

Dokumentacja

Rozpoznawanie znaków z rejestracji samochodu

Julia Seweryn
Jakub Wójcik
Jakub Urbański
Mateusz Cichostępski

1. Opis projektu

Tematem realizowanego projektu jest rozpoznawanie znaków z tablicy rejestracyjnej ze zdjęcia samochodu.

Założeniem programu było odnalezienie tablicy rejestracyjnej na obrazie, wyodrębnienie poszczególnych znaków oraz rozpoznanie ich.

2. Instrukcja instalacji i konfiguracji

Do poprawnego uruchomienia programu wymagany jest Python3 oraz pip. Należy zainstalować wszystkie potrzebne biblioteki komendą

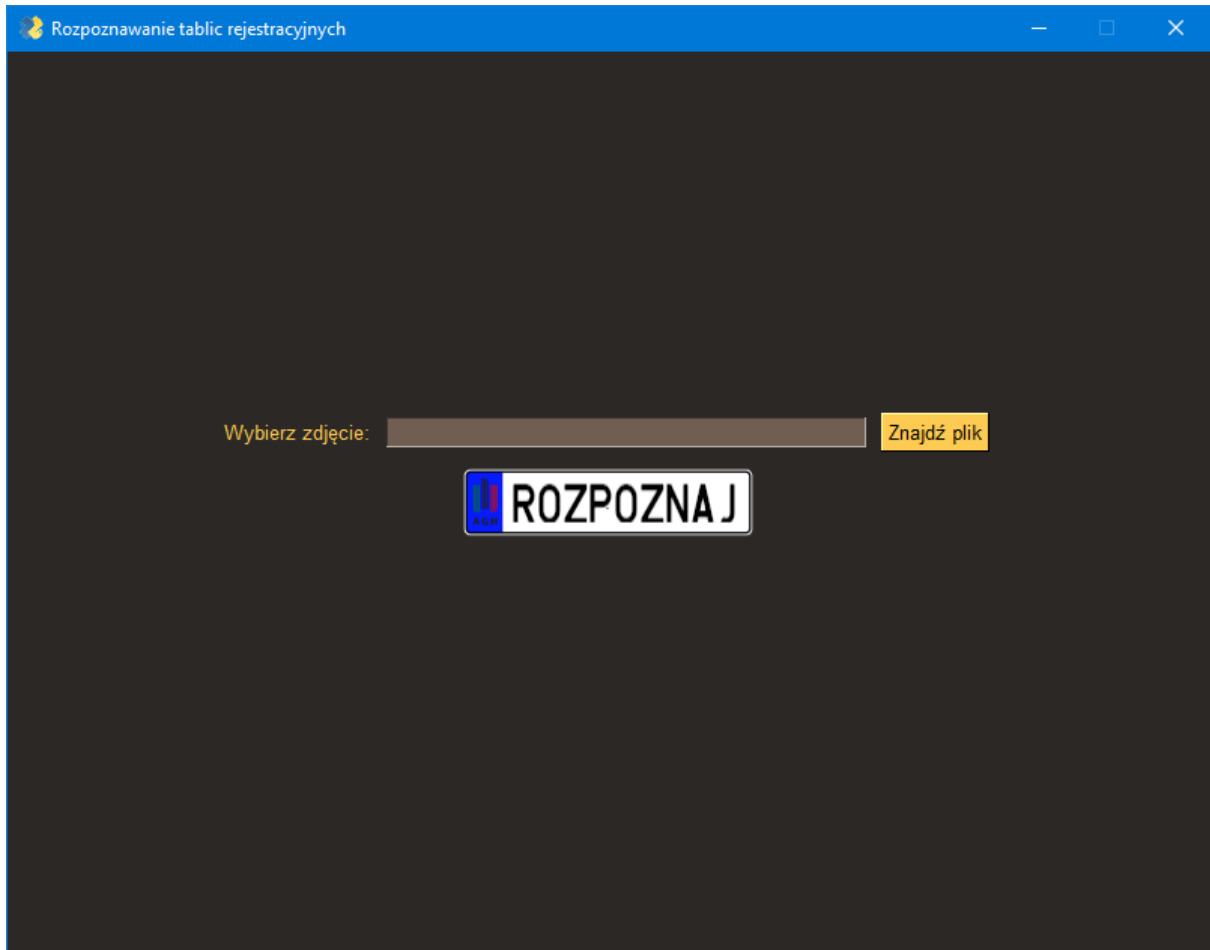
`pip install -r requirements.txt`

3. Instrukcja obsługi programu

Po wykonaniu operacji z punktu 2, można przejść do uruchomienia programu, co wykonano za pomocą komendy

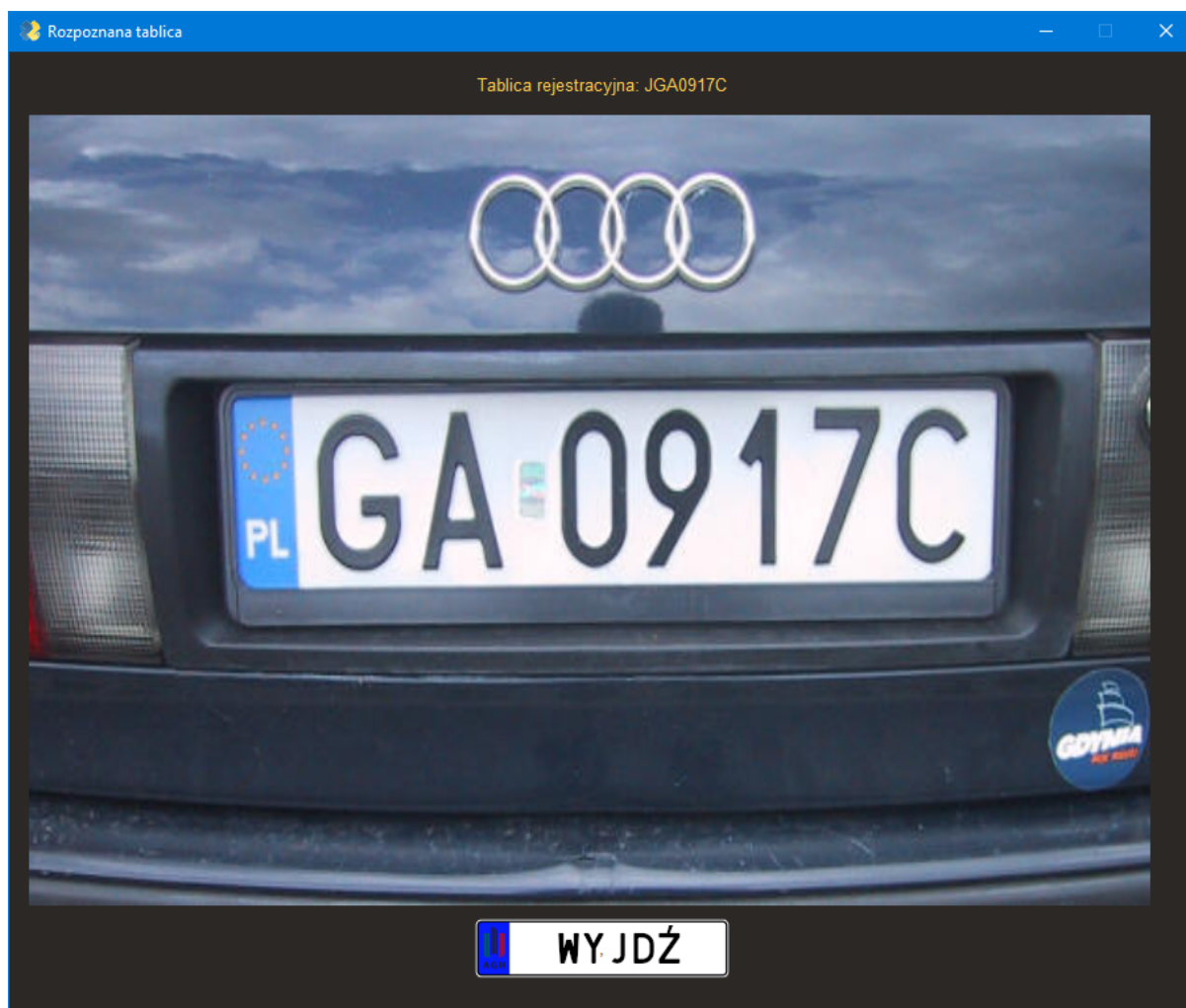
python3 main.py,

po wywołaniu której otrzymano poniższe okno:



Po kliknięciu przycisku **Znajdź plik**, z dowolnego katalogu można wybrać plik ze zdjęciem, który będzie przebadany po naciśnięciu **ROZPOZNAJ**. Po zakończeniu zadania otrzymujemy interfejs, na którym w górnej części obrazu zapisane są znaki z rejestracji, a poniżej wyświetlone jest badane zdjęcie.

Przykładowy wynik programu:



Przycisk **WYJDŹ** pozwala na zakończenie programu po wykonaniu badania.

4. Opis działania programu

Do stworzenia i wytrenowania sieci neuronowej z pomocą *tensorflow* skorzystano z dostępnego dataset'u *License Plate Digits Classification Dataset*. Posiada ona 35 folderów podzielonych na poszczególne znaki na tablicy rejestracyjnej. A każdy folder po 1030 przykładowych zdjęć z danym znakiem.

Funkcja ***trainModel()*** odpowiada za wytrenowanie sieci neuronowej. Dane z folderu *LicensePlatesCharacters* są wczytywane i dzielone na dane trenujące i walidujące. Wykorzystano gotowe funkcje do optymalnego i efektywnego stworzenia sieci neronowej. Nauczono model oraz zapisano go do *licensePlatesReader.model*.

Do stworzenia interfejsu graficznego użyto PySimpleGui. Zdjęcie pobierane jest przy użyciu funkcji `cv2.imread()`, która obsługuje formaty takie jak: .jpg, .bmp, .webp oraz .png.

Podstawowe 3 funkcje, które odpowiadają za działanie programu, to:

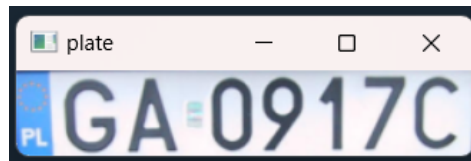
- *findPlate()*
- *findBoxes()*
- *readCharacters()*

- a. *findPlate* - funkcja odpowiada za wykonywanie przekształceń pozwalających na odnalezienie na obrazie konturów, a następnie wyznaczenie takiego konturu, który będzie miał 4 boki, oraz wycięcie go z obrazu.



Kolejne przekształcenia wewnątrz funkcji.

Obraz, zwracany przez funkcję *findPlate()*:

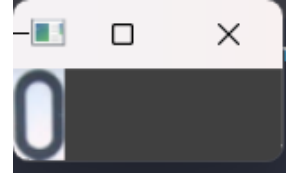
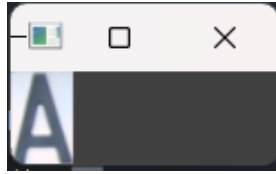
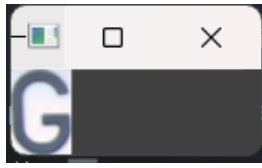


- b. ***findBoxes()*** - funkcja przeprowadza kolejne przekształcenia na obrazie zwróconym przez *findPlate()*, a następnie wydziela prostokątne obszary dookoła każdej z liter. Obszary zbyt małe, lub te o mocno odbiegających od normy proporcjach zostają odrzucone. Funkcja zwraca tablicę zawierającą informacje o wyznaczonych bounding boxach.



- c. ***readCharacters()*** - funkcja jako argumenty przyjmuje wyciętą wcześniej z obrazu tablicę, oraz listę wyznaczonych bounding boxów, które sortuje. Następnie dla każdego obszaru próbuje rozpoznać znajdujący się w jego wnętrzu znak i dopisuje go do tablicy, którą zwraca po zbadaniu wszystkich obszarów.

Trzecim argumentem funkcji jest boolean, który ustawiony jako true, uruchamia dodatkową opcję, jaką jest wyświetlanie kolejno sprawdzanych obszarów:



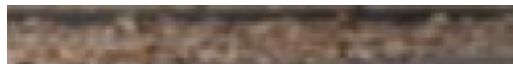
Po zakończeniu badania obrazu, na interfejsie pojawia się pierwotne zdjęcie samochodu, oraz nad nim, znaki sczytane z tablicy rejestracyjnej. (Kolejne znaki zapisane w tablicy zwróconej przez funkcję `readCharacters()`)

5. Co nie działa?

- a. Złe rozpoznanie obszaru z tablicą rejestracyjną na aucie



Rys. : Obraz z błędnie rozpoznanym obszarem z tablicą.



Rys. : Źle rozpoznany obszar tablicy rejestracyjnej.

- b. Złe rozpoznanie znaków na tablicy rejestracyjnej



Rys. : Obraz dla źle rozpoznanych znaków.

Dla powyższego przykładu algorytm rozpoznał sekwencje znaków: "HR26DX8997". Pomylił literę K z X oraz cyfrę 3 z 9.

- c. Odczytanie nadmiarowego znaku z np. flagi rejestracji



Rys. x: Obraz z odczytany nadmiarowym znakiem.



Rys. x: Obraz binarny wyciętej tablicy z nadmiarowym znakiem.

Program odczytał rejestrację "JGA0917C". Obszar ze symbolem *UE* oraz *PL* odczytał jako znak *J*.

6. Podział zadań

- Julia Seweryn
 - Dokumentacja
 - Wyznaczanie obszarów dookoła poszczególnych liter
- Jakub Wójcik
 - Dokumentacja
 - Wycinanie tablic
- Jakub Urbański
 - GUI
- Mateusz Cichostępski
 - Sieć neuronowa
 - Testowanie
 - Dokumentacja

7. Źródła

https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
<https://www.kaggle.com/datasets/aladdinss/license-plate-digits-classification-dataset>
<https://www.pythonpool.com/cv2-boundingrect/>
https://www.tensorflow.org/guide/keras/sequential_model
https://www.tensorflow.org/api_docs/python/tf/keras/models/load_model
<https://answers.opencv.org/questions/>
<https://www.pysimplegui.org/en/latest/>
<https://licenseplatemania.com/>