

Face Mask Detection

Założeniem projektu jest wykrywanie twarzy oraz detekcja maski w czasie rzeczywistym. Program radzi sobie dobrze w różnych warunkach oświetleniowych oraz jest w stanie wykrywać wiele osób jednocześnie.

Struktura Projektu:

Projekt podzielony jest na 2 główne foldery: *"scripts"* oraz *"src"*. W folderze *"src"* znajdują się pliki, które definiują wszystkie aspekty techniczne programu jak uczenie się modelu, gotowy zapisany model, exceptions, rysowanie kwadratów wokół wykrytych twarzy oraz tym podobne metody pomocnicze. Folder *"scripts"* natomiast zawiera skrypt, który pozwala nam na wytrenowanie i ewentualne zapisanie nowego modelu jak i ten do odpalenia przygotowanego wcześniej modelu wraz z odpaleniem kamerki i przetestowaniem go w czasie rzeczywistym.

Techniki wykrywania twarzy oraz maski używane podczas działania programu opierają się o wykorzystanie konwolucyjnych sieci neuronowych (**CNN**). Samo wykrycie twarzy opiera się o gotowy model **MTCNN (Multi-task Cascaded Convolutional Neural Network)**, który jest gotowym modelem głębokiego uczenia wykorzystywanym do ów zagadnienia. Do wykrycia maski (jest to klasyfikacja binarna - with_mask ; without_mask) używam klasy **Sequential** z **API Keras'a**, która pozwala na stworzenie prostego liniowego modelu. Sama budowa modelu została stworzona w oparciu o książkę **"Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow 2nd Edition"** - str. 461 dla problemu dotyczącego **Fashion-MNIST**.

Dokładniejszy Opis Modelu & Trening

Jak wspomniane wcześniej, model opiera się o klasę **Sequential** z **API Keras'a**. Model posiada 3 warstwy konwolucyjne, zwiększając dla następnej ilość filtrów dwukrotnie. Rozmiar filtra = (7x7) tylko dla pierwszej warstwy, by skupić się na wykryciu większych, bardziej ogólnych wzorców. Dodatkowo, po każdej z warstw model jest poddawany poolingowi (MaxPooling). Po przejściu przez warstwy konwolucyjne, wektor zostaje spłaszczony i użyty jako wejście do klasycznych sieci neuronowych (2 warstwy (**NN**), by przy użyciu **funkcji sigmoidalnej (unipolarnej)** na końcu ostatecznie zwrócić prawdopodobieństwo wykrycia maski.

Do treningu modelu, została użyta technika **augmentacji danych**, by zapobiec **overfitting'owi** oraz lepiej przygotować model na "niedoskonałe dane".

Przygotowany, wytrenowany model został załączony do projektu, jednak możliwe jest również zmienienie ustawień, wytrenowanie oraz zapisanie nowego modelu.

Dla zbioru treningowego został użyty gotowy dataset z Kaggle:

<https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>

Funkcje Pomocnicze

Projekt zawiera funkcje pomocnicze służące m.in. ładowaniu zdjęć, ekstrakcji twarzy, zamiany ich na wymaganą dla modelu formę, zapewnieniu równowagi między częstotliwością występowania klas (with_mask, without_mask) oraz końcowemu przygotowaniu ich pod trening, wraz z ich wcześniejszą **normalizacją**.

Część funkcji - m.in. wczytywanie zdjęć w **BGR/RGB**, czy przechwytywanie obrazu z kamery - zostało zaimportowanych z **OpenCV**.