

doi:10.15199/48.2022.01.41

## Możliwości przetwarzania sekwencji wizyjnych w systemach wbudowanych

**Streszczenie.** W artykule przedstawiono wyniki badań eksperymentalnych procesu segmentacji sekwencji wizyjnych z wykorzystaniem systemów wbudowanych. Przetestowano wydajność rozwiązań opartych o mikrokomputer Raspberry Pi 4B oraz platformę Nvidia Jetson Nano pod kątem możliwości ich implementacji w platformie pomiarowej do automatycznego badania jakości działania lamp lotniskowych. Porównano szybkość przetwarzania dla różnych rozdzielczości obrazu oraz wymagania związane z zasilaniem modułów.

**Abstract.** The article presents the results of experimental research on the video segmentation process using two different embedded systems. The performance of solutions based on the Raspberry Pi 4B microcomputer and the Nvidia Jetson Nano platform was tested for the possibility of their implementation in a measurement platform for automatic testing of the quality of airport lamps. The processing speed for different image resolutions and the module power requirements were compared. (**Possibilities of processing video sequences in embedded systems**).

**Słowa kluczowe:** system wizyjny, system wbudowany, przetwarzanie obrazu, analiza wydajności

**Keywords:** vision system, embedded system, image processing, performance analysis

### Wstęp

Systemy identyfikacyjne oparte o rozpoznawanie poszczególnych znaków i zdarzeń w sekwencjach wideo stają się coraz bardziej powszechne i stosowane w wielu zagadnieniach. Są one oparte zarówno na klasycznych metodach przetwarzania obrazu, jak i na sieciach neuronowych, których zastosowanie staje się coraz popularniejsze. Kamery montowane są w samochodach i pojazdach autonomicznych [1, 2], w celu monitoringu miejskiego [3], ale również w specjalistycznych zastosowaniach, takich jak inspekcja lamp lotniskowych [4, 5]. Systemy takie pomagają operatorom w obserwacji przestrzeni miejskich lub otoczenia maszyn oraz w nadzorowaniu bieżących zdarzeń i reagowaniu na nie. W pewnych przypadkach są w stanie całkowicie wyeliminować czynnik ludzki, automatycznie wydając konkretne polecenia pozostałym systemom.

Przetwarzanie sekwencji wizyjnych wymaga użycia jednostek obliczeniowych o różnej mocy, w zależności od wybranego algorytmu i dokonywanych operacji na obrazie [6]. W zastosowaniach mobilnych istotne jest znalezienie kompromisu pomiędzy kompaktowym rozmiarem systemu, poborem mocy oraz wydajnością komputera. W celu doboru najlepszego sprzętu możliwego do zastosowania w poszczególnych przypadkach, niezbędne jest dokonanie analizy porównawczej dostępnych systemów wbudowanych, które, dzięki swoim niewielkim rozmiarom i stosunkowo zmniejszonemu zapotrzebowaniu energetycznemu, mogą być zamontowane w mobilnych urządzeniach pomiarowych, w tym przypadku w specjalistycznej konstrukcji platformy do badania jakości działania lamp lotniskowych (rys. 1).

Konieczność budowy platformy oraz kontroli lamp lotniskowych wynika z rosnących wymagań agencji bezpieczeństwa lotniczego, które wprowadzają coraz bardziej rygorystyczne przepisy dotyczące konieczności kontroli oświetlenia lotniskowego [7]. Lampy znajdujące się na płaszczyznach lotniskowych odgrywają bardzo istotną rolę w nawigacji lotniczej, ponieważ ich widzialność determinuje możliwość wykonania poszczególnych operacji lotniczych.

Wraz ze wzrostem klasy zastosowanych systemów nawigacyjnych na lotnisku, konieczne jest przeprowadzanie bardziej szczegółowych kontroli oświetlenia, do czego niezbędna jest budowa niezawodnych systemów,

mogących dokonać wiarygodnych pomiarów w krótkim czasie pomiędzy wykonywanymi operacjami lotniczymi.

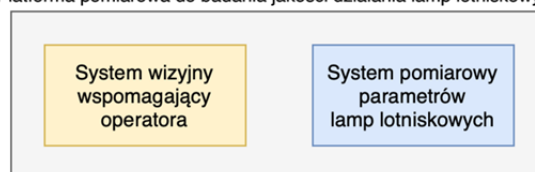


Rys. 1 Platforma pomiarowa do badania jakości działania lamp lotniskowych

Aby umożliwić poprawny najazd platformy nad lampę wbudowaną w drogę startową, konieczne jest zastosowanie wsparcia operatora w postaci systemu wizyjnego, który poprawnie zidentyfikuje oznaczenia zastosowane na drodze startowej i innych płaszczyznach lotniskowych, a następnie wyda komunikat kierowcy o dokonaniu niezbędnej korekty toru jazdy w celu poprawy dokonywanego pomiaru. Ze względu na budowę platformy pomiarowej do badania jakości działania lamp lotniskowych w postaci przyczepy samochodowej, jej widoczność jest ograniczona i bez wspomaganie zewnętrznego nie ma możliwości dokonania prawidłowego pomiaru. Specyfika drogi startowej, która jest znacznie szersza od dróg publicznych, dodatkowo utrudnia znalezienie właściwego punktu odniesienia.

Na rys. 2 przedstawiono schemat zasady współdziałania zastosowanych w platformie pomiarowej systemów. Sekcje te działają niezależnie od siebie, jednak wyniki pracy obydwu z nich stanowią punkt odniesienia dla operatora w celu oceny poprawności dokonanego pomiaru.

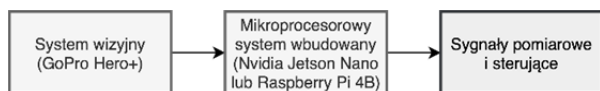
Platforma pomiarowa do badania jakości działania lamp lotniskowych



Rys. 2 Schemat ideowy obrazujący zastosowane systemy w platformie pomiarowej do badania jakości działania lamp lotniskowych

## System wizyjny

Zasada działania systemu wizyjnego wspomagającego operatora w platformie pomiarowej do badania jakości działania lamp lotniskowych przedstawiona została na rys. 3. Danymi wejściowymi do systemu są sekwencje wizyjne pozyskane z kamery GoPro Hero+. Tak pobrane dane są przetwarzane przez system wbudowany poprzez wytrenowane sieci neuronowe bądź klasyczne metody przetwarzania obrazu, w celu uzyskania wyników, czyli np. informacji o optymalnym torze jazdy poprzez wykrywanie linii oraz oznakowania poziomego znajdującego się na płaszczyznach lotniskowych.



Rys. 3 Schemat ideowy przetwarzania sekwencji wizyjnych

Przetwarzanie sygnałów wizyjnych często wykorzystuje techniki Tiny Machine Learning (TinyML) [8]. Popularnymi urządzeniami w takich rozwiązaniach są Arduino Nano 33 BLE, a także SparkFun Edge. Pierwszy z nich, dzięki zastosowaniu mikrokontrolera nRF52480 charakteryzuje się wysoką energooszczędnością i jest w stanie pracować z maksymalną częstotliwością 64 MHz. Mikrokontroler został wyposażony w 1 MB pamięci flash oraz 256 kB pamięci RAM. SparkFun Edge posiada procesor 32-bitowy ARM Cortex-M4F o taktowaniu 48 MHz, 1 MB pamięci Flash oraz 384 KB SRAM. Urządzenia te nastawione są jednak głównie do przetwarzania danych pobieranych z akcelerometrów. W przypadku sygnałów wizyjnych adekwatnym jest wykorzystanie modułów o większej wydajności, takich jak np. Raspberry Pi 4B lub Nvidia Jetson Nano, których charakterystyki zostały przedstawione w kolejnych rozdziałach.

## Wykorzystane sensory wizyjne

Do testów i nagrań w Porcie Lotniczym Poznań – Ławica użyto kamery GoPro Hero+. Ze względu na swoją budowę, jest ona odporna na zmienne zewnętrzne warunki atmosferyczne oraz umożliwia odpowiednie i bezpieczne jej zamontowanie do konstrukcji platformy pomiarowej.

Kamera GoPro Hero+ pozwala na rejestrowanie obrazu w maksymalnej rozdzielczości 1920×1080 pikseli przy 60 klatkach na sekundę. Posiada sensor CMOS o rozdzielczości 8 megapikseli, a wartość przysłony wynosi f/2.8. Kamera ta stosuje kodowanie na poziomie L4.1 (H264/MPEG-4 AVC). Zastosowanie systemu wykrywającego słabe oświetlenie pozwala automatycznie dostosować parametry akwizycji, optymalizując tym samym wydajność kamery. Rejestracja obrazu dokonywana jest na wymiennych nośnikach danych w postaci kart pamięci microSD, a dzięki wbudowanemu modułowi WiFi możliwe jest także przekazywanie obrazu w czasie rzeczywistym [9].



Rys. 4 Przykładowa stopklatka z nagrań wykonanych w Porcie Lotniczym Poznań–Ławica przy pomocy kamery GoPro Hero+

Na rys. 4 przedstawiono stopklatkę z filmu zarejestrowanego przez kamerę GoPro Hero+. Widoczne

jest na niej oznakowanie poziome stosowane na płaszczyznach lotniskowych w postaci: osi centralnej drogi startowej (białe linie przerywane) oraz tory zjazdowe i osie centralne dróg kołowania (linie żółte ciągłe).

## Charakterystyka wykorzystanych systemów wbudowanych

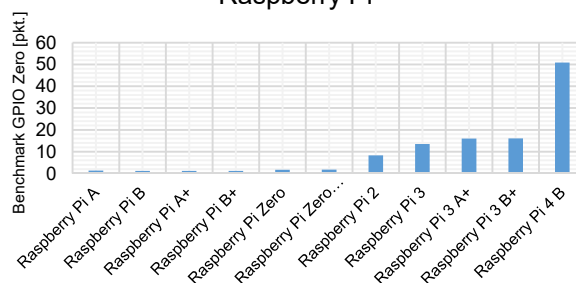
W ramach prac i testów zdecydowano się na wykorzystanie oraz przetestowanie dwóch mikrokomputerów w postaci Raspberry Pi 4B (rys. 5) oraz Nvidia Jetson Nano. Systemy te są często wykorzystywane w uczeniu maszynowym jako jednostki centralne samochodów autonomicznych [10, 11] oraz do wykrywania zdarzeń na obrazie [12].



Rys. 5 Mikrokomputer Raspberry Pi 4B

Pierwszy z nich charakteryzuje się procesorem Broadcom BCM2711 64-bit Quad-Core ARM Cortex-A72 o taktowaniu 1,5 GHz i architekturze ARMv8-A. Testowany model wyposażony został w 8 GB pamięci RAM LPDDR4. Rysunek 6 przedstawia porównanie wydajności reprezentowanej przez poszczególne modele z serii Raspberry Pi [9].

## Wydajność poszczególnych modeli Raspberry Pi



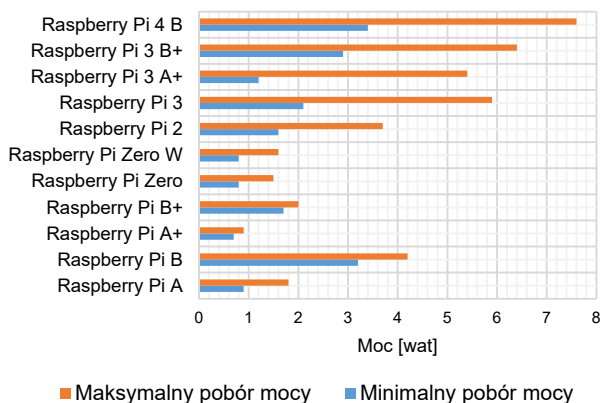
Rys. 6 Porównanie wydajności poszczególnych modeli Raspberry Pi

Analiza wykresu pokazuje, że moduły od typu A do Zero W dysponują podobną mocą obliczeniową, która znacząco odbiega od pozostałych modeli wyposażonych w wydajniejsze układy w architekturze ARM. Modele od 2 do 3 B+ prezentują tendencję wzrostową wydajności, jednak model 4B deklasuje swoich poprzedników. Ze względu na konieczność przetwarzania sekwencji wizyjnych, niezbędny był wybór właśnie tej jednostki, charakteryzującej się największą mocą obliczeniową z całej rodziny.

Na rysunku 7 przedstawiono natomiast porównanie poborów mocy omówionej rodziny Raspberry Pi [9]. Wraz ze wzrostem wydajności urządzenia, rośnie także zapotrzebowanie energetyczne. Raspberry Pi 4 B charakteryzuje się największą wydajnością, jednak przy tym potrzebuje największej mocy do pracy i wykonywania niezbędnych obliczeń. Najnowsze modele charakteryzują się znacznym wzrostem wydajności, przy jednoczesnym ograniczonym zapotrzebowaniu energetycznym przy niskich obciążeniach w porównaniu do swoich

poprzedników. Ponad trzykrotna poprawa mocy obliczeniowej nie niesie za sobą znaczącego wzrostu zapotrzebowania na energię elektryczną.

### Porównanie poboru mocy przez modele Raspberry Pi



Rys. 7 Porównanie poboru mocy przez modele Raspberry Pi

Drugim testowanym urządzeniem był mikrokomputer Nvidia Jetson Nano (rys. 8). Jednostka ta posiada czterordzeniowy procesor ARM A57 o taktowaniu 1,43 GHz oraz 2 GB 64-bitowej pamięci RAM LPDDR4. NVIDIA Jetson Nano została wyposażona również w 128-rdzeniowy procesor graficzny Maxwell o taktowaniu 921 MHz. Dzięki temu jednostka ta, w zestawieniu z Raspberry Pi 4B, posiada o wiele większe możliwości wykonywania operacji opartych o uczenie maszynowe. W zależności od używanego modelu sieci neuronowych, wydajność Nvidia Jetson Nano może być w tym porównaniu większa nawet trzykrotnie [9, 13]. Mikrokomputer ten umożliwia także pracę w dwóch trybach: energooszczędnym (5W) oraz MAXN (10W).



Rys. 8 Mikrokomputer Nvidia Jetson Nano

Do testów laboratoryjnych wykorzystano także o wiele wydajniejszą maszynę wirtualną posiadającą procesor Intel Core i7-7700HQ CPU 2.80GHz – 3.8 GHz × 4, 15.6 GB pamięci RAM oraz kartę graficzną SVGA3D (host: NVIDIA GeForce GTX 1050Ti (4096 MB memory)). Systemem był Ubuntu 18.04.5 LTS 64-bit. Jednostka ta została zastosowana w celu zobrazowania uzyskanych wyników eksperymentalnych wybranych systemów wbudowanych.

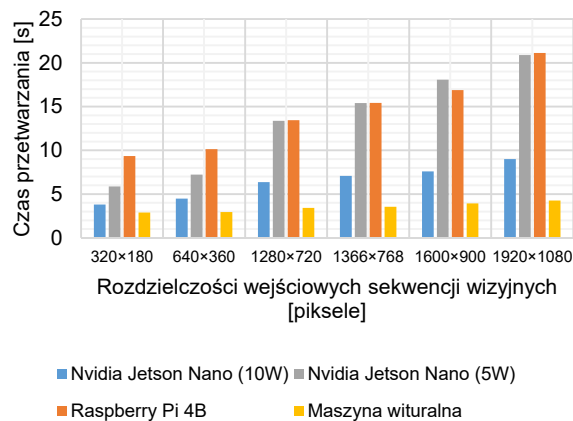
### Porównanie wydajności Raspberry Pi 4B oraz Nvidia Jetson Nano

W ramach prac eksperymentalnych związanych z porównaniem możliwości efektywnego przetwarzania sygnałów wizyjnych przygotowano nagrania o łącznej długości 3 godz. 33 min. Sekwencje wizyjne przedstawiające płaszczyzny lotniskowe wraz z oznaczeniami poziomymi zostały zarejestrowane w Porcie Lotniczym Poznań–Ławica. Jak wspomniano w rozdziale

wprowadzającym, celem eksperymentu była segmentacja linii na drodze startowej portu lotniczego w celu określenia poprawnego toru jazdy.

Porównanie szybkości przetwarzania sekwencji wizyjnych w systemach wbudowanych w postaci Raspberry Pi 4B oraz Nvidia Jetson Nano dokonane zostało na 10-sekundowych fragmentach, wybranych losowo spośród 1278 części, na które zostało podzielone całe nagranie zarejestrowane w porcie lotniczym. Dodatkowo, każdy z fragmentów przeskalowany został do 6 rozdzielczości: 1920×1080 (Full HD), 1600×900 (HD+), 1366×768 (HD), 1280×720 (WXGA), 640×360 (nHD), 320×180. Następnie, na obydwu mikrokomputerach uruchomiono algorytmy wykrywające linie, oparte o klasyczne przetwarzanie obrazów, na przygotowanych wcześniej sekwencjach wizyjnych. Założeniem programów było wyodrębnienie z obrazu przedziału odcieni kolorów linii, które w przypadku oznaczeń lotniskowych są białe lub żółte, co stanowi dobry kontrast wraz z szarym lub ciemnoszarym tłem nawierzchni. Wyizolowanie z obrazu obszarów żółtych i białych zostało poprzedzone zmianą rozdzielczości oraz przestrzeni barw z RGB na HSV, aby uniknąć zakłóceń w postaci nierównomiernego oświetlenia, które mogłoby skutkować pojawieniem się ciemniejszej lub jaśniejszej skali kolorów. Po zdefiniowaniu odcieni odpowiedzialnych za te dwa kolory ustawiono także odpowiednie progi saturacji i składowej będącej mocą światła białego. W ten sposób uzyskano maskę, na której wykryte kolory są pikselami o wartości 255, czyli mają kolor biały, a pozostała część klatki ma wartość 0, czyli kolor czarny. Tak przygotowana tablica może posłużyć jako wejście do algorytmu Canny'ego do wykrywania krawędzi [14]. Drugi algorytm został dodatkowo wyposażony w filtrację wykrytych krawędzi pod względem ich orientacji za pomocą maski Scharra dla kąta 0 stopni.

### Porównanie szybkości przetwarzania sekwencji wizyjnych w zależności od rozdzielczości (alg. 1)



Rys. 9 Porównanie szybkości przetwarzania sekwencji wizyjnych przez mikrokomputery Nvidia Jetson Nano raz Raspberry Pi 4B w zależności od rozdzielczości dla algorytmu nr 1

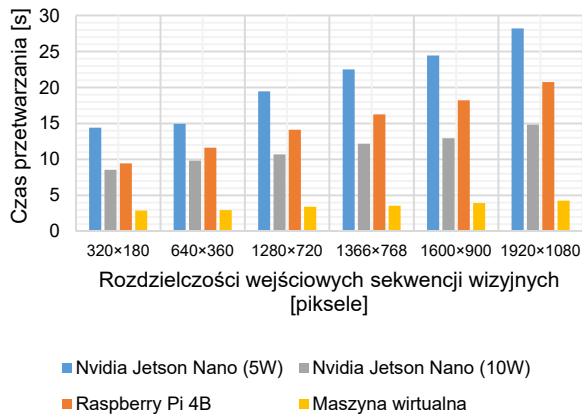
Na rys. 9 i 10 przedstawione zostało porównanie średnich szybkości przetwarzania 10 sekundowych fragmentów w zależności od ich rozdzielczości i wykorzystanego systemu wbudowanego dla poszczególnych algorytmów.

Pierwszy algorytm, z racji mniejszej złożoności obliczeniowej charakteryzuje się krótszymi czasami przetwarzania od drugiego algorytmu. Wraz ze wzrostem rozdzielczości następuje wydłużenie okresu działania programu. Różnice wydajności maszyny wirtualnej oraz Nvidia Jetson Nano w trybie MaxN dla najmniejszych



rozdzielczości są niewielkie w porównaniu do pozostałych badanych przypadków. W porównaniu tym najgorszy wynik uzyskał mikrokomputer Raspberry Pi 4B, którego charakterystyka odpowiada Nvidia Jetson Nano w trybie ograniczonej wydajności (5W).

### Porównanie szybkości przetwarzania sekwencji wizyjnych w zależności od rozdzielczości (alg. 2)



Rys. 10 Porównanie szybkości przetwarzania sekwencji wizyjnych przez mikrokomputery Nvidia Jetson Nano raz Raspberry Pi 4B w zależności od rozdzielczości dla algorytmu nr 2

Złożoność obliczeniowa drugiego algorytmu pokazuje wydłużenie czasu przetwarzania danych wejściowych w systemach wbudowanych. W tym przypadku najlepsze rezultaty również uzyskała Nvidia Jetson Nano w trybie MaxN. Tryb 5W każdorazowo okazał się być wolniejszy od programu uruchamianego na mikrokomputerze Raspberry Pi 4B.

Z przeprowadzonych doświadczeń jednoznacznie wynika, że operacje dokonywane przez Nvidia Jetson Nano są szybsze nie tylko w ramach działania sieci neuronowych, ale również w obszarze klasycznego przetwarzania obrazów. Warto zaznaczyć, że Nvidia Jetson Nano posiada wspomniany wcześniej układ graficzny usprawniający w znacznym stopniu uczenia maszynowe, jednak procesor zamontowany w tej jednostce charakteryzuje się mniejszym taktowaniem niż w przypadku Raspberry Pi 4B, co w wielu przypadkach nie ma negatywnego wpływu na osiągnięte wyniki.

Warto zaznaczyć, że wykorzystanie pełnego potencjału Nvidia Jetson Nano możliwe jest przy zadaniach realizowanych z wykorzystaniem uczenia maszynowego, które wymaga użycia specyficznych bibliotek i dostosowanego kodu.

### Podsumowanie

Wyniki przeprowadzonych eksperymentów pokazują, że zastosowanie systemów wbudowanych do analizy sekwencji wizyjnych nie ustępuje swoją wydajnością w znaczący sposób w stosunku do jednostek o większej mocy obliczeniowej. Należy jednak podkreślić fakt, iż uzyskane wyniki były możliwe przy znacznie mniejszym poborze energii.

Mikrokomputer Raspberry Pi 4B, mimo gorszych wyników w porównaniu z drugim badanym modelem Nvidia Jetson Nano, biorą pod uwagę jego zapotrzebowanie energetyczne, które stanowi maksymalnie 75% wartości konkurenta, osiąga zadowalające rezultaty. Ze względu

jednak na charakter pracy i potencjalne możliwości rozszerzenia funkcjonalności, racjonalnym wydaje się być użycie Nvidia Jetson Nano jako jednostki centralnej omawianego systemu wizyjnego.

**Autor:** mgr inż. Jakub Suder, Politechnika Poznańska, Instytut Automatyki i Robotyki, Zakład Układów Elektronicznych i Przetwarzania Sygnałów, ul. Jana Pawła II 24, 60-965 Poznań, E-mail: [jakub.suder@put.poznan.pl](mailto:jakub.suder@put.poznan.pl)

### LITERATURA

- [1] Arshad N. M., Razak N. A., Vision-based detection technique for effective line-tracking autonomus vehicle, IEEE 8th International Colloquium on Signal Processing and its Applications, 2012, pp. 441-445, doi: 10.1109/CSPA.2012.6194765
- [2] Barua B., Gomes C., Baghe S., Sisodia J., A Self-Driving Car Implementation using Computer Vision for Detection and Navigation, 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 271-274, doi: 10.1109/ICCS45141.2019.9065627
- [3] Podbucki K., Suder J., Marciniak T., Dąbrowski A., CCTV based system for detection of anti-virus masks, 2020 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2020, pp. 87-91, doi: 10.23919/SPA50552.2020.9241303
- [4] Podbucki K., Suder J., Marciniak T., Dąbrowski A., Elektroniczna matryca pomiarowa do badania lamp lotniskowych, Przegląd Elektrotechniczny, No. 02/2021, pp. 47-51, doi: 10.15199/48.2021.02.12
- [5] Suder J., Maciejewski P., Podbucki K., Marciniak T., Dąbrowski A., Platforma pomiarowa do badania jakości działania lamp lotniskowych, Pomiary Automatyka Robotyka, R. 23, No. 2/2019, pp. 5-13, doi: 10.14313/PAR\_232/5
- [6] Suder, J.; Podbucki, K.; Marciniak, T.; Dąbrowski, A. Low Complexity Lane Detection Methods for Light Photometry System. Electronics 2021, 10, 1665. <https://doi.org/10.3390/electronics10141665>
- [7] Certification Specifications (CS) and Guideline Material (GM) for Aerodrome Design Edition 3, Annex to Decision No. 2016/027/R of the EASA Executive Director, European Aviation Safety Agency, 2016
- [8] Warden P., Situnayake D., TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low power microcontrollers, O'Reilly Media, 2020
- [9] Raspberry Pi 4, 3A+, Zero W - specs, benchmarks & thermal tests, available online: <https://magpi.raspberrypi.org/articles/raspberry-pi-specs-benchmarks>
- [10] Vijitkunsawat W., Chantngarm P., Comparison of Machine Learning Algorithm's on Self-Driving Car Navigation using Nvidia Jetson Nano, 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2020, pp. 201-204.
- [11] Jain A. K., Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 1630-1635.
- [12] Çivik E., Yüzgeç U., Deep Learning Based Continuous Real-Time Driver Fatigue Detection for Embedded System, 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1-4.
- [13] Süzen A. A., Duman B., Şen B., Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN, 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-5.
- [14] Sun Z., Vision Based Lane Detection for Self-Driving Car, 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), 2020, pp. 635-638.