dr n. tech. Dorota Anna OSZUTOWSKA-MAZUREK[a,b], dr hab. inż. Przemysław MAZUREK[c]

[a] Wyższa Szkoła Techniczno-Ekonomiczna w Szczecinie, Wydział Systemów Automotive
Higher School of Technology and Economics in Szczecin, Faculty of Automotive Systems

[b] Pomorski Uniwersytet Medyczny, Wydział Nauk o Zdrowiu, Zakład Epidemiologii i Zarządzania
Pomeranian Medical University, Faculty of Health Sciences, Department of Epidemiology and Management

[c] Zachodniopomorski Uniwersytet Technologiczny w Szczecinie, Wydział Elektryczny,
Katedra Przetwarzania Sygnałów i Inżynierii Multimedialnej
West Pomeranian University of Technology in Szczecin, Faculty of Electrical Engineering,
Department of Signal Processing and Multimedia Engineering

ZASTOSOWANIE SIECI KONWOLUCYJNEJ GŁĘBOKIEGO UCZENIA W DETEKCJI POJAZDÓW

Streszczenie

Wstęp i cel: Detekcja pojazdów na znaczenie w bezpieczeństwie ruchu drogowego oraz programowaniu pojazdów autonomicznych. Celem pracy jest detekcja pojazdów odróżniająca obrazy pojazdów od innych obrazów nie zawierających pojazdów.
Materiał i metody: W pracy wykorzystano bazę pojazdów zawierającą obrazy ekstrahowane z sekwencji wideo, które przetwarzano za pomocą sieci konwolucyjnej głębokiego uczenia.
Wyniki: Uzyskana sieć konwolucyjna charakteryzuje się bardzo dobrymi parametrami, krzywa PSNR względem kroku uczenia rośnie co oznacza, że zachodzi proces odszumiania kerneli w całym procesie uczenia.
Wniosek: Proponowana metoda może być wykorzystana w programowaniu pojazdów autonomicznych oraz implementacji w Inteligentnych Systemach Transportowych ITS do detekcji pojazdów; bazuje na uczeniu a nie na projektowaniu algorytmu syntetycznego, dzięki temu jest potrzebny relatywnie krótki czas opracowania klasyfikatora.

Słowa kluczowe: Sieć konwolucyjna, głębokie uczenie, detekcja pojazdów, przetwarzanie obrazów.
(Otrzymano: 04.12.2017; Zrecenzowano: 11.12.2017; Zaakceptowano: 18.12.2017)

USE OF DEEP LEARNING CONVOLUTIONAL NETWORK IN VEHICLE DETECTIONMEARS

Abstract

Introduction and aim: *Vehicle detection plays essential role in road safety and automatic vehicle programming. The aim of study is vehicle detection distinguishing car and non-car images*
Material and methods: *Vehicle database images extracted from video sequences were processed by deep learning convolutional network.*
Results*: Obtained convolutional network is characterised by very good parameters, PSNR curve indicates denoising of kernels in learning process.*
Conclusion: *Proposed method is potentially useful in autonomic vehicles programming and Intelligent Transportation Systems (ITS) for vehicles detection. The solution is based on learning, not on synthetic algorithm design, thanks to this, a relatively short time of classifier development is needed.*

Keywords: *Image processing, deep learning, convolutional neural network, vehicle detection.*
(Received: 04.12.2017; Revised: 11.12.2017; Accepted: 18.12.2017)

1. Wstęp i cele

Detekcja pojazdów jest jednym z ważnych obszarów badań w kontekście bezpieczeństwa ruchu drogowego oraz postępu technologicznego związanego z rozwojem pojazdów autonomicznych oraz Inteligentnych Systemów Transportowych (ITS). Opracowano różne metody detekcji pojazdów, wśród których uwagę autorów zwróciły rozwiązania bazujące na sztucznej inteligencji

W pracy [7] zaproponowano rozpoznawanie pojazdu z widoku z przodu z wykorzystaniem głębokiego uczenia *(ang. deep learning).* Jako metodę deep learning wykorzystano RBM *(ang. Restricted Boltzmann Machines)* dla obrazów binarnych. W artykule [5] przedstawiono sieć konwolucyjną głębokiego uczenia do uczenia nadzorowanego dla detekcji pojazdów. Dla procesu klasyfikacji stopniowo dodawano kolejne warstwy ma wyjściu. Z kolei w pracy [3] zaprezentowano system który służy do wyszukiwania wielu pojazdów na drodze. Zastosowano dwie sieci konwolucyjne, jedna która wykorzystuje detekcje na podstawie koloru, a druga na podstawie marki i rodzaju modelu pojazdu. W pracy [6] wykorzystano sieć konwolucyjną do detekcji pojazdu z innego pojazdu. W tego typu konfiguracji część obrazu zawiera obiekty spoza drogi, dlatego wykorzystano detekcję pasów drogi w celu ograniczenia obszaru analizy w celu odrzucenia detekcji fałszywych powiązanych z obiektami spoza drogi.

Celem niniejszej pracy jest detekcja pojazdów odróżniająca obrazy pojazdów od innych obrazów nie zawierających pojazdów.

2. Materiały i metody

W pracy wykorzystano bazę, która zawiera obrazy ekstrahowanych z sekwencji wideo (pozyskane z kamerą patrzącą w przód zamontowaną na pojeździe) [9]. Baza składa się z 3425 obrazów pojazdów - konkretnie tyłów pojazdów, wykonanych z różnych ujęć, dodatkowo 3900 obrazów było ekstrahowanych z sekwencji drogowych, które nie zawierały pojazdów. Jedna z istotnych cech wpływających na wygląd pojazdu od tyłu jest pozycja obserwowanego pojazdu względem kamery. Zatem baza oddziela obrazy dla czterech różnych regionów w odniesieniu do pozycji: pojazd blisko lub w średniej odległości od kamery w centrum kadru; pojazd blisko lub w średniej odległości od kamery w lewej części kadru; pojazd blisko lub w średniej odległości od kamery w prawej części kadru; pojazd w dużej odległości.

Uwzględnienie tego typu przypadku jest istotne z uwagi na obserwowanie pojazdów na sąsiednich pasach (wariant z lokalizacją z lewej / prawej strony kadru).

Dodatkowo obrazy były ekstrahowane w ten sposób, że nieperfekcyjnie pasowały do konturu pojazdów w celu uczynienia klasyfikacji bardziej odpornej na przesunięcia w hipotezie stanu generalnego; część obrazów zawierała całe pojazdy podczas gdy inne tylko zawierały pojazdy częściowo [1], [9].

Obrazy miały rozdzielczość 64×64 piksele i były wycięte z sekwencji 360×256 pikseli nagrań z autostrad Madrytu, Brukseli, Turynu. Baza jest otwarta dla innych badaczy, co jest opisane na stronie bazy [9]

Oprócz obrazów własnej kolekcji badacze udostępniający bazę uwzględnili mały zbiór obrazów z innych baz w celu zaokrąglenia obrazów do 40000 obrazów pojazdów oraz 4000 obrazów nie zawierających pojazdów. Uwzględniono dodatkowe bazy, takie jak: Caltech Database [13] i [4] oraz the TU Graz-02 Database [11], [14].

Kompletne zestawy obrazów były wyselekcjonowane w ten sposób, że były wykonane w różnych warunkach atmosferycznych. Z 2000 obrazów przeznaczonych na każdy rejon (1000 obrazów pojazdów versus 1000 obrazów nie-pojazdów), 20% było wykonanych w czasie słonecznej pogody, 20% w dni pochmurne, 20% w tzw. średnich warunkach (nie bardzo słoneczne i nie bardzo pochmurne), 20% w słabych warunkach oświetleniowych, 10% pod-

czas lekkiego deszczu, 5% z niską rozdzielczością kamery, a 2,5% w tunelach, które mają sztuczne oświetlenie. Przykłady obrazów z bazy są przedstawione na rysunkach 1 oraz 2.

Rys. 1. Przykładowe pojazdy [9]
Fig. 1. Examples of cars [9]

Rys. 2. Przykładowe tła (nie-pojazdy) [9]
Fig. 2. Examples of background (non-cars) [9]

W pracy zastosowano sieć konwolucyjną głębokiego uczenia do detekcji pojazdów. Proces uczenia konwolucyjnych sieci neuronowych jest podobny do uczenia konwencjonalnych sieci neuronowych. Zaletą zastosowania sieci konwolucyjnej z użyciem kerneli jest przetwarzanie danych pomiędzy warstwami. Występują warstwy dedykowane konwolucji, a liczba wag jest zredukowana, ponieważ część wag jest współdzielona dla obliczenia dla wyjścia piksela. Zwielokrotnione kernele są uczone w konkretnej konwolucyjnej warstwie, dlatego efektem jest wiele obrazów na wyjściu. W związku ze zwielokrotnieniem przychodzących danych, stosowane są warstwy typowe dla sieci konwolucyjnych, na przykład jest kilka typów warstw redukujących, inaczej głosujących [10] *(ang. pooling)* [8], które odpowiadają za redukcję da-

nych (skalowanie obrazu do niższej rozdzielczości, zastosowanie średniej lub maksymalnej wartości dla lokalnych grup pikseli). W warstwie redukującej realizowana jest filtracja statystyczną w obrębie maski o żądanych rozmiarach poprzez wyznaczenie wybranej statystyki, np. wartości maksymalnej (tzw. MaxPooling przedstawiony na rys. 3) [2], [12] Sieć konwolucyjna zawiera kilka konwolucyjnych warstw redukujących, ale po normalizacji wielokanałowej *(ang. Cross-channel Normalisation)* następuje przetwarzane z zastosowaniem nieliniowej funkcji aktywacji. Ma to związek z tym, że konwolucja i redukcja prowadzą operacje liniowe i zwielokrotnienie tych warstw powinno być zredukowane do pojedynczej liniowej warstwy. Dodanie nieliniowości daje możliwość realizacji złożonej transformacji wejście-wyjście. Przykładową funkcja aktywacji jest *ReLU (ang. Rectified Linear Unit),* której zaletą jest to, że bezpośrednia funkcja oraz pochodne mogą być implementowane bezpośrednio w kodzie bez specjalnych obliczeń. W tej warstwie są eliminowane wartości ujemne będące wynikiem działania warstwy konwolucyjnej. Działanie funkcji ReLU polega to na tym, że dodatnie wartości, są pozostawione są bez zmian a ujemne zamieniane na zerowe [2], [12]. Takie działanie funkcji jest ważne dla zachowania pozytywnego dopasowania kernela do konkretnego obrazu i supresji negatywnie skorelowanego obrazu.

Struktura sieci konwolucyjnej jest wielowarstwowa, więc schemat przetwarzania mieści się w ramach technik głębokiego uczenia. Ostatnie warstwy sieci konwolucyjnej są klasycznymi sieciami neuronowymi z pełnymi połączeniami *(ang. Full Connection, FC),* (Rys. 3). Koncepcja sieci konwolucyjnej bazuje na przetwarzaniu obrazu zaczynając od wykrywania detali w pierwszej warstwie, do bardziej detekcji skomplikowanych struktur wyższego rzędu. Pary treningowe są wybierane losowo z dużej bazy, ponieważ jest wiele wag w sieci. Warto wspomnieć, że liczba wag jest o wiele mniejsza w porównaniu do pełnych połączeń z niezależnymi wagami w konwencjonalnej sieci neuronowej.

Liczba wag i warstw nie jest optymalna dla poszczególnych zadań rozpoznawania i przetwarzania obrazów w porównaniu z konwencjonalnymi sieciami neuronowymi. Podstawową zaletą sieci konwolucyjnych jest szybszy czas uczeniu, ponieważ konwencjonalna siec neuronowa ma złożone relacje pomiędzy wagami i wymaganym czasem uczenia [12].

W ostatnich dwóch warstwach z pełnymi połączeniami (FC) realizowany jest proces klasyfikacji. W celu ograniczenia wystąpienia bardzo dużych wartości na wyjściu zastosowano normalizację za pomocą warstwy SoftMax [8], dzięki czemu wartości wyjściowe są ograniczone do zakresu 0-1.

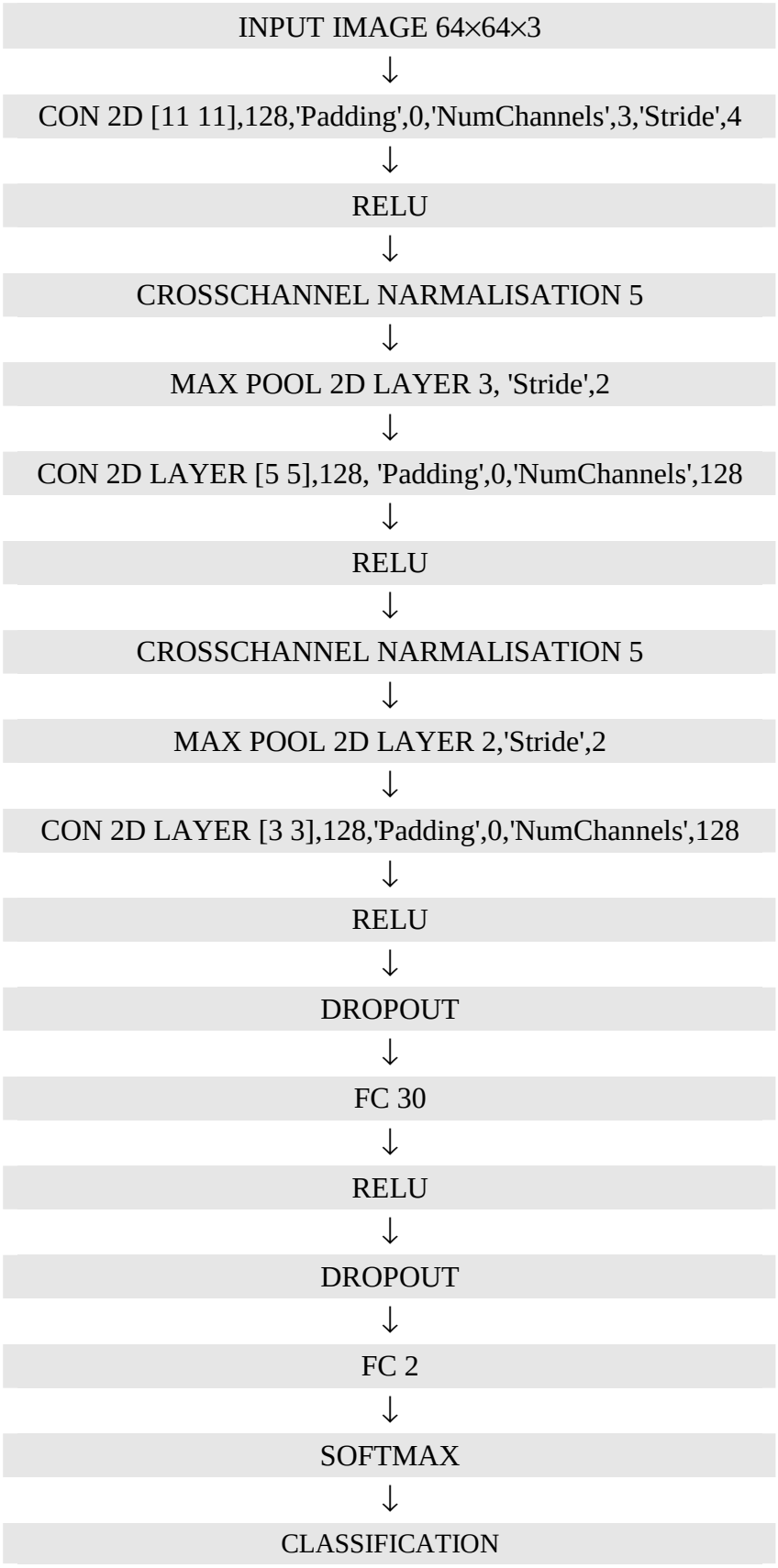
Na wyjściu otrzymywane są dwie wartości binarne – samochód albo inny obiekt, przez porównanie większościowe wartości obu wyjść.

Uczenie przebiega od zera z losowego punktu startowego, wykorzystano algorytm SGDM *(ang. Stochastic Gradient Descent with Momentum),* który jest wariantem metody SGD, polegającym na uwzględnieniu ostatniej dokonanej zmiany wagi w obecnie dokonywanej [10]

Obrazy wejściowe miały rozdzielczość 64×64 pikseli. Z analizowanych obrazów 50% wykorzystano do uczenia oraz 50% do testowania, przy czym każde 50% dotyczy poszczególnych grup-pojazd, tło.

W procesie uczenia wykorzystano środowisko MATLAB w połączeniu z kartą graficzną GPU GeForce GTX TITAN X (Maxwell) z 3072 rdzeniami CUDA, 12GB pamięci GPU oraz 384-bit interfejsem pamięci. MATLAB wykorzystywał bibliotekę NVidia cuDNN, która jest biblioteką dedykowaną przetwarzaniu danych w sieci konwolucyjnch i jest optymalizowana przez producenta GPU. CPU zastosowano do argumentacji obrazów wejściowych pary treningowej (obroty +/-10 stopni, lustrzana odbicie lewo-prawo, pochylanie obrazu *(ang. shear)* poziome +/- 5 stopni, skalowanie obrazu 50-200% niezależnie horyzontalnie i wertykalnie).

Ponieważ zbiór uczący jest bardzo duży, zwłaszcza po wprowadzaniu augmentacji i nie można załadować go jednocześnie do pamięci karty GPU, to konieczne stało się wykorzystanie uczenia losowymi ale mniejszymi fragmentami tego zbioru *(ang. mini-batch).*

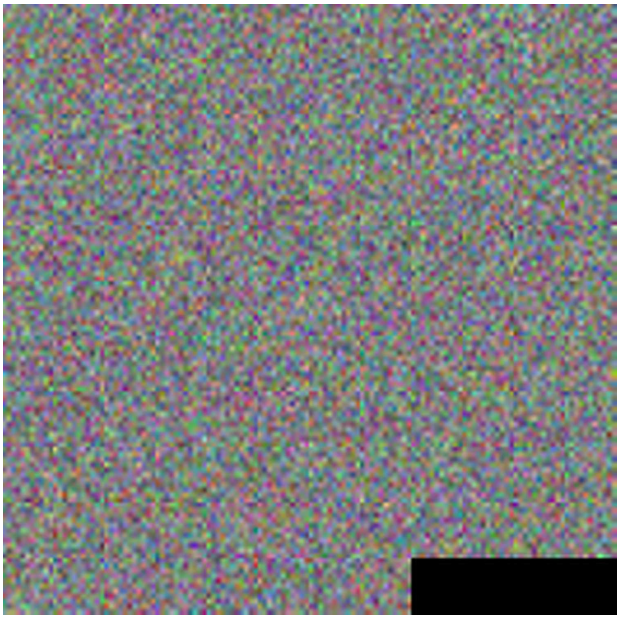


Rys. 3. Schemat sieci konwolucyjnej
Źródło: Opracowanie własne

Fig. 3. Scheme of convolutional network
Source: Elaboration of the Authors

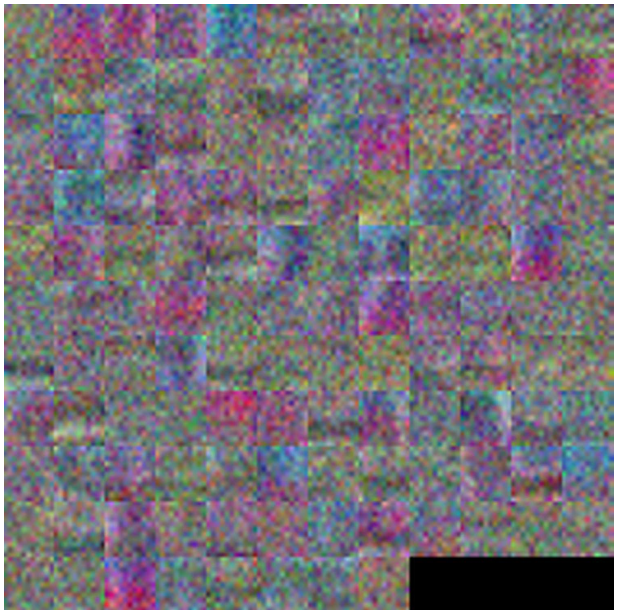
3. Wyniki

Proces rozpoczyna się od losowania wszystkich wag, co widać jako szum na obrazie zbiorczym kerneli pierwszej warstwy (Rys. 4). Kernele przedstawiono zbiorczo na jednym obrazku. Prawidłowy proces uczenia powinien doprowadzić do powstania kerneli odpowiadających za wykrywanie drobnych elementów pojazdu. Kernele nie powinny być zaszumione. Jakość kerneli wskazuje pośrednio na jakość klasyfikatora bazującego na sieci konwolucyjnej. Na kolejnych rysunkach (Rys. 5, Rys. 6, Rys. 7) pokazano wygląd kerneli dla wybranych momentów uczenia n = 10, 100, 1000, odpowiednio.



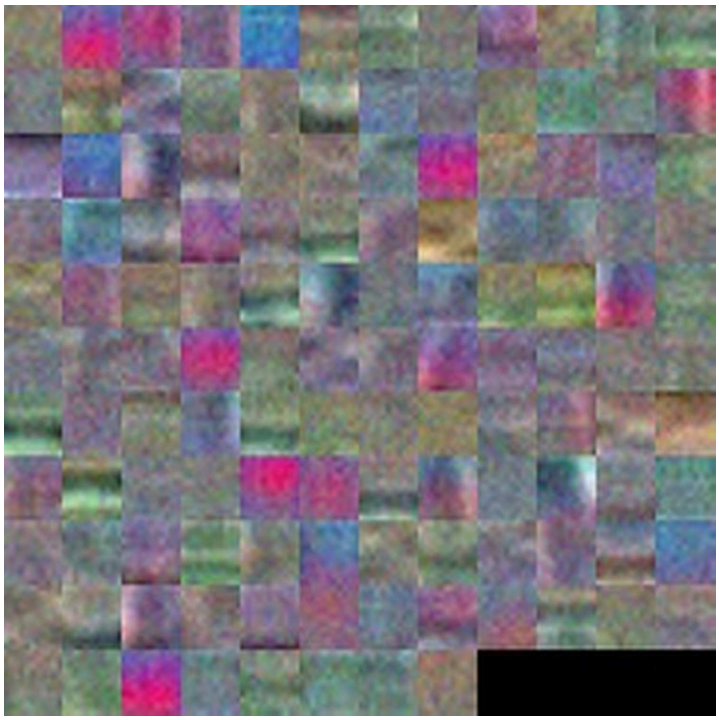
Rys. 4. *Obrazy kerneli detekcji w pierwszej warstwie konwolucyjnej (po inicjalizacji)*
Źródło: Opracowanie własne

Fig. 4. Images of kernels detection in first convolutional layer *(after initialisation)*
Source: Elaboration of the Authors



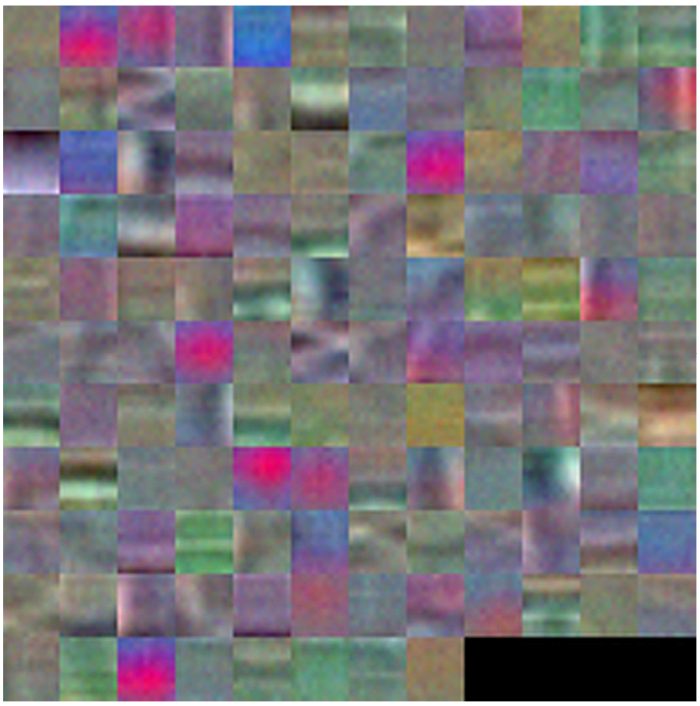
Rys. 5. *Obrazy kerneli detekcji w pierwszej warstwie konwolucyjnej (dla kroku uczenia n=10)*
Źródło: Opracowanie własne

Fig. 5. Images of kernels detection in first convolutional layer *(for learning step n=10)*
Source: Elaboration of the Authors



Rys. 6. *Obrazy kerneli detekcji w pierwszej warstwie konwolucyjnej (dla kroku uczenia n=100)*
Źródło: Opracowanie własne

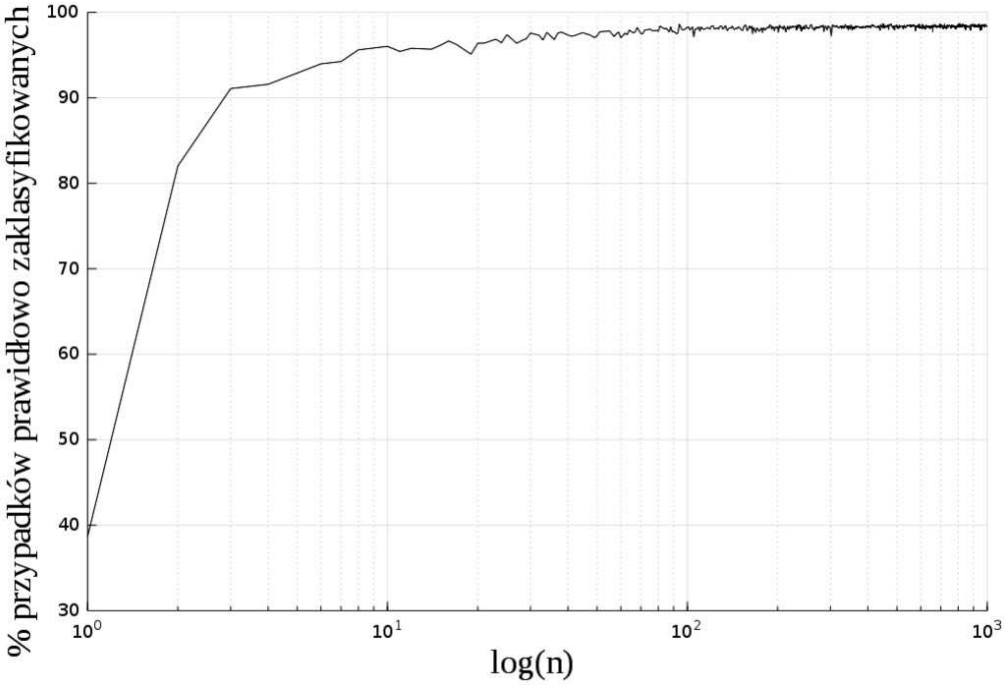
Fig.6. Images of kernels detection in first convolutional layer *(for learning step n=100)*
Source: Elaboration of the Authors



Rys. 7. *Obrazy kerneli detekcji w pierwszej warstwie konwolucyjnej (dla kroku uczenia n=1000)*
Źródło: Opracowanie własne

Fig. 7. Images of kernels detection in first convolutional layer *(for learning step n=1000)*
Source: Elaboration of the Authors

Proces uczenia został przerwany po 1000 krokach i uzyskano krzywą uczenia jak przestawiono na rysunku 8.



Rys. 8. Krzywa uczenia z błędami testowania na podstawie zbioru testującego
Źródło: Opracowanie własne

Fig. 8. Learning curve with testing errors based on test set
Source: Elaboration of the Authors

Ponieważ proces uczenia wiąże się z redukcją szumu kerneli, to można wykorzystać do analizy procesu uczenia jedno z kryteriów do porównywania obrazu – PSNR *(ang. Peak Signal-to-Noise-Ratio)* opisane następującym wzorem:

$$\mathrm{PSNR} = 10\log\left[\frac{(2^k - 1)^2}{\mathrm{MSE}}\right] \tag{1}$$

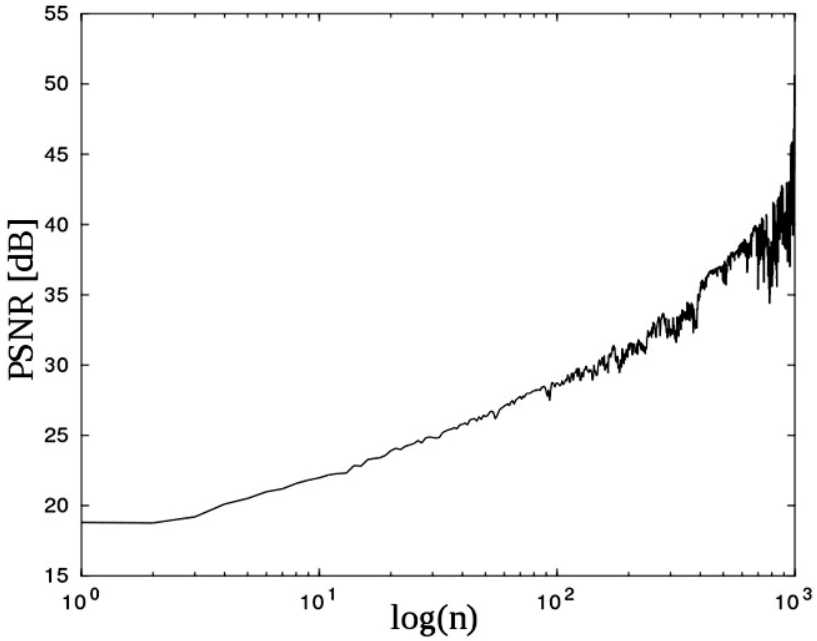
gdzie k jest liczbą bitów użytych do kodowania stopni szarości obrazu, a MSE *(ang. Mean Square Error)* jest błędem średniokwadratowym między parą obrazów. Obrazem referencyjnym jest ostatni obraz uzyskany z procesu uczenia, a porównanie wykonuje się ze wszystkimi kernelami zapisanymi na dysku (dla wszystkich n). Zmiana wartości PSNR pokazana jest na rysunku 9.

Kolejną metodą analizy jakości pracy sieci jest sprawdzenie jest zachowania dla zmiany rodzaju obrazu. Obrazy uczące i testujące są obrazami rzeczywistymi kolorowymi. Wprowadzenie obrazów w odcieniach szarości nie powinno skutkować dużymi błędami. Oznacza to że sieć bazuje na cechach luminancji oraz detekcji kształtu. Wyniki przedstawiono w tabeli 1.

Tab. 1. Porównanie jakości rozpoznawania dla obrazów kolorowych i w odcieniach szarości
Tab. 1. Comparison of identification quality for colour and grayscale images

	Ilość przypadków	Ilość prawidłowych rozpoznań
Pojazd kolorowy	3425	3416
Nie-pojazd kolorowy	3900	3865
Pojazd szary	3425	3400
Nie-pojazd szary	3900	3187

Źródło: Opracowanie własne / Source: Elaboration of the Authors



Rys. 9. Krzywa uczenia dla PSNR
Źródło: Opracowanie własne

Fig. 9. Learning curve for PSNR
Source: Elaboration of the Authors

4. Dyskusja

Uzyskana sieć kowolucyjna charakteryzuje bardzo dobrymi parametrami, co pokazują wyniki z tabeli 1, gdzie jedynie dla obrazu nie-pojazdów w odcieniach szarości jest gorszy wynik. W pozostałych przypadkach w kilkunastu przypadkach jest błędy wynik. Krzywa uczenia jest o typowym kształcie (Rys. 8).

Obrazy kerneli zawierają obrazy linii lub krzywych (Rys. 7), a w szczególności obrazy czerwonych świateł tylnych pojazdu. Są to typowe obrazy podstawowych cech na jakie jest rozbijany obraz w pierwsze warstwie konwolucyjnej. Poza czerwonymi światłami zasadniczo nie ma obrazów o innych dominujących kolorach co znaczy, że sieć stara się w mniejszym stopniu wykorzystywać informację o kolorze.

Test dla obrazów w odcieniach szarości pokazuje, że istotnie kolor nie stanowi elementu decydującego, dla detekcji pojazdu. Możliwe, że ma większe znaczenie dla obrazów zawierających tło (np. z uwagi na trawę na poboczu), jednak wymaga to dalszych badań.

W pracy zaproponowano badanie PSNR względem kroku uczenia. Krzywa ta rośnie (rośnie stosunek sygnału do szumu) co oznacza, że istotnie zachodzi proces odszumiania kerneli w całym procesie uczenia.

5. Wnioski

- Proponowana metoda może być wykorzystana w programowaniu pojazdów autonomicznych.
- Rozwiązanie bazuje na uczeniu a nie na projektowaniu algorytmu syntetycznego, dzięki temu jest krótki czas opracowania klasyfikatora
- Metoda pozwala może być zaimplementowana w Inteligentnych Systemach Transportowych ITS do detekcji pojazdów.

Acknowledgments: We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

Literatura

[1] Arróspide J., Salgado L., Nieto M.: *Video analysis based vehicle detection and tracking using an MCMC sampling framework.* EURASIP Journal on Advances in Signal Processing, Vol. 2012, Article ID 2012: 2, January.

[2] Chmielińska J., Jakubowski J.: *Zastosowanie sieci konwolucyjnej do wykrywania wybranych symptomów zmęczenia kierowcy.* Przegląd Elektrotechniczny, ISSN 0033-2097, R. 93 NR 10, s. 6-10, 2017.

[3] Dehghan A., Masood S.Z., Shu G., Ortiz E.G.: *View Independent Vehicle Make, Model and Color Recognition Using Convolutional Neural Network.* arXiv:1702.01721v1 [cs.CV], 2017.

[4] Fergus R., Perona P., Zisserman A.: [in:] *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 16-22 June 2003.

[5] Feyzabadi S.: *Joint Deep Learning for Car Detection.* arXiv:1412.7854v2 [cs.CV], 2016.

[6] Forczmański P, Nowosielski A.: *Deep learning approach to detection of preceding vehicle in advanced driver assistance*. Mikulski J. (eds) Challenge of Transport Telematics. TST 2016. Communications in Computer and Information Science, Vol. 640. Springer, Cham 2016.

[7] Gao Y., Lee H.J.: *Moving car detection and model recognition based on deep learning,* Advanced Science and Technology Letters. Vol. 90, pp. 57-61, Multimedia 2015.

[8] Goodfellow I., Bengio Y., Courville A.: *Deep learning. Systemy uczące się.* Wydawnictwo Naukowe PWN, Warszawa 2018.

[9] *https://www.gti.ssr.upm.es/data/Vehicle_database.html, dostęp 1 grudnia 2017*

[10] Odrzywołek K.: *Wykorzystanie głębokich sieci neuronowych w weryfikacji mówcy*. Praca magisterska, Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, Kraków 2016.

[11] Opelt A., Pinz A.: [in:] *Proceedings of the 14th Scandinavian Conference on Image Analysis*, Joensuu, Finland, 19-22 June 2005.

[12] Oszutowska-Mazurek D., Knap O.: *The use of deep learning for segmentation of bone marrow histological images, Artificial intelligence trends in intelligent systems.* Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol. 1, Springer 2017.

[13] The Caltech Database (Computational Vision at California Institute of Technology, Pasadena), http://www.vision.caltech.edu/html-files/archive.html. Accessed 14 May 2011.

[14] The TU Graz-02 Database (Graz University of Technology), http://www.emt.tugraz.at/~pinz/data/GRAZ_02/. Accessed 14 May 2011.