# Autonomiczny Pojazd

Wykonawcy:
Konstanty Odważny,
Jędrzej Szczerbal,
Paweł Chumski

Promotor:
dr hab. inż. Tomasz Pajchrowski

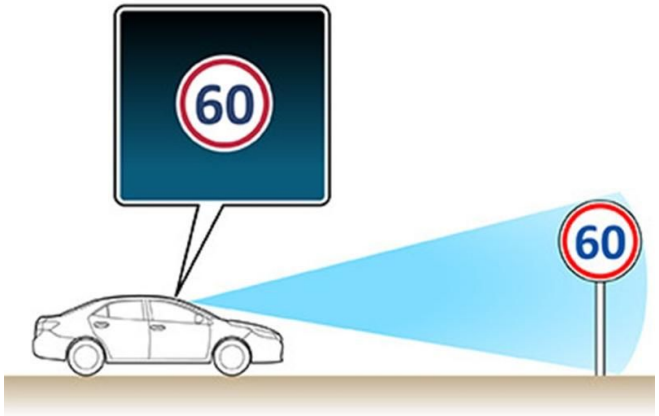
# Współpraca z kołem naukowym RAI

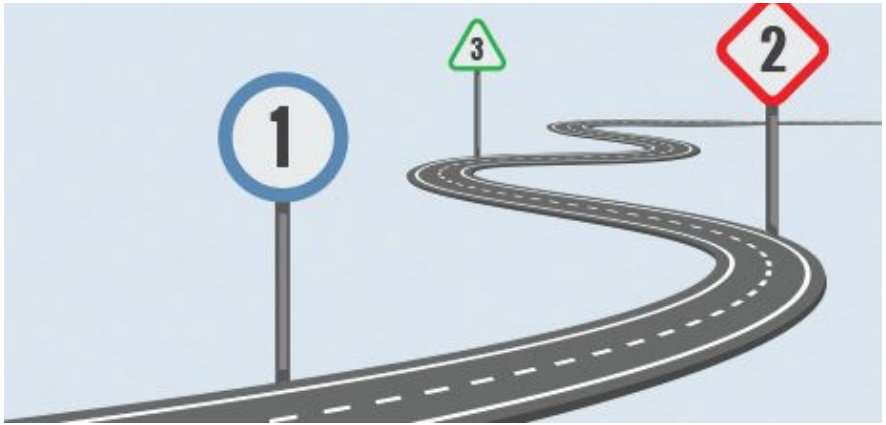
# Plan prezentacji

1. Koncepcja projektu
2. Szczegóły techniczne platformy pojazdu
3. Zastosowane technologie
4. Śledzenie linii
5. Własny kontroler
6. Rozpoznawanie obrazu
7. Interfejs użytkownika

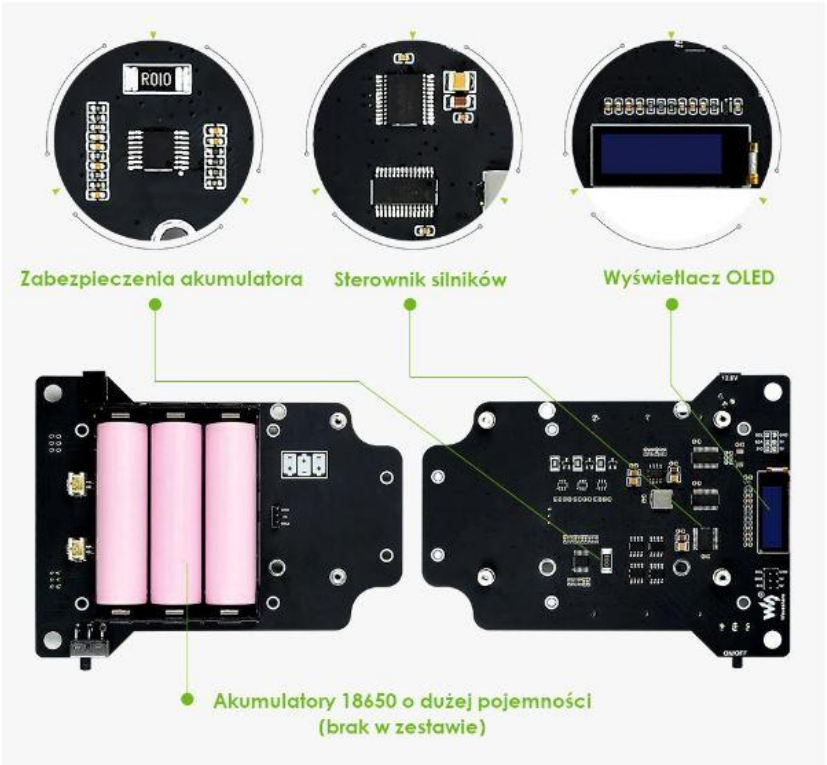
# Koncepcja projektu

- Stworzenie pojazdu poruszającego się po przygotowanej planszy, który będzie potrafił rozpoznać znaki i odpowiednio się do nich zastosować
- Możliwość sterowania zdalnego za pomocą pada
- Umiejętność zapamiętania stworzonej trasy poprzez jazdę po niej za pomocą kontrolera



# Nvidia Jetracer





Serwo z metalowymi zebatkami

Metalowy silnik

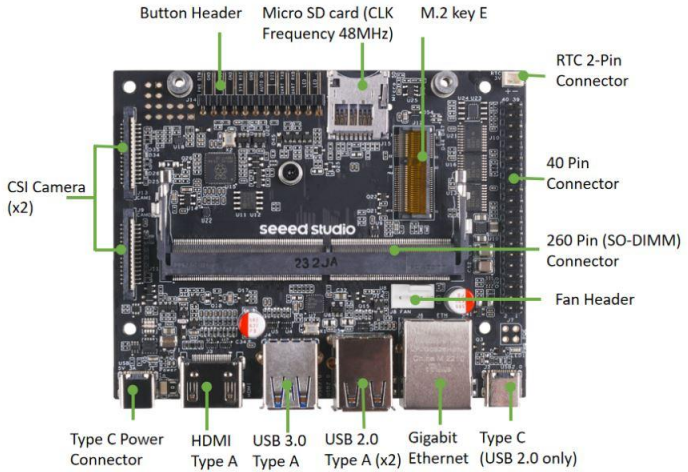
Moduł komunikacji bezprzewodowej

# Jetson Nano i CSI camera

## Jetson Nano Developer Kit Specification

| | |
|---|---|
| **GPU** | 128-core NVIDIA Maxwell™ architecture GPU |
| **CPU** | Quad-core ARM® Cortex®-A57 MPCore processor |
| **Memory** | 4GB 64-bit LPDDR4 |
| **Storage** | microSD (Card not included) |
| **Video Encode** | 1x 4K30 \| 2x 1080p60 \| 4x 1080p30 \| 9x 720p30 (H.264/H.265) |

Top View



# Zastosowane technologie

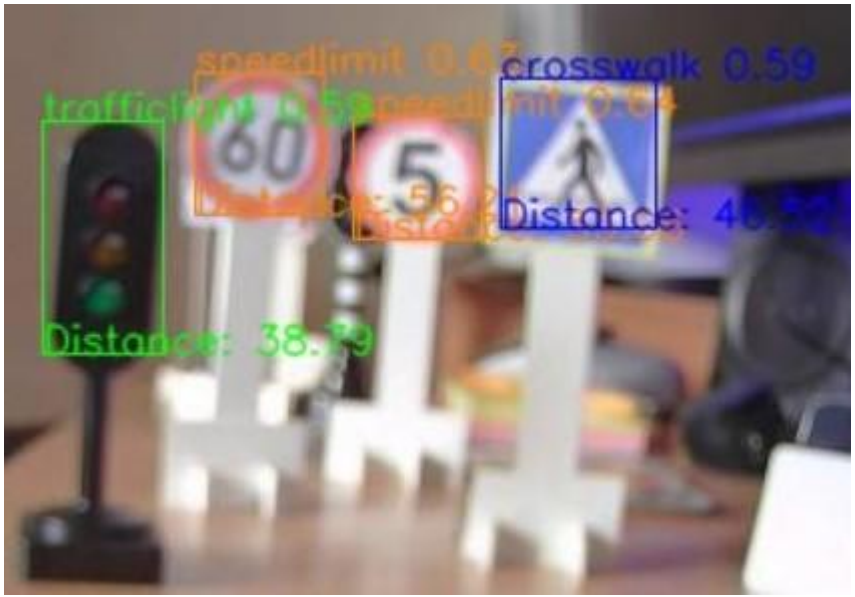

YOLOv5



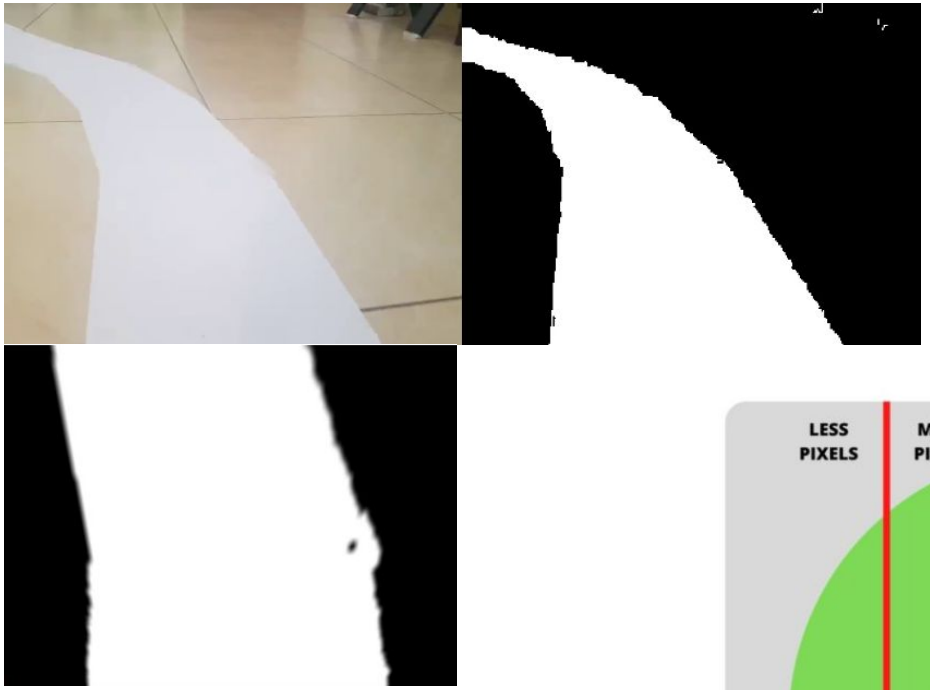

PyTorch

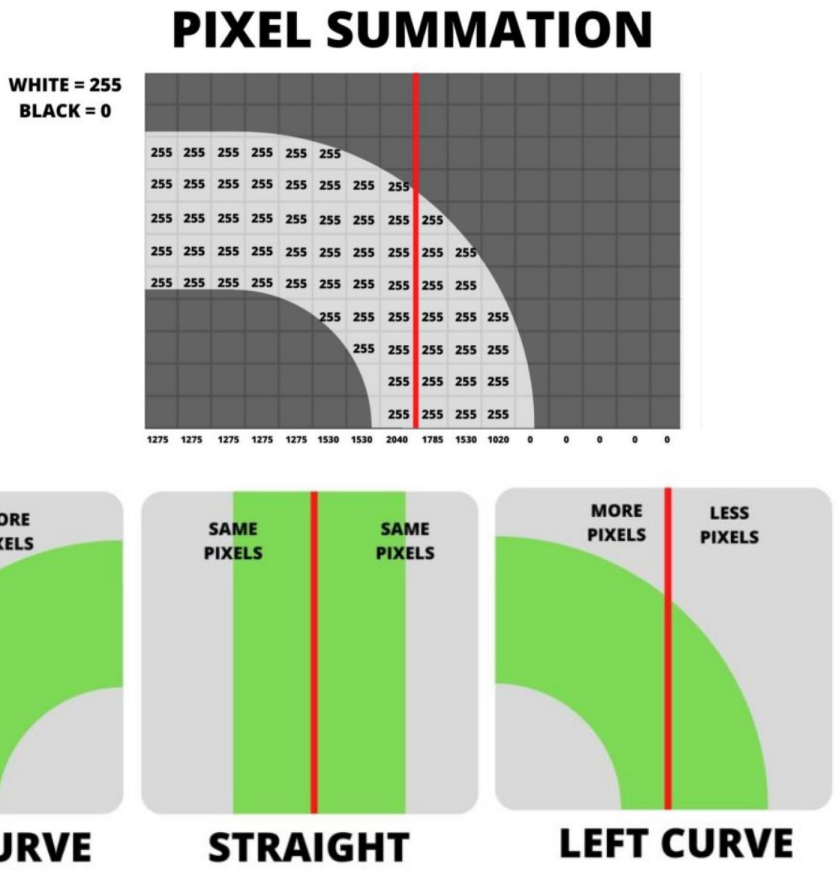
# OpenCV

- Biblioteka typu open-source służąca do przetwarzania obrazów oraz machine learning'u
- Zintegrowana i kompatybilna z wieloma innymi bibliotekami np. numpy
- Można używać jej w wielu językach programowania



# Śledzenie linii





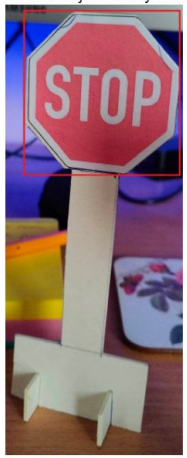
# YOLO - You Only Look Once

1. Lokalizacja obiektu
2. Klasyfikacja obiektu
3. Rozpoznanie obiektu

Klasyfikacja


“Stop”

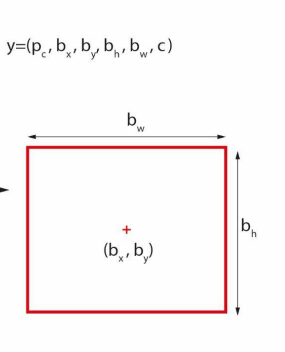
Lokalizacja z klasyfikacją



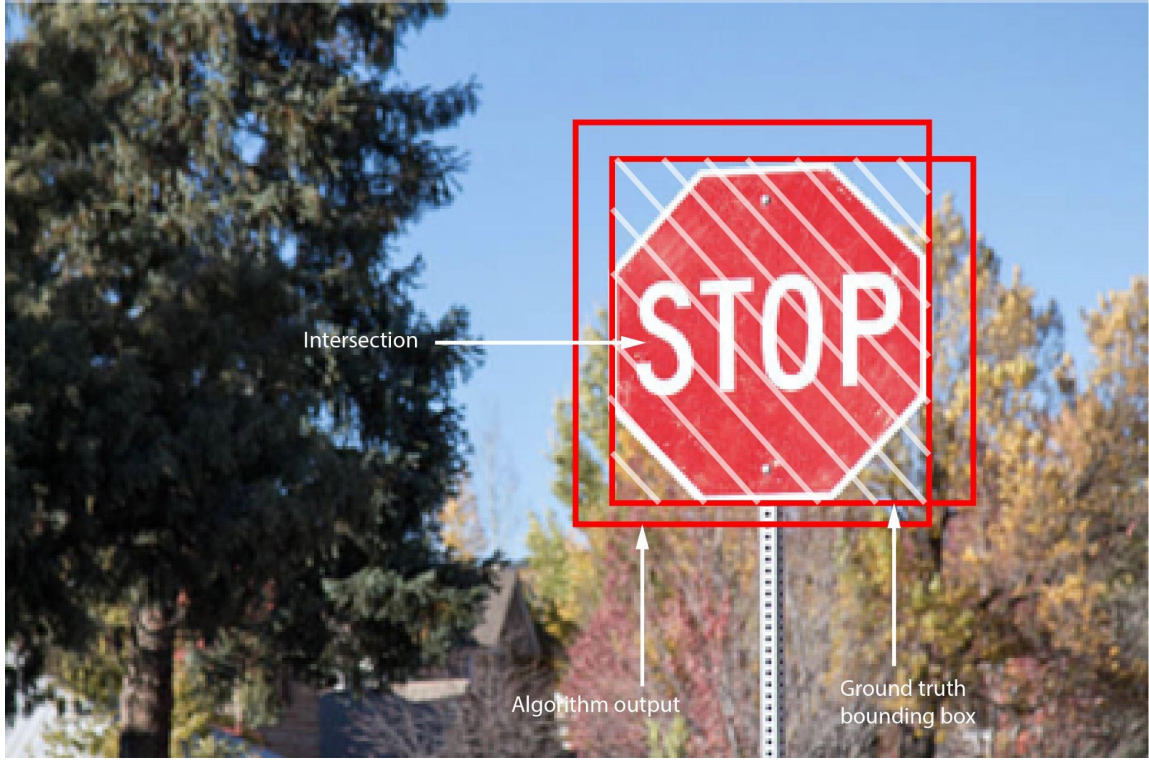

“Stop”

Detekcja



# Yolo v5




$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$TP$ = True positive

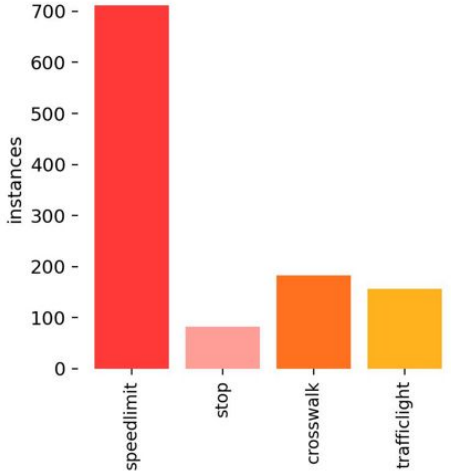
$TN$ = True negative

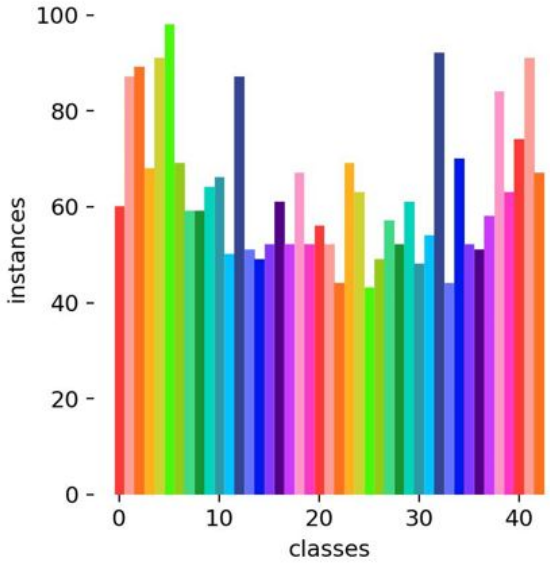
$FP$ = False positive

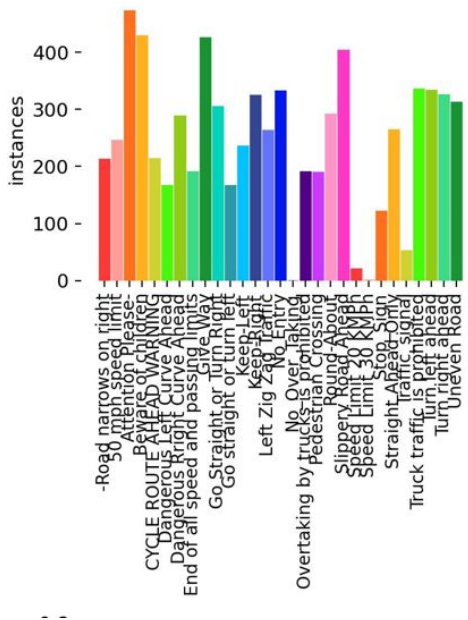
$FN$ = False negative

# Wybór zbioru danych

- Wybór danych z platformy Kaggle lub Roboflow
- Zwrócenie uwagi na ilość klas oraz zbalansowanie danych



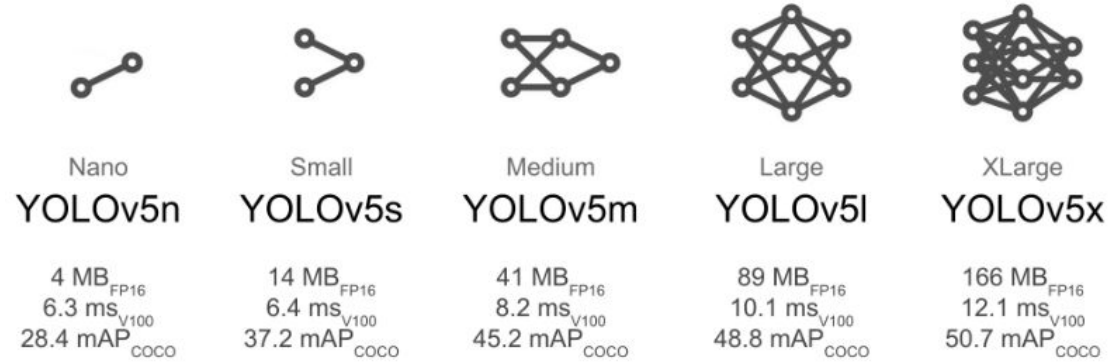
# Przygotowanie zbioru uczącego

- Podział na obrazy oraz pliki opisujące (labels) w formacie .txt, gdzie każdy plik jest zapisany w postaci wzorca:

```
[Class Number] [center in x] [center in y] [Width] [Height]
```

```
2 0.7378676470588236 0.5125 0.030147058823529412 0.055
```

- Podział zbioru na zbiór treningowy oraz walidacyjny, w proporcji 90% - treningowy, 10% - walidacyjny
- Utworzenie pliku konfiguracyjnego .yaml
- Wybór wcześniej wytrenowanego modelu Yolov5: yolov5x,yolov5l, yolov5m, yolov5s, yolov5n



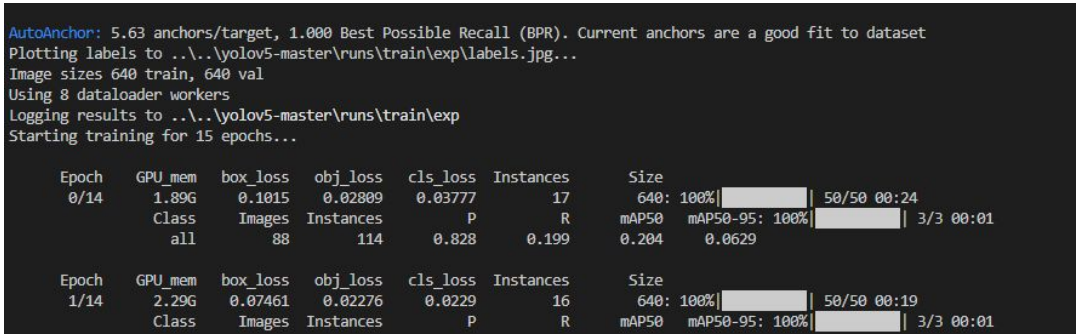

```yaml
train: ../data/images/training/
val: ../data/images/validation/

# number of classes
nc: 4

# class names
names: ['speedlimit', 'stop', 'crosswalk', 'trafficlight']
```

# Trenowanie modelu

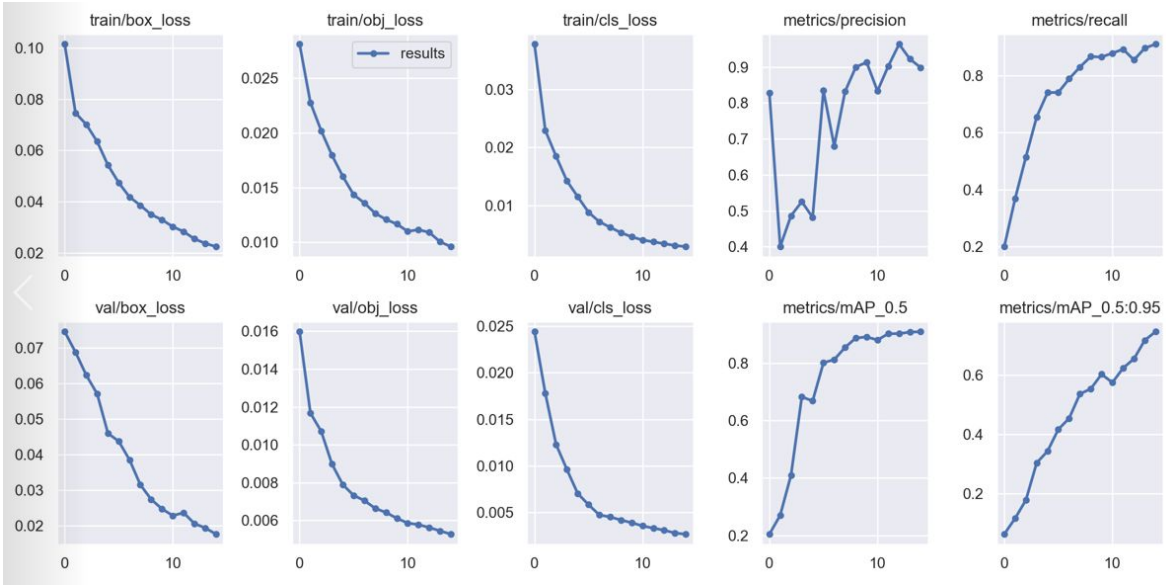
- Wybór wielkości obrazów wejściowych (standardowo 640x640 px )
- Wybór precyzji: FP32, FP16 lub INT8
- Wybór ilości epok szkolenia (ang. epochs) oraz ilości obrazów pobieranych za jednym razem do trenowania (ang. batch size)
- Walidacja - testowanie w trakcie trenowania
- Trenowanie z użyciem GPU
- Ocena jakości modelu, wskaźniki jakości, wykresy



# Wybór modelu

Kryteria:

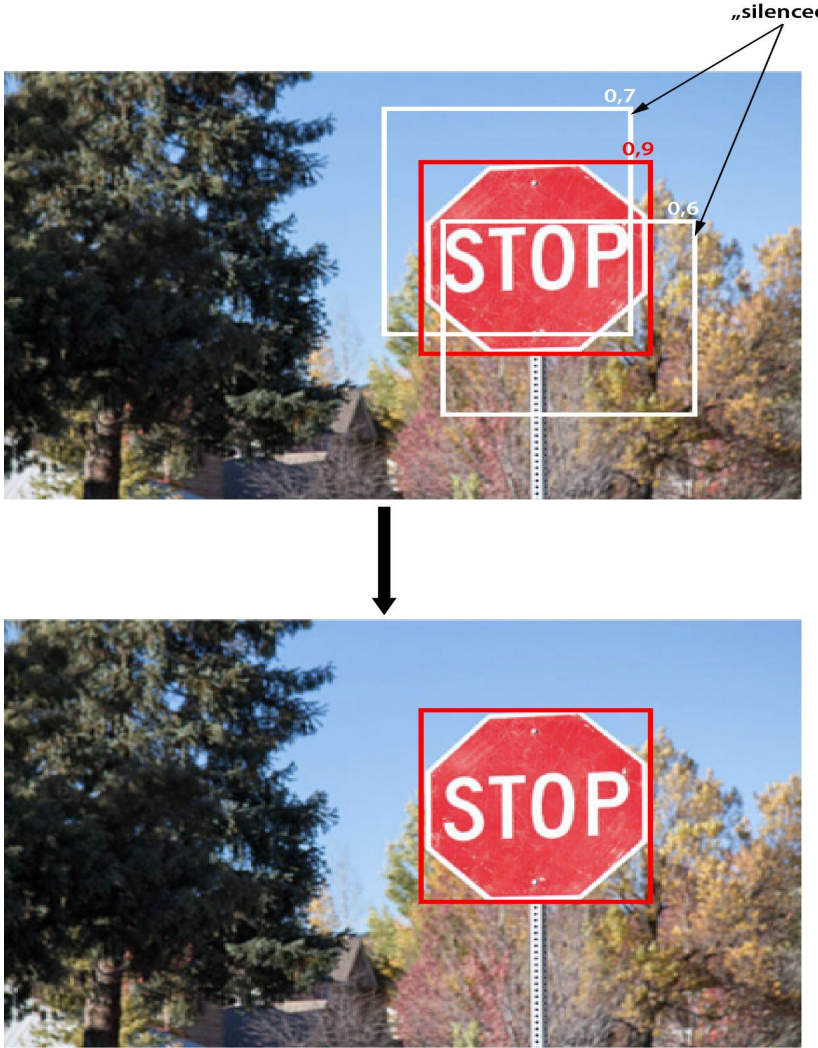
- Dopasowanie modelu
  - Nadmierne dopasowanie
  - Niedopasowanie
- Ilość klatek na sekundę osiągnięta na platformie (ang. FPS)
- Funkcje strat (ang. loss)
- Funkcje metryki - mAP (mean average precision)



# Testowanie modelu

Kroki rozpoznawania obrazu:

- Przygotowanie obrazu ( image preprocess) - zmiana wielkości, normalizacja
- Załadowanie modelu sieci neuronowej
- Predykcja - Próg ufności, NMS - non-maximum suppression
- Zaznaczenie rozpoznanego obiektu na obrazie za pomocą ramki ograniczającej (ang. bounding box)

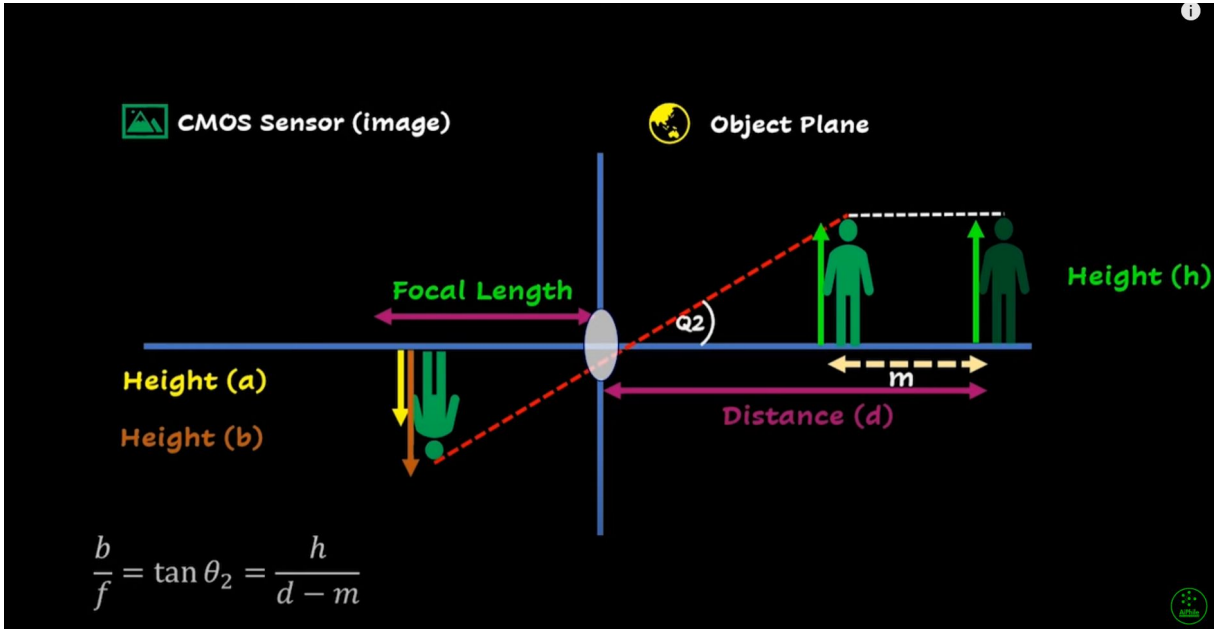


# Testowanie modelu

- Testowanie na obrazach:


# Estymacja odległości od obiektu

1. Odczytanie wartość ogniskowej obiektywu kamery ( ang. focal length).
2. Obliczanie odległości na podstawie ramek znalezionych w fazie rozpoznawania obiektów.



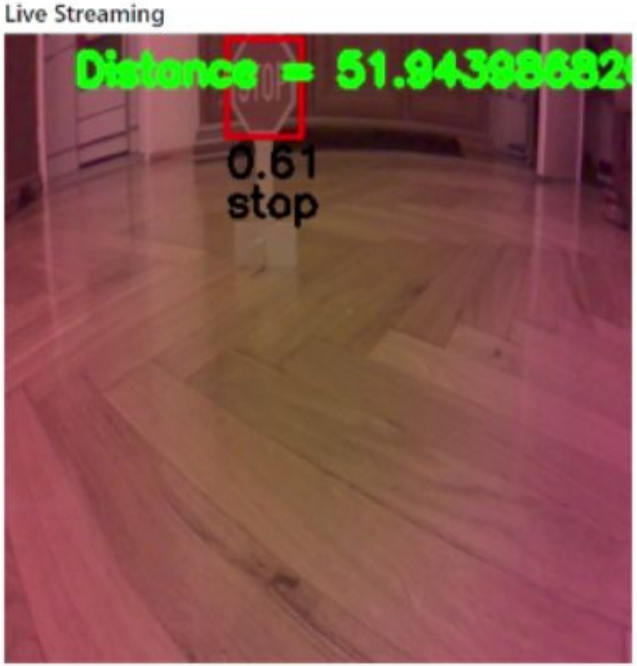
# Testowanie modelu

# Interfejs użytkownika

Strona klienta REST, stworzona za pomocą biblioteki Flask

Zalety:

- Łatwość integracji z resztą zadań
- Całość projektu w języku Python
- Możliwość obserwacji obrazu z kamery na żywo



# Własny kontroler

Założenia wstępne:

- komunikacja BT oraz Wifi
- podstawowe sterowanie pojazdem
- przełączanie trybu ręcznego oraz automatycznego
- odbieranie obrazu z kamery
- zasilanie z opcją ładowania
- możliwość programowania w języku C++
- elastyczność pod przyszłe funkcjonalności

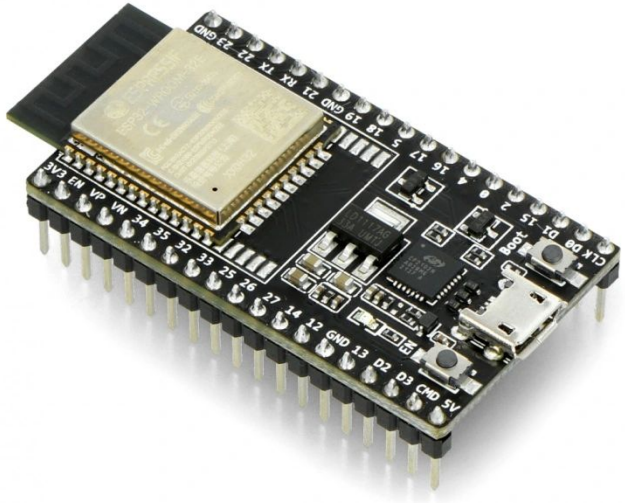
# Mikrokontroler

Wybrany model: ESP32-WROOM-32D

- procesor dual-core Xtensa 32-bit LX6 - 240MHz
- 448kB pamięci ROM oraz 520kB pamięci SRAM oraz 4MB pamięci flash
- obsługa magistral cyfrowych, m.in.: I2C, **SPI**, I2S, PWM, UART, IR, CAN
- Standard BT: Bluetooth v4.2 BR/EDR oraz BLE
- Protokoły Wi-Fi: 802.11 b/g/n/d/e/i/k/r
- przetwornik ADC oraz DAC

ESP32 devkitc v4:

- konwerter USB/UART(Silabs CP2102)
- stabilizator AMS1117
- Wyprowadzenia GPIO o rozstawie 2,54 mm



# Komunikacja

Radiowa w standardzie Bluetooth:

- standard BL 4.2 oraz BLE
- przepustowość do:
  1500Kbps(BT classic)
  700Kbps(BLE)
- połączenie master-slave/server-client
- zasięg <100m

Połączenie LAN przez wifi:

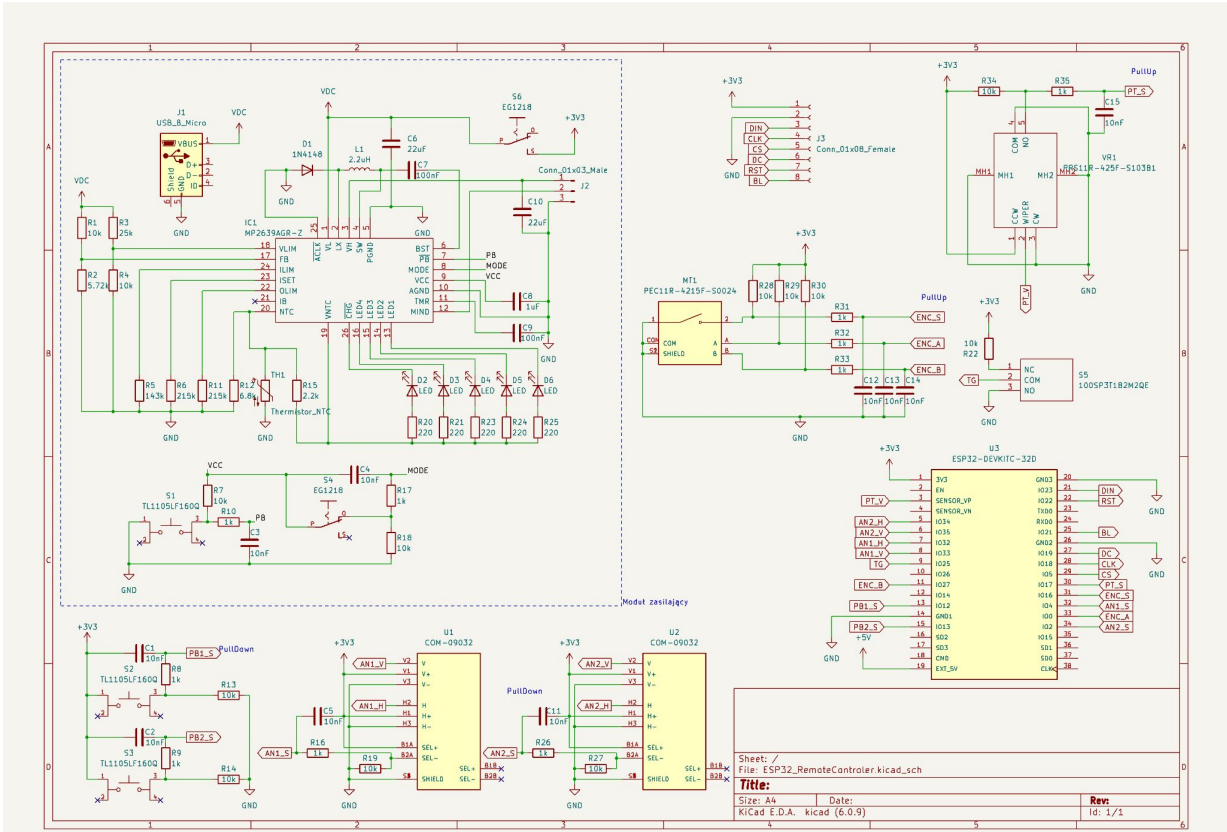
- częstotliwość 2.4GHz
- standardy 802.11 b/g/n
- przepustowość do 150Mbps

Możliwość jednoczesnej pracy obydwu standardów komunikacji przy odpowiednich ustawieniach.

# Schemat ideowy

Podstawowe elementy:

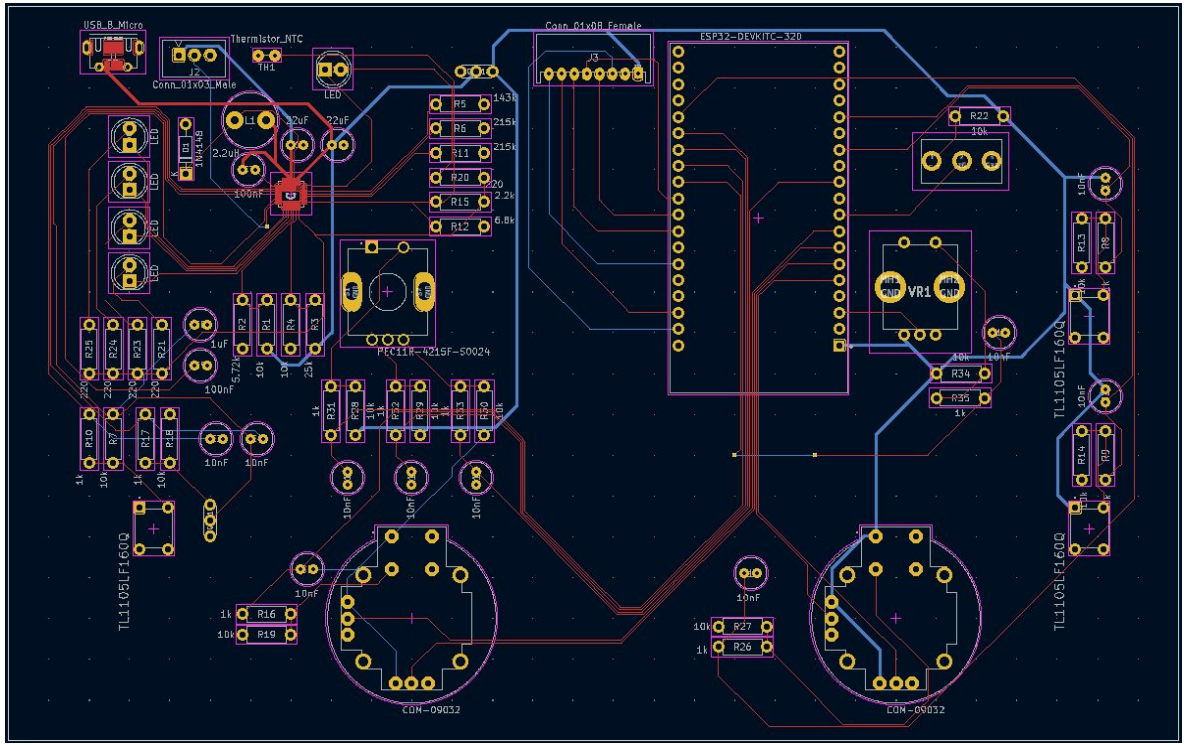
- wyświetlacz LCD TFT(SPI)
- joystick
- enkoder inkrementalny
- potencjometr
- przełącznik
- przyciski
- układ zasilający MP2639AGR -Z
- port micro USB B



# Wstępny projekt PCB

Wersja prototypowa zawiera głównie elementy THT. Ułożenie komponentów pasywnych oraz ich wartości wybrano uwzględniając zalecenia producenta.

Wybrano kompromis między ergonomią urządzenia oraz ograniczeniem miejsca i możliwości lutowniczych.



# Program

- Język: C++
- Wykorzystane środowisko: arduino IDE
- Biblioteki: "BluetoothSerial.h"/”WiFi.h”
- Rola master/client w zależności od wybranej komunikacji
- Funkcjonalność:
- Odczyt wartości z urządzeń wejściowych
- Wysyłanie ramki danych do pojazdu
- Żądanie obrazu z kamery, wyświetlanie oraz zapis do pamięci flash

# Dziękujemy za uwagę