



**Wydział Matematyki
i Nauk Informatycznych**

POLITECHNIKA WARSZAWSKA

Rozpoznawanie znaków drogowych

Warsztaty z Techniki Uczenia Maszynowego

Mateusz Chmurzyński

Mateusz Chudek

Marcin Skrzypczak

Jakub Stachyra

Jakub Winiarski

Wersja 2.0

26 kwietnia 2024

Spis treści

1	Historia zmian	2
2	Wstęp	2
3	Dane do uczenia	2
4	Wykorzystane technologie i języki	2
5	Podział pracy	2
6	Pierwszy sprint	3
7	Podsumowanie sprint 1	3
7.1	Przygotowanie danych i pierwsze modele	3
7.2	Potencjalne usprawnienia	3
8	Bibliografia	4

1 Historia zmian

Data	Autor	Opis	Wersja
2024-03-08	Wszyscy	Dokumentacja wstępna	1.0
2024-04-26	M. Skrzypczak	Sekcja 7	2.0
2024-04-26	M. Chmurzyński	Sekcja 7	2.0

Tabela 2: Historia zmian

2 Wstęp

Celem projektu jest stworzenie aplikacji służącej do rozpoznawania znaków drogowych na obrazach oraz nagraniach wideo (na przykład z kamer samochodowych). Taki system mógłby być podstawą do systemów rozpoznających znaki drogowe w samochodach autonomicznych, bądź wyświetlających informacje na wyświetlaczach head-up display. Planowane jest zaimplementowanie przetwarzania wideo w czasie rzeczywistym, a w celu zbadania przydatności takiego rozwiązania zostaną przeprowadzone odpowiednie eksperymenty, a na ich podstawie zostanie wykonana analiza danych.

3 Dane do uczenia

Przykładowymi źródłami danych, które mogą zostać wykorzystane w projekcie są zbiory przybliżonych zdjęć polskich znaków drogowych dostępnych na stronie <https://www.kaggle.com/datasets/chriskjm/polish-traffic-signs-dataset> oraz <https://www.kaggle.com/code/keneler/starter-polish-traffic-signs-dataset-4b78942b-3/input>. Inny zbiór danych zawierający zdjęcia zrobione z dalszej odległości, który może zostać użyty w projekcie znajduje się pod adresem <https://github.com/mikgor/DRIVER-ASSISTANT/tree/master/data>. Najbardziej problematyczną częścią będzie zaimplementowanie algorytmu rozpoznającego znaki w czasie rzeczywistym. Pierwszą próbą rozwiązania tego problemu będzie dostosowanie sieci rozpoznającej znaki na statycznych obrazach do wideo. Jeśli ten pomysł okaże się niemożliwy do zrealizowania, zostanie wybrane inne rozwiązanie.

Ciekawym pomysłem wydaje się wykorzystanie filmów wrzucanych przez internautów na kanale Stop Cham na portalu YouTube: <https://www.youtube.com/@STOPCHAM>. Dane zostaną wykorzystane jako zbiór testowy.

4 Wykorzystane technologie i języki

Do części rozpoznającej znaki drogowe planowane jest użycie algorytmu YOLOv8. W przypadku problemów z efektywnym działaniem algorytmu zostanie wykorzystany inny model uczenia maszynowego. Do implementacji backendu posłuży Python wraz z frameworkiem FastAPI. Natomiast do aplikacji mobilnej użyty zostanie framework Flutter.

5 Podział pracy

Przykładowy podział pracy przy tworzeniu projektu:

1. aplikacja mobilna - 1 osoba stworzy aplikację, wykorzystującą wytrenowaną sieć neuronową do rozpoznawania znaków drogowych w czasie rzeczywistym
2. trenowanie sieci neuronowej - 2 osoby przeanalizują dostępną dokumentację i podobne projekty oraz stworzą i wytrenują wspomniany model
3. dokumentacja, datasety, badanie modelu - 2 osoby udokumentuje architekturę systemu, opisz wyniki testów oraz zestawienie efektywności modelu w różnych warunkach

6 Pierwszy sprint

Poniżej przedstawiono plan pracy zespołu na pierwszy sprint.

Osoba	Zadanie
Mateusz Chmurzyński	Przygotowanie zbiorów danych. Etykietowanie oraz wgranie odpowiednich plików do Google Colab.
Mateusz Chudek	Stworzenie aplikacji mobilnej. Przygotowanie widoków do robienia i wgrywania zdjęć oraz przetwarzania wideo.
Marcin Skrzypczak	Przygotowanie zbiorów danych. Etykietowanie oraz wgranie odpowiednich plików do Google Colab.
Jakub Stachyra	Wytrenowanie pierwszego modelu sieci neuronowej rozpoznającego znaki.
Jakub Winiarski	Wytrenowanie pierwszego modelu sieci neuronowej rozpoznającego znaki.

Tabela 3: Plan pracy na pierwszy sprint.

Jako, że powyższy projekt jest jednocześnie zadaniem zaliczeniowym przedmiotu na studiach, w celu jak największego poszerzenia wiedzy, uczestnicy projektu będą wymieniali się zadaniami, tak aby każdy spróbował czegoś nowego.

7 Podsumowanie sprint 1

7.1 Przygotowanie danych i pierwsze modele

Przed rozpoczęciem trenowania modelu musieliśmy lekko przeorganizować dostępne dane. Znaki drogowe były rozmieszczone w folderach z nazwą danego znaku a ponieważ do algorytmu YOLO potrzebne są w jednym folderze zdjęcia a w drugim etykiety dla prostokątów należało napisać odpowiedni skrypt który przeorganizuje dane i wygeneruje etykiety. Po tym wytrenowaliśmy pierwsze modele i sprawdziliśmy jak sobie radzą ze zdjęciami znaków drogowych oraz w czasie rzeczywistym z włączoną kamerką.

7.2 Potencjalne usprawnienia

1. Rozdzielenie detekcji oraz rozpoznawania - z powodu braku przystosowania danych treningowych do problemu rozpoznawania, model może mieć problem spowodowany dużą liczbą znanych znaków. Zbadamy możliwość rozdzielenia zadania pomiędzy dwa mniejsze modele - pierwszy wyspecjalizowany w detekcji znaków drogowych, drugi zajmujący się rozpoznawaniem konkretnych znaków na podstawie fragmentów wskazanych przez poprzedni model.
2. Preprocessing danych - zbadamy jak standaryzacja danych wpłynie na skuteczność modelu np. skalowanie do tego samego rozmiaru

3. Wykorzystanie GoogleColab lub AzureML - umożliwi nam to długotrwałe trenowanie modelu

8 Bibliografia

- [1] Ultralytics Inc. Ultralytics YOLOv8 docs. <https://docs.flutter.dev/>. Dostęp: 2024-03-08.
- [2] Google, Inc. Flutter docs. <https://docs.flutter.dev/>. Dostęp: 2024-03-08.
- [3] Sebastián Ramírez. FastAPI docs. <https://fastapi.tiangolo.com/>. Dostęp: 2024-03-08.