



AGH

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

Dokumentacja projektu
Analiza i Przetwarzanie Obrazów

Mateusz Barnacki
Agnieszka Lasek
Aleksandra Kuś
Piotr Matiaszewski
Piotr Jasiński
Rafał Jakubczyk

Kraków, 22.06.2022

1. Opis problemu

Celem projektu jest stworzenie programu, który na podstawie danych wejściowych w postaci nagrań wideo z kamery samochodowej będzie w stanie określić aktualną lokalizację pojazdu. Określenie położenia pojazdu nastąpi poprzez analizę elementów obrazu, m.in. tablice rejestracyjne, znaki drogowe, plakaty, analiza położenia na drodze pozostałych pojazdów uczestniczących w ruchu drogowym.

Inspiracją powstania projektu jest popularna gra Geoguessr (<https://www.geoguessr.com/>), w której użytkownik ma za zadanie zgadywać, w jakim rejonie świata znajduje się scena przedstawiona na obrazie.

Projekt powinien udostępniać podstawowy graficzny interfejs użytkownika.

2. Implementacja rozwiązania

Algorytmy wyszukiwania informacji o lokalizacji sceny przedstawionej na nagraniu, które zostały zaimplementowane w projekcie, dzielą się na trzy główne grupy.

a. Wykrywanie tablic rejestracyjnych

Biblioteka OpenCV pozwala na wykrywanie obiektów z użyciem tzw. kaskad Haar'a (ang. Haar Cascades). Jest to jeden z najbardziej efektywnych sposobów detekcji obiektów na obrazie, co zostało wykorzystane w tym projekcie. Jego działanie polega na wykrywaniu cech charakterystycznych, a przechodząc przez obraz i porównując fragmenty obrazu, próbuje dopasować następujące grupy cech:

- cechy krawędzi,
- cechy linii,
- cechy otoczenia centrum.

Klasyfikator Haara można uczyć samemu, dostarczając pliki uczące albo skorzystać z jednego z predefiniowanych modeli:

<https://github.com/opencv/opencv/tree/master/data/haarcascades>

W projekcie do wykrycia tablic rejestracyjnych został wykorzystany plik *haarcascade_russian_plate_number.xml*.

Po zainicjowaniu kaskady odpowiednim plikiem, w celu wykrycia obiektu skorzystano z funkcji *detectMultiScale*. Przyjmuje dwa parametry - obraz w skali szarości, na którym ma zostać znaleziony obiekt oraz parametr skali mówiący o tym, o jaki współczynnik zmniejszane są obrazy w kolejnych fazach pracy algorytmu.

Algorytm zwraca obszar na oryginalnym zdjęciu, w którym potencjalnie znajduje się wykryty obiekt. Obszar ten jest opisany przez cztery liczby - dwie opisujące położenie piksela będącego prawym lewym rogiem znalezionej obszar oraz dwie określające długość i szerokość obszaru.

Po wykryciu położenia tablic rejestracyjnych, tworzone są kopie obszarów zawierające tablice i przekazywane są do narzędzia do optycznego rozpoznawania znaków - OCR. W projekcie zostało wykorzystane EasyOCR, która jest dostępna na githubie: <https://github.com/JaidedAI/EasyOCR>.

Wywołanie funkcji *readtext* z tej biblioteki zwraca znaleziony ciąg znaków (ang. *characters*). Listę pożądaných znaków zawężono do cyfr oraz dużych liter, żeby poprawniej odczytywało numer tablicy rejestracyjnej.

Finalnie, mając ciąg znaków znajdujący się na potencjalnie wykrytej tablicy rejestracyjnej, wysyłane jest zapytanie do darmowej strony: <https://www.ofesauto.es/en/know-the-nationality-of-a-vehicle-through-its-plate-number/>. Następnie, przy pomocy biblioteki BeautifulSoup, odczytywany jest wynik - lista państwa, do których może należeć znaleziony numer rejestracyjny pojazdu wraz z prawdopodobieństwem.

b. Przewidywanie na podstawie wykrytego tekstu

EasyOCR w projekcie służy również do wykrywania tekstu na omijanych znakach drogowych czy też plakatach.

Na podstawie całego znalezionej napisu w danej klatce przewidywany jest język. Wykorzystany do tego został moduł Googletrans, który implementuje Google Translate API. Translator, tak jak w znanej aplikacji, proponuje język, w którym tekst został napisany.

Znaleziony tekst używany jest także do szukania informacji na temat dokładniejszej lokalizacji, czyli nazwy miasta czy ulicy. Służy do tego pakiet Geocoder wraz z dostawcą OpenStreetMap. Wykryte napisy są przesyłane do OSM API i na podstawie odpowiedzi zbierane są państwa, w których może leżeć poszukiwana miejscowość.

c. Wykrywanie, po której stronie jezdni porusza się samochód

Detekcja strony jezdni, po której porusza się pojazd została zaimplementowana w oparciu o wykryte na obrazie linie.

Założenie, o które oparto implementację jest następujące: w przypadku gdy samochód porusza się po prawej stronie jezdni, z powodu perspektywy kamery,

większość linii na każdej klatce powinna być skierowana w prawą stronę (przeciwnie w przypadku ruchu lewostronnego).

Do wykrywania linii na przekazanej klatce wykorzystano funkcję OpenCV HoughLines.

3. Wykorzystane źródła danych

Do przetestowania stworzonej aplikacji potrzebne były krótkie filmy nagrane za pomocą kamer samochodowych. Można wyróżnić 2 główne źródła danych:

- **Youtube**

Strona udostępnia nieograniczoną liczbę nagrań, które mogą zostać wykorzystane do przetestowania zaimplementowanych w ramach aplikacji algorytmów. Zdecydowaną zaletą tego serwisu jest możliwość doboru jakości w celu jak najlepszego wykorzystania zaimplementowanych algorytmów. Spośród wad należy zwrócić szczególną uwagę na potrzebę dodatkowej obróbki nagrań opartej głównie na wycięciu krótszych fragmentów filmu.

- **Driving Event Camera Dataset (Samsung DVS Gen3)**

Zbiór danych udostępniony przez Uniwersytet w Zurychu. Filmy zostały nagrane za pomocą kamery Samsung DVS Gen3 oraz kamery Huawei P20 Pro, które były przymocowane do przedniej szyby pojazdu. Nagrania prezentują przejazd samochodem po ulicach Zurychu. Niekwestionowaną zaletą jest długość dostępnych filmów, które trwają zazwyczaj nie dłużej niż 2min.

Do testowania projektu wybrano następujące filmy:

- <https://youtu.be/UzmCmQ1-Rlg?t=1327> Ateny, Grecja
- <https://youtu.be/Hx44jrWHaUE?t=950> Autostrada A3, Niemcy
- https://youtu.be/4_WhLB16w3g?t=525 Warszawa, Polska
- https://download.ifi.uzh.ch/rpg/web/data/E2VID/datasets/driving_gen3/external_videos/back8.mp4 Zurych, Szwajcaria

4. Uruchomienie programu

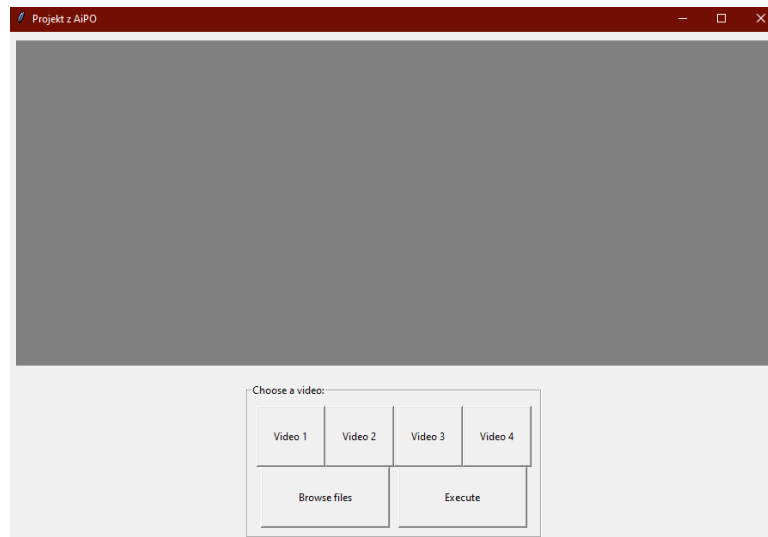
Projekt został napisany w języku Python, a do interfejsu użytkownika wykorzystano paczkę Tkinter. W celu uruchomienia programu należy zatem wykonać następującą komendę:

```
pip install -r requirements.txt
```

Następnie należy wykonać komendę:

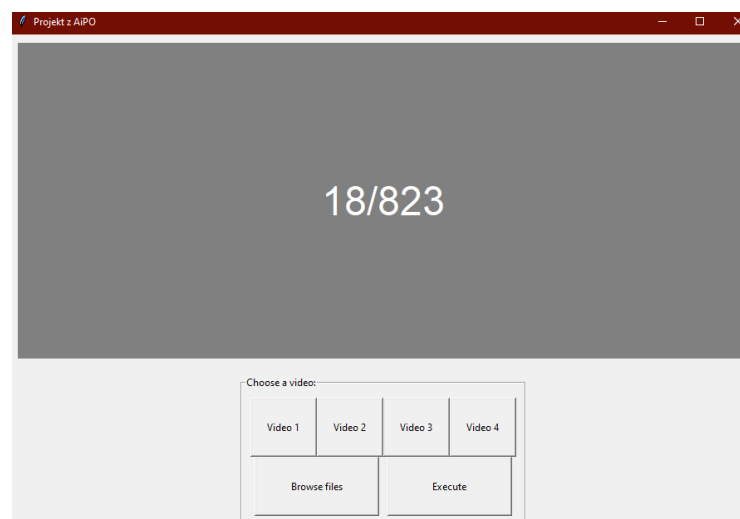
python main.py

Aplikację testowano w wersji Python 3.9.6, a dostęp do GPU znacznie przyspieszy jego działanie.



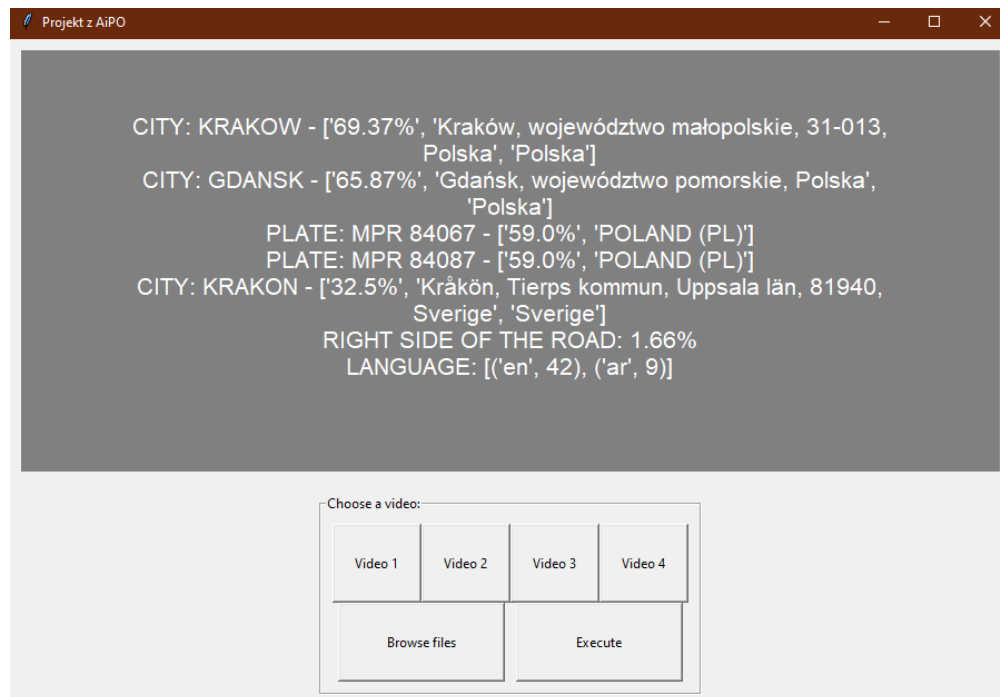
Zdjęcie 1. Podstawowy widok aplikacji

Użytkownik ma opcję wyboru 4 przykładowych filmów do przetestowania aplikacji. Może również wybrać własne video. Po załadowaniu filmu, należy kliknąć 'Execute', które rozpocznie analizę.



Zdjęcie 2. Aplikacja w trakcie analizy video

Podczas zbierania informacji z filmu, program pokazuje licznik z aktualnym postępem. Na końcu pokazywane są wyniki w formie tekstowej.



Zdjęcie 3. Przykładowy wynik

5. Wyniki

1. Wideo 1: Ateny, Grecja

- Analiza tablic rejestracyjnych

Nie wykryto tablic rejestracyjnych.

- Przewidywanie na podstawie wykrytego tekstu

Program przeanalizował wszelkie napisy znajdujące się na testowanej sekwencji wideo. Z tych najczęściej wykrywanych, znaleziono następujące prawdopodobnie występujące nazwy miast:

- Omonia, Cypr z prawdopodobieństwem 40%
- Solas, Wielka Brytania z prawdopodobieństwem 34%
- Madre de Dios, Peru z prawdopodobieństwem 46%

Na podstawie wykrytych napisów określono prawdopodobne języki w danej lokalizacji jako angielski i grecki.

- Wykrywanie po której stronie jezdni porusza się samochód

Aplikacja wyznaczyła prawdopodobieństwo ruchu prawostronnego na poziomie 25%. Niska wartość prawdopodobieństwa jest najprawdopodobniej skutkiem słabo widocznych linii na jezdni.

Podsumowanie zwróconych informacji:

- prawdopodobne miasta to Omonia, Solas lub Madre de Dios
- prawdopodobieństwo występowania ruchu prawostronnego to 25%
- języki w danej lokalizacji zostały sklasyfikowane jako angielski i grecki

2. Wideo 2: Autostrada A3, Niemcy

- Analiza tablic rejestracyjnych

Nie wykryto tablic rejestracyjnych.

- Przewidywanie na podstawie wykrytego tekstu

Program przeanalizował wszelkie napisy znajdujące się na testowanej sekwencji wideo. Ze znalezionych napisów, następujące zostały sklasyfikowane jako prawdopodobne nazwy miast:

- Bad Camberg, Niemcy z prawdopodobieństwem 62%
- Frankfurt, Niemcy z prawdopodobieństwem 79%
- Brechen, Niemcy z prawdopodobieństwem 47%

Na podstawie wykrytych napisów określono prawdopodobne języki w danej lokalizacji jako angielski i niemiecki.

- Wykrywanie po której stronie jezdni porusza się samochód

Aplikacja wyznaczyła prawdopodobieństwo ruchu prawostronnego na poziomie 33%. Na testowanym filmie nie widać na jezdni żadnych znaków wskazujących na ruch prawostronny.

Podsumowanie zwróconych informacji:

- Bad Camberg, Frankfurt, Brechen to trzy miasta o wysokim prawdopodobieństwie wystąpienia danej lokalizacji położone na terenie Niemiec
- prawdopodobieństwo występowania ruchu prawostronnego to 33%
- języki w danej lokalizacji zostały sklasyfikowane jako angielski i niemiecki

3. Wideo 3: Warszawa, Polska

- Analiza tablic rejestracyjnych

Program w 80% przypadków wykrył tablice rejestracyjne pozostałych pojazdów uczestniczących w ruchu drogowym jako tablice wydawane na terenie Polski. Prawdopodobieństwo przypisania tablic do Polski zostało oszacowane na 59% w trzech przypadkach oraz 100% w jednym przypadku.

- Przewidywanie na podstawie wykrytego tekstu

Program przeanalizował wszelkie napisy znajdujące się na testowanej sekwencji wideo. Dwa ze znalezionych napisów zostały określone jako prawdopodobne nazwy miast. Jednym z wyszukanych miast jest Kraków, czego prawdopodobieństwo określono na 69%. Drugim z wyszukanych miast jest Gdańsk z prawdopodobieństwem 66%. Na podstawie wykrytych napisów określono język w danej lokalizacji jako angielski i arabski.

- Wykrywanie po której stronie jezdni porusza się samochód

Aplikacja wyznaczyła prawdopodobieństwo ruchu prawostronnego na poziomie 1%. Na wykorzystanej sekwencji wideo praktycznie nie widać żadnych linii na jezdni. W związku z tym program nie może prawidłowo sklasyfikować rodzaju ruchu drogowego.

Podsumowanie zwróconych informacji:

- Kraków lub Gdańsk, miasta położone w Polsce
- kraj pochodzenia tablic rejestracyjnych mijanych pojazdów został sklasyfikowany w 80% jako Polska
- prawdopodobieństwo występowania ruchu prawostronnego to 1%
- języki w danej lokalizacji zostały sklasyfikowane jako angielski i arabski

4. Wideo 4: Zurych, Szwajcaria

- Analiza tablic rejestracyjnych

Program w 80% przypadków wykrył tablice rejestracyjne pozostałych pojazdów uczestniczących w ruchu drogowym jako tablice wydawane na terenie Szwajcarii. Prawdopodobieństwo przynależności do tego kraju dla każdej z tablic wyniosło 66.1%.

- Przewidywanie na podstawie wykrytego tekstu

Program przeanalizował wszelkie napisy znajdujące się na testowanej sekwencji wideo. Jeden ze znalezionych napisów został określony jako prawdopodobna nazwa miasta. Wyszukanym miastem jest Wipkingen znajdujące się w pobliżu Zurychu w Szwajcarii. Na podstawie analizy napisów określono prawdopodobne języki jako angielski oraz arabski.

- Wykrywanie po której stronie jezdni porusza się samochód

Aplikacja wyznaczyła prawdopodobieństwo ruchu prawostronnego na poziomie 15%. Najprawdopodobniej taki rezultat wynika ze słabej widoczności linii na jezdni. Ponadto na przebytym odcinku drogi jest tylko jeden znak na jezdni oznaczający skręt w prawo.

Podsumowanie zwróconych informacji:

- Wipkingen, miasto położone w Szwajcarii w okolicach Zurychu
- tablice rejestracyjne mijanych pojazdów zostały sklasyfikowane w 80% jako Szwajcaria
- ruch został określony jako prawostronny
- języki w danej lokalizacji zostały sklasyfikowane jako angielski i arabski

6. Wnioski

Projekt okazał się dużym wyzwaniem. Najbardziej kłopotliwe okazało się wykrywanie tekstu na obrazie, które jest podstawą części predykcji. Wykorzystanie darmowych API również nie zapewnia dobrej jakości rezultatów. Program jednak zazwyczaj prawidłowo określa kraj, w którym znajduje się analizowana lokalizacja oraz przynajmniej jedno z miast w danym państwie. Problematiczne okazało się również klasyfikowanie języków, które są używane w danej lokalizacji. Wykorzystane API za każdym razem na pierwszym miejscu zwraca język angielski. Rzeczywisty język wykorzystywany w danym miejscu jest zazwyczaj zwracany na drugim miejscu. Zapewne jest to spowodowane podstawowym językiem, który próbuje wykryć EasyOCR. Duże ograniczenia wykorzystywanych darmowych rozwiązań mają swoje konsekwencje w znalezionych lokalizacjach, jednak w każdym testowanym filmie przynajmniej jedna cecha była poprawna.

7. Bibliografia

7.1 https://rpg.ifi.uzh.ch/event_driving_datasets.html