

System wizyjny do automatycznego rozpoznawania zwierząt na nagraniach z wideorejestratorów samochodowych

Streszczenie. W niniejszym artykule zaproponowano system do automatycznego rozpoznawania zwierząt na obrazach z nagrań wideorejestratorów samochodowych wykorzystujący sztuczne sieci neuronowe. Na przygotowanej bazie obrazów przeprowadzono trening, walidację i testy pod kątem rozpoznawania m.in. krów, dzików, saren oraz innych, także mniejszych, zwierząt spotykanych na polskich drogach. Osiągnięto satysfakcjonujące wyniki rozpoznawania nawet na obrazach o niskiej jakości. Do obsługi systemu przygotowano aplikację użytkownika umożliwiającą przeprowadzenie testów rozpoznawania na pojedynczych zdjęciach i plikach wideo.

Abstract. This article proposes a system for automatic recognition of animals based on images from car video recorders using artificial neural networks. Training, validation and tests were conducted on the prepared image database in terms of recognizing, among others: cows, wild boars, roe deer and other, also smaller, animals encountered on Polish roads. Satisfactory recognition results were achieved even on low-quality image frames. A user application has been prepared to operate the system, enabling recognition tests to be conducted on photos and video files. (Vision system for automatic recognition of animals on images from car video recorders).

Słowa kluczowe: rozpoznawanie zwierząt, wideorejestrator samochodowy, system wizyjny, sztuczne sieci neuronowe.

Keywords: animal recognition, car video recorder, vision system, artificial neural networks.

1. Wprowadzenie

Wideorejestratory samochodowe są coraz popularniejszym wyposażeniem pojazdów [1, 2]. W przypadku zdarzenia drogowego, zarejestrowany przez wideorejestrator materiał umożliwia odtworzenie przebiegu zdarzenia i wskazanie przyczyny, np. do dalszego postępowania sądowego. Współczesne wideorejestratory oferują możliwość nagrywania przebiegu tras nawet w jakości 4K, tj. z rozdzielczością 3840 na 2160 pikseli [3]. Mając na uwadze różnice w umiejscowieniu, budowie i funkcjonalności poszczególnych urządzeń, można wyróżnić wideorejestratory wbudowane oraz akcesoryjne.

Wideorejestratory wbudowane [4] pozwalają na integrację z systemami pojazdu i korzystania z dodatkowych danych, np. pochodzących z innych czujników lub sekwencji obrazów z różnie umiejscowionych kamer, w celu monitorowania pojazdu i środowiska wokół pojazdu. Dostęp do danych może być również uzyskany za pomocą smartfona.

Urządzenia montowane dodatkowo mogą być łatwo wymieniane wraz z zachodzącym postępem technologicznym. Dodatkowy wideorejestrator umieszczony na przedniej szybie może jednak ograniczać obszar widzenia kierowcy. Oprócz urządzeń dedykowanych, można również korzystać z kamer wbudowanych w smartfony.

Wideorejestratory mogą być także wykorzystane w ramach systemów wizyjnych służących do analizy obrazów i asystowania kierowcy. Obiekty z obrazów z kamer samochodowych mogą być rozpoznawane za pomocą tzw. spłotowych sztucznych sieci neuronowych (ang. convolutional neural networks, CNN), w tym architektury YOLO (ang. You Only Look Once) [5]. Przykładowo, pojazdy i osoby wyodrębnione z sekwencji obrazów, w połączeniu z informacjami o prędkości pojazdu, pozwalają na dokładne przewidywanie subiektywnego ryzyka występującego w ruchu drogowym, postrzeganego wśród populacji, nawet lepiej niż poszczególne uczestnicy obserwujący obrazy [6].

Na podstawie sekwencji obrazów z kamer samochodowych można wykrywać kolizje i wypadki drogowe. W pracy [7] zaproponowano identyfikację samochodów za pomocą sieci neuronowej o architekturze YOLO i, na podstawie powierzchni obiektów, wyznaczanie odległość do nich [7]. Inne rozwiązania to np. wykorzystanie spłotowej sieci neuronowej bazującej na regionach (ang. region based convolutional neural networks, RCNN) do wykrywania obiektów,

przewidywania trajektorii i szacowania możliwych lokalizacji uczestników ruchu drogowego [8].

Od 2022 roku, zgodnie z rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej [9], nowe samochody muszą być wyposażone w układ do rejestracji danych na temat zdarzeń, tzw. „czarną skrzynkę”. Jest to wyposażenie pojazdu zbierające dane takie jak m.in. prędkość, informacje o procesie hamowania, czy też pozycję pojazdu przed, w trakcie i po zdarzeniu [9]. W rozporządzeniu nie wyszczególniono jednak rejestracji ani obrazu, ani dźwięku.

Zwiększenie ilości zbieranych informacji, tj. uzupełnienie ich o nagrania wideo z kamer samochodowych, mogłoby pomóc przeanalizować przebieg zdarzeń drogowych w celu właściwej rekonstrukcji faktów i oceny sprawy. Znany jest przypadek śmierci osoby, początkowo uznanej za nieśczęśliwy wypadek, a ostatecznie, na podstawie nagrania wideo, na którym widoczny był celowy i nagły bieg ofiary w kierunku środka jezdni, uznanej za samobójstwo [10].

Wśród wypadków szczególne znaczenie mają zdarzenia drogowe z udziałem zwierząt, zarówno dzikich jak i domowych. W ostatnich latach zanotowano wzrost liczby tego typu zdarzeń w Polsce [11]. Często skutkują one obrażeniami osób, zwierząt oraz uszkodzeniami pojazdów. Z perspektywy kierowcy nie jest łatwe dostrzeżenie zwierzęcia na tyle szybko, żeby wykonać manewr umożliwiający uniknięcie wypadku [12]. Jednakże systemy bezpieczeństwa we współczesnych samochodach, np. służące do awaryjnego hamowania, mogą być aktywowane na podstawie odpowiednich sygnałów dochodzących z czujników wizyjnych.

Rozpoznawanie zwierząt może służyć także do innych celów, np. do monitorowania przyrody. W tego rodzaju zastosowaniach wykorzystuje się zdjęcia ze stacjonarnych kamer fotopułapek i spłotowe sieci neuronowe o różnych architekturach takich jak: YOLO, RCNN, ResNet (ang. residual neural network, resztkowa sieć neuronowa), AlexNet, VGG (ang. Visual Geometry Group), NiN (ang. Network in Network), czy też GoogleLeNet [13–15].

Obecnie można spotkać różne systemy do automatycznego wykrywania zwierząt przeznaczone dla pojazdów. Mogą one wykorzystywać czujniki światła widzialnego [16–18] albo pasywne lub aktywne czujniki podczerwieni [12, 19–21]. Sporą część systemów zaprojektowano z myślą o uniknięciu zderzeń z dużymi zwierzętami lub gdy rozmiar zwierzęcia przekracza pewną minimalną wielkość.

W sytuacji powszechnej dostępności wideorejestratorów samochodowych, celowa wydaje się być implementacja systemu do rozpoznawania zwierząt, uwzględniającego detekcję oraz ich klasyfikację pod względem rozmiaru i gatunków występujących typowo w Polsce. Zastosowanie podziału na klasy pozwoli na precyzyjne i adekwatne do sytuacji ostrzeżenie kierowcy i, potencjalnie, po odpowiedniej adaptacji aplikacji, na stosowne działanie poszczególnych systemów bezpieczeństwa czynnego i biernego pojazdu. Tym samym poprawi się ochrona osób znajdujących się w pojeździe i innych uczestników ruchu.

W niniejszej pracy zaproponowano system, wraz z dedykowaną aplikacją, do automatycznego rozpoznawania wybranych zwierząt na podstawie analizy obrazów z wideorejestratorów samochodowych. Rozpoznawanie wizerunków zwierząt wykorzystuje sztuczne sieci neuronowe. Badania eksperymentalne wykonano na specjalnie przygotowanej do tego bazy obrazów.

2. Baza obrazów zwierząt

W celu wytrenowania i przetestowania sztucznych sieci neuronowych do zakładanego zadania przygotowano odpowiednią bazę obrazów. Przyjęto, że, nie będą wykorzystywane ani artystyczne zdjęcia zwierząt, ani obrazy z fotopułapek, ani obrazy wygenerowane sztucznie za pomocą metod sztucznej inteligencji, tylko rzeczywiste obrazy pochodzące z nagrań wideorejestratorów samochodowych.

Zrezygnowano z samodzielnego wykonania ujęć przedstawiających zwierzęta w inscenizowanych sytuacjach na drodze przed pojazdem. Napotkanie odpowiedniej liczby zwierząt mogłoby być niezwykle czasochłonne, a niewskazane i niebezpieczne byłoby sztucznie wywoływanie tego typu sytuacji, np. przez poganianie, zwłaszcza dzikich, zwierząt. Skorzystano więc z serwisu internetowego z filmami zawierającymi zbiory scen ze zwierzętami i innymi obiektami. Do przeprowadzenia badań wybrano 16 zbiorów (kompilacji) scen. Łączny czas trwania nagrań wyniósł ponad 112 minut, a liczba scen (zdarzeń) znacznie przekroczyła sto.

Z wybranych materiałów wyodrębniono klatki z wizerunkami zwierząt. Wybrano klatki, które przedstawiały zwierzęta w okolicy dróg, ulic lub krajobrazu około miejskiego. Kryterium wyboru miało odwzorowywać warunki panujące podczas jazdy samochodem (w widoku z perspektywy poruszającego się pojazdu). Jeżeli z pojedynczej sceny pochodziła więcej niż jedna klatka, to z tej sceny były wybierane maksymalnie trzy klatki, gdzie zwierzęta różniły się wyglądem, np. zwierzę było w różnych miejscach i pozycjach (przykładowo nogi u szybko poruszającej sarny były rozprostowane i ugięte), czy też było różne oświetlenie.

W skład bazy danych, wchodziły obrazy zawierające wizerunki dzików, krów, saren oraz innych, mniejszych, zwierząt takich jak psy, czy ptaki. Oprócz zwierząt na obrazach widoczne były inne obiekty. Obrazy mogły zawierać więcej niż jeden obiekt. Liczbę obiektów w przygotowanej bazie przedstawiono w Tabeli 1. Zbiór obrazów został podzielony na rozłączne zbiory: treningowy, walidacyjny oraz testowy. Oznacza to, że te same klatki nie powtarzały się w zbiorach.

Aby uzyskać możliwie rzetelne wyniki testów skuteczności, do zbioru testowego dodano obrazy nie zawierające reprezentantów wymienionych wyżej klas, w tym przedstawiające ludzi. Założono, że system ma nie wykrywać ludzi, więc do treningu i walidacji nie użyto zdjęć z ich wizerunkami.

Podczas tworzeniu zbioru obrazów zastosowano proces augmentacji, który, w przypadku niewielkich zbiorów, poprawia skuteczność działania algorytmów uczenia maszynowego. Do zbioru treningowego i walidacyjnego dodano

przekształcone (obrócenie obrazu, odbicie lustrzane oraz rozmycie gaussowskie) kopie obrazów.

Ostatecznie, po augmentacji, w bazie znalazły się zdjęcia przedstawiające 1095 obiektów. W przypadku eksperymentów bez augmentacji zbiór testowy obiektów stanowił 50% wszystkich obiektów (ok. 31% bez odniesienia w postaci ludzi), a w przypadku z augmentacją 20% wszystkich obiektów (ok. 10% bez ludzi).

Ostatnim krokiem niezbędnym do poprawnego przygotowania bazy uczącej dla przygotowanych modeli było opisanie, tzw. „otagowanie” obrazów (ang. data labeling). Wynikowe pliki tekstowe zawierały informację o rodzajach obiektów wraz położeniem i rozmiarem prostokątnego obrysu tych obiektów.

Tabela 1. Liczba obrazów i obiektów w przygotowanej bazie

cecha	klasy						suma
	dzik	krowa	sarna	inne (małe) zwierzę	brak /inne	czło wiek	
trening bez augmentacji	28/48	16/27	43/43	24/29	-/-	-/-	111 /147
trening z augmentacją	114 /203	67 /122	173 /182	98 /112	-/-	-/-	452 /619
walidacja bez augmentacji	13/20	8/19	19/22	11/11	-/-	-/-	51 /72
walidacja z augmentacją	50/69	29/62	75/78	42/48	-/-	-/-	196 /257
test (bez augmentacji)	15/24	8/28	26/30	15/15	25/-	25 /122	114 /219
suma bez augmentacji	56/92	32/74	88/95	50/55	25/-	25 /122	276 /438
suma z augmentacją	179 /296	104 /212	274 /290	155 /175	25/-	25 /122	762 /1095

3. Architektura sztucznych sieci neuronowych

Do implementacji sztucznej sieci neuronowej i przeprowadzenia eksperymentów wybrano model YOLO w wersji 4 (YOLOv4), wykorzystujący architekturę CNN [5, 22]. Proces treningu sieci neuronowej został przeprowadzony na platformie Google Colaboratory [23], która pozwala na użycie jednostek obliczeniowych w tzw. chmurze obliczeniowej.

Przed rozpoczęciem treningu, ustawiono liczbę iteracji procesu treningu oraz wymiary obrazów. W przypadku wszystkich zdjęć z przygotowanej bazy danych zastosowano stałą rozdzielczość 416 na 416 pikseli. Pozostałe parametry pozostawiono w ustawieniach domyślnych.

W każdej iteracji przeprowadzanego procesu treningu przedstawiana była m.in. liczba wykonanych iteracji, czas pozostały do ukończenia nauki (ang. hours left) oraz średnia strata dla błędu treningu (ang. average loss). Efektem treningu był plik z wagami wyuczonej sieci oraz wykres przedstawiający wartość funkcji strat w funkcji czasu.

Aby wykorzystać otrzymane rezultaty treningu modeli sieci neuronowej, zostało zaimplementowane oprogramowanie w języku Python [24] z wykorzystaniem biblioteki OpenCV [25]. Język programowania Python, ze względu na skryptowy charakter, pozwalał na szybkie i sprawne zarządzanie zdjęciami z przygotowanej bazy obrazów. Biblioteka OpenCV oferuje duży zasób funkcji do przetwarzaniu obrazów. Ponadto, zawiera ona moduł „dnn” (ang. deep neural networks) pozwalający na obsługę wytrenowanych modeli sieci neuronowych.

Podczas wczytywania wytrenowanych modeli sieci neuronowych, podawanymi zmiennymi były uzyskane wagi modelu sieci, plik konfiguracyjny oraz plik tekstowy, zawierający nazwy klas. Wytrenowany model załadowano korzystając z wcześniej wspomnianego modułu „dnn” biblioteki OpenCV.

Wynikiem działania przygotowanego oprogramowania było przedstawienie wybranego obrazu ze zbioru testowego

z wprowadzonymi oznaczeniami dotyczącymi rozpoznawania poszczególnych zwierząt. Każdy wykryty obiekt posiadał przypisaną informację na temat klasy oraz wartość procentową wskazującą pewność prezentowanego wyniku.

4. Eksperyment

W ramach eksperymentu przeprowadzono testy skuteczności przygotowanego oprogramowania na utworzonej bazie danych. Opracowano 3 modele sieci neuronowych YOLOv4, wytrenowane na zbiorze niepoddanym augmentacji i poddanym augmentacji. Augmentację przeprowadzono w dwóch wariantach: pełnym i w zredukowanym (tzw. „Tiny”). Wariant zredukowany przygotowany był celem wykorzystania w aplikacji testowej dla użytkownika. W tym wariancie przypadku proces detekcji był przyspieszony, a liczba wag wytrenowanej sieci istotnie zredukowana.

Testy wykonane zostały na zbiorze testowym, w skład którego wchodziły zdjęcia dzików, krów, saren, innych małych zwierząt, ludzi oraz zdjęcia nie zawierające żadnych z wcześniej wymienionych obiektów. Miało to na celu sprawdzenie, jak dobrze poszczególne modele sieci radzą sobie z rozpoznawaniem zwierząt w sytuacjach, gdy na obrazach widnieją nie tylko zwierzęta, ale również inne obiekty.

Trzy wymienione modele wytrenowano i przetestowano w zakresie od 1000 do 8000 iteracji. Testy były wykonywane dla treningów z krokiem 1000 iteracji. Uzyskane wyniki dokładności rozpoznawania przedstawiono w tabeli 2.

Tabela 2. Średnia strata dla błędu treningu i dokładność testu dla sieci wytrenowanych na przestrzeni 1000-8000 iteracji treningu

liczba iteracji treningu	bez augmentacji		z augmentacją		z augmentacją „Tiny”	
	średnia strata	dokładność [%]	średnia strata	dokładność [%]	średnia strata	dokładność [%]
1000	0,80	80,33	1,35	76,23	0,60	36,89
2000	0,32	82,79	0,70	82,79	0,23	71,31
3000	0,25	82,79	0,50	81,97	0,20	71,31
4000	0,20	82,79	0,40	81,97	0,18	72,95
5000	0,18	74,59	0,35	80,65	0,16	70,49
6000	0,17	80,33	0,30	80,33	0,13	71,31
7000	0,10	81,15	0,29	81,15	0,10	72,13
8000	0,10	81,97	0,28	83,61	0,08	72,13

Najwyższa dokładność, wynosząca ok. 83%, wystąpiła dla modelu z obrazami nie poddanymi augmentacji, dla 4000 iteracji, gdy wartość średniej straty podczas uczenia wyniosła 0,2. Dla modelu z obrazami poddanymi augmentacji, osiągnięto nieznacznie większą maksymalną dokładność, wynoszącą ok. 84%. Wynik ten uzyskano przy liczbie iteracji treningu równej 8000 i średniej stracie wynoszącej 0,28. Wyniki dla modelu „Tiny”, okazały gorsze niż w przypadku poprzednich modeli i wynosiły ok. 73% dokładności. Wartość ta pozwala na uznanie ich za akceptowalne.

Dla zbioru testowego zawierającego zwierzęta nie zanotowano fałszywego wykrycia zwierzęcia, rozumianego za uznanie za zwierzę obiektu, który nie był zwierzęciem. Na 25 zdjęciach nie zawierających zwierząt (klasa "brak/inne"), na których zdarzały się występować np. pojazdy na drodze, w zależności od rodzaju treningu i liczby iteracji, uzyskano od 0 do 2 fałszywych wykryć zwierząt. Na zdjęciach testowych ze 122 osobami (klasa "człowiek"), uzyskano od 3 do 17 przypadków fałszywego wykrycia zwierząt.

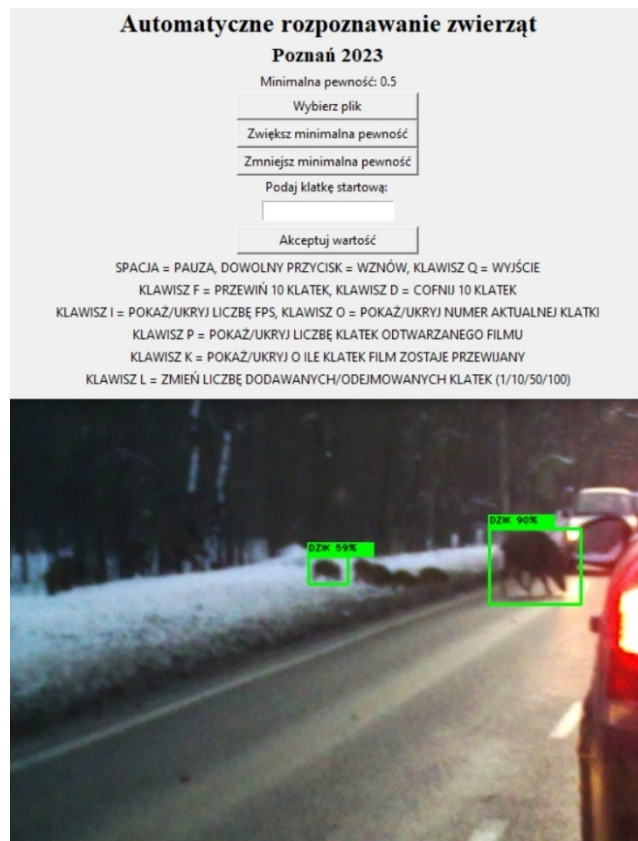
Błędy podczas rozpoznawania wynikały głównie z dwóch przyczyn: niskiej jakości nagrań z wideorejestratorów (niewielka rozdzielczość, duża utrata szczegółów wynikająca z wysokich stopni kompresji użytych kodeków, pojawiające się artefakty, np. efekty blokowe) oraz z podobieństwa reprezentantów poszczególnych klas i obiektów.

5. Aplikacja użytkownika

Przygotowano testową aplikację na komputer klasy PC pracujący w systemie operacyjnym Windows. Pozwala ona na przeprowadzenie procesu rozpoznawania zwierząt na wybranym przez użytkownika pliku graficznym lub pliku wideo pochodzącym z wideorejestratora.

Program został zaimplementowany w języku Python z wykorzystaniem bibliotek OpenCV oraz Tkinter [24]. Druga wymieniona biblioteka umożliwia szybkie tworzenie interfejsu graficznego.

W aplikacji użytkownik może ustawić minimalną wartość pewności wymaganą do detekcji. Przyjmuje ona wartości mieszczące się w przedziale od 0 do 1, gdzie 0 jest zdarzeniem niemożliwym, a 1 jest zdarzeniem pewnym. Program pozwala również, za pomocą skrótów klawiaturowych, na przewijanie wyświetlanego pliku wideo, zatrzymywanie go oraz pokazywanie lub ukrywanie bieżących parametrów wyświetlanych na obrazie podczas odtwarzania. W przypadku pliku wideo, w lewym górnym rogu, przez cały czas projekcji może być pokazywana liczba klatek na sekundę (ang. frames per second, FPS), która jest odzwierciedleniem szybkości przetwarzania, oraz numer aktualnej klatki wideo, całkowita liczba klatek nagrania i liczba przewijanych klatek. Na rysunku 1 przedstawiono zrzut ekranu menu interfejsu użytkownika oraz okno działania aplikacji testowej po wczytaniu przykładowego obrazu.



Rys.1. Przykładowy zrzut ekranu menu interfejsu użytkownika oraz okna działania aplikacji testowej

W przypadku wczytania filmu o rozdzielczości wejściowej 1280 na 720 pikseli zakodowanego przy użyciu kodeka H264 – MPEG-4 AVC, szybkość przetwarzania obrazów na komputerze przenośnym, przy włączonych wszystkich opcjach wyświetlania informacji bezpośrednio na obrazie, wyniosła ok. 1 klatki na sekundę.

6. Podsumowanie i wnioski

Przeprowadzone testy skuteczności przygotowanego oprogramowania, pozwoliły stwierdzić, że dodanie obrazów po augmentacji wpływa na wydłużenie czasu treningu, jak również na poprawę wyników działania otrzymanych modeli. Najlepsza uzyskana dokładność rozpoznawania czterech klas zwierząt wyniosła ok. 84%.

Należy podkreślić, że wyniki te uzyskano na relatywnie niskiej jakości nagraniach zwierząt pochodzących z wideorejestраторów samochodowych.

Przygotowano również aplikację testową z graficznym interfejsem użytkownika, umożliwiającą przetestowanie działania systemu na obrazach i plikach wideo wybranych przez użytkownika, również spoza przygotowanej bazy danych.

W przyszłości, w celu zwiększeniu skuteczności rozpoznawania zwierząt planowane jest powiększenie bazy obrazów treningowych, np. wraz z pojawiającymi się w serwisach udostępniających filmy, nowymi nagraniami pochodzącymi z wideorejestраторów.

W przypadku większej liczby obrazów, można zwiększyć liczbę klas zwierząt, wyszczególniając kolejne gatunki. Aplikacja testowa może być przygotowana również w wersji przeznaczonej bezpośrednio na wideorejestраторy akcesoryjne lub wbudowane. Tym samym mogłaby służyć jako narzędzie ostrzegające kierowcę i pomagające dostosować działanie odpowiednich systemów poprawiających bezpieczeństwo adekwatnie do zagrożenia.

Badania sfinansowano z subwencji badawczej 0211/SBAD/0223.

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. Paweł Pawłowski, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: pawel.pawlowski@put.poznan.pl; inż. Błażej Trzcziński, Politechnika Poznańska, Instytut Automatyki i Robotyki, ul. Piotrowo 3a, 60-965 Poznań, E-mail: blazej.trzcinski@student.put.poznan.pl.

LITERATURA

- [1] Grand View Research, Dashboard Camera Market Size, Share&Trends Analysis Report by Technology (Basic, Smart), By Product (1-, 2-Channel), By Application, By Video Quality, By Distribution Channel, And Segment Forecasts, 2022 - 2030, <https://www.grandviewresearch.com/industry-analysis/dashboard-camera-market> (dostęp: 17.03.2023).
- [2] Piątek P., Samochodowe wideorejestраторy. Krajowy rynek okiem ekspertów, serwis Motofakty, Polska Press, <https://motofakty.pl/samochodowe-wideorejestраторy-krajowy-rynek-okiem-ekspertow/ar/c4-16117439>, 13.01.2022 (dostęp: 17.03.2023).
- [3] Cobra, Dash Cam Footage Video Quality, <https://www.cobra.com/blogs/news/dash-cam-footage> (dostęp: 17.03.2023).
- [4] Naylor S., Best cars with an in-built dash cam, BuyCar, <https://www.buyacar.co.uk/cars/used-cars/2291/best-cars-with-an-in-built-dash-cam>, 29.10.2021 (dostęp: 19.03.2023).
- [5] Redmon J., Divvala S., Girshick R., Farhadi A., You Only Look Once: Unified, Real-Time Object Detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, Las Vegas, NV, USA, 27-30 June 2016, <https://doi.org/10.1109/CVPR.2016.91>.
- [6] de Winter J., Hoogmoed J., Stapel J., Dodou D., Bazilinskyy P., Predicting perceived risk of traffic scenes using computer vision, Transportation Research Part F: Traffic Psychology and Behaviour, Volume 93, Pages 235-247, February 2023.
- [7] Goenka U., Jagetia A., Patil P., Singh A., Sharma T., Saini P., Threat Detection in Self-Driving Vehicles Using Computer Vision, In: Doriya, R., Soni, B., Shukla, A., Gao, XZ. (eds) Machine Learning, Image Processing, Network Security and Data Sciences. Lecture Notes in Electrical Engineering, vol. 946, Springer, Singapore, 2023, https://doi.org/10.1007/978-981-19-5868-7_45.
- [8] Yao Y., Xu M., Wang Y., Crandall D. J., Atkins E. M., Unsupervised Traffic Accident Detection in First-Person Videos, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 273-280, Macau, China, 2019, <https://doi.org/10.1109/IROS40897.2019.8967556>.
- [9] Dziennik Urzędowy Unii Europejskiej, Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2019/2144 z dnia 27 listopada 2019 r. w sprawie wymogów dotyczących homologacji typu pojazdów silnikowych (...), 16.12.2019.
- [10] Giovannini E., Giorgetti A., Pelletti G., Giusti A., Garagnani M., Pascali J. P., Pelotti S., Fais P., Importance of dashboard camera (Dash Cam) analysis in fatal vehicle-pedestrian crash reconstruction, Forensic Science, Medicine and Pathology, 17, pp. 379–387, 2021, <https://doi.org/10.1007/s12024-021-00382-0>.
- [11] Tereszkievicz K., Choroszy K., Analiza zdarzeń drogowych z udziałem zwierząt w Polsce w latach 2006-2015, Autobusy : technika, eksploatacja, systemy transportowe, Instytut Naukowo-Wydawniczy "SPATIUM", R. 17, nr 12, 467–473, 2016.
- [12] Sibanda V., Mpofo K., Trimble J., Zengeni N., Design of an Animal Detection System for Motor Vehicle Drivers, Procedia CIRP, vol. 84, pp. 755-760, 2019, ISSN 2212-8271, <https://doi.org/10.1016/j.procir.2019.04.175>.
- [13] Nguyen H., Maclagan S. J., Nguyen T. D., Nguyen T., Flemons P., Andrews K., Ritchie E. G., Phung D., Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring, 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, pp. 40-49, 2017, <https://doi.org/10.1109/DSAA.2017.31>.
- [14] Norouzzadeh M. S., Nguyen A., Kosmala M., Swanson A., Packer C., Clune J., Automatically identifying wild animals in camera trap images with deep learning, Proceedings of the National Academy of Sciences, vol. 115, no. 25, 2018, <https://doi.org/10.1073/pnas.1719367115>.
- [15] Tan M., Chao W., Cheng, J.-K., Zhou M., Ma Y., Jiang X., Ge J., Yu L., Feng, L., Animal Detection and Classification from Camera Trap Images Using Different Mainstream Object Detection Architectures, Animals, 12, no. 15: 1976, 2022, <https://doi.org/10.3390/ani12151976>.
- [16] Prabhu R., Viswanathan N., Automated Roadside COW Animal Detection and Collision Prevention System, International Journal of Advanced Research in Science, Communication and Technology, pp. 145-149, 2021, <https://doi.org/10.48175/IJARSC-2321>.
- [17] Adams E., Volvo's Cars Now Spot Moose and Hit the Brakes for You, Wired, Condé Nast, <https://www.wired.com/2017/01/volvos-cars-now-spot-moose-hit-brakes/>, 27.01.2017 (dostęp: 19.01.2023).
- [18] Volvo, Pedestrian, Cyclist & Large Animal Detection, https://volvo.custhelp.com/app/answers/detail/a_id/9778/~pedestrian%2C-cyclist-%26-large-animal-detection, 11.04.2019 (dostęp: 19.01.2023).
- [19] BMW X5 Owners & Service Manuals, BMW X5: Night Vision with pedestrian and animal detection, https://www.bmw5info.com/night_vision_with_pedestrian_and_animal_detection-1786.html (dostęp: 21.04.2023).
- [20] Dr. Ing. h.c. F. Porsche AG, Assistance systems, Comprehensive systems increase comfort and safety, Night Vision Assist with thermal imaging camera, <https://presskit.porsche.de/models/en/porsche-cayenne/topic/assistance-systems.html> (dostęp: 21.03.2023).
- [21] Fuths T., The End of Night, Porsche Cars North America, <https://www.porsche.com/usa/aboutporsche/christophorusmagazine/archive/382/articleoverview/article05/> (dostęp: 21.03.2023).
- [22] Peiyuan Jiang, DajiErgu, Fangyao Liu, Ying Cai, Bo Ma, A Review of Yolo Algorithm Developments, Procedia Computer Science, vol. 199, pp. 1066-1073, 2022, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2022.01.135>.
- [23] Google Colaboratory, Google Research, <https://colab.research.google.com> (dostęp: 26.03.2023).
- [24] Python Software Foundation, Python, <https://www.python.org> (dostęp: 26.03.2023).
- [25] OpenCV team, OpenCV, <https://opencv.org> (dostęp: 26.03.2023).