

テーマ番号	1EP001				
プロジェクト テーマ	和文	機械学習を用いた電車の車両タイプ判別システムの開発		指導教員	鷹合 大輔 准教授
	英文	Development of a Train Type Identification System Using Machine Learning			
プロジェクト メンバー	4EP1-68 野崎 悠渡 (NOZAKI Yuto) 4EP4-75 田村 優祐 (TAMURA Yusuke)				

Abstract There are nearly 100 types of train cars for JR's conventional lines alone. There are nearly 100 types of train cars on JR's conventional lines alone. It is difficult for those who do not have much knowledge of trains to distinguish similar train car types. In this project This project aims to create a system to identify train car types using YOLO. The classification and discrimination models were created using the same image dataset, and the discrimination results were similar. A web application was also created using the created model. Created a web application and server using nodejs to output discrimination results .

Keywords YOLO, Machine Learning, node.js, python-shell, python

1. はじめに

電車の車両タイプは JR の在来線だけでも 100 種類近く存在している。電車の知識がある人は一目見るだけでその電車の車両タイプを判別できるが、電車の知識があまりない人は似ている電車の車両タイプを判別することが難しい。本プロジェクトでは簡単に画像や動画に写っている車両タイプが何なのかを分類または識別できるシステムを開発する。

2. 画像の分類と識別について

画像の分類とは、画像が特定のカテゴリやクラスに属するかどうかを判別する作業である。例えば、画像に写っているのが猫か犬かのクラスに分けることである。

識別とは画像のどこに何が写っているのかを判別するプロセスを指す。1 枚の画像に複数の物体が存在する場合も識別はできる。動画に写っている電車の車両タイプも判別することができる。

3. システム概要

本システムは、ユーザに画像または動画をブラウザ上で入力してもらい、それをサーバ上で画像認識を用いて処理し、結果をブラウザで表示する Web アプリケーションである。Web サイトのテンプレートには w3.school [1] のものを使用した。システム概要を図 1 に示す。

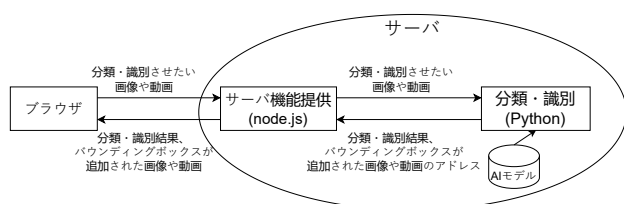


図 1 システム概要
Fig.1 System Overview

4. システムの機能

本システムは、以下の 3 つの機能を提供する。

- 電車の画像の分類
- 電車の画像の識別
- 電車の動画の識別

4.1 電車の画像の分類

出力後の画面を図 2 に示す。この機能では、ユーザがブラウザからアップロードした画像に含まれる電車の種類を分類する。分類の結果、最も可能性が高いものを Web ページに出力する。

4.2 電車の画像の識別

出力後の画面を図 3 に示す。この機能では、ユーザがブラウザからアップロードした画像に含まれる電車の位置と種類を識別する。識別の結果、バウンディングボックスが追加された画像を Web ページに表示する。

4.3 電車の動画の識別

出力後の画面を図 4 に示す。この機能では、ユーザがブラウザからアップロードした動画に含まれる電車の位置と種類を識別する。識別の結果、バウンディングボックスが追加された動画を Web ページに表示する。



図 2 画像の分類 Fig.2 Image Classification
図 3 画像の識別 Fig.3 Image Identification
図 4 動画の識別 Fig.4 Video Identification

5. 動作確認と現状

学内ネットワークから Web アプリケーションに接続すると画像の分類、画像の識別、動画の識別の動作を確認できた。しかしスマホで 4G 回線に接続し、学内の VPN を用いた際の結果が返される速度と学内ネットワークを用いた際の結果が返される時間に大きな差が出てしまう。今後、メモリ使用量や、CPU 使用量、ネットワーク使用量について調査する。

6. 判別モデルの開発の流れ

電車が写っている画像を集めて、データセットを作成し、学習をするという流れで判別モデルを開発する。

6.1 データ収集

YouTube で特定の電車のみが映っている 1～3 種類の動画を保存し、指定した枚数分のランダムなフレームを保存する。保存した画像を識別し電車が映っている画像だけを保存してトレーニングデータとバリデーションデータを収集した。テストデータは様々なウェブサイトから手作業で 17 種類の各車両の画像を 10 枚ずつ収集した。各車両タイプの画像の保存枚数を図 7 に示す。

6.2 データセットの作成

本プロジェクトで作成するモデルは識別モデル、分類モデルの二種類である。分類モデルのデータセット内の画像をアノテーションして、識別モデル用のデータセットを作成した。

6.3 モデルの学習

作成したデータセットと YOLOv8 [2] を用いてモデルの学習を行い、分類モデルと識別モデルを作成した。分類モデルでは静止画での判別しかできない。動画から車両タイプを判別するために識別モデルを作成した。

7. 作成したモデルの評価と考察

7.1 分類モデルの評価

作成した分類モデルを用いてテストデータセットの分類を行って得られた混同行列を図 6 に示す。縦軸は正解の車両タイプを表し、横軸は作成したモデルが予想した車両タイプを表す。

7.2 識別モデルの評価

作成した識別モデルを用いてテストデータセットの識別を行って得られた混同行列を図 7 に示す。識別時には、どの車両タイプにも当てはまらないと識別されることがある。その場合の車両タイプは 0 として識別モデルの混同行列を作成した。

7.3 考察

正解率は電車によって異なることがわかる。287 系と 683 系のように外見が似ている電車だと、誤判別していることが多かった。287 系を図 8、683 系を図 9 に示す。データセットの画像の枚数が少ない車両タイプの判別結果が必ず悪くはならなかった。画像の枚数が判別結果に影響を与えるのではなく、車体の特徴が鮮明に写っている画像の枚数が判別結果に影響を与えられられる。判別精度の向上のために、データセットとして質の悪い画像を大量に集めるのではなく、車体の特徴が鮮明に写っている画像を車両タイプごとに集める必要があったと考えられる。

同じ画像を使って、2 種類のデータセットを作成した。分類結果の図と識別結果の図が似ているため、車両タイプを知るためのシステムを作る際には、識別モデルのみを作成することで画像と動画の両方に対応した車両判別システムを作成でき、電車の車両タイプを知ることができるように考える。

8. まとめ

本プロジェクトでは、機械学習を用いた電車の車両タイプを判別するため、2 種類のモデルを作成した。また、電車が写っている画像や動画をサーバ上で、分類、識別を行い判別結果を出力する Web アプリを作成した。

本システムを利用することで電車の知識がない人でも画像または動画に写る一部の車両タイプが何なのかを知ることができる。4G 回線と学内の VPN を利用して判別した場合と、学内ネットワークを利用して判別した場合は、結果が返ってくる時間に大きな差があった。今後、メモリ使用量や CPU 使用量、ネットワーク使用量について調査をする。

参考文献

- [1] "W3.CSS Templates", https://www.w3schools.com/w3css/w3css_templates.asp
- [2] "Ultralytics YOLOv8 ドキュメント", <https://docs.ultralytics.com/>

cs.ultralytics.com/ja

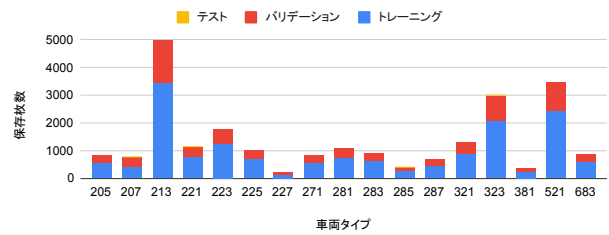


図 5 各車両タイプの保存枚数
Fig.5 Number of images stored for each type

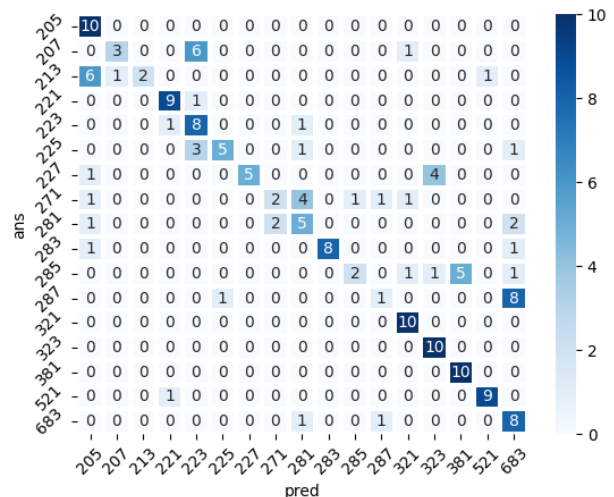


図 6 分類モデルの混同行列
Fig.6 Confusion matrix of classification models

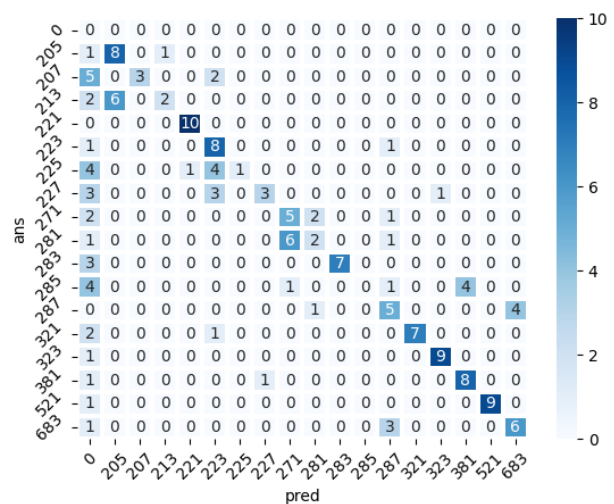


図 7 識別モデルの混同行列
Fig.7 Confusion matrix of identification models



図 8 287 系
Fig.8 287 series



図 9 683 系
Fig.9 683 series