# Smart Dom – YOLO + MQTT

**Piotr Deda, Łukasz Wajda**

## Wstęp

YOLO jest modelem wykorzystującym uczenie maszynowe do detekcji obiektów na obrazach. Celem projektu jest stworzenie przykładowego programu wykorzystującego YOLO w tematyce smart dom. Stworzony program pozwala na wysyłanie liczby osób i zwierząt domowych wykrytych na kamerze za pomocą protokołu MQTT. Do demonstracji programu stworzona została przykładowa automatyzacja w Home Assistant za pomocą Node-RED.

W projekcie użyto następujących technologii:

- Python (https://www.python.org)
- Ultralytics YOLOv8 (https://github.com/ultralytics/ultralytics)
- OpenCV (https://opencv.org)
- Home Assistant (https://www.home-assistant.io)
- Protokół MQTT (https://mqtt.org)
- Node-RED (https://nodered.orgt)

## Detekcja obiektów YOLO

Biblioteka YOLO używa głębokich konwolucyjnych sieci neuronowych do wykrywania obiektów. Do tego celu udostępnia komendy wiersza poleceń oraz funkcje języka Python pozwalające na detekcję i analizę wyników bezpośrednio w programie. Program działa w sposób mocno zautomatyzowany, potrafiąc na przykład samemu pobrać jeden z domyślnych wytrenowanych modeli, czy dokonać detekcji na zdjęciu lub filmie automatycznie pobranym z zadanego linku.

Posiada pięć trybów działania, pozwalających między innymi na śledzenie toru ruchu obiektów czy wykrywanie poz ciała. W projekcie zastosowano prosty tryb `detect`, który opatruje wykryte obiekty bounding boxem wraz z procentową wartością pewności. Poniżej zobrazowano przykładowe użycie tego trybu za pomocą komendy:

```
yolo detect predict model=yolov8n.pt source='https://live.staticflickr.com/65535/
50586034981_5e499a6934_c_d.jpg'
```


Figure 1: Przykładowy efekt wywołania komendy `yolo`

# Opis programu

Program używa domyślnego modelu YOLOv8 w wersji nano (*YOLOv8n*) do wykrywania ilości osób i zwierząt (psów i kotów) na poszczególnych klatkach obrazu z kamery, a następnie wysyła je jako wiadomości MQTT o tematach odpowiednio `yolo/people_count` i `yolo/animal_count` na skonfigurowany serwer MQTT.

W celu ograniczania ilości przesyłanych wiadomości oraz wykorzystania zasobów, program domyślnie działa w trybie ze spowolnieniem. Gdy w zasięgu kamery nie znajdują się żadni ludzie, detekcja odbywa się z pełną prędkością, dla zapewnienia maksymalnie szybkiej informacji o pojawieniu się osoby. W momencie kiedy dowolna liczba osób zostanie wykryta, program wchodzi w spowolniony tryb i wykrywa osoby jedynie co pewien odstęp czasu (konfigurowalny, domyślnie 2 sekundy). Przechodzenie w tryb spowolnienia można wyłączyć, jednak wtedy liczba wykrytych osób może się chaotycznie zmieniać, a program w ciągu kilku sekund będzie wysyłał nawet kilkadziesiąt wiadomości.

# Konfiguracja programu

Konfiguracja projektu odbywa się poprzez edycję pliku `config.ini` zamieszczonego w katalogu programu.

```
[MQTT]
Host = example.com
Port = 1883
Username = user
Password = pass

[YOLO]
Camera = 0
WaitTime = 2
RapidMode = false
Preview = false
Threshold = 0.25
```

Kategoria `[MQTT]` zawiera dane potrzebne do połączenia się z serwerem MQTT Home Assistanta (lub dowolnym innym). W celu zachowania poufności danych takie jak hasło, można je nadpisać za pomocą zmiennych środowiskowych, np.:

```
set MQTT_HOST=example.com
set MQTT_PORT=1883
set MQTT_USERNAME=user
set MQTT_PASSWORD=pass
```

Kategoria `[YOLO]` pozwala na zmianę ustawień dotyczących detekcji obiektów:

- Camera – numer kamery używanej do detekcji (zazwyczaj `0` dla pierwszej kamery)
- WaitTime – czas oczekiwania w sekundach gdy wykryto ludzi na obrazie
- RapidMode – wartość `true` wyłącza kompletnie czas oczekiwania
- Preview – wartość `true` włącza okno OpenCV z podglądem obiektów wykrytych przez YOLO
- Threshold – procentowy poziom pewności modelu do zaliczenia obiektu jako osoby

## Opis przykładowej automatyzacji

Automatyzacja przygotowana jest do uruchamiania w pracowni 209 na wcześniej skonfigurowanym urządzeniu. Wykorzystuje dwie żarówki używane wcześniej na zajęciach, które zapalają się z jasnością zależną od ilości wykrytych zwierząt (żarówka “pierwsza”) i ludzi (żarówka “druga”). Jeśli liczba wykrytych obiektów wynosi 0, odpowiednia żarówka gaśnie.

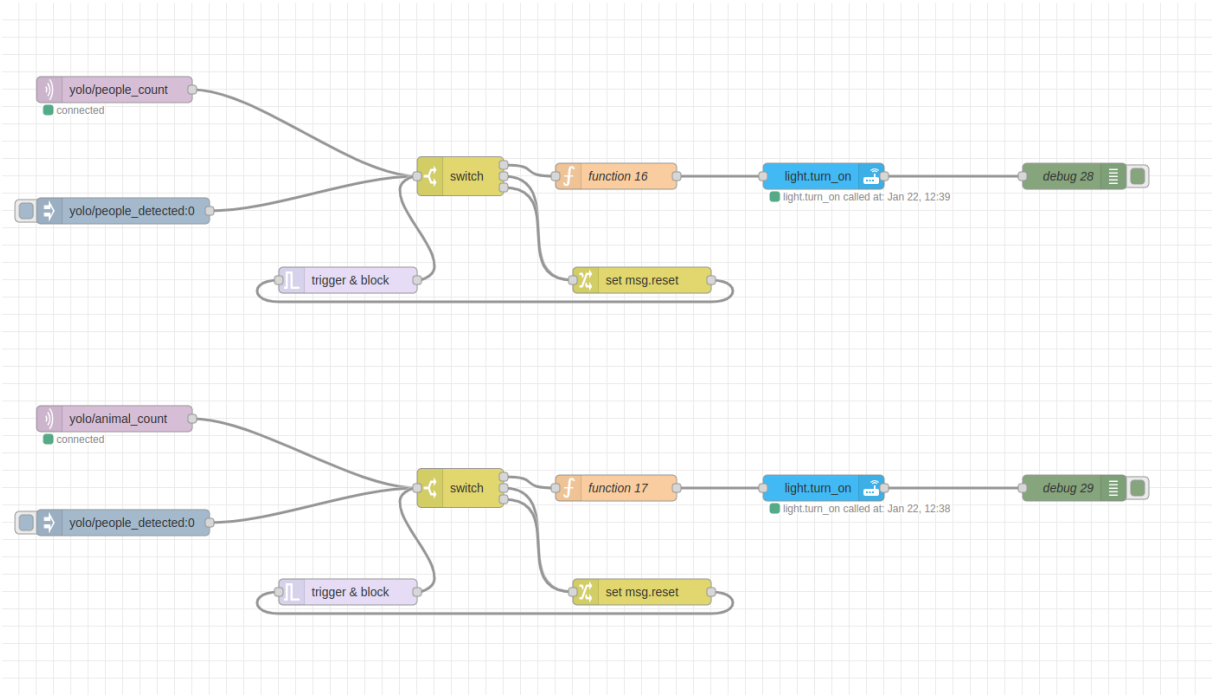


Figure 2: Schemat połączeń w przykładowej automatyzacji