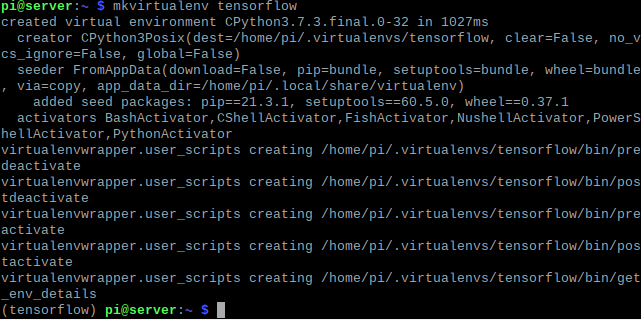
# Detekcja obiektów na Raspberry Pi za pomocą TensorFlow Lite:

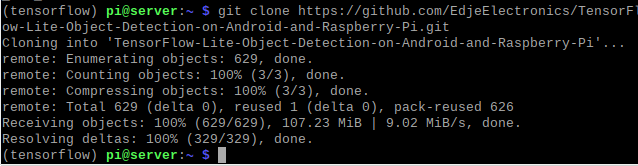
**TensorFlow – otwarta biblioteka programistyczna napisana przez Google Brain Team. Używa się jej w uczeniu maszynowym i głębokich sieciach neuronowych. Biblioteka może wykorzystywać do działania karty graficzne, procesory jak i mikroprocesory z akceleratorami AI. Biblioteka składa się z wielu modułów, w jej najniższej warstwie jest rozproszony silnik wykonawczy – ang. Distributed Execution Engine, wyżej znajdują się frontendy napisane w kilku językach programowania m.in. w Pythonie. Powyżej zostałą umieszczona warstwa API, która zapewnia prostszy interface dla powszechnie używanych warstw w modelach głębokiego uczenia. W niniejszym przykładzie została przedstawiona wersja biblioteki jako „Lite”. Jej kluczowe cechy to:**

* **Optymalizacja pod kątem uczenia maszynowego rozwiązujące kluczowe ograniczenia: opóźnienie, prywatność, łączność, rozmiar oraz zużycie energii.**
* **Obsługa wielu platform, obejmująca urządzenia z systemem Android i iOS**
* **Obsługa różnych języków programowania m.in.: Java, Swift, C++ i Python**
* **Wysoka wydajność z akceleracją sprzętową**
* **Kompleksowe przykłady typowych zadań uczenia maszynowego takich jak: klasyfikacja obrazów, wykrywanie obiektów, szacowanie pozycji, odpowiadanie na pytania, klasyfikacja tekst itp.**
* **Plik binarny TensorFlow Lite ma ~1MB.**

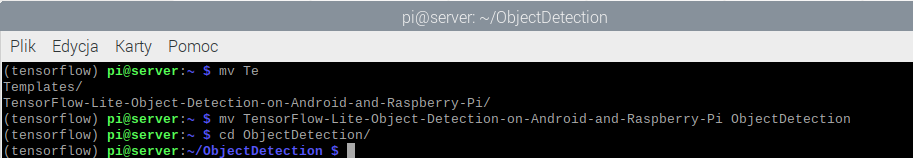
Tworzymy nowe wirtualne środowisko:



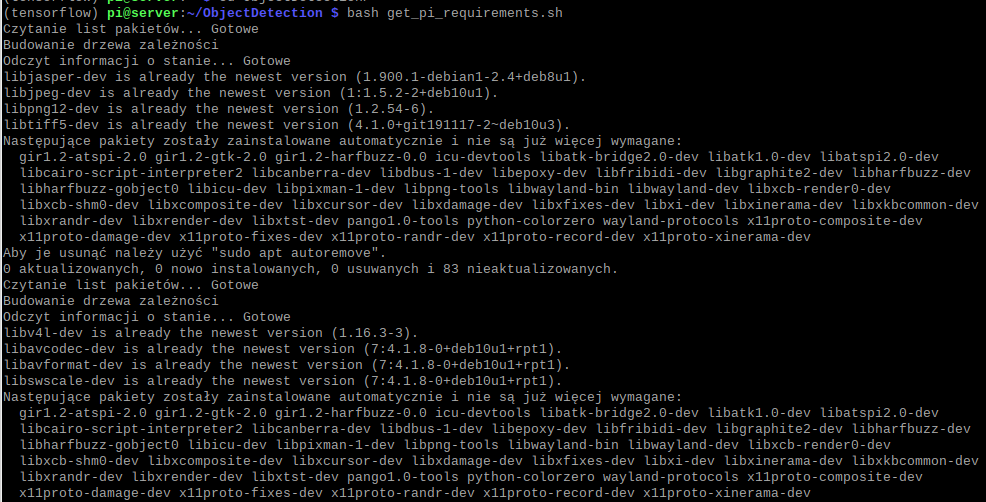
Pobieramy repozytorium github:



Zmieniamy nazwę katalogu na krótszą:



Instalujemy OpenCV oraz TensorFlow poprzez dostępny skrypt:

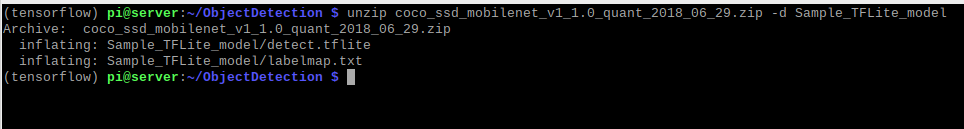


Następnie skonfigurujemy model wykrywania, który będzie używany z TensorFlow Lite. Z modelem wykrywania są powiązane dwa pliki: plik detect.tflite (który jest samym modelem) oraz plik labelmap.txt (który zawiera mapę etykiet dla modelu). Google dostarcza przykładowy model wykrywania obiektów SSDLite-MobileNet-v2, który jest wytrenowany z zestawu danych MSCOCO i przekonwertowany do działania w TensorFlow Lite. Może wykryć i zidentyfikować 80 różnych popularnych obiektów, takich jak ludzie, samochody, kubki itp.

Pobieranie modelu:



Rozpakowujemy pliki:



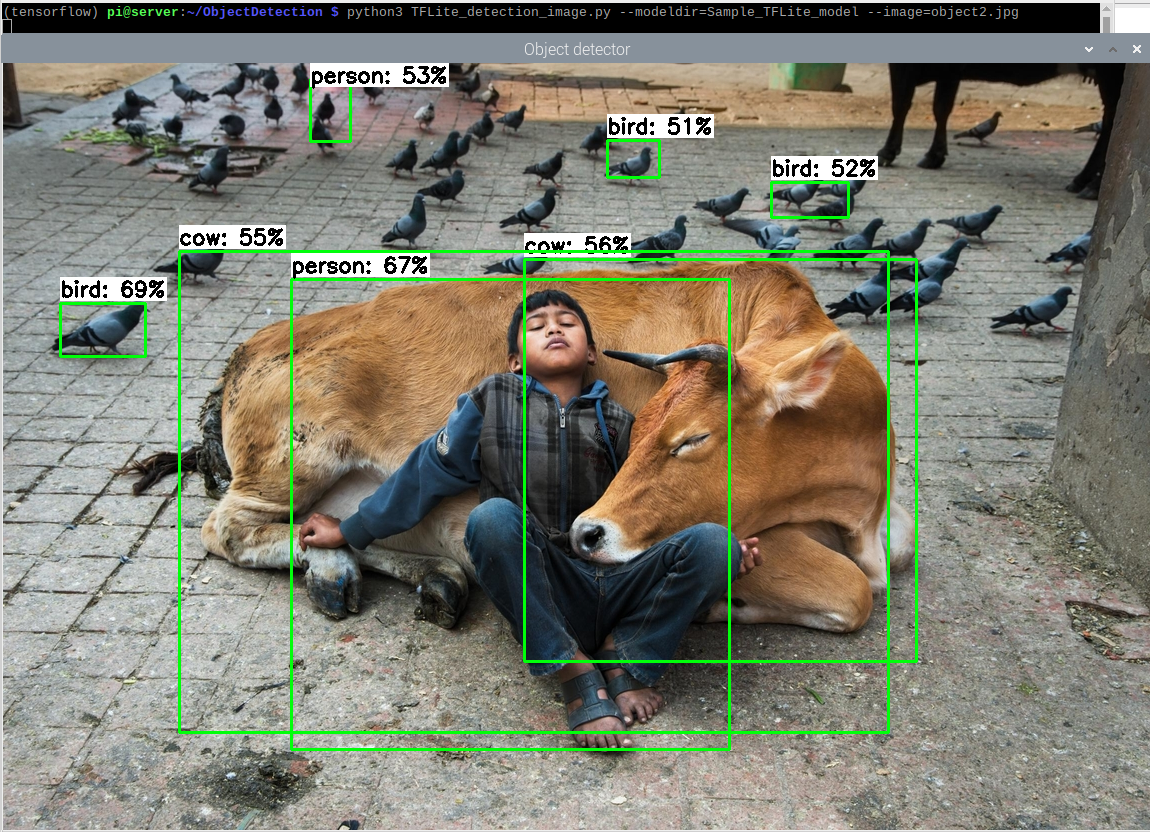
Urchamianie i sprawdzanie działania skrytów:

Najpierw sprawdzamy jak działa ze statycznymi obrazami poniżej oryginalny obraz:



Wynik działania skrytpu

$ python3 TFLite\_detection\_image.py --modeldir=Sample\_TFLite\_model –image=object2.jp

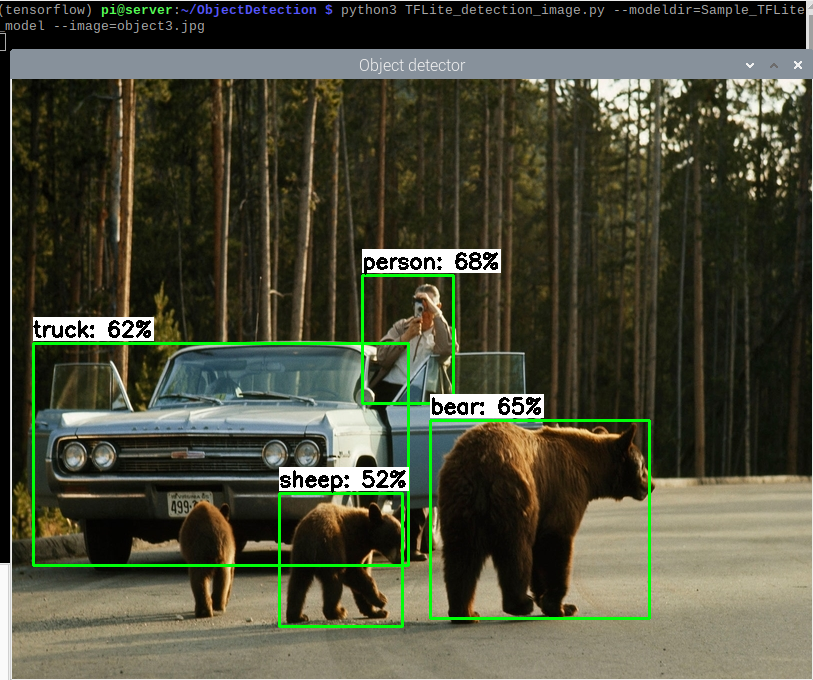


Do załadowania określonego zdjęcia wykorzystujem,y parametr –image=nazwa.obrazu

Kolejny przykład:



Wynik skryptu:



Zobaczymy teraz jak tensorflow radzi sobie z obrazem w zasie rzeczywistym do tego testu wywołamy obraz z kamery podłączonej do naszego raspberry pi:

**$python3 TFLite\_detection\_webcam.py –modeldir=Sample\_TFLite\_model**

[Obraz zawierający tekst, osoba, wewnątrz

Opis wygenerowany automatycznie](real_time.mkv)