 2019.
- [29] Nowruzi F.E., Kolhatkar D., Kapoor P., Al Hassanat F., Heravi E.J., Laganiere R., Rebut J., Malik W., *Deep Open Space Segmentation using Automotive Radar.* In Proceedings of the 2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Linz, 2020.
- [30] Zhang X., Liang F., Chen X., *A Target Detection Method of Automotive Millimeter Wave Radar Based on Deep Learning*, 2023 8th International Conference on Communication, Image and Signal Processing (CCISP), Chengdu, China, 2023.
- [31] Servadei L., Sun H., Ott J., *Label-aware ranked loss for robust people counting using automotive in-cabin radar*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

- [32] Santra A., Hazra S., *Deep Learning Applications of Short Range Radars,* Artech House Publishers, 2020.
- [33] Sun Z., Ke Q., Rahmani H., Bennamoun M., Wang G. Liu J., *Human Action Recognition From Various Data Modalities: A Review*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, s. 3200-3225, 2023.
- [33] Shrestha A., Li H., Le J. Fioranelli F., *Continuous Human Activity Classification From FMCW Radar With Bi-LSTM Networks*. In IEEE Sensors Journal, vol. 20, s. 13607-13619, 2020.
- [34] Kim Y., Moon T., *Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks.* In IEEE Geoscience and Remote Sensing Letters, vol. 13, s. 8-12, 2016.
- [36] Molchanov P., Gupta S., Kim K., Pulli K. Multi-sensor system for driver's handgesture recognition. Proc. IEEE Int. Conf. Workshops Autom. Face Gesture Recognit., 2015.
- [37] Kim Y., Toomajian B., *Hand gesture recognition using micro-Doppler signatures with convolutional neural network*, IEEE Access, vol. 4, s. 7125–30, 2016.
- [38] Jokanovic B., Amin M., *Fall Detection Using Deep Learning in Range Doppler Radars*. IEEE Transactions on Aerospace and Electronic Systems, vol. 54, s.180–189, 2018.
- [39] Joy J., John A. T., Anna Alex A., Nair A. S., Sreesh P. R., Manuel A., Elderly Fall Detection System Using mm-Wave Radar Sensor. 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2024.
- [40] Ma L., Wu J., Zhang J., Wu Z., Jeon G., Zhang Y., Wu T., Research on sea clutter reflectivity using deep learning model in industry 4.0. IEEE Trans. Ind. Informat., vol. 16, s. 5929–5937, 2020.
- [41] Cifola L., Harmanny R., *Target/clutter disentanglement using deep adversarial training on micro-Doppler signatures*. In Proc. 16th Eur. Radar Conf. (EuRAD), 2019.
- [42] Lepetit P., Ly C., Barthes L., Mallet C., Viltard N., Lemaitre Y., Rottner L., Using deep learning for restoration of precipitation echoes in radar data. IEEE Trans. Geosci. Remote Sens., early access, 2021.
- [43] Hong S.-J., Yi Y.-G., Jo J., Seo B.-S., 'Classification of radar signals with convolutional neural networks. In Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN), 2018.
- [44] Lee G.-H., Jo J., Park C. H., Jamming prediction for radar signals using machine learning methods. Secur. Commun. Netw., 2020.
- [45] Walenczykowska M., Kawalec A., Application of Continuous Wavelet Transform and Artificial Naural Network for Automatic Radar Signal Recognition. Sensors, 22, 7434, 2022. https://doi.org/10.3390/s22197434

- [46] Walenczykowska M., Kawalec A., Krenc K., *An Application of Analytic Wavelet Transform and Convolutional Neural Network for Radar Intrapulse Modulation Recognition.* Sensors, 23, 1986, 2023. https://doi.org/ 10.3390/s23041986.
- [47] Orduyilmaz A., Yar E., Kocamis M. B., Serin M., Efe M., *Machine learning-based radar waveform classification for cognitive EW*. Signal, Image Video Process., vol. 15, s. 1–10, 2021.
- [48] Pan Z., Wang S., Zhu M., Li Y., Automatic waveform recognition of overlapping LPI radar signals based on multi-instance multilabel learning. IEEE Signal Process. Lett., vol. 27, s. 1275–1279, 2020.
- [49] Guo J., Wang L., Zhu D., Hu C., *Compact convolutional autoencoder for SAR target recognition*. IET Radar, Sonar Navigat., vol. 14, s. 967–972, 2020.
- [50] Guo Y., Du L., Wei D., Li C., Robust SAR automatic target recognition via adversarial learning. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, s. 716–729, 2021.
- [51] Ding J., Chen B., Liu H., Huang M., *Convolutional neural network with data augmentation for SAR target recognition*. IEEE Geosci. Remote Sens. Lett., vol. 13, s. 364–368, 2016.
- [52] Angelov, A., Robertson, A., Murray-Smith, R., Fioranelli, F., *Practical classification of different moving targets using automotive radar and deep neural networks*. IET RadarSonar Navig. 12, 1082–1089, 2018.
- [53] Joao, S., Zimmer, A., Brandmeier, T., Freire, R., *Pedestrian recognition using micro Doppler effects of radar signals based on machine learning and multiobjective optimization.* Expert Systems with Applications. 136, 304-315, 2019.
- [54] Pfaltz, J. L., Sequential operations in digital picture processing. Journal of the ACM, 13(4), 471-494, 1966.
- [55] MacQueen, J., Some Methods for Classification and Analysis of MultiVariate Observations. Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, 281-297, 1967.
- [56] Hadhrami, E. A., Mufti, M. A., Taha, B., Werghi, N., Ground Moving Radar Targets Classification Based on Spectrogram Images Using Convolutional Neural Networks. Proceedings of the 19th International Radar Symposium (IRS), Bonn, Germany, pp. 1-9, 2018.
- [57] Dadon, Y. D., Yamin, S., Feintuch, S., Permuter, H. H., Bilik, I., Taberkian, J., Moving Target Classification Based on micro-Doppler Signatures Via Deep Learning. Proceedings of the IEEE Radar Conference (RadarConf21), Atlanta, GA, USA,1-6, 2021.
- [58] Buchman, D., Drozdov, M., Krilavičius, T., Maskeliūnas, R., Damaševičius, R., *Pedestrian and Animal Recognition Using Doppler Radar Signature and Deep Learning*. Sensors, 22, 3456, 2022.
- [59] Pietrasiński, J., *Wybrane problemy przetwarzania sygnałów radarowych*, WAT, Warszawa, 2015.
- [60] Skolnik, M. Radar Handbook, Third Edition, McGraw-Hill, New York, USA, 2008.

- [61] Mahafza, B. R., Elsherbeni, A. Z. *Matlab Simulations for Radar Systems Design*, New York, USA, 2004.
- [62] Rypulak, A., *Sensory obrazowe bezzałogowych statków powietrznych*, Lotnicza Akademia Wojskowa, Dęblin, Polska, 2023.
- [63] Jankiraman, M., *FMCW Radar Design*, Artech Hause, Massachusetts, USA, 2018.
- [64] Richards, M. A., *Fundamentals of Radar Signal Processing*, McGraw-Hill Professional, 2014.
- [65] Budge, M., German, S., *Basic Radar Analysis*, Second Edition, Artech Hause, Massachusetts, USA, 2020.
- [66] uRAD by Anteral, *uRAD USB v1.2 24 GHz radar, velocity and 1D positioning Datasheet*, uRAD by Anteral, 2022.
- [67] Zieliński, T., *Cyfrowe przetwarzanie sygnałów od teorii do zastosowań*, WKŁ, Warszawa, 2005.
- [68] Ślesicki, B., Ślesicka, A., A New Method for Traffic Participant Recognition Using Doppler Radar Signature and Convolutional Neural Networks. Sensors, 24, 3832, 2024.
- [69] Ślesicki, B., Ślesicka, A., Kawalec, A., Walenczykowska, M., Improving Recognition of Road Users via Doppler Radar Data and Deep Learning Convolutional Networks. Electronics 13, 4070, 2024.
- [70] Dobrowolski, A., *Transformacje sygnałów od teorii do praktyki*, Wydawnictwo BTC, Legionowo, 2018.
- [71] Wang, W., Zhang, W., Guo H. *A safety-based behavioral approaching model with various driving characteristics*. Transportation Research Part C-Emerging Technologies, vol. 19, no. 6, pp. 1202–1214, 2011.
- [72] Gandhi, T., Trivedi, M. M., Pedestrian protection systems: issues, survey, and challenges. IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 3, pp. 413–430, 2007.
- [73] Guo, L., Ge, P.-S., Zhang, M.-H., Li, L.-H., Zhao, Y.-B. Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine. Expert Systems with Applications, vol. 39, no. 4, pp. 4274– 4286, 2012.
- [74] Goodfellow, I., Bengio, Y., Courville, A. Deep Learning, MIT Press, 2016.
- [75] Cubuk, E. D., Zoph, B. Mane D., *AutoAugment: Learning Augmentation Policies from Data*, arXiv: 1805,09501, 2019.
- [76] Kwasigroch, A., Grochowski, M., Mikołajczyk, A. *Neural Architecture Search for Skin Lesion Classification*, IEEE Access, t. 8, s. 9061–9071, 2020.
- [77] Kwasigroch, A., Grochowski, M., Mikolajczyk, M. Deep Neural Network Architecture Search Using Network Morphism, Proceedings of 24th

International Conference on Methods and Models in Automation and Robotics (MMAR), s. 30–35, 2019.

Spis rysunków

- 1.1. Kompleksowy podział metod całościowych sztucznej inteligencji [11].
- Porównanie dokładności najlepszych modeli sieci CNN na przestrzeni lat 2016-2024 w konkursie ImageNet [19].
- 1.3. Matematyczny model neuronu [10].
- 1.4. Przebiegi wymienionych funkcji aktywacji.
- 1.5. Schematyczna budowa wielowarstwowej sieci neuronowej.
- 1.6. Przykładowa operacja splotu w CNN.
- 1.7. Warstwy grupujące w CNN.
- 1.8. Przykładowa struktura konwolucyjnej sieci neuronowej.
- 2.1. Stosowane reprezentacje danych radarowych [22].
- 2.2. Efekt zjawiska mikro-Dopplera na przykładzie pływaka [23].
- 2.3. Zestawienie publikacji dotyczących zastosowania głębokich sieci neuronowych w radiolokacji w latach 2014-2024.
- 2.4. Struktury sieci neuronowych: I uproszczona sieć VGG16, II sieć ResNet-50 oraz III - połączenie CNN z LSTM [51].
- 2.5. Uzasadnienie wyboru metody SVM w badaniach [52].
- 2.6. Struktura sieci CNN w artykule [57].
- 2.7. Metody zwiększania zbioru danych treningowych w artykule [57]. A – obraz oryginalny, B – losowe przesunięcie częstotliwości, C - losowe przesunięcie w czasie, D – odwrócenie w poziomie, E – odwrócenie w pionie, F – dodanie szumów.
- 2.8. Struktura sieci CNN w artykule [58].
- 3.1. Schemat blokowy radaru FMCW [62].
- 3.2. Modulacja częstotliwości na przykładzie przebiegu piłokształtnego [62].
- 3.3. Radar uRad USB v1.2 wykorzystany do badań [64].
- 3.4. Widok graficznego interfejsu użytkownika oprogramowania radaru uRAD USB v1.2. [64].
- 3.5. Przykład zastosowania transformacji STFT.
- 3.6. Przykład zastosowania transformacji STFT dla różnej długości ramki.
- 4.1. Symulowany model radaru oraz obiektów w środowisku Matlab [68].
- 4.2. Symulowany model pieszego [68].
- 4.3. Symulowany model rowerzysty [68].

- 4.4. Diagram przepływu symulowanych danych [68].
- 4.5. Schemat pozyskiwania danych z radaru i generowania spektrogramów [69].
- 4.6. Struktura pliku I.txt.
- 4.7. Zamiana surowych sygnałów I oraz Q na spektrogram za pomocą STFT.
- 4.8. Struktura zaproponowanej głębokiej sieci neuronowej do badań symulacyjnych.
- 4.9. Struktura zaproponowanej głębokiej sieci neuronowej do badań praktycznych.
- 4.10. Schemat opracowanej metody klasyfikacji obiektów.
- 5.1. Zbiór klas zastosowanych w eksperymentach numerycznych.
- 5.2. Krzywa uczenia sieci dla 10 epok, radar FMCW 77 GHz.
- 5.3. Stanowisko pomiarowe wykorzystane do zbadania opracowanej metody klasyfikacji obiektów.
- 5.4. Zbiór klas zastosowanych w eksperymentach praktycznych.

Spis tabel

- 1.1. Macierz pomyłek stosowana do klasyfikacji dwuklasowej.
- 1.2. Macierz pomyłek stosowana do klasyfikacji wieloklasowej.
- 2.1. Zbiorcze zestawienie zastosowania metod sztucznej inteligencji do wykrywania obiektów w radiolokacji.
- 3.1. Zestawienie właściwości radaru uRAD USB v1.2.
- 4.1. Zestawienie parametrów symulowanych modeli [68].
- 5.1. Macierz pomyłek dla najlepszego przypadku– radar FMCW 77 GHz.
- 5.2. Zestawienie wyników symulacji radar FMCW 77 GHz.
- 5.3. Macierz pomyłek dla najlepszego przypadku– radar FMCW 24 GHz.
- 5.4. Zestawienie wyników symulacji- radar FMCW 24 GHz.
- 5.5. Macierz pomyłek dla najlepszego przypadku badań.
- 5.6. Zestawienie wyników badań [69].
- 5.7. Porównanie dokładności różnych metod klasyfikacji obiektów.

Wykaz najważniejszych publikacji autora rozprawy

- [1] **Ślesicki, B.**, Kawalec, A., Ślesicka, A. *The Study of the Possibility of Applying Parallel Programming to the Algorithms of Space-Time Adaptive Processing.* Problems of Mechatronics. Armament, Aviation, Safety Engineering, 13 (3): 27-42, 2022. **70 pkt.**
- [2] Ślesicka, A., Kawalec, A., **Ślesicki, B**. *Capon-like method for direction of arrival estimation using TDM MIMO RADAR*. Metrology And Measurement Systems 2022; Vol. 30, No. 3. 2023. **100 pkt.**
- [3] Kawalec, A., Ślesicka, A., **Ślesicki, B**. A New Statistical Method for Determining the Clutter Covariance Matrix in Spatial–Temporal Adaptive Processing of a Radar Signal. Sensors, 23, 4280, 2023. **100 pkt.**
- [4] Ślesicka, A., Ślesicki, B. The concept of disrupting anti-radiation missiles in a radar decoy system. Aviation and Security Issues, 3(1), 167–181, 2023.
 100 pkt.
- [5] Ślesicki, B., Ślesicka, A., Kawalec, A. Improve the safety of air transport, especially in militarized terrain, by use of side looking airborne radar and space time adaptive processing. Scientific Journal of Silesian University of Technology. Series Transport, 123, 335-346, 2024. 100 pkt.
- [6] Ślesicka, A., **Ślesicki, B**., Iwański, J., Truskowski, A. *Application of space-time adaptive signal processing in radiolocation*. Aviation and Security Issues, 4(2), 2024. **100 pkt.**
- [7] **Ślesicki, B**., Ślesicka, A., A New Method for Traffic Participant Recognition Using Doppler Radar Signature and Convolutional Neural Networks. Sensors, 24, 3832, 2024. **100 pkt.**
- [8] **Ślesicki, B**., Ślesicka, A., Kawalec, A., Walenczykowska, M., *Improving Recognition of Road Users via Doppler Radar Data and Deep Learning Convolutional Networks*. Electronics 13, 4070, 2024. **100 pkt**.

Kody źródłowe zastosowanych programów i funkcji

Generowanie echa sygnału odbitego od pieszego/pieszych

```
function [WynikDlaPieszych] = Piesi(npulse, tm, c,fc, fs, txWave,
yLocLimit, xLocLimit,LiczbaPieszych,OkresProbkowania,posr,velr)
xPedRec = complex(zeros(round(fs*tm),npulse));
WynikDlaPieszych =
complex(zeros(size(xPedRec,1),size(xPedRec,2),LiczbaPieszych));
%% Generowanie sygnałów dla pieszych
for ii = 1:LiczbaPieszych
    % Parametry pieszego, położenie, wzrost, prędkość poruszania się
   Lokalizacja = [xLocLimit(1) + (xLocLimit(2)-xLocLimit(1))*rand;
   yLocLimit(1) + (yLocLimit(2)-yLocLimit(1))*rand; 0];
   Wzrost = 1.5 + (2-1.5)*rand;
   Predkosc = rand^{*}(0.6)+0.1;
   Kierunek = -180 + 360*rand;
   % Wbudowany moduł działania modelu pieszego
   Pieszy = backscatterPedestrian(...
                'InitialPosition',Lokalizacja,...
                'InitialHeading',Kierunek,...
                'PropagationSpeed', c,...
                'OperatingFrequency', fc,...
                'Height',Wzrost,...
                'WalkingSpeed',Predkosc);
   % Kanał propagacji sygnału
   channel ped = phased.FreeSpace(...
                'PropagationSpeed', c,...
                'OperatingFrequency', fc,...
                'TwoWayPropagation', true,...
                'SampleRate',fs);
   % Symulacja modelu pieszego, nadawanie ciągu impulsów przez radar
   for m = 1:npulse
     [posPed,velPed,axPed] = move(Pieszy,OkresProbkowania,Kierunek);
     [~,angrPed] = rangeangle(posr,posPed,axPed);
     xPedCh =
     channel ped(repmat(txWave,1,size(posPed,2)),posr,posPed,velr,
     velPed);
     xPed = reflect(Pieszy,xPedCh,angrPed);
     xPedRec(:,m) = xPed;
   end
WynikDlaPieszych(:,:,ii) = conj(dechirp(xPedRec,txWave));
end
```

Generowanie echa sygnału odbitego od rowerzysty

```
function [WynikDlaRowerzystow] = Rowerzysci(npulse, tm, c,fc, fs,
txWave, yLocLimit,
LocLimit, LiczbaRowerzystow, OkresProbkowania, posr, velr)
xBicRec = complex(zeros(round(fs*tm),npulse));
WynikDlaRowerzystow =
complex(zeros(size(xBicRec,1),size(xBicRec,2),LiczbaRowerzystow));
%% Generowanie sygnałów dla rowerzystów
for ii = 1:LiczbaRowerzystow
   Lokalizacja = [xLocLimit(1) + (xLocLimit(2)-xLocLimit(1))*rand;
   yLocLimit(1) + (yLocLimit(2)-yLocLimit(1))*rand; 0];
   Predkosc = 1 + (10-1)*rand;
   Kierunek = -180 + 360*rand;
   Stosunek obrotow kola do pedalu = 0.5 + (6-0.5)*rand;
   LiczbaSzprych = 36;
   Coast = rand<0.5;</pre>
   Rowerzysta =
   backscatterBicyclist('InitialPosition',Lokalizacja,...
                'InitialHeading',Kierunek,...
                'Speed', Predkosc, ...
                'PropagationSpeed', c,...
                'OperatingFrequency', fc,...
   'GearTransmissionRatio', Stosunek_obrotow_kola_do_pedalu,...
                'NumWheelSpokes',LiczbaSzprych,...
                'Coast',Coast);
   channel bic = phased.FreeSpace('PropagationSpeed', c,...
                   'OperatingFrequency', fc,...
                   'TwoWayPropagation', true,...
                   'SampleRate',fs);
   for m = 1:npulse
        [posBic,velBic,axBic] =
       move(Rowerzysta,OkresProbkowania,Kierunek);
        [~,angrBic] = rangeangle(posr,posBic,axBic);
       xBicCh =
        channel bic(repmat(txWave,1,size(posBic,2)),posr,posBic,velr,
       velBic);
       xBic = reflect(Rowerzysta,xBicCh,angrBic);
       xBicRec(:,m) = xBic;
   end
   WynikDlaRowerzystow(:,:,ii) = conj(dechirp(xBicRec,txWave));
end
```
Generowanie echa sygnału odbitego od samochodu

```
function [WynikDlaSamochodow] = Samochody(npulse, tm, c,fc, fs,
txWave, yLocLimit,
xLocLimit,OkresProbkowania,posr,velr,MaxPredkoscSamochodu,LiczbaSamo
chodow)
xCarRec = complex(zeros(round(fs*tm),npulse));
WynikDlaSamochodow =
complex(zeros(size(xCarRec,1),size(xCarRec,2),LiczbaSamochodow));
%% Generowanie sygnałów dla samochodów
 for ii = 1:LiczbaSamochodow
 Lokalizacja = [xLocLimit(1) + (xLocLimit(2)-xLocLimit(1))*rand;
                yLocLimit(1) + (yLocLimit(2)-yLocLimit(1))*rand;
               01
 Predkosc = [-MaxPredkoscSamochodu + (MaxPredkoscSamochodu +
  MaxPredkoscSamochodu)*rand; -MaxPredkoscSamochodu
  +(MaxPredkoscSamochodu+MaxPredkoscSamochodu)*rand; 0]
 Samochod = phased.Platform('InitialPosition',Lokalizacja,...
                                'Velocity', Predkosc,...
                                'OrientationAxesOutputPort',true);
 phased.RadarTarget('PropagationSpeed', c, 'OperatingFrequency', fc,
  'MeanRCS',10);
 chan car = phased.FreeSpace('PropagationSpeed', c,...
            'OperatingFrequency',fc,...
             'TwoWayPropagation', true, 'SampleRate', fs);
   for m = 1:npulse
     [posCar,velCar,~] = Samochod(OkresProbkowania);
      xCarCh = chan car(repmat(txWave,1,size(posCar,2)),...
      posr,posCar,velr,velCar);
      xCar = carTgt(xCarCh);
      xCarRec(:,m) = xCar;
   end
   WynikDlaSamochodow(:,:,ii) = conj(dechirp(xCarRec,txWave));
 end
end
```

Definicja funkcji wyznaczającej spektrogram STFT

```
function [S,T,F] = Sygnatury(RodzajObiektu,OkresProbkowania)
M = 200;
beta = 6;
w = kaiser(M,beta);
R = floor(1.7*(M-1)/(beta+1));
noverlap = M-R;
[S, F, T] =
stft(squeeze(sum(RodzajObiektu,1)),1/OkresProbkowania,'Window',w,'FF
TLength',M*2,'OverlapLength',noverlap);
S = przetwarzanie(S);
end
function S = przetwarzanie(S)
    S = 10*log10(abs(S));
    for ii = 1:size(S,3)
        zs = S(:,:,ii);
        zs = (zs - min(zs(:)))/(max(zs(:))-min(zs(:)));
        S(:,:,ii) = zs;
    end
end
```

Generowanie spektrogramów STFT dla danych z radaru uRAD

```
f0=24.004e9;
c=3e8;
% Wczytanie danych ze wskazanego pliku CW I oraz CW Q
path = uigetdir;
filename Idata=[path '\I.txt'];
uRad Data I = readtable(filename Idata
,'FileType','text','Delimiter',' ');
filename Odata=[path '\0.txt'];
uRad Data Q = readtable(filename_Qdata
,'FileType','text','Delimiter',' ');
% Odczyt parametru Ns - ilość próbek w ramce
[Rump num,Ns]=size(uRad Data Q);
Ns=Ns-2;
uRad Data I(:,Ns+1)=[];
uRad Data Q(:,Ns+1)=[];
time I=uRad Data I(:,Ns+1);
uRad Data I(:,Ns+1)=[];
time Q=uRad Data Q(:,Ns+1);
uRad Data Q(:,Ns+1)=[];
% Przekształecenie danych z tabeli do macierzy
uRad Data I = table2array(uRad Data I);
uRad Data Q = table2array(uRad Data Q);
% Wygenerowanie wektora czasu
time = linspace(0, length(uRad Data I)/f0, length(uRad Data I));
% Wvznaczenie STFT
MaxPredkosc = 78.1;
DlugoscFali = c/fo;
oversamplingFactor = 1.11;
fmax = (2*MaxPredkosc/DlugoscFali)*oversamplingFactor;
OkresProbkowania = 1/(2*fmax);
Sygnatury(sig data,OkresProbkowania);
figure(1) imagesc(T sig_data,F sig_data,S sig data (:,:,1));
folderpath = strcat('C:\Dane\uRAD\');
filenameindex = 1;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(1),filename)
```

Program do generacji spektrogramów dla poszczególnych klas obiektów

```
%% Parametry radaru
c = 3e8;
fc = 24e9; %
OkresPowtarzaniaImpulsow = 1e-6;
PozycjaRadaru = [0;0;0];
PrekoscRadaru = [0;0;0];
Radar =
phased.Platform('InitialPosition', PozycjaRadaru, 'Velocity', PrekoscRa
daru,'OrientationAxesOutputPort',true);
SzerokoscPasma = 250e6;
CzestotliwoscProbkowania = SzerokoscPasma*2;
wav =
phased.FMCWWaveform('SampleRate',CzestotliwoscProbkowania,'SweepTime
 ,OkresPowtarzaniaImpulsow, 'SweepBandwidth', SzerokoscPasma);
Nadajnik = phased.Transmitter('PeakPower',5,'Gain',25);
txWave = Nadajnik(wav());
%% Ustawienia symulacji
MaxPredkoscRowerzysty = 3;
MaxPredkoscSamochodu = 10;
DlugoscFali = c/fc;
oversamplingFactor = 1.11;
fmax = (2*MaxPredkoscRowerzysty/DlugoscFali)*oversamplingFactor;
OkresProbkowania = 1/(2*fmax);
CzasTrwaniaSymulacji = 10;
LiczbaImpulsow = floor(CzasTrwaniaSymulacji/OkresProbkowania);
[posr,velr,~] = Radar(OkresProbkowania);
%% Obszar obserwowanego terenu
Yobszar = [-10, 10];
Xobszar = [5, 45];
for i=1:1000 %ustalenie ilości obrazów
%% jeden pieszy
   LiczbaPieszych = 1;
   WynikDlaPieszych = Piesi(LiczbaImpulsow,
   OkresPowtarzaniaImpulsow, c,fc, CzestotliwoscProbkowania, txWave,
   Yobszar, Xobszar, LiczbaPieszych, OkresProbkowania, posr, velr);
   [SPed,T,F] = Sygnatury(WynikDlaPieszych,OkresProbkowania);
   figure(i)
   imagesc(T,F,SPed(:,:,1));
   folderpath = strcat('C:\Dane\Pedestrian\');
   filenameindex = i;
   filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
   saveas(figure(i),filename)
```

%% jeden rowerzysta

```
LiczbaRowerzystow = 1;
WynikDlaRowerzystow = Rowerzysci(LiczbaImpulsow,
OkresPowtarzaniaImpulsow, c,fc, CzestotliwoscProbkowania, txWave,
Yobszar, Xobszar,LiczbaRowerzystow,OkresProbkowania,posr,velr);
[SBic,TBic1,FBic1] =
Sygnatury(WynikDlaRowerzystow,OkresProbkowania);
figure(i)
imagesc(TBic1,FBic1,SBic(:,:,1));
folderpath = strcat('C:\Dane\Bicycle\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
```

%% jeden samochód

```
LiczbaSamochodow = 1;
WynikDlaSamochodow = Samochody(LiczbaImpulsow,
OkresPowtarzaniaImpulsow, c,fc, CzestotliwoscProbkowania, txWave,
Yobszar,
Xobszar,OkresProbkowania,posr,velr,MaxPredkoscSamochodu,LiczbaSam
ochodow);
[SCar,TCar1,FCar1] =
Sygnatury(WynikDlaSamochodow,OkresProbkowania);
figure(i)
imagesc(TCar1,FCar1,SCar(:,:,1));
folderpath = strcat('C:\Dane\Car\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
```

%% dwóch pieszych

```
LiczbaPieszych2 = 2;
WynikDlaPieszych2 = Piesi(LiczbaImpulsow,
OkresPowtarzaniaImpulsow, c,fc, CzestotliwoscProbkowania, txWave,
Yobszar, Xobszar,LiczbaPieszych2,OkresProbkowania,posr,velr);
[SPed2,T2,F2] = Sygnatury(WynikDlaPieszych2,OkresProbkowania);
figure(i)
imagesc(T2,F2,SPed2(:,:,1));
folderpath = strcat('C:\Dane\Pedestrian_Pedestrian\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
```

%% trzech pieszych

```
LiczbaPieszych3 = 3;
WynikDlaPieszych3 = Piesi(LiczbaImpulsow,
OkresPowtarzaniaImpulsow, c,fc, CzestotliwoscProbkowania, txWave,
Yobszar, Xobszar,LiczbaPieszych3,OkresProbkowania,posr,velr);
[SPed3,T3,F3] = Sygnatury(WynikDlaPieszych3,OkresProbkowania);
```

```
figure(i)
imagesc(T3,F3,SPed3(:,:,1));
folderpath =
strcat('C:\Dane\Pedestrian Pedestrian \');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
%% jeden pieszy + jeden rowerzysta
JedenPieszyJedenRower = complex(zeros(size(WynikDlaPieszych)));
for ii = 1:size(WynikDlaPieszych,3)
JedenPieszyJedenRower(:,:,ii) = WynikDlaPieszych(:,:,ii) +
WynikDlaRowerzystow(:,:,ii);
end
[SJedenPieszyJedenRower, TJedenPieszyJedenRower, FJedenPieszyJedenR
ower] = Sygnatury(JedenPieszyJedenRower,OkresProbkowania);
figure(i)
imagesc(TJedenPieszyJedenRower,FJedenPieszyJedenRower,SJedenPiesz
yJedenRower(:,:,1));
folderpath = strcat('C:\Dane\Pedestrian Bicycle\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
%% jeden pieszy + jeden samochód
JedenPieszyJedenSamochod =
complex(zeros(size(WynikDlaPieszych)));
for ii = 1:size(WynikDlaPieszych,3)
JedenPieszyJedenSamochod(:,:,ii) = WynikDlaPieszych(:,:,ii) +
WynikDlaSamochodow(:,:,ii);
end
[SJedenPieszyJedenSamochod, TJedenPieszyJedenSamochod, FJedenPieszy
JedenSamochod] =
Sygnatury(JedenPieszyJedenSamochod,OkresProbkowania);
figure(i)
imagesc(TJedenPieszyJedenSamochod,FJedenPieszyJedenSamochod,SJede
nPieszyJedenSamochod(:,:,1));
folderpath = strcat('C:\Dane\Pedestrian Car\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
%% jeden rowerzysta + jeden samochód
JedenRowerJedenSamochod = complex(zeros(size(WynikDlaPieszych)));
for ii = 1:size(WynikDlaPieszych,3)
JedenRowerJedenSamochod(:,:,ii) = WynikDlaRowerzystow(:,:,ii) +
WynikDlaSamochodow(:,:,ii);
end
```

```
[SJedenRowerJedenSamochod, TJedenRowerJedenSamochod, FJedenRowerJed
enSamochod] =
Sygnatury(JedenRowerJedenSamochod,OkresProbkowania);
figure(i)
imagesc(TJedenRowerJedenSamochod,FJedenRowerJedenSamochod,SJedenR
owerJedenSamochod(:,:,1));
folderpath = strcat('C:\Dane\Car Bicycle\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
%% dwóch pieszych + jeden rowerzysta
DwowchPieszychJedenRower =
complex(zeros(size(WynikDlaPieszych)));
for ii = 1:size(WynikDlaPieszych,3)
        DwowchPieszychJedenRower(:,:,ii) =
WynikDlaPieszych2(:,:,ii) + WynikDlaRowerzystow(:,:,ii);
end
[SDwowchPieszychJedenRower, TDwowchPieszychJedenRower, FDwowchPiesz
ychJedenRower] =
Sygnatury(DwowchPieszychJedenRower,OkresProbkowania);
figure(i)
imagesc(TDwowchPieszychJedenRower,FDwowchPieszychJedenRower,SDwow
chPieszychJedenRower(:,:,1));
folderpath = strcat('C:\Dane\Pedestrian Pedestrian Bicycle\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
%% dwóch pieszych + jeden samochód
DwowchPieszychJedenSamochod =
complex(zeros(size(WynikDlaPieszych)));
for ii = 1:size(WynikDlaPieszych,3)
DwowchPieszychJedenSamochod(:,:,ii) = WynikDlaPieszych2(:,:,ii) +
WynikDlaSamochodow(:,:,ii);
end
[SDwowchPieszychJedenSamochod, TDwowchPieszychJedenSamochod, FDwowc
hPieszychJedenSamochod] =
Sygnatury(DwowchPieszychJedenSamochod,OkresProbkowania);
figure(i)
imagesc(TDwowchPieszychJedenSamochod,FDwowchPieszychJedenSamochod
,SDwowchPieszychJedenSamochod(:,:,1));
folderpath = strcat('C:\Dane\Pedestrian_Pedestrian_Car\');
filenameindex = i;
filename = strcat(folderpath, sprintf('%d.jpg', filenameindex));
saveas(figure(i),filename)
close all
```

```
end
```

Wybrane transformacje spektrogramów

```
[nazwa pliku, sciezka] = uigetfile({'*.jpg;*.png;*.bmp;*.tif'},
'Wybierz obraz');
obraz = imread(fullfile(sciezka, nazwa pliku));
figure;
imshow(obraz);
title('Oryginalny obraz');
Folder_zapisu = fullfile(sciezka, 'Obrazy_przetransformowane');
%% Transformacje geometryczne
% Przesunięcie obrazu o określoną liczbę pikseli
LiczbaPikseliPoziom = 5;
LiczbaPikseliPionowo = 3;
PrzesuniecieObrazu = maketform('affine', [1 0 0; 0 1 0;
LiczbaPikseliPoziom LiczbaPikseliPionowo 1]);
Obraz przetransformowany geometrycznie = imtransform(obraz,
PrzesuniecieObrazu, 'XData', [1 size(obraz,2)], 'YData', [1
size(obraz,1)]);
figure;
imshow(Obraz przetransformowany geometrycznie);
title('Obraz przesuniety');
imwrite(Obraz przetransformowany geometrycznie,
fullfile(Folder zapisu, 'Obraz przesuniety.png'));
% Skalowanie obrazu
wspolczynnik skalowania = 0.05;
PrzeskalowanieObrazu = maketform('affine', [wspolczynnik skalowania
0 0; 0 wspolczynnik skalowania 0; 0 0 1]);
ObrazPrzeskalowany = imtransform(obraz, PrzeskalowanieObrazu);
figure:
imshow(ObrazPrzeskalowany);
title('Obraz skalowany');
imwrite(ObrazPrzeskalowany, fullfile(Folder zapisu,
'Obraz skalowany.png'));
%% Operacje na wartościach pikseli
% Dodanie szumu
ObrazZaszumiony = imnoise(obraz, 'gaussian', 0, 0.01);
figure;
imshow(ObrazZaszumiony);
title('Obraz z dodanym szumem Gaussowskim');
imwrite(ObrazZaszumiony, fullfile(Folder zapisu,
'Obraz_z_szumem.png'));
% Zmiana jasności
ZwiekszenieJasnosci = 5;
ObrazJasniejszy = obraz + ZwiekszenieJasnosci;
ObrazJasniejszy = min(ObrazJasniejszy, 255);
```

figure; imshow(ObrazJasniejszy); title('Obraz o zwiększonej jasności'); imwrite(ObrazJasniejszy, fullfile(Folder_zapisu, 'Obraz_zwiekszona_jasnosc.png'));

% Zmiana kontrastu WspolczynnikKontrastu = 1.01; ObrazKontrast = imadjust(obraz, [], [], WspolczynnikKontrastu); figure; imshow(ObrazKontrast); title('Obraz o zwiększonym kontraście'); imwrite(ObrazKontrast, fullfile(Folder_zapisu, 'Obraz_zwiekszony_kontrast.png'));

Budowa, proces uczenia i klasyfikacja CNN

```
%% Pobranie danych
Sciezka = 'C:\Dane';
ZbiorDanych = fullfile(Sciezka,'77 GHz');
Obrazy = imageDatastore(ZbiorDanych,
'IncludeSubfolders',true,'LabelSource','foldernames');
%% Podział zbiorów
IloscObrazyTreningowe = 0.7;
IloscObrazyWalidacyjne = 0.15;
[ObrazyTreningowe,ObrazyWalidacyjne,ObrazyTestowe] =
splitEachLabel(imds,IloscObrazyTreningowe,IloscObrazyWalidacyjne,
'randomize');
RozmiarObrazu = [656,875,3];
%% Sieć CNN
Warstwy = [
    imageInputLayer([RozmiarObrazu(1) RozmiarObrazu(2) 3])
    convolution2dLayer(5,16, 'Padding', 'same')
    batchNormalizationLaver
    reluLaver
    maxPooling2dLayer(5,'Stride',2)
    convolution2dLayer(5,32,'Padding','same')
    batchNormalizationLayer
    reluLaver
    averagePooling2dLayer(2,'Stride',2)
    convolution2dLayer(5,64, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer
    averagePooling2dLayer(2,'Stride',2)
    convolution2dLayer(5,128,'Padding','same')
    batchNormalizationLayer
    reluLayer
    averagePooling2dLayer(2, 'Stride', 2)
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer
        ];
```

```
%% Proces uczenia
OpcjeProcesuUczenia = trainingOptions('sgdm', ...
    'InitialLearnRate',0.001, ...
    'MaxEpochs',10, ...
    'Shuffle','every-epoch', ...
    'MiniBatchSize',48,...
    'ValidationData',ObrazyWalidacyjne, ...
    'ValidationFrequency',15, ...
    'Verbose', false, ...
    'CheckpointPath',tempdir,...
    'Plots', 'training-progress');
%% Uruchomienie sieci
SiecCNN =
trainNetwork(ObrazyTreningowe,Warstwy,OpcjeProcesuUczenia);
%% Wyniki uczenia sieci
KlasaPredykowana = classify(SiecCNN,ObrazyTestowe);
KlasaTestowa = ObrazyTestowe.Labels;
Dokladnosc = sum(KlasaPredykowana ==
KlasaTestowa)/numel(KlasaTestowa)
figure(1)
MacierzPomylek = confusionchart(KlasaPredykowana, KlasaTestowa);
MacierzPomylek.RowSummary = 'row-normalized';
MacierzPomylek.Title = 'Macierz pomylek dla klasyfikacji obrazow';
```