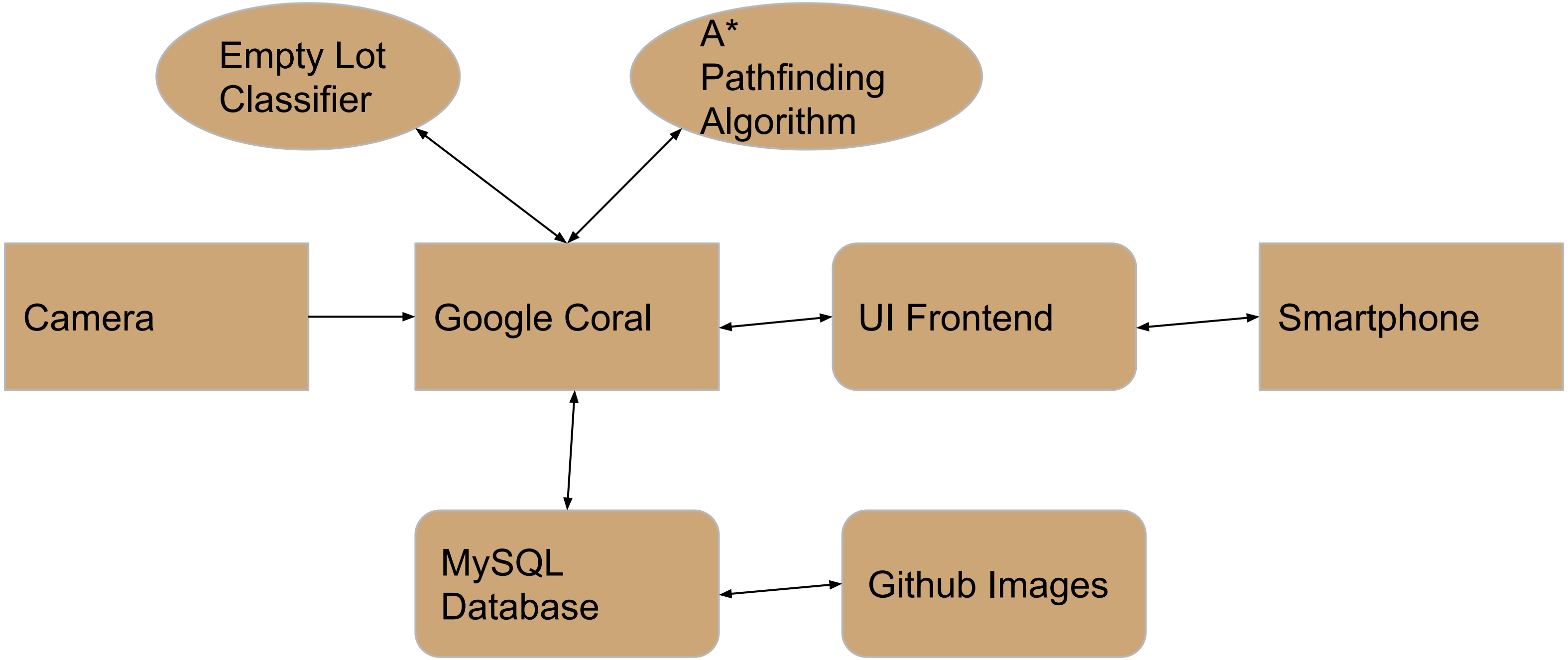
iValet Design Review

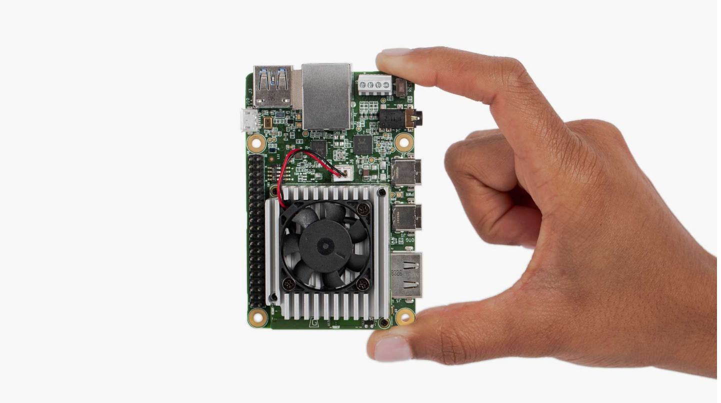
Members: Wei Xiong Toh, Faiza Yousuf, Yunchu Feng, Kelin Yu

Full Design Flowchart

- ● Image Processing
- ▢ Interface
- ■ Hardware



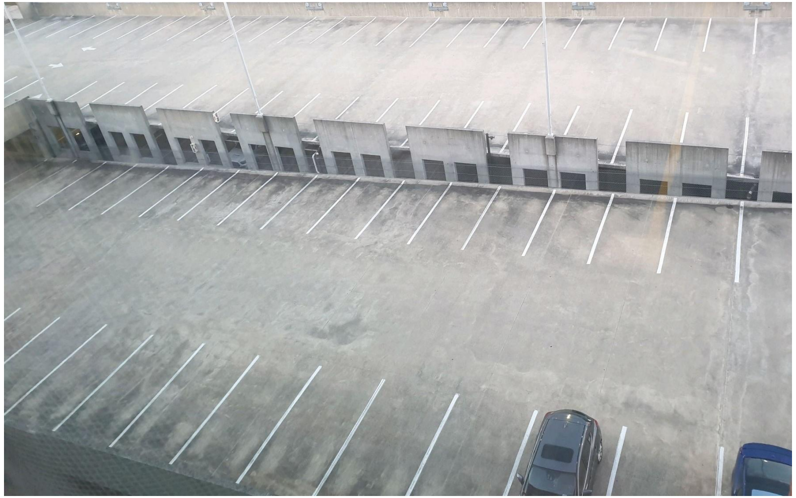
Google Coral

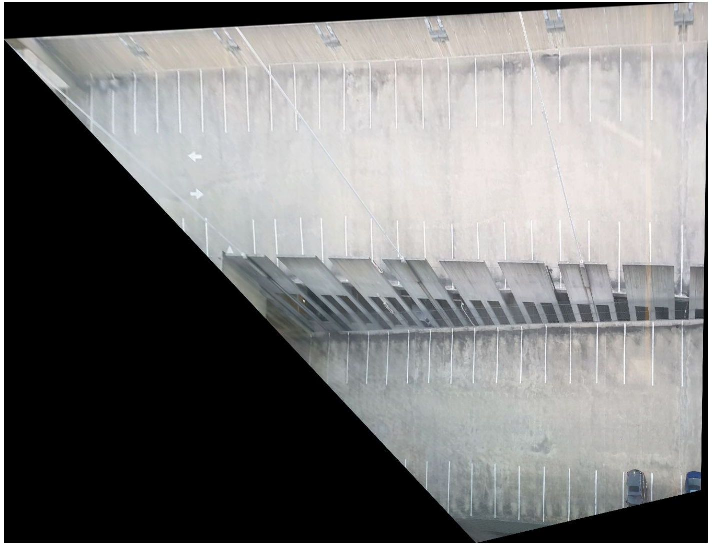


- Coral Dev Board is a single-board computer ideal for prototyping new projects that demand fast on-device inferencing for machine learning models.
- Set it up with Mendel Development Tool.
- Used it to run our detection model and path-planning algorithm and generate SQL Database.
- Can connect it to the coral camera and use the streaming server to watch the output images from our model.

Image Rectification

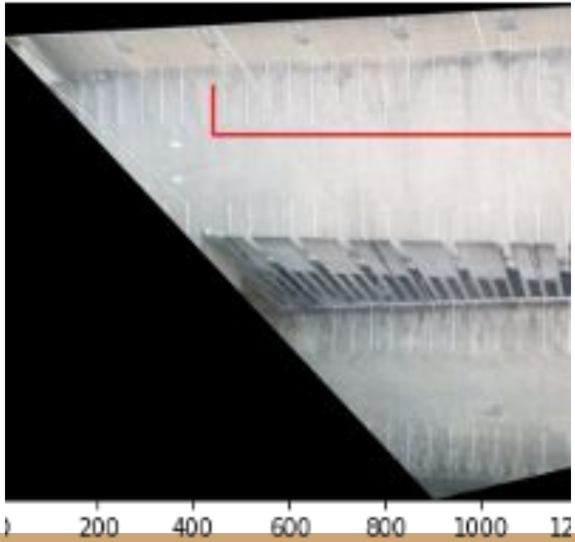
- Algorithm identifies vanishing points with RANSAC on edges detected
- Computes homography matrix based on vanishing points
- Warps images such that edges are parallel





A* Pathfinding

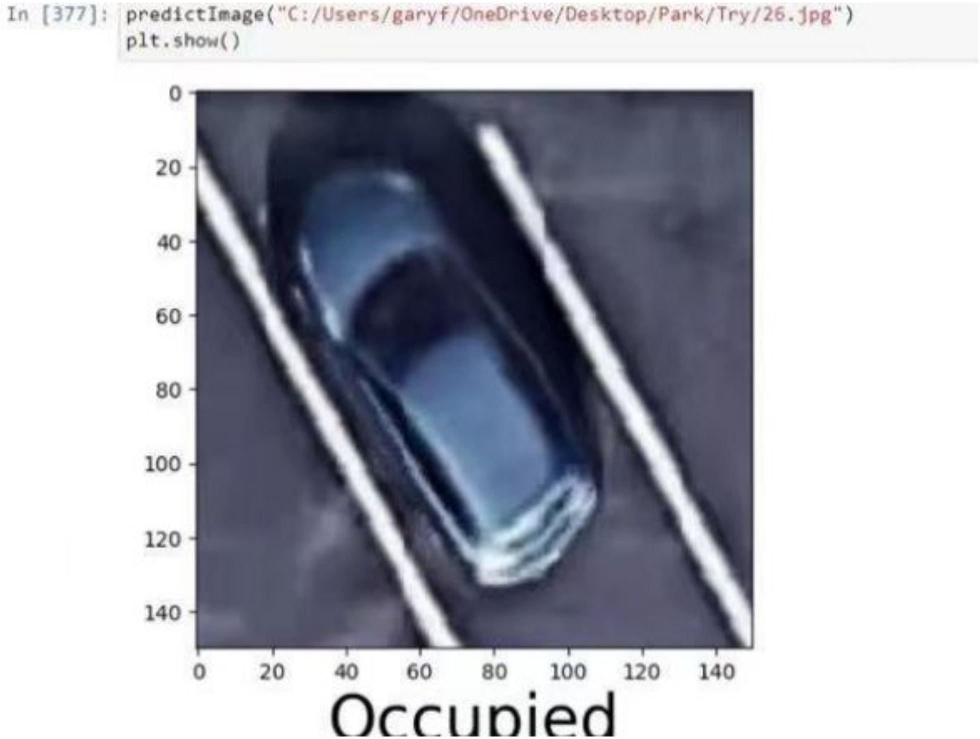
- Takes in image mask of available path, obstacles, starting and ending points
- Uses heuristic search to identify shortest path from start to end points



Empty Lot Classifier

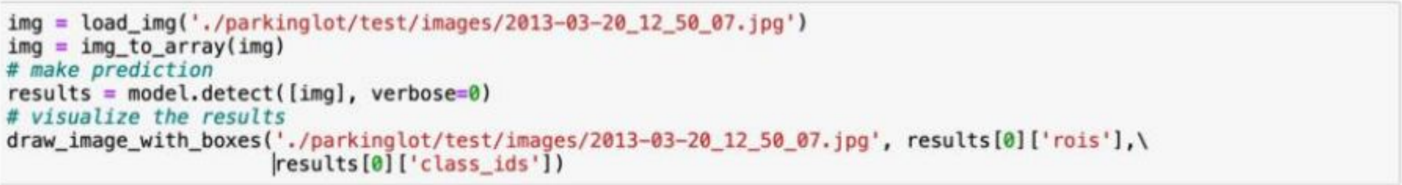
- The two classifier that is going to be used are binary CNN and segmentation
- Segmented data will be passed into binary for availability
- Pre-trained weights will be passed into Coral board

Empty Lot Classifier

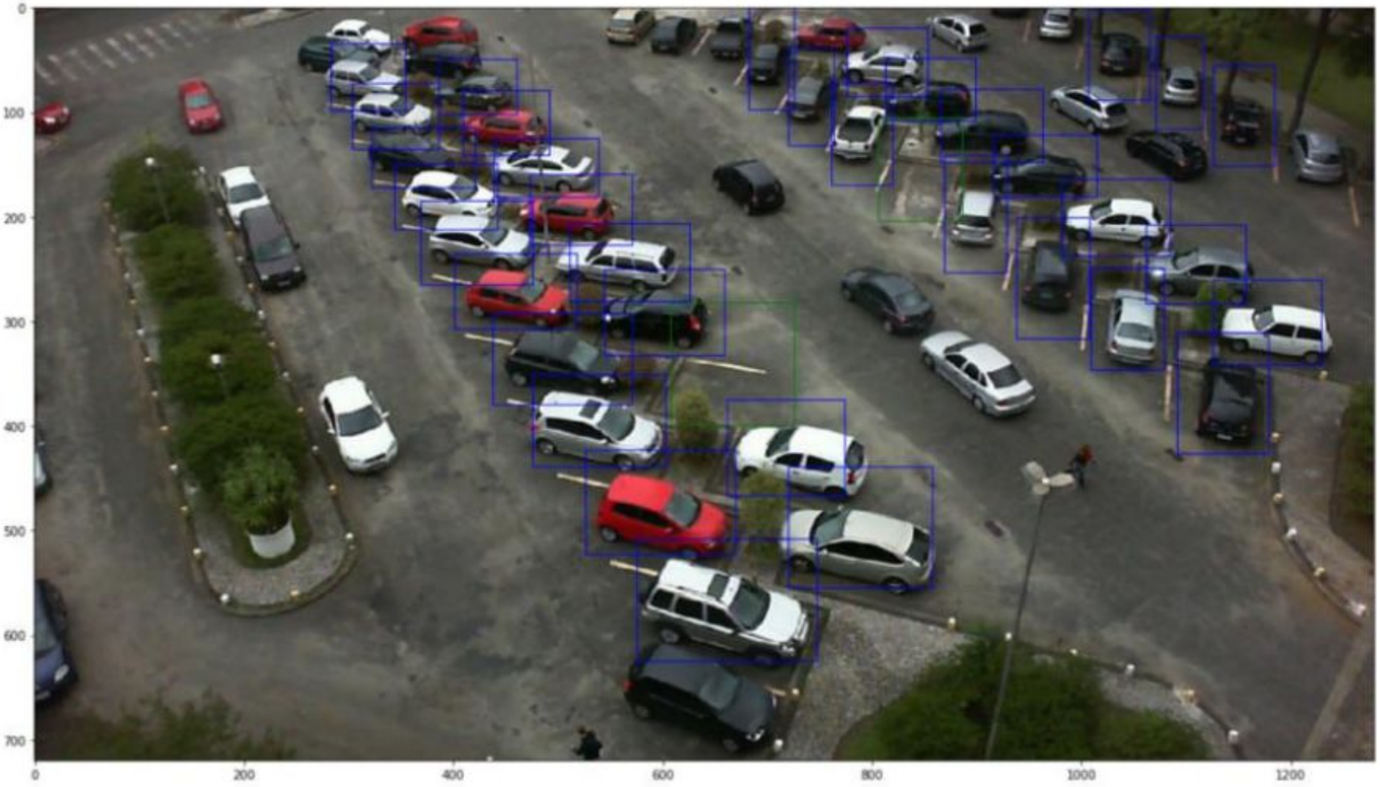


```
In [377]: predictImage("C:/Users/garyf/OneDrive/Desktop/Park/Try/26.jpg")
          plt.show()
```

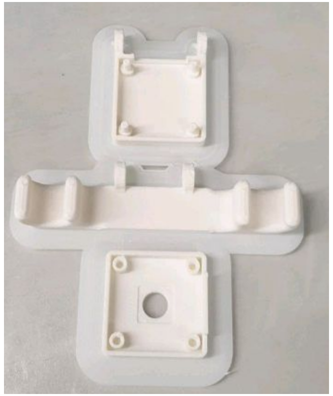
Occupied

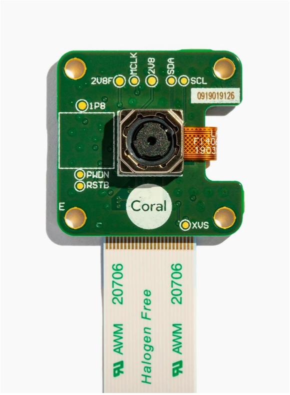


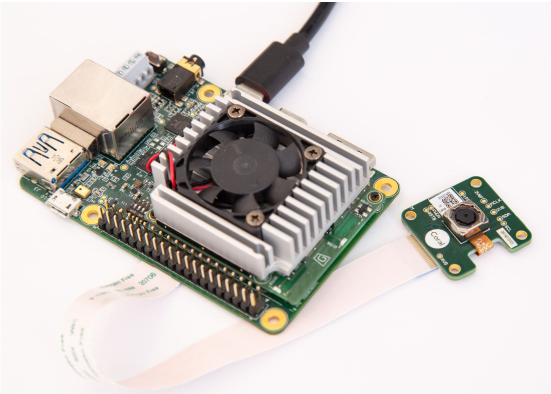
```python
img = load_img('./parkinglot/test/images/2013-03-20_12_50_07.jpg')
img = img_to_array(img)
# make prediction
results = model.detect([img], verbose=0)
# visualize the results
draw_image_with_boxes('./parkinglot/test/images/2013-03-20_12_50_07.jpg', results[0]['rois'],\
                      results[0]['class_ids'])
```



Camera





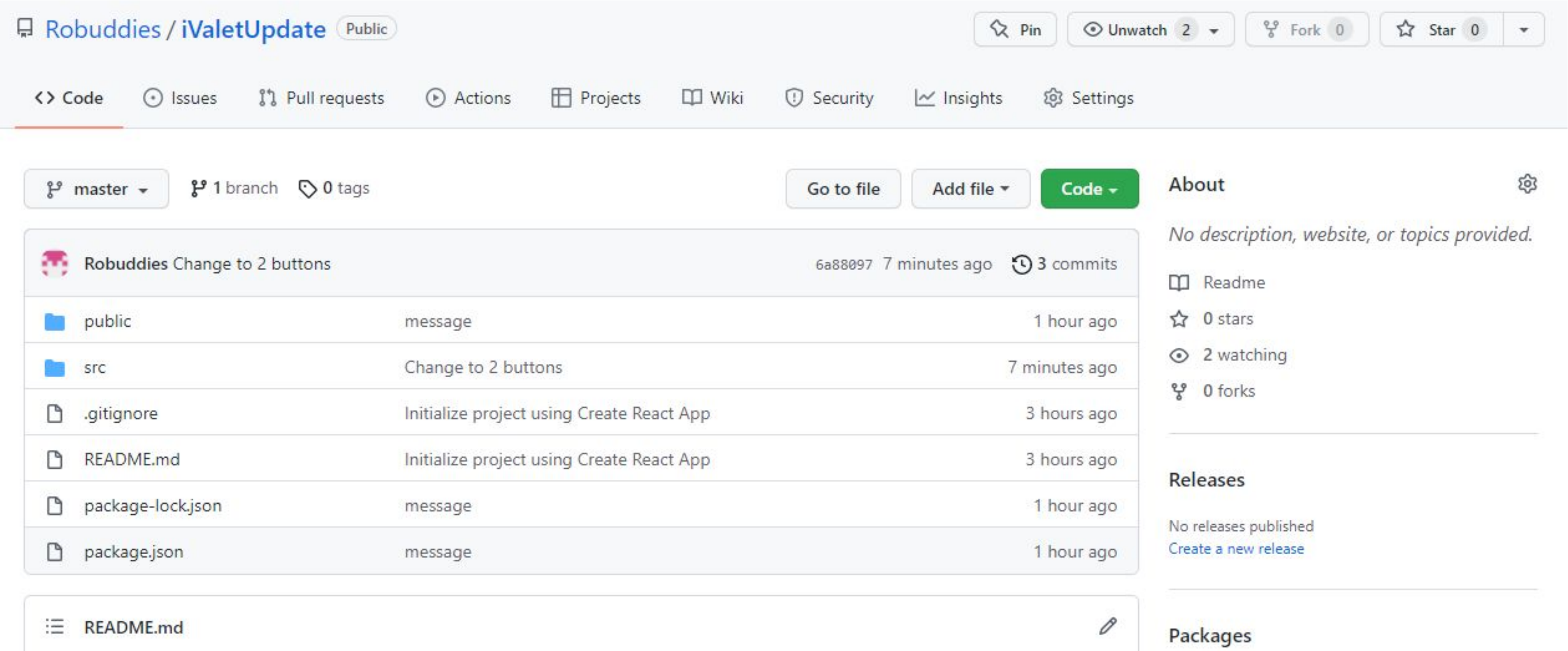


- Use the Coral camera with our Dev Board in this project.
- The coral camera is a 5-pixel camera compatible with the Dev board. It can easily bring an image input to our model.
- Used to take images of parking lots and bring them as image input into our detection model. We can see processed images from the streaming server.
- Built a mount to fix it at an appropriate angle.

UI Frontend

- Hosted through automatic deploys on Netlify - deploys triggered through GitHub updates
- https://www.ivalet-crc.com
- Built with React Library - JavaScript Library for front-end development
  - Uses Material-UI framework for styling React components
  - Uses React-Router-Dom@v6 to navigate between pages

UI Frontend - GitHub



Robuddies / iValetUpdate Public

Robuddies Change to 2 buttons 6a88097 7 minutes ago 3 commits

public	message	1 hour ago
src	Change to 2 buttons	7 minutes ago
.gitignore	Initialize project using Create React App	3 hours ago
README.md	Initialize project using Create React App	3 hours ago
package-lock.json	message	1 hour ago
package.json	message	1 hour ago

README.md

About

No description, website, or topics provided.

Readme

0 stars

2 watching

0 forks

Releases

No releases published

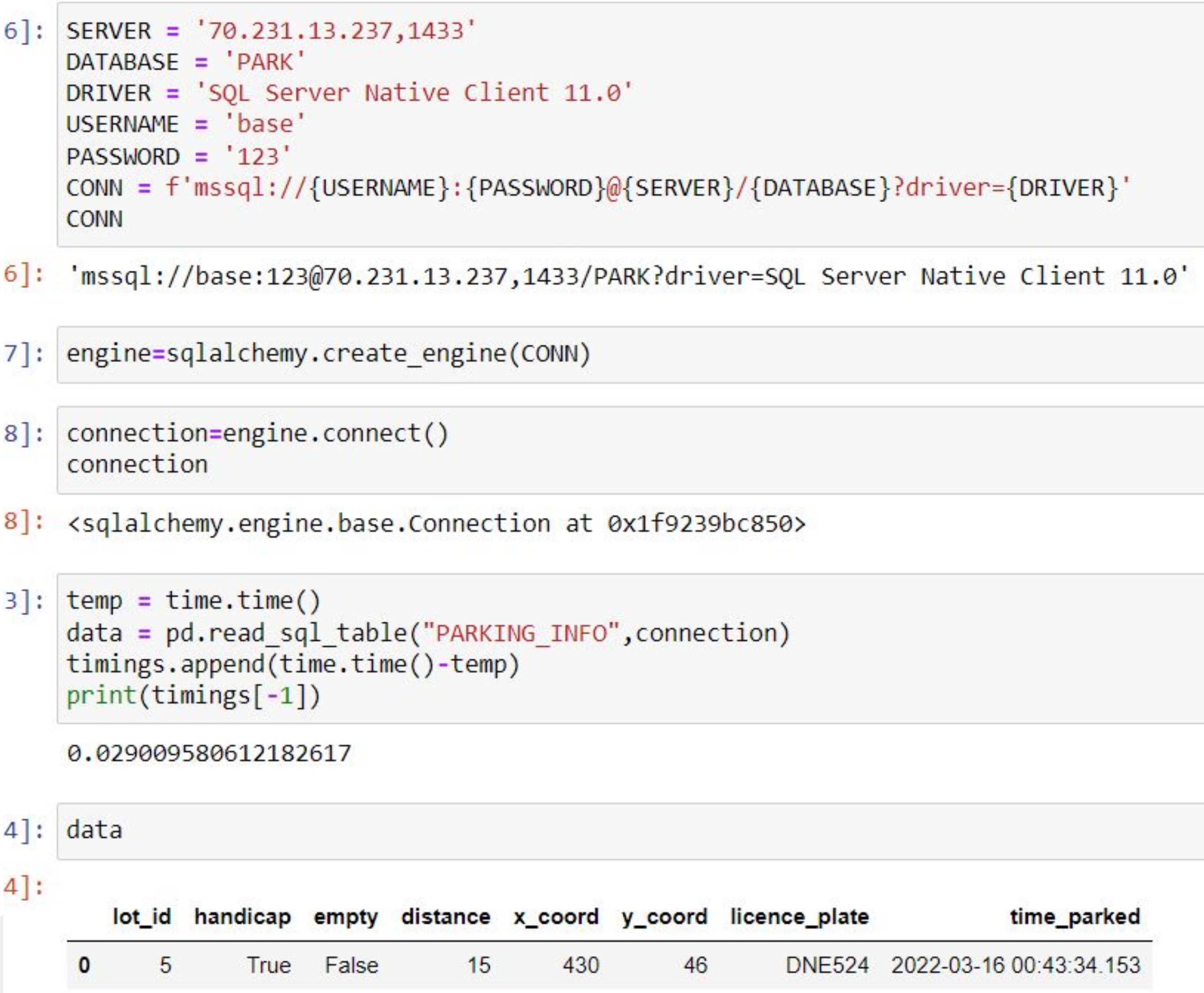
Create a new release

Packages

MSSQL

- Host all the necessary data for Frontend, Ex: “lot_id” column is the data used for image connection with GitHub
- Server hosted on member’s computer and can be accessed with python using the sqlAlchemy library
- Port Forwarding and and server’s exact IP is required for the access.

MSSQL



```
6]: SERVER = '70.231.13.237,1433'
    DATABASE = 'PARK'
    DRIVER = 'SQL Server Native Client 11.0'
    USERNAME = 'base'
    PASSWORD = '123'
    CONN = f'mssql://{USERNAME}:{PASSWORD}@{SERVER}/{DATABASE}?driver={DRIVER}'
    CONN

6]: 'mssql://base:123@70.231.13.237,1433/PARK?driver=SQL Server Native Client 11.0'
```

```
7]: engine=sqlalchemy.create_engine(CONN)
```

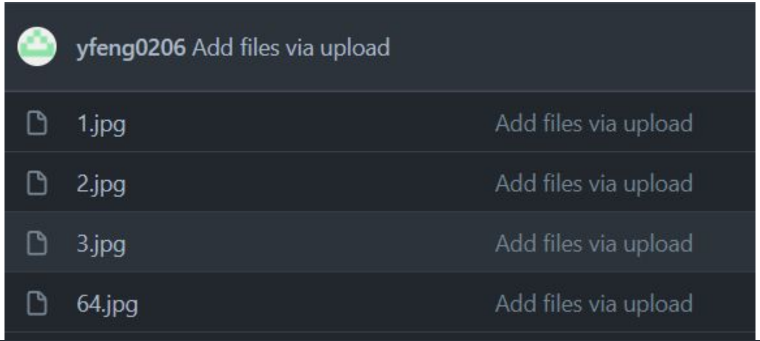
```
8]: connection=engine.connect()
    connection

8]: <sqlalchemy.engine.base.Connection at 0x1f9239bc850>
```

```
3]: temp = time.time()
    data = pd.read_sql_table("PARKING_INFO",connection)
    timings.append(time.time()-temp)
    print(timings[-1])

    0.029009580612182617
```

```
4]: data

4]:
```

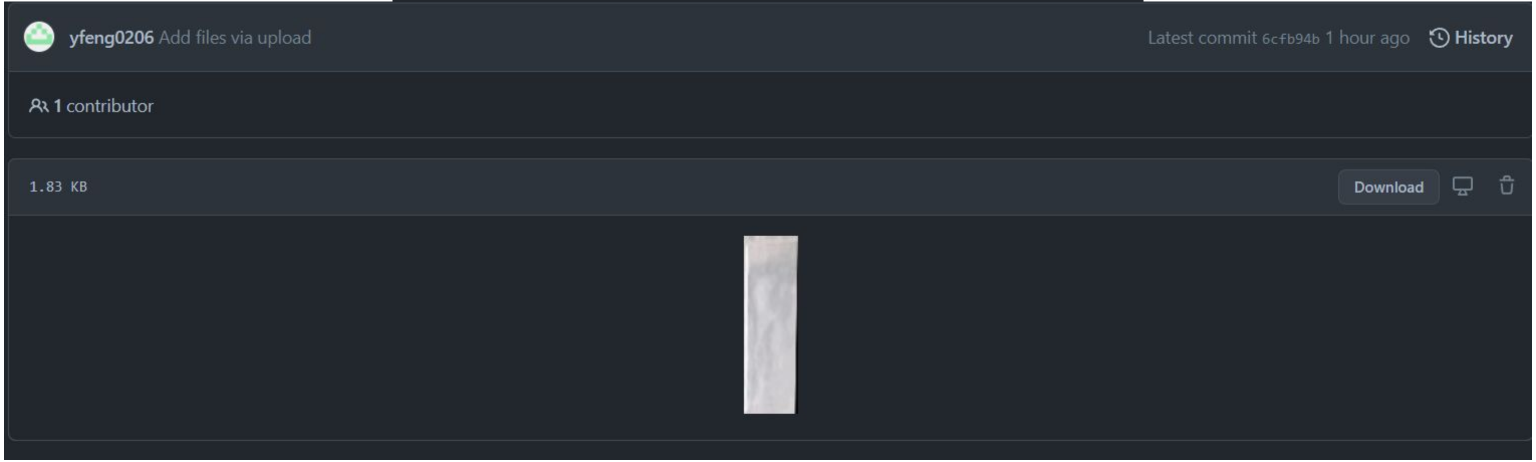
	lot_id	handicap	empty	distance	x_coord	y_coord	licence_plate	time_parked
0	5	True	False	15	430	46	DNE524	2022-03-16 00:43:34.153

Github Images



yfeng0206 Add files via upload

1.jpg	Add files via upload
2.jpg	Add files via upload
3.jpg	Add files via upload
64.jpg	Add files via upload



yfeng0206 Add files via upload

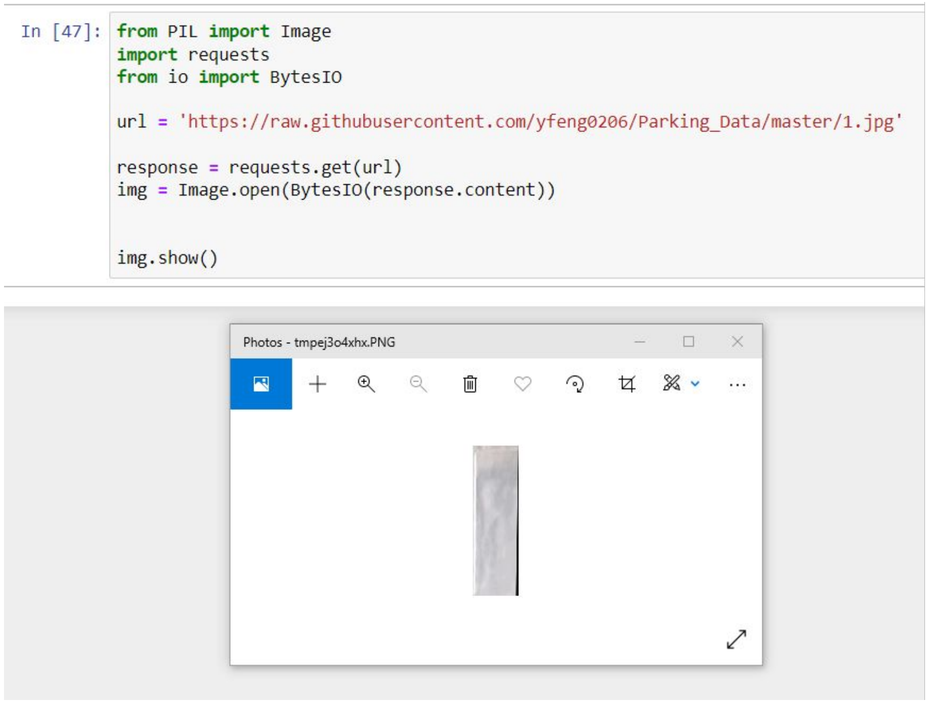
Latest commit 6cfb94b 1 hour ago

1 contributor

1.83 KB

- To avoid overwhelming disk space of Coral

1) Receive “lot_id” from SQL
2) Get image from GitHub matching Lot ID
3) Use that image to calculate the path



```
In [47]: from PIL import Image
         import requests
         from io import BytesIO

         url = 'https://raw.githubusercontent.com/yfeng0206/Parking_Data/master/1.jpg'

         response = requests.get(url)
         img = Image.open(BytesIO(response.content))


         img.show()
```