Julian BALCEREK[1], Adam KONIECZKA[2], Paweł PAWŁOWSKI[3], Cyprian DANKOWSKI[4], Mateusz FIRLEJ[4], Piotr FULARA[4]

Politechnika Poznańska, Wydział Automatyki, Robotyki i Elektrotechniki (1, 2, 3, 4)
ORCID: 1. 0000-0003-1317-9866; 2. 0000-0002-0362-3006; 3. 0000-0001-5373-5148

doi:10.15199/48.2022.09.47

Automatyczne rozpoznawanie parametrów kół pojazdu

Streszczenie. *W niniejszym artykule zaproponowano system do automatycznego rozpoznawania parametrów kół pojazdu wykorzystujący sztuczne sieci neuronowe. Przeprowadzono trening, walidację i testy na przygotowanej bazie obrazów pod kątem rozpoznawania oznaczeń opon samochodów osobowych. Oprócz oznaczeń, wykrywane są także uszkodzenia kół, poziom ciśnienia powietrza i stan bieżnika opon. Do wygodnej obsługi systemu przygotowano aplikację mobilną.*

Abstract*. This article proposes a system for the automatic recognition of vehicle wheel parameters, which uses artificial neural networks. Training, validation and tests were conducted on the prepared image database for the recognition of passenger car tire markings. Our system also detects damage to the wheels as well as the level of air pressure and the condition of the tire tread. A mobile application has been prepared for the convenient operation of the system. (**Automatic recognition of vehicle wheel parameters**).*

Słowa kluczowe: rozpoznawanie parametrów kół, system wizyjny, aplikacja mobilna, sztuczne sieci neuronowe.
Keywords: recognition of parameters of the vehicle wheels, vision system, mobile application, artificial neural networks.

1. Wprowadzenie

Koła pojazdu stanowią elementy, których stan techniczny jest kluczowy z punktu widzenia bezpieczeństwa. Przykładowo, w Polsce w 2019 roku zły stan ogumienia stanowił ponad 30% defektów wykrytych w autach biorących udział w wypadkach drogowych spowodowanych złym stanem technicznym pojazdów [1]. Zagadnienie bieżącej kontroli wizyjnej stanu opon i felg, uzupełniającej regularną diagnostykę, stanowi więc istotny element obsługi pojazdu przez użytkownika, a sam użytkownik mógłby korzystać z odpowiednich systemów wspomagających opartych na czujnikach wizyjnych.

Przykładowy system wizyjny do odczytywania oznaczeń na oponach, w którym do rozpoznawania tekstu zastosowano kaskadę klasyfikatorów konwolucyjnych sieci neuronowych (CNN), powstał w formie stacjonarnego urządzenia [2]. Można posłużyć się bardziej praktyczną, z punktu widzenia użytkownika, aplikacją mobilną, o nazwie Tire Check [3]. Posiada ona jednak funkcjonalność ograniczoną jedynie do zbadania opony z przodu i to wyłącznie pod względem określenia wysokości bieżnika. Z kolei w artykule [4], opisującym kompleksową aplikację mobilną do wspierania użytkowników w obsłudze pojazdu (zawierającą możliwość rozpoznawania oznaczeń pojazdu, kontrolek na desce rozdzielczej i elementów pod maską), nie uwzględniono rozpoznawania oznaczeń opon.

W niniejszej pracy zaproponowano system wraz z dedykowaną aplikacją mobilną do automatycznego rozpoznawania parametrów kół pojazdu. Rozpoznawanie symboli i cech opony wykorzystuje sztuczne sieci neuronowe. Badania eksperymentalne wykonano na specjalnie przygotowanej do tego bazie obrazów.

2. Baza obrazów kół pojazdów

Bazę zdjęć (opisaną w Tabeli 1) zawierającą różne koła samochodowe w różnym stanie przygotowano w celu treningu sztucznych sieci neuronowych oraz przeprowadzenia testów ich skuteczności. Ze względu na przeznaczenie, obrazy podzielono na: część zawierającą oznaczenia opon, uszkodzenia kół, zróżnicowanie ciśnienia powietrza oraz stan bieżnika opon. Zdjęcia zostały zarejestrowane w różnych warunkach atmosferycznych oraz oświetleniowych. Fotografie przeznaczone do rozpoznawania oznaczeń, wykrywania uszkodzeń oraz klasyfikacji poziomu ciśnienia powietrza w oponie wykonano od boku, natomiast zdjęcia bieżnika – na wprost opony. Uzyskane obrazy podzielono na grupy treningowe, walidacyjne i testowe.

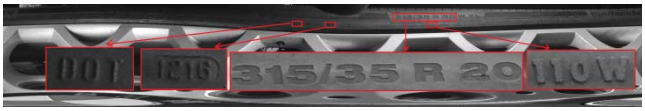
Tabela 1. Baza obrazów kół pojazdów

cecha	liczba zdjęć	uwagi
oznaczenia opon	174	z 13 różnych pojazdów; trening: 124, walidacja: 8, test: 42
uszkodzenia kół	238	222 z przynajmniej jednym widocznym defektem; trening: 184, walidacja: 22, test: 32
poziom ciśnienia powietrza w oponach	568	423 z odpowiednim ciśnieniem, 145 z niewłaściwym; trening: 418, walidacja: 73, test: 77
wysokość bieżnika opon	813	656 z właściwą wysokością bieżnika, 157 z niewłaściwą; trening: 601, walidacja 105; test: 107
łącznie:	1793	–

Przykładowe zdjęcia do rozpoznawania oznaczeń opon pokazano na rysunku 1.

Rys. 1. Przykładowe zdjęcia opon do rozpoznawania oznaczeń (z powiększonymi oznaczeniami)

Dodatkowo, w części bazy przygotowanej do przetwarzania przez sieci neuronowe zamieszczono obrazy po zmianie układu współrzędnych (rys. 2). Powstały one podczas wstępnego przetwarzania na potrzeby treningu, walidacji i testów.



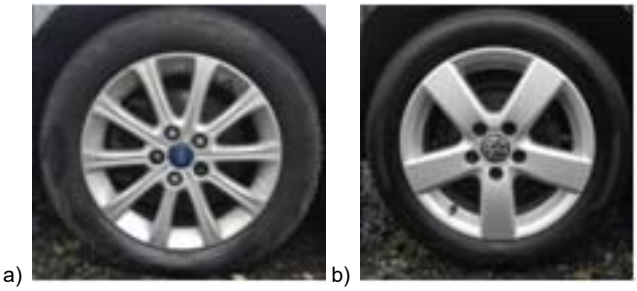
Rys. 2. Przykładowe zdjęcie napisu na oponie z bazy po zmianie układu współrzędnych (z powiększonymi oznaczeniami)

W części bazy przeznaczonej do rozpoznawania uszkodzonych kół, oprócz 16 zdjęć kół bez uszkodzeń, przeznaczonych do rozpoznania fałszywej detekcji, wszystkie pozostałe zawierały przynajmniej jeden widoczny defekt (zarów-

no na oponach jak i felgach). Przykładowe zdjęcia kół z uszkodzeniami przedstawiono na rysunku 3.

Rys. 3. Przykładowe zdjęcia uszkodzonych kół

Fotografie związane z poziomem ciśnienia powietrza w oponach powstały w taki sposób, aby obraz obejmował cały obwód koła pojazdu. Przyjęto założenie o zakwalifikowaniu opony do grupy o zbyt niskiej wartości ciśnienia tylko wtedy, kiedy taki stan był wyraźnie widoczny (rys. 4).



Rys. 4. Przykładowe zdjęcia kół: a) ze zbyt niskim, b) z właściwym ciśnieniem powietrza w oponie

Zdjęcia przeznaczone do klasyfikacji poziomu bieżnika (rys. 5) wykonywano w taki sposób, aby jego stan był najlepiej był widoczny. Za opony należące do klasy zużytych uznano te z bieżnikiem, którego wysokość osiągnęła lub zbliżała się do wysokości znacznika zużycia bieżnika (TWI, z ang. Tread Wear Indicator – 1,6 mm).

Rys. 5. Przykładowe zdjęcia opon: a) ze zużytym, b) z właściwym stanem bieżnika

3. Architektura sztucznych sieci neuronowych

Zaroponowany system przygotowano w oparciu o m.in.: język programowania Python [5] i środowisko programistyczne PyCharm [6], biblioteki OpenCV [7], TensorFlow [8] i Keras [9], język programowania Java [10] oraz środowisko programistyczne Android Studio [11].

Do zadania rozpoznawania oznaczeń opon wytrenowano sieć YOLOv3 [12]. Tu wykorzystano platformę Google Colaboratory, umożliwiającą wykonywanie obliczeń w chmurze [13]. Przed przesłaniem danych do sieci neuronowej zamieniano układ współrzędnych, aby oznaczenia umieszczone na okręgu przedstawić w jednej linii. Następnie, podczas skanowania obrazu, zwracany był typ wykrytego pola tekstowego oraz jego współrzędne i prawdopodobieństwo występowania. Z racji tego, że obraz był przetwarzany fragmentami, wynik skanowania podlegał filtracji pod kątem powtarzania się wykrytych oznaczeń. Jeżeli oznaczenia się powtarzały, wybierane były te o większym prawdopodobieństwie występowania. Następnie, druga sieć neuronowa służyła do wykrywania znaków na wskazanych polach tekstowych.

W przypadku rozpoznawania uszkodzeń kół, ze względu na ograniczenia sprzętowe, wszystkie zdjęcia przeskalowano do rozdzielczości 640×640 pikseli. Przed procesem treningu sieci ręcznie oznaczono uszkodzenia i nadano im etykiety określające rodzaj defektu – wykorzystano do tego program LabelImg [14]. Uszkodzenia kół rozpoznawano z użyciem sieci SSD MobileNetv2. Trening tej sieci został wykonany w wirtualnym środowisku oprogramowania Anaconda [15].

W przypadku klasyfikacji poziomu powietrza w oponach, do treningu sieci wykorzystano zdjęcia kół przycięte w taki sposób, aby zminimalizować wpływ tła. Model sieci wytrenowany został przy użyciu skryptu napisanego w języku programowania Python, w środowisku programistycznym PyCharm [6]. Użyto do tego bibliotek TensorFlow i Keras [16].

Sieć neuronowa do predykcji poziomu zużycia bieżnika powstała z wykorzystaniem tych samych narzędzi programistycznych, co sieć do klasyfikacji poziomu ciśnienia powietrza [16]. Ze względu na stosunkowo niewielką bazę zdjęć prezentujących to zagadnienie, w celu poprawy dokładności treningu sieci, zdecydowano się na automatyczne rozszerzenie bazy treningowej (z ang. data augmentation). Pozwoliło to na wielokrotne użycie tych samych zdjęć, ale obróconych o pewien losowy kąt z przedziału od -18° do 18°.

4. Eksperyment

Dla ułatwienia przeprowadzenia testów skuteczności wszystkich przygotowanych sieci neuronowych, przygotowano w języku Python narzędzie, które pozwalało na losowy wybór zdjęć z przygotowanej bazy. Opracowany program dawał możliwość wyboru liczby testów oraz obrazów osobno dla każdego z typów klasyfikacji.

Narzędzie do testów zostało podzielone na cztery moduły. Każdy z nich odpowiadał za inny rodzaj klasyfikacji. Dla każdego modułu wykorzystano osobną część bazy zdjęć do testowania.

Pierwszy z nich służył do odczytu parametrów opony, drugi do weryfikacji uszkodzeń opony (narzędzie to wybierało zdjęcia z odpowiedniej części bazy oraz rysowało na nich prostokąty w miejscach uszkodzenia – wraz z klasyfikacją znalezionego defektu). Trzeci moduł był odpowiedzialny za sprawdzenie ciśnienia w oponie – wyniki tego testu wyświetlane były w postaci typowej macierzy pomyłek, która jest często używana do oceny jakości klasyfikacji. Moduł weryfikujący stan bieżnika działał analogicznie do modelu sprawdzającego ciśnienie, z tą różnicą, że zdjęcia były wybierane z innej części bazy (spośród zdjęć wykonanych z przodu opony).

Moduły do odczytu parametrów opony oraz sprawdzenia uszkodzeń koła były testowane częściowo manualnie, gdyż ostatecznie było to mniej czasochłonne niż przygotowanie etykiet dla testu automatycznego.

Testy skuteczności przeprowadzono z użyciem opracowanego oprogramowania. Dotyczyły one rozpoznawania oznaczeń opon, wykrywania uszkodzeń kół oraz klasyfikacji poziomu ciśnienia powietrza i zużycia bieżnika opon (wyniki prezentują Tabele 2–5).

W wyniku przeprowadzonych na przygotowanej bazie obrazów testów uzyskano ponad 95% skuteczności rozpo-

znawania znaków w odniesieniu do liczby prawidłowo wykrytych znaków na oponie. Skuteczność wykrywania uszkodzeń opon wyniosła ponad 70%, a w przypadku uszkodzeń felg była na poziomie 90%. Dokładność klasyfikacji ciśnienia w oponie wyniosła około 78%, a dokładność klasyfikacji wysokości bieżnika – ponad 84%.

Tabela 2. Wyniki rozpoznawania oznaczeń na oponach

parametr	wartość
liczba pól tekstowych (#1)	305
liczba poprawnie wykrytych pól tekstowych (#2)	223
liczba fałszywie wykrytych pól tekstowych	12
liczba znaków w poprawnie wykrytych polach tekstowych (#3)	402
liczba poprawnie wykrytych znaków w poprawnie wykrytych polach tekstowych (#4)	386
liczba fałszywie wykrytych znaków w poprawnie wykrytych polach tekstowych	1
liczba poprawnie rozpoznanych znaków w poprawnie wykrytych polach tekstowych (#5)	368
100%·(#2)/(#1)	73,12%
100%·(#4)/(#3)	96,02%
100%·(#5)/(#4)	95,34%

Tabela 3. Wyniki wykrywania uszkodzeń kół

parametr	wartość
liczba zdjęć	32
liczba zdjęć z uszkodzeniami	16
liczba zdjęć z prawidłowo wykrytym przynajmniej jednym uszkodzeniem	14
liczba zdjęć bez uszkodzeń, na których fałszywie wykryto przynajmniej jedno uszkodzenie	0
liczba wszystkich uszkodzeń na wszystkich zdjęciach (#1)	37 (opony: 17, felgi: 20)
liczba prawidłowo wykrytych uszkodzeń (#2)	30 (opony: 12, felgi: 18)
liczba fałszywie wykrytych uszkodzeń	0
100%·(#2)/(#1)	81,08%
100%·(#2)/(#1) dla opon	70,59%
100%·(#2)/(#1) dla felg	90,00%

Tabela 4. Wyniki klasyfikacji poziomu ciśnienia w oponach

parametr	wartość
liczba obrazów testowych	77
pozytywne (niewłaściwy poziom ciśnienia)	27
negatywne (właściwy poziom ciśnienia)	50
dokładność	77,92%

Tabela 5. Wyniki klasyfikacji wysokości bieżnika

parametr	wartość
liczba obrazów testowych	107
pozytywne (niewłaściwa wysokość bieżnika)	21
negatywne (właściwa wysokość bieżnika)	86
dokładność	84,11%

W wyniku przeprowadzonych na przygotowanej bazie obrazów testów uzyskano ponad 95% skuteczności rozpoznawania znaków w odniesieniu do liczby prawidłowo wykrytych znaków na oponie. Skuteczność wykrywania uszkodzeń opon wyniosła ponad 70%, a w przypadku uszkodzeń felg była na poziomie 90%. Dokładność klasyfikacji ciśnienia w oponie wyniosła około 78%, a dokładność klasyfikacji wysokości bieżnika – ponad 84%.

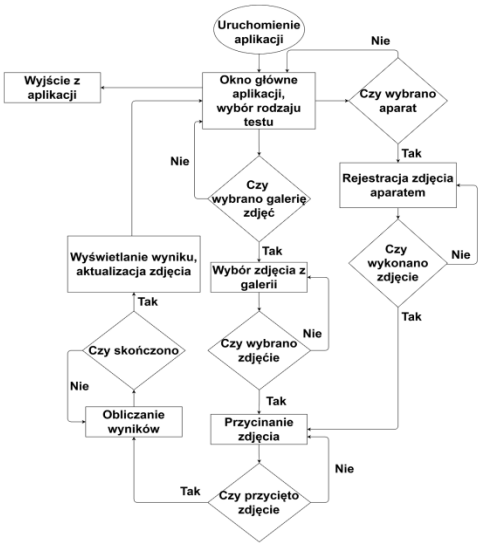
5. Aplikacja mobilna

Do wygodnej obsługi systemu przygotowano dedykowaną aplikację mobilną działającą w systemie operacyjnym Android. Dla aplikacji określono następujące wymagania funkcjonalne: możliwość wykonania zdjęcia z poziomu aplikacji, możliwość przycięcia zdjęcia (aby dopasować obraz do krawędzi opony), możliwość wyboru zdjęcia z galerii, możliwość sprawdzenia uszkodzeń i wartości ciśnienia oraz oznaczeń opony, wyświetlenie wykrytych parametrów oraz liczby uszkodzeń koła, zaznaczenie oraz opis uszkodzeń na zdjęciu koła.

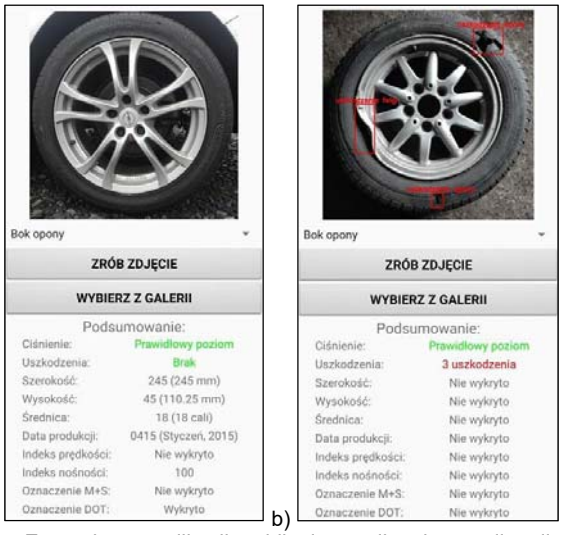
Określono także pozostałe wymagania implementowanej aplikacji: urządzenie mobilne z zainstalowanym systemem Android 6.0 lub wyższym oraz posiadające aparat fotograficzny wraz z podstawowym oprogramowaniem do jego obsługi, aplikacja ma działać niezależnie od wielkości ekranu, brak wymagań dotyczących czasu działania, brak konieczności dostępu do sieci Internet oraz przynajmniej 1 GB wolnej pamięci wewnętrznej urządzenia.

Schemat działania aplikacji wraz z czynnościami wykonywanymi przez użytkownika przedstawiono na rysunku 6.

Po uruchomieniu aplikacji użytkownik ma możliwość wyboru rodzaju testu ze względu na widok opony (od boku lub z przodu). Następnie uruchamia aparat lub wybiera zdjęcie z galerii urządzenia i uzyskany obraz przycina do właściwych wymiarów. Później uruchamia obliczenia – ich wynik jest wyświetlany najpóźniej w ciągu kilkunastu sekund.



Rys. 6. Schemat blokowy działania przygotowanej aplikacji mobilnej



a) b)

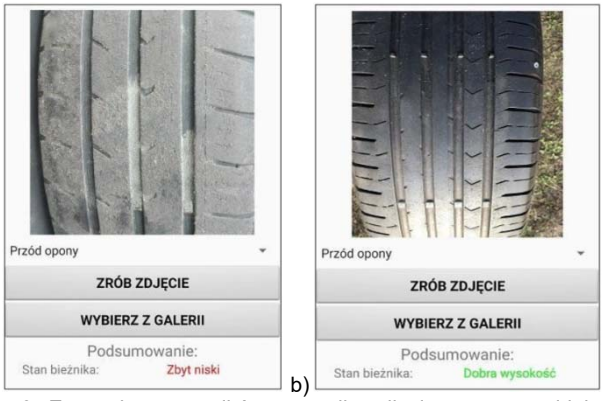
Rys. 7. Zrzut ekranu aplikacji mobilnej z wynikami testu dla zdjęcia koła w widoku z boku: a) z odczytanymi parametrami b) z zaznaczonymi uszkodzeniami

Przykładowe zrzuty ekranu aplikacji ukazujące wyniki testu dla zdjęcia koła w widoku z boku przedstawiono na rysunku 7a. Do wybranych wykrytych oznaczeń aplikacja automatycznie dopisuje podpowiedzi dla użytkownika. Na

prezentowanym przykładzie odczytana wartość wysokości równa 45 oznacza 110,25 mm. Brak wykrycia oznaczenia „M+S” informuje o tym, że opona nie jest przeznaczona do trudnych warunków atmosferycznych. Wykrycie prawidłowego poziomu ciśnienia powietrza w oponie zostało zaznaczone kolorem zielonym. Stan bieżnika również został uznany za prawidłowy i oznaczony odpowiednim kolorem.

Podsumowanie, w którym znajdują się wszystkie wyniki, jest przesuwaną listą. Takie rozwiązanie jest wygodniejsze dla użytkowników posiadających urządzenia z mniejszym ekranem. Przykładowy test wykonany dla uszkodzonego koła został przedstawiony na rysunku 7b.

Zrzut ekranu aplikacji przedstawiony na rysunku 8a, pokazuje przykładowy wynik testu sprawdzającego oponę w widoku z przodu. Dla zdjęcia z przykładu stan bieżnika został oceniony jako nieprawidłowy, co zostało wyróżnione kolorem czerwonym. Drugi przykładowy wynik testu dla zdjęcia opony w widoku z przodu został przedstawiony na rysunku 8b. Dla zdjęcia z tego przykładu stan bieżnika został oceniony jako prawidłowy, co zostało wyróżnione kolorem zielonym.



a) b)

Rys. 8. Zrzut ekranu wyników testu dla zdjęcia opony w widoku z przodu: a) zbyt niski poziom bieżnika, b) właściwy poziom bieżnika

6. Podsumowanie i wnioski

Wyniki testów pokazują, że można, z co najmniej dobrą skutecznością, odczytywać znaki na oponach i rozpoznawać stan techniczny kół pojazdów. Warte podkreślenia jest uzyskanie takich wyników pomimo niewielkiego kontrastu pomiędzy oznaczeniami opon a resztą opony (oznaczenia nie są nanoszone osobnym kolorem).

Wykrywanie uszkodzeń kół ma kluczowe znaczenie z punktu widzenia bezpieczeństwa, zwłaszcza, że nie ma innej możliwości automatycznego wykrywania takich uszkodzeń. Z racji tego, że w pojazdach powszechnie są stosowane czujniki ciśnienia, niższa skuteczność klasyfikacji poziomu ciśnienia nie powinna być znaczącym problemem. Należy podkreślić, że wyniki uzyskano pomimo występowania zmiennego oświetlenia i pewnych różnic w kątach, pod którymi były wykonywane zdjęcia.

W przyszłości skuteczność działania oprogramowania można zwiększyć między innymi poprzez wykonanie treningu sieci neuronowej na większej bazie obrazów. Opracowana aplikacja mobilna mogłaby być ważnym narzędziem wspomagającym użytkownika w obsłudze pojazdu.

Badania sfinansowano z subwencji badawczej 0211/SBAD/0222 oraz z projektu 0211/PRKE/6428.

***Autorzy**: dr inż. Julian Balcerek, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: julian.balcerek@put.poznan.pl; dr inż. Adam Konieczka, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: adam.konieczka@put.poznan.pl; dr inż. Paweł Pawłowski, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: pawel.pawlowski@put.poznan.pl; inż. Cyprian Dankowski, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: cyprian.dankowski@student.put.poznan.pl; inż. Mateusz Firlej, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: mateusz.firlej@student.put.poznan.pl; inż. Piotr Fulara, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: piotr.fulara@student.put.poznan.pl.*

LITERATURA

[1] Symon E., Komenda Główna Policji Biuro Ruchu Drogowego, Wypadki drogowe w Polsce w 2019 roku, Warszawa 2020, https://statystyka.policja.pl/st/ruch-drogowy/76562,wypadki-drogowe-raporty-roczne.html (dostęp: 30.03.2022).

[2] Kazmi W., Nabney I., Vogiatzis G., Rose P., Codd A., An Efficient Industrial System for Vehicle Tyre (Tire) Detection and Text Recognition Using Deep Learning, https://core.ac.uk/download/pdf/286712293.pdf (dostęp: 30.03.2022).

[3] Sklep Google Play, Tire Check – Analyse your Tyre 2018, https://play.google.com/store/apps/details?id=ie.assettrac.tirecheck.tirecheck&hl=pl (dostęp: 30.03.2022).

[4] Balcerek J., Hinc M., Jalowski Ł., Michalak J., Rabiza M., Konieczka A., Vision-based mobile application for supporting the user in the vehicle operation, Proc. of Signal Processing − Algorithms, Architectures, Arrangements, and Applications SPA'2019, IEEE International Conference, pp. 250−255, Poznań, Poland, 18−20 September 2019.

[5] Python Software Foundation, Python, 2022, https://www.python.org/about/ (dostęp: 30.03.2022).

[6] JetBrains s.r.o., PyCharm The Python IDE for Professional Developers, 2022, https://www.jetbrains.com/pycharm/ (dostęp: 30.03.2022).

[7] OpenCV team, OpenCV, 2022, https://opencv.org (dostęp: 30.03.2022).

[8] TensorFlow, TensorFlow Lite converter, https://www.tensorflow.org/lite/convert (dostęp: 30.03.2022).

[9] Unruh A., What is the TensorFlow machine intelligence platform?, 09.11.2017, https://opensource.com/article/17/11/intro-tensorflow (dostęp: 30.03.2022).

[10] Schildt H., Java. Kompendium programisty, Helion, 2005.

[11] Google Developers, Android Developers, Meet Android Studio, 24.03.2022, https://developer.android.com/studio/intro (dostęp: 30.03.2022).

[12] Redmon J., Farhadi A., YOLOv3: An Incremental Improvement, https://arxiv.org/pdf/1804.02767.pdf (dostęp: 30.03 2022).

[13] Colaboratory, Witamy w Colaboratory, https://colab.research.google.com/notebooks/intro.ipynb (dostęp: 30.03.2022).

[14] Tzutalin, Program LabelImg, https://tzutalin.github.io/labelImg/ (dostęp: 30.03.2022).

[15] Anaconda Inc., Anaconda Distribution, 2022, https://www.anaconda.com/products/individual (dostęp: 30.03.2022).

[16] TensorFlow, Image classification, https://www.tensorflow.org/tutorials/images/classification (dostęp: 30.03.2022).