令和５年度 プロジェクトデザインIII

機械学習を用いた電車の車両タイプの判別システムの開発

4EP1-68
のざきゆうと
野崎悠渡

4EP5-15
たむらのぶなが
田村信長

令和5年9月22日

KIT 金沢工業大学

1. はじめに – 背景と目的 –
建前編

- 現在，何が問題か（あるいは将来，何が問題になるか）を書く．
  世の中には似たようなものがたくさん存在している（動物や車，植物など）
  詳しく知ろうとしたときに，今見ているものが何なのか判別するまでに大きな労力が必要とされている．
- その問題に対処するためには，どのようなものがあればよいか（あるいは取り組みが必要）かを書く．
  知りたいと思っているモノの写真から，それが何なのか判別できるシステムがあればこれまでよりも簡単に知ることができる．

- 本プロジェクトでは何を使ってどんなものを作っているかを書く．
  本プロジェクトではYOLOv8を用いて，モノの識別をするシステムの開発を行う

1. はじめに–背景と目的–
本音編

- 走っている電車が何なのか簡単に知りたい.
  紛らわしいのが多すぎる.
  種類が多すぎる.
  図鑑とにらめっこするのがダルいという問題がある.
- この問題を解決するためには, 画像から分類や識別ができるシステムがあれば良い.
- YOLOを使って作っちゃうぞ

発表の流れ

1. はじめに – 背景と目的 –
2. システム概要
3. 評価
4. むすび
5. `http://www.fujitsu.co.jp`

2. システム概要

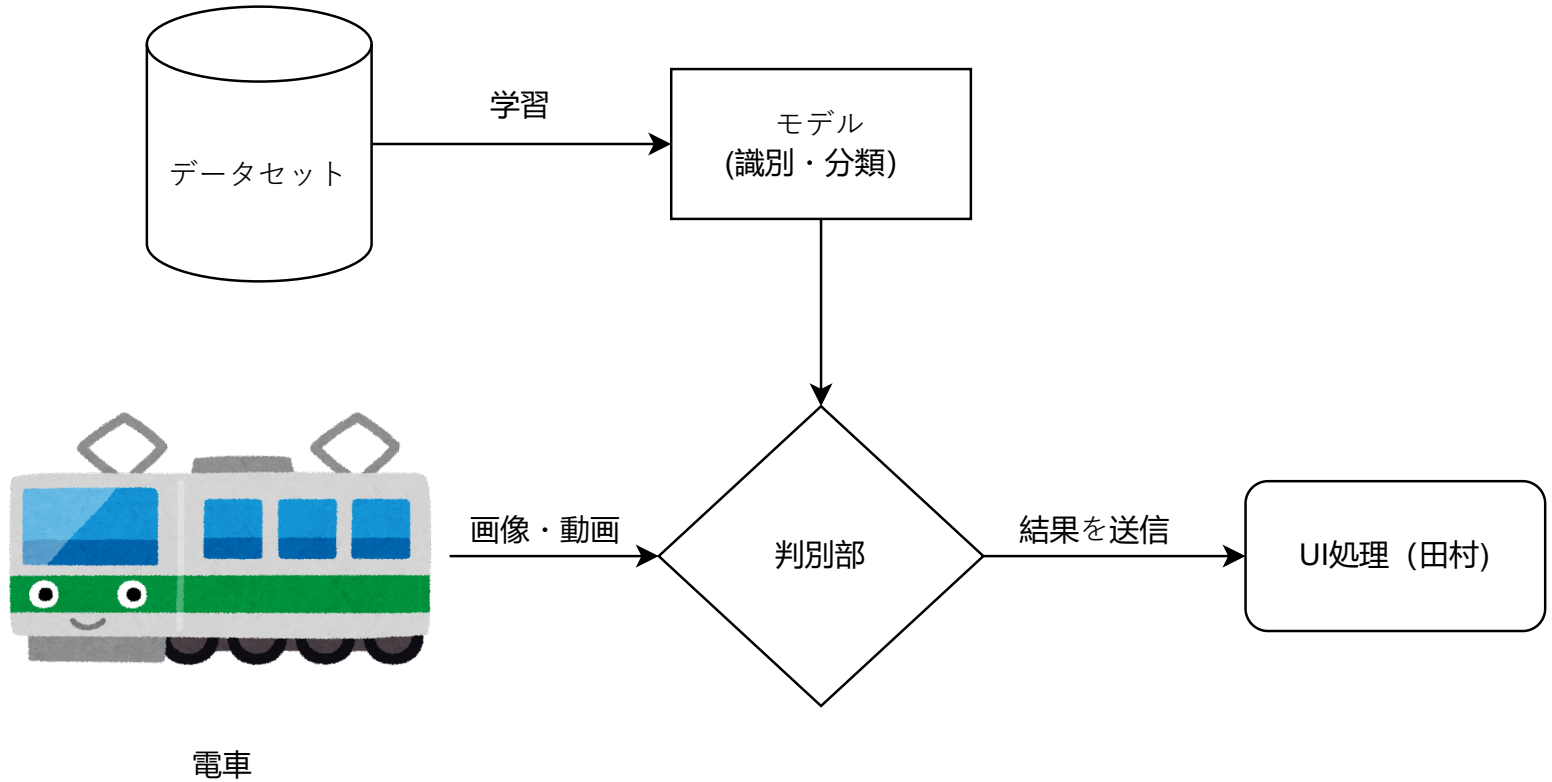

図 1: システム概要図

YOLOとは

YOLOはリアルタイムオブジェクト検出アルゴリズムである．

You Only Look Onceの略で，人間のようにひと目見ただけで物体検出ができる．

YOLOv8では，物体の識別と分類，セグメンテーションができる．

図 2: 識別

図 3: 分類

図 4: セグメンテーション

2. 現時点までで行ったこと

画像収集の簡略化

YoutubeのURLと車両タイプを指定することで，YouTubeの動画から車両が写っている部分を切り抜く.

集めるj画像の工夫

バウンディングボックスの中身のみ切り抜き，データセットに追加した

モデルの学習

収集した画像からデータセットを作成して，モデルの学習を行った.

- 電車と新幹線を識別するモデル(YOLOv5)
- ３種類の似ている電車を識別するモデル(YOLOv5)

- 電車や新幹線の各車両を分類するモデル(YOLOv8)

3. 評価・考察

- このスライドでは何をどのような方法で評価したかを明記し，結果をグラフで示すこと（表よりグラフのほうが良い）.
- システムが動いている様子がわかるようにデモ映像を流すこと（デモ映像には字幕をつけたりするなどしてわかりやすくすること）.
- 評価の際は，改良の前後でどうなったかを示す．あるいは他の手法などと比較してどうなのかを示すことも必要.
- 結果について考察も示すこと.

3. 評価・考察に書く内容

- アノテーションをする際に，バウンディングボックスの中身だけにすると不具合がおきたこと
  ボックスの中身だけで学習させる際に何故かエポック数が１００回になっていた．うまくいかないと思っていたがシンプルに学習回数が足りなかった？
  背景画像の有無で物体認識の精度が悪くなったので，データセットの画像には背景画像が必要なのではないのかな
- 電車と新幹線の識別モデルの作成→電車は百種類近くあるのでうまく行かなかった
- 似ている電車を識別モデルを作成する際に，データセットの画像を増やす

だけではうまくいかなかったこと

- 新幹線の分類はまぁまぁ上手くいったこと

4. むすび

- 何のために何を作成したかを改めて書く.
- 現時点での評価結果，考察を簡潔に書く.
- 来月の報告までに何をするか計画を書く.

ここからおまけ

- PDFファイルと同じフォルダにdemo002.mp4があれば再生できる.
- YOUTUBEで再生
- https://youtu.be/74agBeJxdFI

リスト 1: test2.c

```
1  #include <avr/io.h>
2  #include <avr/wdt.h>
3  int main(void)
4  {
5      DDRC = 0x30; // PC5/4を出力ピンに設定
6      PORTC = 0x10; // PC5/4の出力をL/Hに設定
7      for (;;) {
8          wdt_reset(); // ウォッチドックタイマをリセット
9      }
```

```
10      return 0;
11  }
```

リスト 2: test2.py

```
1  from time import sleep
2  from random import randint
3
4  while True:
5      input('push ENTER key')
6      r = randint(1,6)
7      print( r )
8      sleep(0.5)
```

UNIXv1 におけるタスク切り替えが行われるタイミング

① みなさん

② こんにちは

- まんじゅう
- りんご

③ お元気で
またあうひまで

```
$ gcc test.c ↵
 (*_*)
 (*_*)
```

ここでCTL+Cを押す

謝辞 本研究はJSPS科研費21Kxxxxxxxxx助成を受けた

文献

[1] K.Thompson，D.M.Ritchie，**"The UNIX Time-Sharing System"**，Communications of the ACM, Vol.17, No.7, 1974.

[2] Digital Equipment Corporation: **PDP11/20-15-r20 Processor Handbook**, 1971.

[3] T.R. Bashkow, **"Study of UNIX: Preliminary Release of Unix Implementation Document"**, `http://minnie.tuhs.org/Archive/Distributions/Research/Dennis_v1/PreliminaryUnixImplementationDocument_Jun72.pdf`, Jun. 1972.

[4] simh, **"The Computer History Simulation Project"**, `https://github.com/simh/simh`, 参照Mar.14, 2022.

[5] W.Toomey, **"First Edition Unix: Its Creation and Restoration"**, IEEE Annals of the History of Computing, 32 (3), pp.74-82, 2010.

[6] Diomidis.Spinellis，**"unix-history-repo"**, https://github.com/dspinellis/unix-history-repo/tree/Research-V1, 参照Mar.14, 2022.

[7] Digital Equipment Copporation: **PDP11 Peripherals HandBook**, 1972.