

テーマ番号	1EP001				
プロジェクト テーマ	和文	機械学習用いた電車の車両タイプの判別システムの開発		指導教員	鷹合 大輔 准教授
	英文	A Construction Method of Optimum Integer-to-integer Transform based on an Error Propagation Model			
プロジェクト メンバー	4EP1 68 野崎 悠渡 (NOZAKI Yuto)    4EP5-11 松永 久秀 (MATSUNAGA Hisahide) 4EP5-29 筒井 順慶 (TSUTSUI Junkei)    4EP5-100 百地 丹波 (MOMOCHI Tanba)				

**Abstract** Describe about 5 lines of abstract in English here. Describe about 5 lines of abstract in English here. Describe about 5 lines of abstract in English here. Describe about 5 lines of abstract in English here. Describe about 5 lines of abstract in English here. (何が問題で、それをどんな手法で取り組んで、どういう結果であったかなどを英語で要約して下さい) Describe about 5 lines of abstract in English here. Describe about 5 lines of abstract in English here. Describe about 5 lines of abstract in English here.

**Keywords** Qwerty1, Qwerty2, Qwerty3, Qwerty4, Qwerty5

## 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。多くの人は電車を見て電車だと認識することは可能だが、その電車の車両タイプまでを判断できる人は少ない。電車について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断できるが、大多数の人は似ている電車の車両タイプを判断することが難しい。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が何なのかを判別できるシステムを開発する。

## 2. システム概要

開発するシステムの概要を図 1 に示す。

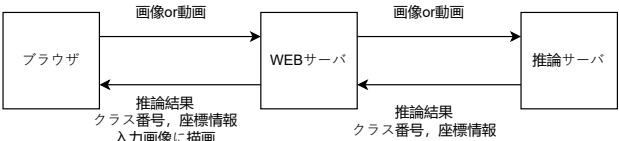


図 1 後で変える  
Fig.1 aaaaa

## 3. 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。

### 3.1 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て、指定した枚数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て、電車が映っているものだけ保存する。動画ごとに電車が写っている時間が異なるため、集めた画像は車両タイプごとに異なる。保存した画像の枚数を図 2 に示す。

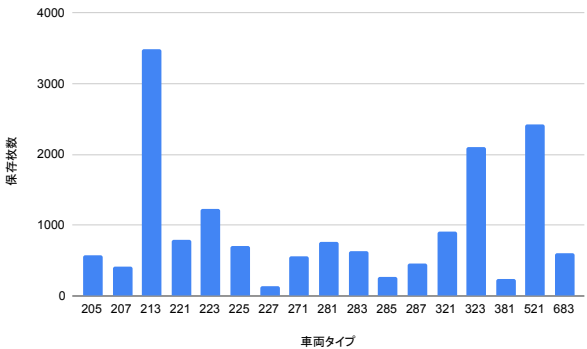


図 2 各車両タイプの保存枚数  
Fig.4 hozonnmisuuuade

### 3.2 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。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 枚ずつ集めて、テストデータセットも作成した。

#### 3.2.1 分類とは

分類はデータやオブジェクトを異なるクラスやカテゴリに分けるプロセスを指す。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、車が飛行機などのクラスに分類することがある。

#### 3.2.2 識別とは

識別とは画像のどこに何が写っているのかを判断するプロセスを指す。1 枚の画像に複数の物体が写っていても識別はできる。

#### 3.2.3 アノテーションとは

アノテーションとは、機械学習の分類の一つである教師あり学習において、分析対象データにラベルを付与するプロセスである。画像にバウンディングボックスと呼ばれる四角形を描画しクラス番号を指定する。バウンディングボックスを描画することでその画像に写っている物体の座標情報を取得することができる。クラス番号とは、判別したいもののリストを作成し、画像に写っている物体に対応した、リストのインデックスのことである。アノテーションをした結果は、識別モデルの学習時に使用する。

### 3.3 モデルの学習

YOLOv8 を用いてモデルを作成した。

#### 3.3.1 YOLOv8 の概要

YOLO とは You Only Look Once の略で、人間のようにより一瞥だけで物体検出ができることを指している。データセットを作成し学習させることで、任意の物体のみ検出させることが可能である。YOLOv8 は YOLO シリーズの最新バージョンであり、ディープラーニングとコンピュータビジョンの最先端の進歩に基づいており、速度と精度の両方で比類のない性能を提供している。

#### 3.3.2 学習の実行

学習を進めると徐々に性能が上がっていき性能が向上しなくなると学習が途中で中断される。学習は中断されるまで続けたため、学習回数はモデルによって差がある。

### 3.4 モデルの使い方

識別モデルと分類モデルをどのように使えば結果が出力されるのかを説明する。python のコードを貼り付ける？ 田村のパートに記載してもいいかもしれない

## 4. 作成したモデルの評価

### 4.1 分類モデルの性能評価

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た結果を図 3 に示す。

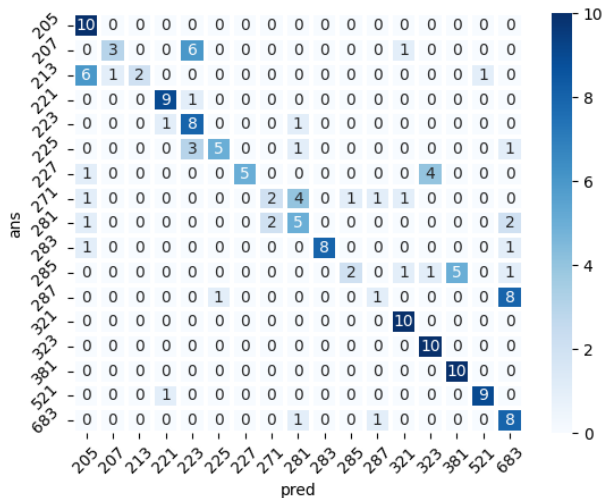


図 3 性能評価  
Fig.3 seinouhyouka

### 4.2 識別モデルの性能評価

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た結果を図 4 に示す。識別失敗とは間違った車両タイプだと判断していることを指し、識別不能とはどの車両タイプにも当てはまらなと判断していることを指す。

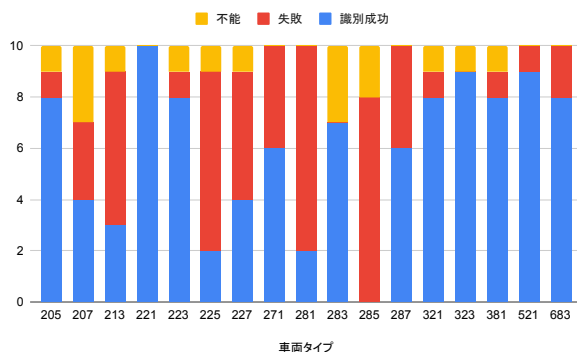


図 4 テストデータ識別結果  
Fig.4 ababefn

## 5. 考察

正解率は電車によって異なることがわかる。特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。

各車両タイプの画像の枚数に差があったことが判別結果に影響を与えていると考えられる。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ていると考えられる。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。

\_\_\_\_\_ここまで野崎\_\_\_\_\_

\_\_\_\_\_ここから田村\_\_\_\_\_

## 6. システムについて

### 6.1 システムの動作要件

本システムでは、ノートパソコンに Linux サーバのセットアップを行い、サーバとして使用した。ネットワークは、学内の固定 IP を割り当ててもらい、学内ネットワークを使用した。サーバのスペックは以下の通りである。

#### 6.1.1 サーバのスペック

- OS: Ubuntu 22.04.3 LTS
- メモリ: 4GB
- CPU: Intel® Core™ i7
- GPU: NVIDIA GeForce GTX 470M
- HDD: 80GB

### 6.2 システムの開発環境

本システムの開発には、以下の言語やツールを使用した。

#### 6.2.1 使用言語

- フロントエンド:
  - html
  - JavaScript
  - css
- バックエンド:
  - JavaScript (Node.js)
  - Python

#### 6.2.2 使用ツール

- エディタ: VSCode
- ブラウザ: Google Chrome 119

### 6.3 システムの機能詳細

本システムは、以下の 3 つの機能を提供する。

#### 6.3.1 電車の画像の分類

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 HTML ページに出力する。

#### 6.3.2 電車の画像の識別

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 HTML ページに表示する。

#### 6.3.3 電車の動画の識別

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 HTML ページに表示する。

## 7. システムの速度検証

画像・動画の解像度、各機能ごとの応答速度（解像度の調整が困難なためスマホのサイズでの検証にはしない）（GPU で動作した際には GPU 固定）GPU マシンが

動作した場合、解像度一定で、各機能の CPU,GPU 速度を検証

### 7.1 システムの考察 (GPU 動作時) (仮)

GPU を使用したときと CPU を使用したときでは動作の速度が大きく異なることが分かる。また、入力された画像や動画の解像度によっても速度が異なることが分かった。

### 7.2 システムの考察 (GPU 非動作時) (仮)

入力された画像や動画の解像度によっても速度が異なることが分かった。

## 8. システムの考察

-----ここまで田村-----

## 9. まとめ

### 参考文献

- [1] "Ultralytics YOLOv8 ドキュメント",<https://docs.ultralytics.com/ja>
- [2] "アノテーションとは - 定義と重要性, 必要な準備や注意点を解説",<https://www.dir.co.jp/world/entry/solution/annotation>