Jakub KONOPIŃSKI[1], Krystyna BARAN[2], Jakub ŁAZIK[3], Piotr GÓRAL[4], Paweł PAWŁOWSKI[5]

Zespół Szkół Techniczno-Elektronicznych w Kaliszu (1,2,3) Politechnika Poznańska, Instytut Automatyki i Robotyki (4,5), ORCID: 1. 0009-0005-3252-2430, 4. 0000-0001-8999-693X, 5. 0000-0001-5373-5148,

doi:10.15199/48.2024.10.41

Sztuczna sieć neuronowa kontra technika algorytmiczna w zadaniu klasyfikacji kształtów

Streszczenie. *W artykule przedstawiono wyniki badań porównawczych zadania klasyfikacji prostych kształtów. Porównano opracowany model sztucznej sieci neuronowej typu CNN z techniką algorytmiczną dokonującą detekcji krawędzi algorytmem Canny'ego i klasyfikującą obiekty na podstawie liczby i wzajemnego położenia rozpoznanych krawędzi. Do eksperymentów przygotowano zbiór danych składający się z 2162 zdjęć reprezentujących przedmioty o kształtach: prostokąta, koła i trójkąta. Sieć neuronowa uzyskała dokładność klasyfikacji równą 85%, a technika algorytmiczna 77%. Porównanie czasu działania pokazało jednak wyższość techniki algorytmicznej: działała ona 8 razy szybciej. Rozwiązanie może znajdować zastosowania do segregacji obiektów na liniach produkcyjnych i być zaimplementowane na komputerze jednoukładowym.*

Abstract*. The article presents results of comparative research on the task of classifying simple shapes. The developed model of an artificial neural network of the CNN type was compared with an algorithmic technique that detects edges using the Canny algorithm and classifies objects based on the number and relative position of recognized edges. A data set consisting of 2162 photos representing objects with the shapes of a rectangle, a circle and a triangle was prepared for the experiments. The neural network achieved a classification accuracy of 85% and the algorithmic technique 77%. However, a comparison of the processing time showed the superiority of the algorithmic technique: it worked 8 times faster. The solution can be used for the segregation of objects on production lines and be implemented on a single-chip computer. (**Artificial Neural Network vs. Algorithmic Technique in Shape Classification Task**)*

Słowa kluczowe: detekcja kształtu, sztuczne sieci neuronowe, algorytm Canny'ego, komputer jednoukładowy
Keywords: shape detection, artificial neural networks, Canny algorithm, single-chip computer

Wstęp

Rozwój sztucznej inteligencji przyczynił się do znacznego postępu w dziedzinie nowych technologii, szczególnie w obszarze automatyki, robotyki i analizy obrazów [1]. Algorytmy uczenia maszynowego, stając się powszechnie stosowanym narzędziem, rewolucjonizują sposób, w jaki rozwiązujemy problemy związane z analizą danych wizyjnych [2.3].

Dotychczas, dominującym podejściem do analizy obrazów cyfrowych były algorytmy detekcji [4] i klasyfikacji [5], które mimo skuteczności w wielu zastosowaniach, wykazują ograniczenia w zakresie adaptacji do zmieniających się warunków środowiskowych oraz wrażliwość na zakłócenia powstałe w procesie akwizycji obrazu.

W ramach tych metod kluczową rolę odgrywały, algorytmy detekcji krawędzi, oparte na filtrach [6] i gradientach [7]. Metody takie jak algorytmy Canny'ego, i Sobela nadal pozostają skutecznymi narzędziami w wykrywaniu krawędzi obiektów na podstawie zmian tonalnych na obrazie. Algorytm Canny'ego, opiera się na wykrywaniu lokalnych maksimów w obrazie po filtracji gradientowej [8], podczas gdy algorytm Sobela dokonuje dwukrotnej filtracji obrazu, wykrywając krawędzie poziome i pionowe [9].

Rozwój sztucznych sieci neuronowych umożliwił rozpoznawanie wzorców na obrazach, niezależnie od warunków środowiskowych [10]. Sieci te, zbudowane z wielu warstw neuronów, potrafią samodzielnie wyodrębniać istotne cechy z danych treningowych, co pozwala na wykrywanie i klasyfikację obiektów z dużą dokładnością [11]. Efektywność tych rozwiązań jest ściśle związana z architekturą sieci, rozmiarem i zróżnicowaniem zbioru treningowego oraz z poziomem wytrenowania modelu. Im większy zbiór danych treningowych oraz im dokładniejszy model, tym lepiej sieć potrafi radzić sobie z różnorodnymi zadaniami.

W miarę postępu w miniaturyzacji elektroniki oraz coraz szerszego zastosowania Internetu rzeczy, wykorzystanie sztucznej inteligencji staje się coraz powszechniejsze nawet w małych układach cyfrowych [12]. W związku z tym istnieje rosnące zainteresowanie sztucznymi sieciami neuronowymi w celu zwiększenia skuteczności i wszechstronności analizy obrazów w systemach wizyjnych [13]. Przykładowymi zastosowaniami są systemy monitoringu, diagnostyka medyczna [14], analiza obrazów satelitarnych [15] czy też automatyczne wykrywanie obiektów na drogach dla systemów wspomagania kierowcy [16].

Celem niniejszej pracy było opracowanie modelu sztucznej splotowej sieci neuronowej (ang. CNN Convolutional Neural Network), który został wykorzystywany do klasyfikacji kształtu obiektów na obrazach. Zaproponowano architekturę sieci CNN, odpowiednio dostosowaną do specyfiki zadania. Model ten został poddany szczegółowej analizie i porównaniu jego skuteczność z dotychczas stosowanymi prostymi algorytmami klasyfikującymi, używającymi informacji o liczbie krawędzi obiektu.

W pracy dodatkowo zaproponowano wykorzystanie wytrenowanego modelu jako narzędzia do budowy systemu automatycznej detekcji i sortowania obiektów na linii produkcyjnej, z możliwością implementacji na komputerze jednoukładowym.

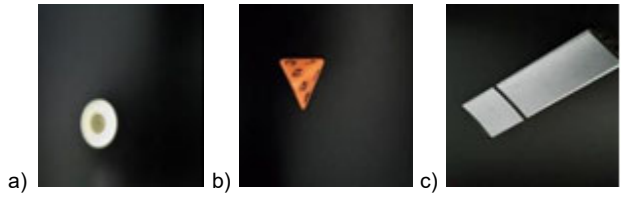
Artykuł został przygotowany w ramach współpracy Międzywydziałowego Studenckiego Koła Naukowego Decybel Politechniki Poznańskiej i Zespołu Szkół Techniczno-Elektronicznych w Kaliszu.

Metodyka badań

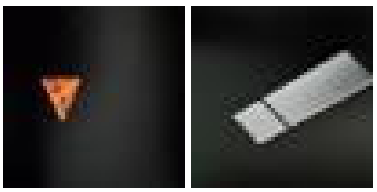
W celu przeprowadzenia treningu sieci neuronowej oraz oceny skuteczności algorytmów detekcji, przygotowano zbiór danych składający się z 2162 zdjęć reprezentujących przedmioty o kształtach: prostokąta, koła i trójkąta. Zdjęcia zostały wykonane przy użyciu aparatu fotograficznego o rozdzielczości 50 megapikseli, wbudowanego w smartfon zamontowany na statywie w odległości około 20 cm od obiektów. Przykładowe zdjęcia obiektów zostały zaprezentowane na rysunku 1.

Dla zróżnicowania charakterystyki zbioru, około 60% zdjęć przedstawiało obiekty na czarnym tle, co miało na celu ułatwienie ewentualnego procesu binaryzacji obrazu oraz zwiększenie kontrastu między obiektem a tłem. Przed analizą zdjęcia zostały przeskalowane do rozdzielczości

50x50 pikseli. Przykładowe efekt zmiany rozdzielczości przedstawiono na rysunku 2.



Rys. 1. Przykładowe zdjęcia obiektów poddanych detekcji:
a) czoło wtyczki zasilacza na ciemnym tle,
b) pomarańczowa czworościenna kostka na czarnym tle,
c) szary pendrive na czarnym tle.



Rys. 2. Zdjęcie po zmianie rozdzielczości obrazu. (na podstawie Rys.1 b, c)

Na potrzeby symulacji różnorodnych warunków oraz potencjalnych błędów w procesie akwizycji obrazów, poddano zbiór losowo wybranych 200 zdjęć działaniu filtru rozmywającego [17].

Całkowity zbiór danych, po odpowiednim przygotowaniu, został podzielony na trzy rozłączne podzbiory: treningowy, walidacyjny i testowy, w proporcji odpowiednio 60%, 20% i 20%. Ta standardowa praktyka umożliwia ocenę skuteczności modelu na niezależnych danych testowych, co zapewnia rzetelność wyników eksperymentu [18].

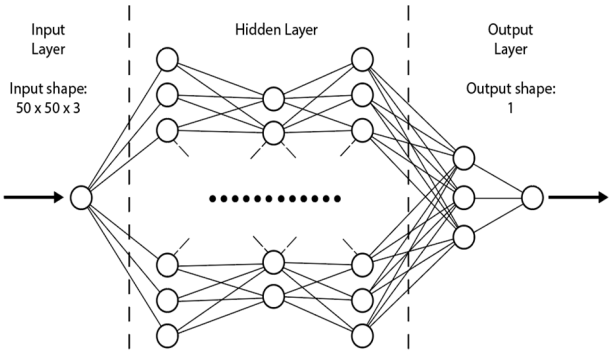
W celu konfrontacji rozwiązania wykorzysującego CNN zaprojektowano aplikację dokonującą klasyfikacji obiektów metodą algorytmiczną. Na wstępie skalowano zdjęcie do rozmiaru 300x300 pikseli w celu zmniejszenia ewentualnych ubytków na badanych elementach. W kolejnym kroku wykrywano krawędzie obiektów z wykorzystaniem algorytmu Canny'ego [19]. Następnie klasyfikowano obiekty na podstawie rozpoznanych krawędzi uwzględniając liczbę i wzajemne położenie krawędzi [20]. W celu poprawienia wydajności wykrywania krawędzi w drugiej iteracji testów rozbudowano program o możliwość aproksymacji liniowej wykrytych konturów. Program został napisany w języku Python 3.12 z wykorzystaniem biblioteki OpenCV [21].

Architektura sieci

Do opracowania modelu sieci neuronowej wykorzystano środowisko programistyczne Visual Studio Code 1.87.1 oraz język programowania Python 3.12. W połączeniu z bibliotekami Tensorflow [22], Keras [23] oraz NumPy [24], umożliwiło to implementację struktury konwolucyjnej sieci neuronowej (CNN, ang convolutional neural network). Do implementacji głębokiej sieci neuronowej oraz zoptymalizowania procesu trenowania użyto TensorFlow, jedno z najważniejszych narzędzi w dziedzinie uczenia maszynowego. Biblioteka Keras zapewniła możliwość implementacji modeli oraz narzędzia do prototypowania i testowania różnych architektur sieci.

Testy przeprowadzono na komputerze o konfiguracji:
- procesor: i5-6400 3,3 GHz
- karta graficzna: GTX 1060 6GB
- pamięć RAM: 16 GB DDR4, 2400 MHz
- system operacyjny Windows 10.

Dane wejściowe (skalowane zdjęcia) reprezentowane są w postaci macierzy o rozmiarze 50x50x3. Trzy warstwy macierzy reprezentują kolory obrazu R, G i B.



Rys. 3. Schemat zrealizowanej CNN (na podstawie [25])

Opracowany model CNN złożony jest z 8 warstw. Rozkład warstw zaprojektowanej sieci wygląda następująco (rysunek 3):

a) Warstwa konwolucyjna (Conv2D) z 32 filtrami o rozmiarze 3x3 i funkcją aktywacji ReLU.
b) Warstwa max pooling (MaxPooling2D) o rozmiarze 2x2.
c) Warstwa konwolucyjna (Conv2D) z 64 filtrami o rozmiarze 3x3 i funkcją aktywacji ReLU.
d) Warstwa max pooling (MaxPooling2D) o rozmiarze 2x2.
e) Warstwa konwolucyjna (Conv2D) z 64 filtrami o rozmiarze 3x3 i funkcją aktywacji ReLU.
f) Warstwa spłaszczająca (Flatten), która konwertuje dane wejściowe do wektora.
g) Warstwa gęsto połączona (Dense) z 64 neuronami i funkcją aktywacji ReLU.
h) Warstwa gęsto połączona (Dense) z 3 neuronami i funkcją aktywacji softmax

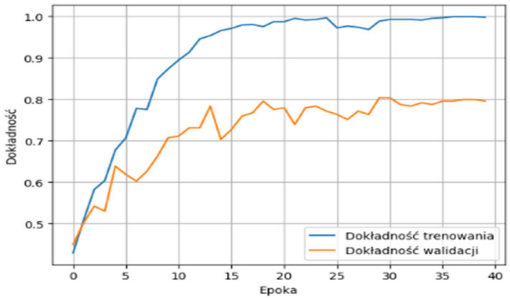
Wytrenowaną sieć neuronową zaimplementowano na mikrokomputerze Raspberry Pi 4 model B 4 GB z systemem operacyjnym Raspberry Pi OS (Linux 6.1.63).

Skuteczność trenowania modelu sztucznej inteligencji została zmierzona w każdej epoce etapu uczenia. Pomiar został przeprowadzony w celu uzyskania wartości straty uczenia, straty walidacyjnej i dokładności klasyfikacji.

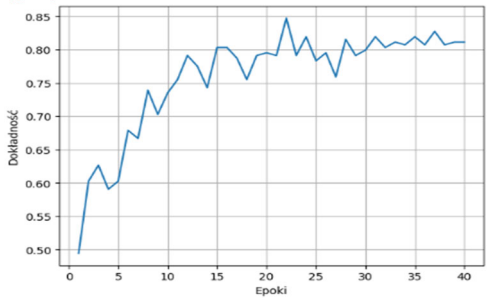
Wyniki dokładności klasyfikacji do trzech zbiorów uzyskane z opracowanego modelu CNN dla zbioru treningowego i walidacyjnego przedstawia rysunek 4, dla zbioru testowego rysunek 5. Na rysunku 6 przedstawiono straty trenowania i walidacji.

Wyniki przeprowadzonych badań wskazują na ustabilizowanie się dokładności detekcji kształtu od około 30-tej epoki uczenia (rysunek 4). W celu potwierdzenia poprawności działania sieci, przeprowadzono testy skuteczności detekcji na zbiorze testowym (rysunek 5). Wyniki tych testów potwierdziły dokładność detekcji na poziomie porównywalnym z dokładnością osiągniętą na zbiorze walidacyjnym.

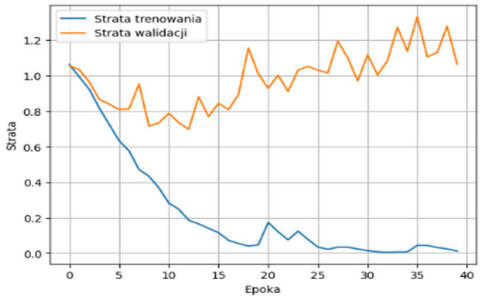
Dokładność w zbiorze testowym powyżej 80% uzyskiwana jest już od 15-tej epoki, a wartość maksymalną 85% osiągnięto dla 22 epoki. Błędy detekcji mogą być wynikiem charakteru oraz rozmiaru zbioru danych treningowych. Widać jednak, że dalsze trenowanie sieci nie przynosi istotnej poprawy wyników.



Rys. 4. Dokładności klasyfikacji dla zbiorów treningowego i walidacyjnego



Rys. 5. Dokładności klasyfikacji dla zbioru testowego



Rys. 6. Wykres straty trenowania i walidacji

Detekcja krawędzi algorytmem Canny'ego

W celu porównania uzyskanej przez CNN dokładności detekcji, na próbce 200 zdjęć przetestowano algorytm Canny'ego, do wykrywania krawędzi. W wyniku działania algorytmu obrazy testowe zostały podzielone na kategorie w zależności od liczby wykrytych krawędzi. Algorytm umożliwił precyzyjne określenie położenia obiektów na podstawie wyznaczonych krawędzi, osiągając skuteczność klasyfikacji na poziomie 69%. Badania przeprowadzono po dokonaniu aproksymacji liniowej wykrytych konturów.

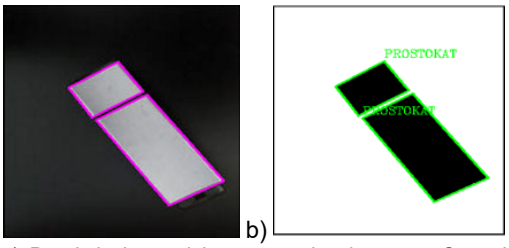
W drugiej wersji testu obrazy wejściowe zostały poddane procesowi binaryzacji. Wartość progu binaryzacji została ustalona przy użyciu techniki progowania adaptacyjnego. Zastosowanie progowania histerezowego przyczyniło się do poprawy dokładności detekcji krawędzi, jednakże nie wpłynęło bezpośrednio na proces klasyfikacji ze względu na powstanie dodatkowych konturów, zwanych pasożytniczymi. W celu usprawnienia klasyfikacji przeprowadzono modyfikację wartości progowania poprzez ręczne dostosowanie ich o wartość wynoszącą ±2%. Regulacja wartości progu w określonym zakresie pozwoliła na udokładnienie procesu binaryzacji, uwzględniając indywidualne cechy poszczególnych obrazów.

Najlepsze rezultaty badań wykazały skuteczność klasyfikacji na poziomie 77%. Dostosowanie progów detekcji indywidualnie do każdego z badanych zdjęć umożliwiłoby uzyskanie lepszych wyników detekcji i klasyfikacji, ale wymagałoby użycia bardziej zaawansowanych procedur. Zestawienie wykrytych kształtów zostało przedstawione w tabeli 1.

Dodatkowo, program został rozszerzony o funkcję prezentacji obrazu przed i po binaryzacji, kolorując krawędzie. Przykład analizy wykrycia krawędzi, binaryzacji i klasyfikacji obiektów został zaprezentowany na rysunkach 7 a) i b).

Tabela 1. Wyniki klasyfikacji z wykorzystaniem algorytmu Canny'ego i z etapem binaryzacji.

Kształt	Wszystkie	Wykryte poprawnie
Prostokąt	73	64
Koło	66	48
Trójkąt	61	42
Łącznie	200	154



a) b)

Rys. 7. a) Detekcja krawędzi przy użyciu algorytmu Canny'ego - zdjęcie przed klasyfikacją, b) Wynik klasyfikacji kształtu z uwzględnieniem binaryzacji zdjęcia.

Wprowadzenie dodatkowego filtru rozmywającego, który w założeniach miał rozmyć krawędzie, przyniosło zmniejszenie skuteczności działania klasyfikatora do poziomu 70,5% (uwzględniając najlepszy wynik w zakresie histerezy). Spadek dokładności został przedstawiony w tabeli 2.

Tabela 2. Wyniki klasyfikacji obrazów rozmytych z etapem binaryzacji.

Kształt	Wszystkie	Wykryte poprawnie
Prostokąt	73	59
Koło	66	46
Trójkąt	61	36
Łącznie	200	141

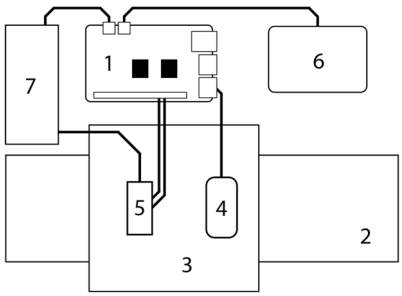
Klasyfikacja obiektów z użyciem CNN oraz metoda algorytmiczna zostały także porównane pod kątem ich efektywności czasowej. W celu dokonania pomiaru czasu wymaganego do klasyfikacji krawędzi, przeprowadzono eksperyment na próbie składającej się ze 100 obrazów, a sam proces powtórzono 10-krotnie w każdym cyklu. Łączny czas wykonania każdej próby został zaprezentowany w tabeli 3.

Na podstawie uzyskanych wyników można stwierdzić, że efektywność czasowa rozwiązania z CNN jest ponad 8-krotnie niższa.

Tabela 3. Wyniki pomiarów czasu klasyfikacji algorytmów.

Nr próby	Czas klasyfikacji próby aplikacją z algorytmem Canny'ego [s]	Czas klasyfikacji próby przy użyciu sieci neuronowych [s]
1	2,718	25,814
2	2,748	22,630
3	2,715	22,219
4	2,776	23,022
5	2,826	21,645
6	2,783	21,158
7	2,825	21,523
8	2,770	23,130
9	2,814	21,590
10	2,855	20,855
Średnia:	2,783 ± 0,047	22,358 ± 1,436

Obecnie prowadzone są prace nad wdrożeniem rozwiązania do komputera jednoukładowego Raspberry Pi 4B w celu opracowania systemu sortującego i wykrywającego wady obiektów na linii produkcyjnej. Schemat blokowy systemu został zaprezentowany na rysunku 8.



Rys. 8. Schemat systemu detekcji i sortowania obiektów: 1. Mikrokomputer Raspberry Pi 4B, 2. Linia produkcyjna, 3. Ciemnia z oświetlaczem centralnym 4. Kamera, 5. Układ manipulatora, 6. Monitor do podglądu w czasie rzeczywistym, 7. Zasilanie

Podsumowanie

Detekcja przy użyciu algorytmu Canny'ego, ze względu na charakter danych wejściowych oraz kontury kształtów, okazała się nieprecyzyjna i wymagała ulepszenia w celu zwiększenia dokładności. W tym celu przeprowadzono binaryzację wykorzystującą progowanie adaptacyjnie, co pozwoliło na bardziej precyzyjne odwzorowanie konturów obiektów. Wyniki eksperymentów wyraźnie pokazały, że algorytm Canny'ego osiągnął większą skuteczność w detekcji kształtów prostokątnych i trójkątnych. Natomiast w przypadku okręgów pojawiły się trudności głównie związane z problemem niezrównoważonej aproksymacji oraz kształtu pierwotnego.

Analiza wyników eksperymentów wskazuje, że zaprojektowany model CNN osiągnął zadowalające rezultaty w zadaniu klasyfikacji kształtów obiektów. Niemniej jednak, istnieje potencjał do dalszej optymalizacji modelu, szczególnie poprzez zwiększenie i urozmaicenie zbioru danych treningowych oraz dostrojenie hiperparametrów sieci. Aby poprawić efektywność czasową, konieczne jest zoptymalizowanie rozwiązania opartego na CNN. Optymalizacja zarówno wydajności klasyfikacji jak i czasu działania, może być trudna w implementacji dla obecnej architektury rozwiązania.

Dzięki niskim kosztom, łatwej dostępności oraz możliwości dostosowania do różnych zastosowań, implementacja wytrenowanej sieci neuronowej na mikrokomputerze Raspberry Pi 4B otwiera szerokie możliwości wykorzystania w automatyzacji procesów produkcyjnych oraz poprawie efektywności i jakości w różnych dziedzinach przemysłu.

Publikację sfinansowano z subwencji badawczej 0211/SBAD/0224.

***Autorzy**: mgr inż. Jakub Konopiński, mgr inż. Krystyna Baran, Jakub Łazik, Zespół Szkół Techniczno-Elektronicznych w Kaliszu ul. Częstochowska 99-105, 62-800 Kalisz, mgr inż. Piotr Góral, dr inż. Paweł Pawłowski, Zakład Układów Elektronicznych i Przetwarzania Sygnałów, Instytut Automatyki i Robotyki, Wydział Automatyki, Robotyki i Elektrotechniki, Politechnika Poznańska, ul. Jana Pawła II 24, 60-965 Poznań ,E-mail: piotr.goral@.put.poznan.pl, pawel.pawlowski@put.poznan.pl,*

LITERATURA

[1] Hankyu M., Optimal Edge-Based Shape Detection, *IEEE transactions on image processing*, Vol. 11, NO. 11, (2002)
[2] Czechowicz A., Mikrut Z., Selekcja podobrazów dla potrzeb dopasowywania zdjęć lotniczych oparta na histogramach gradientu i sieci Neuronowej, *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, Vol. 17a, (2007)
[3] Dena Nadir G., Hashem B., Brain Tumor Detection Using Shape features and Machine Learning Algorithms, *International Journal of Scientific & Engineering Research*, Vol. 6, 12, (2015)
[4] Czechowicz A., Mikrut Z., Analiza przydatności algorytmów detekcji krawędzi w zastosowaniach fotogrametrii bliskiego zasięgu, *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, Vol. 16, (2006)
[5] Kondej M., Putz B., Szybki algorytm dopasowania obrazów dla potrzeb fuzji w czasie rzeczywistym, *Pomiary Automatyka Robotyka* 11/(2010)
[6] Wahyu S., Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak, *ILKOM Jurnal*, Vol. 12 No. 2, (2020), 112-120
[7] Owotogbe J. S., Ibiyemi T. S., Adu B. A., Edge Detection Techniques on Digital Images - A Review, *International Journal of Innovative Science and Research Technology*, Vol 4, 11, (2019)
[8] Zacniewski A., Detekcja kodów kreskowych w obrazach za pomocą filtrów gradientowych i transformacji morfologicznych, *LOGISTYKA* 4, (2015)
[9] Gao W., Zhang X., Yang L., & Liu H., An improved Sobel edge detection, *3rd International Conference on Computer Science and Information Technology*, 5, (2010), 67-71
[10] Marleen de Bruijne,, Machine learning approaches in medical image analysis: From detection to diagnosis, *Medical Image Analysis*, Vol. 33, October (2016), 94-97
[11] Góral P., Pawłowski P., Piniarski K., Dąbrowski, A. Multi-Agent Vision System for Supporting Autonomous Orchard Spraying. *Electronics* (2024), 13, 494. https://doi.org/10.3390/electronics13030494
[12] Ahmad Z., Shahid Khan A., Nisar K., Haider I., Hassan R., Haque M.R., Tarmizi S., Rodrigues J.J.P.C., Anomaly Detection Using Deep Neural Network for IoT Architecture. *Appl*. Sci. (2021), 11, 7050
[13] Yoo, Hyeon-Joong, Deep Convolution Neural Networks in Computer Vision: A Review. IEIE Transactions on Smart Processing and Computing. *The Institute of Electronics Engineers of Korea*, vol. 4, no. 1, (2015)
[14] Marciniak T., Stankiewicz A., Zaradzki P., Neural Networks Application for Accurate Retina Vessel Segmentation from OCT Fundus Reconstruction. *Sensors* (2023), 23, 1870. https://doi.org/10.3390/s23041870
[15] Giuffrida G., et al., The Φ-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation, in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, (2022), 1-14
[16] Pawłowski P., Piniarski K., Dąbrowski A., Highly Efficient Lossless Coding for High Dynamic Range Red, Clear, Clear, Clear Image Sensors. *Sensors* (2021), 21, 653. https://doi.org/10.3390/s21020653
[17] Abdul M, Lateef R., Expansion and Implementation of a 3x3 Sobel and Prewitt Edge Detection Filter to a 5x5 Dimension Filter. (2019)
[18] Géron A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, *O'Reilly Media*, (2019)
[19] Ahmed Shihab Ahmed, Comparative study among sobel, prewitt and Canny edge detection operators used in image Processing, *Journal of Theoretical and Applied Information Technology*, Vol.96. No 19, (2018)
[20] Piniarski K., Pawłowski P., Dąbrowski A., Video Processing Algorithms for Detection of Pedestrians, Comput. Methods Sci. Technol. *CMST*, vol. 21, no. 3, (2015), 141–150, doi: 10.12921/CMST.2015.21.03.005.
[21] Dokumentacja techniczna biblioteki OpenCV: https://docs.opencv.org/4.x/index.html (dostęp z dnia 11.02.2024)
[22] Dokumentacja techniczna biblioteki TensorFlow: https://www.tensorflow.org/api_docs/python/tf/all_symbols (dostęp z dnia 11.02.2024)
[23] Dokumentacja techniczna biblioteki Keras: https://keras.io/api/ (dostęp z dnia 11.02.2024)
[24] Dokumentacja techniczna biblioteki NumPy: https://numpy.org/doc/ (dostęp z dnia 11.02.2024)
[25] Pettit R., Pettit F., Robert & Cheng, Chao & Amos, Christopher. Artificial intelligence, machine learning, and deep learning for clinical outcome prediction. *Emerging Topics in Life Sciences*, 5, (2021) doi: 10.1042/ETLS20210246.