

# TrafficSignRecognition

---

## 1 テーマ

交通標識の意味を教えてくれるCNNモデル

## 2 画像認識システムの概要

### a. ユーザー & 利用シーン

ユーザー：日本の交通ルールを詳しく知らない外国人

利用シーン：交通標識の意味をリアルタイムで教えてくれる

なぜ：自分は今留学生寮に住んでいて、外国人と接する機会が多い。そこで、外国人とが日本にいて不便なことを自分の力で少しでも解決できればと思っていた。そこで留学生の友達がレンタカーを借りれるけど、日本の交通ルールについて知らないと言われ、そのことについて調べてみるとコロナ前は外国人のレンタカーの利用者数が増加して、また外国人のドライバーによる事故が増えていた。そこで、交通標識の意味を運転中に教えてくれるものがあれば便利だと思った。最終的にはウェアブルのメガネ型デバイスに対応させたり、車にカメラを搭載して、AR技術でドライバーが見ることができれば面白いと思う

### b. 課題解決イメージ

目標としてはカメラの前に標識の画像をおくとその画像の意味を教えてくれる。

## 3 システムの詳細

### a. システムの全体像

1. Training用のデータセットを用いて機械学習モデルに学習をさせる
2. 画像認識のモデルをh5オブジェクトとして保存する
3. 動画を静止画の集まりとして、そのそれぞれの静止画についてh5モデルの学習済のデータと比較してどの意味かを考える
4. 比較させた時に正しい確率が95%以上だった場合、出力させる

### b. 実現したい機能

実現したい機能のマイルストーンを以下のようにおく

1. 画像を入力したらどの画像か分類する
2. 動画からリアルタイムで標識を分類する
3. 速度などの数字がついているものについてはその数字がどの数字かも判断できるようにして、どのような数字が出てきても対応できるようにする

### c. 想定される技術的課題

1. テストデータの画像の質がよくないため、より多くの画像数が必要になる。
2. 映像を読み込む際にFPSが高すぎると認識するのにかかる時間と次のフレームを表示させる時間で誤差が生じる

## 4 開発イメージ

### a. 入力画像、ラベル数

今回入力画像として使用するのは International Joint Conference on Neural Networks 2011 に開かれた The German Traffic Sign Benchmarkのデータセットを利用した。理由としては42ラベル、写真は5万枚以上に上り、入力画像の枚数、ラベル数を幅広いレンジで変更できるため、難易度を段階的にあげていくことができ、最適なモデルを作るのには適していると考えた。

The German Traffic Sign Benchmark -> [https://benchmark.ini.rub.de/gtsrb\\_news.html](https://benchmark.ini.rub.de/gtsrb_news.html)

## b.使用する技術

Python3.9.5

Keras version2.4.3

Tensorflow version2.5.0

OpenCV version4.5.36

## c.ファイル階層構造

```

IMAGE_RECONGNITION
├── README.md
├── codes
│   ├── cnn_model.ipynb    //ソースコード格納場所
│   ├── create_datasets.py //使用するAIモデル
│   └── video.py           //使用する画像を決める
├── image_datasets
│   ├── Meta               //webcamに写っている被写体が何かを判別する
│   ├── Meta.csv           //写真データ格納場所
│   ├── Test               //メタデータ
│   ├── Test_use           //Meta含まれるデータの画像の情報が書かれている
│   ├── Test.csv           //テストデータ
│   ├── Train              //使用するテストデータ
│   ├── Train_use          //Test含まれるデータの画像の情報が書かれている
│   └── Train.csv          //学習用データ
├── model                  //使用する学習用データ
│   ├── label.txt          //Trainに含まれるデータのそれぞれの画像の情報が書か
│   ├── my_model.h5        //写真データ格納場所
│   ├── Test               //数値のラベルが書かれている
│   ├── Test_use           //学習済みデータ
│   └── Test.csv           //テストデータ
└── keys.json              //使用するテストデータ
                           //Test含まれるデータの画像の情報が書かれている
                           //パラメータが書かれている

```