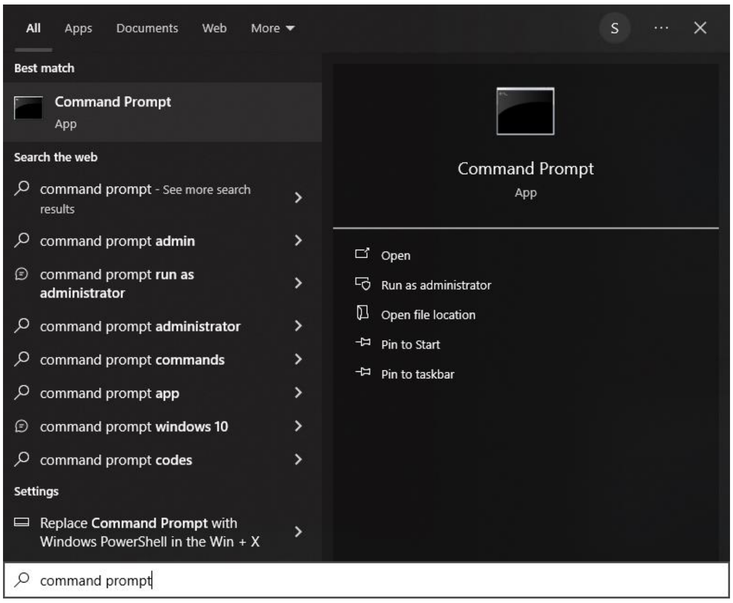
This project is developed based on Python 3.9. Therefore, when downloading Python, please ensure its version is >=3.9. You can download Python from this link: https://www.python.org/downloads/ We highly recommend using a Python IDE to run our project. We suggest using PyCharm. You can download it from this link: https://www.jetbrains.com/pycharm/
1;

How to download all 300 FELO Leaflets (Webscraping)：

*Tested on Windows 10

Pre-requisites:

1. Download Google Chrome
2. Have Python 3 installed
3. Have the necessary packages installed. To install the packages, open Command Prompt via Start. without quotation marks, run '`pip install requests beautifulsoup4 selenium`'



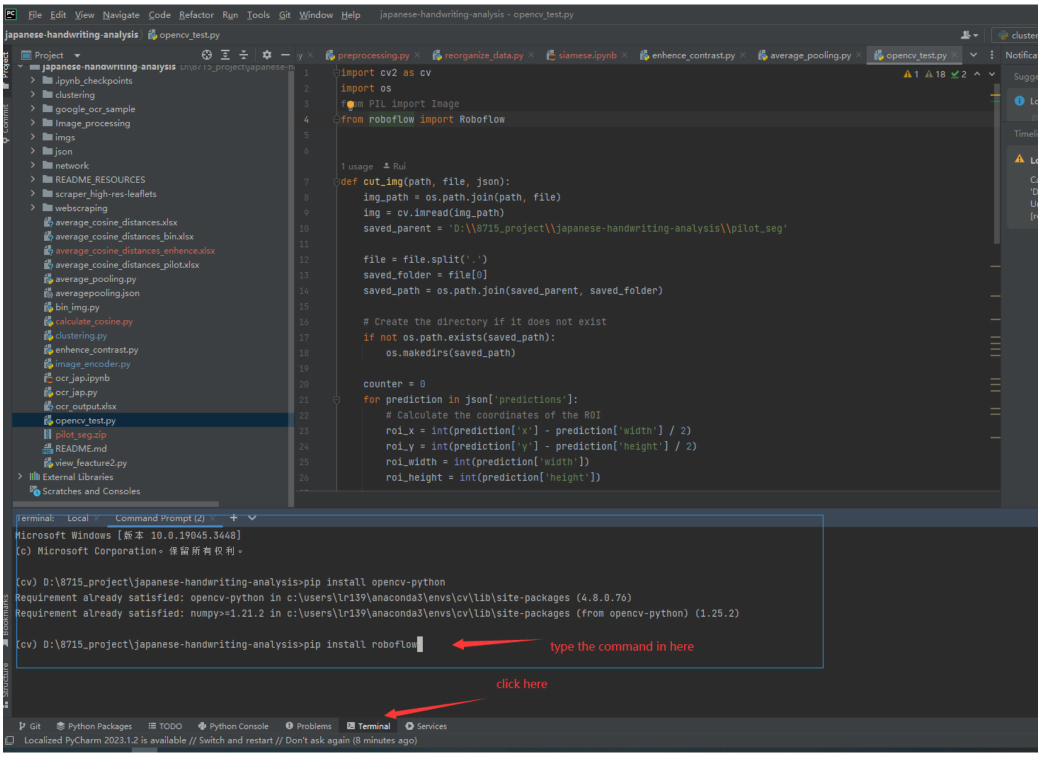
How to download the leaflets:

1. Within the Command Prompt, navigate to /scraper_high-res-leaflets/ folder.
2. Without quotation marks, run '`python scraper.py` ' or '`python3 scraper.py`'

Leaflets will be downloaded within the same directory the script is ran. Based on FELO_leaflets spreadsheet, there should be 300 leaflets downloaded. Download speed may vary depending on internet speed.

Word Segmentation：

- Use the following commands to install the required libraries
  pip install opencv-python
  pip install roboflow


japanese-handwriting-analysis - opencv_test.py
preprocessing.py
reorganize_data.py
siamese.ipynb
enhence_contrast.py
average_pooling.py
opencv_test.py
Terminal: Local Command Prompt (2)
Git
Python Packages
TODO
Python Console
Problems
Terminal
Services
Localized PyCharm 2023.1.2 is available // Switch and restart // Don't ask again (8 minutes ago)

```python
1  import cv2 as cv
2  import os
3  from PIL import Image
4  from roboflow import Roboflow
5
6
7  def cut_img(path, file, json):
8      img_path = os.path.join(path, file)
9      img = cv.imread(img_path)
10     saved_parent = 'D:\\8715_project\\japanese-handwriting-analysis\\pilot_seg'
11
12     file = file.split('.')
13     saved_folder = file[0]
14     saved_path = os.path.join(saved_parent, saved_folder)
15
16     # Create the directory if it does not exist
17     if not os.path.exists(saved_path):
18         os.makedirs(saved_path)
19
20     counter = 0
21     for prediction in json['predictions']:
22         # Calculate the coordinates of the ROI
23         roi_x = int(prediction['x'] - prediction['width'] / 2)
24         roi_y = int(prediction['y'] - prediction['height'] / 2)
25         roi_width = int(prediction['width'])
26         roi_height = int(prediction['height'])
```

```
Microsoft Windows [版本 10.0.19045.3448]
(c) Microsoft Corporation。保留所有权利。

(cv) D:\8715_project\japanese-handwriting-analysis>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\lr139\anaconda3\envs\cv\lib\site-packages (4.8.0.76)
Requirement already satisfied: numpy>=1.21.2 in c:\users\lr139\anaconda3\envs\cv\lib\site-packages (from opencv-python) (1.25.2)

(cv) D:\8715_project\japanese-handwriting-analysis>pip install roboflow
```

type the command in here

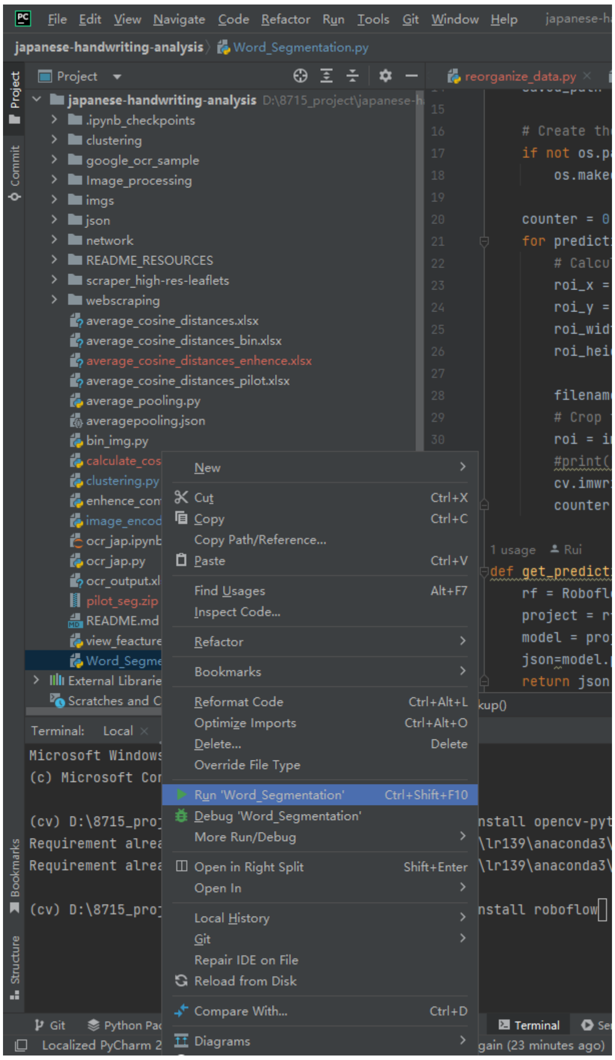
click here

- Open the Word Segmentation.py file. In this file, we will invoke the model from RoboFlow to implement Word Segmentation. What you need to modify are:

  - Line 10's saved_parent: This is the path where the output images will be saved.
  - Line 42's path: This is the path for the input images.
  - Line 36's api key: Refer to the RoboFlow section for this

- The path for the input images should be in the following format:

  -japanese-handwriting-analysis
  ----input_image_folder
  ----------img1.jpg
  ----------img2.jpg
  ----------….jpg
  ----other_folder
  ----Word_Segmentation.py



```
    # Create the
    if not os.p
        os.make

    counter = 0
    for predict
        # Calcu
        roi_x =
        roi_y =
        roi_wid
        roi_hei

        filenam
        # Crop
        roi = i
        #print(
        cv.imwr
        counter

def get_predict
    rf = Robofl
    project = r
    model = pro
    json=model.
    return json
```

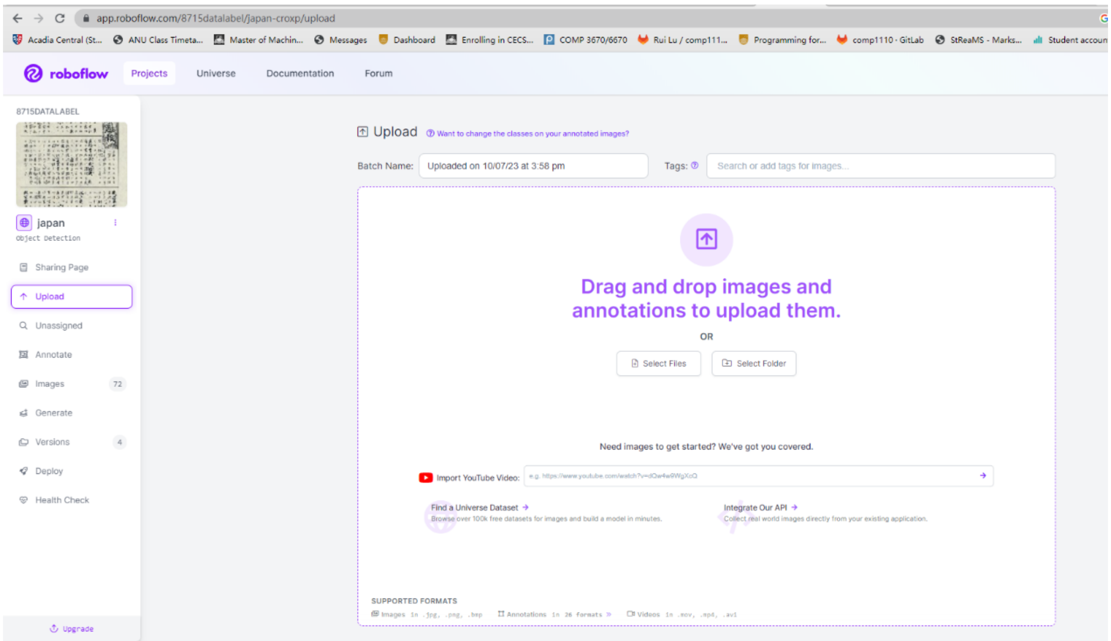
```
Microsoft Windows
(c) Microsoft Cor

(cv) D:\8715_proj                                 nstall opencv-pyt
Requirement alrea                                 \lr139\anaconda3\
Requirement alrea                                 \lr139\anaconda3\

(cv) D:\8715_proj                                 nstall roboflow
```

- In the left navigation pane, select the file and right-click, then choose Run

2: RoboFlow

RoboFlow offers an integrated platform for annotating data, training models, and deploying models. Our current RoboFlow project is nearing its API call limit. Therefore, you might consider creating a new RoboFlow project and transferring the annotated data into it. The annotated data can be found on Google Drive under the filename 'japan.v5i.yolov8'.

Train model using RobowFlow:


app.roboflow.com/8715datalabel/japan-croxp/upload
Acadia Central (St... ANU Class Timeta... Master of Machin... Messages Dashboard Enrolling in CECS... COMP 3670/6670 Rui Lu / comp111... Programming for... comp1110 · GitLab StReaMS - Marks... Student accoun
roboflow
Projects
Universe
Documentation
Forum
8715DATALABEL
japan
Object Detection
Sharing Page
Upload
Unassigned
Annotate
Images 72
Generate
Versions 4
Deploy
Health Check
Upgrade
Select Files
Select Folder

Upload ⓘ Want to change the classes on your annotated images?

Batch Name: Uploaded on 10/07/23 at 3:58 pm

Tags: Search or add tags for images...

Drag and drop images and annotations to upload them.

OR

Need images to get started? We've got you covered.

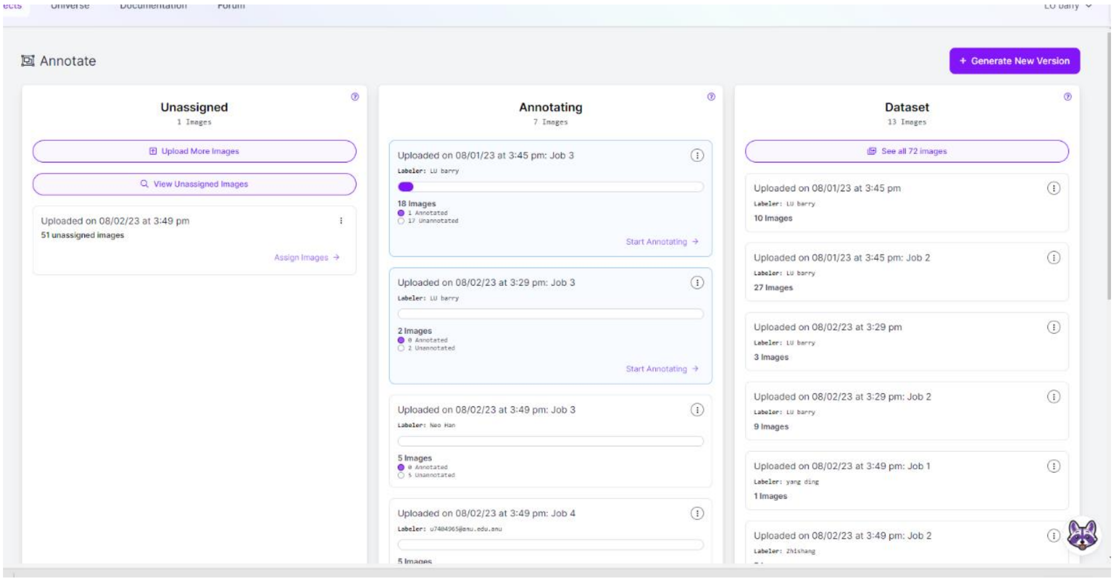
Import YouTube Video: e.g. https://www.youtube.com/watch?v=dQw4w9WgXcQ

Find a Universe Dataset →
Browse over 100k free datasets for images and build a model in minutes.

Integrate Our API →
Collect real world images directly from your existing application.

SUPPORTED FORMATS
Images in .jpg, .png, .bmp · Annotations in 26 formats » · Videos in .mov, .mp4, .avi

First, open your RoboFlow project, then go to the 'upload' page to upload your dataset



Unassigned
1 Images

Uploaded on 08/02/23 at 3:49 pm
51 unassigned images

Annotating
7 Images

Uploaded on 08/01/23 at 3:45 pm: Job 3
Labeler: LU barry
18 Images
1 Annotated
17 Unannotated

Uploaded on 08/02/23 at 3:29 pm: Job 3
Labeler: LU barry
2 Images
0 Annotated
2 Unannotated

Uploaded on 08/02/23 at 3:49 pm: Job 3
Labeler: Neo Han
5 Images
0 Annotated
5 Unannotated

Uploaded on 08/02/23 at 3:49 pm: Job 4
Labeler: u7484965@anu.edu.anu
5 Images

Dataset
13 Images

Uploaded on 08/01/23 at 3:45 pm
Labeler: LU barry
10 Images

Uploaded on 08/01/23 at 3:45 pm: Job 2
Labeler: LU barry
27 Images

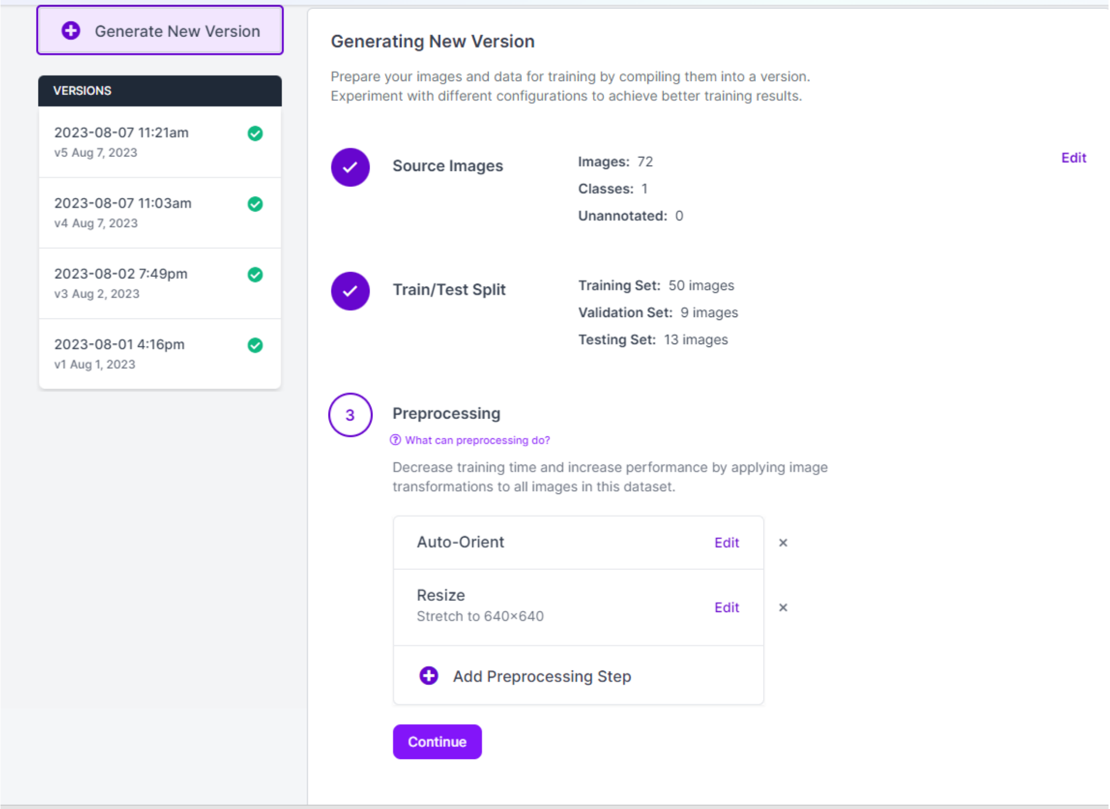
Uploaded on 08/02/23 at 3:29 pm
Labeler: LU barry
3 Images

Uploaded on 08/02/23 at 3:29 pm: Job 2
Labeler: LU barry
9 Images

Uploaded on 08/02/23 at 3:49 pm: Job 1
Labeler: yang ding
1 Images

Uploaded on 08/02/23 at 3:49 pm: Job 2
Labeler: Zhishang

If you want to annotate more data, go to the 'annoted' page. Click on 'assign image' to allocate the images to yourself or your team members. Then, click 'start annotating'. After completing the annotations, click on 'generate new version'.



Generate New Version

VERSIONS

2023-08-07 11:21am ✓
v5 Aug 7, 2023

2023-08-07 11:03am ✓
v4 Aug 7, 2023

2023-08-02 7:49pm ✓
v3 Aug 2, 2023

2023-08-01 4:16pm ✓
v1 Aug 1, 2023

Generating New Version

Prepare your images and data for training by compiling them into a version.
Experiment with different configurations to achieve better training results.

✓ Source Images — **Images:** 72, **Classes:** 1, **Unannotated:** 0 — Edit

✓ Train/Test Split — **Training Set:** 50 images, **Validation Set:** 9 images, **Testing Set:** 13 images

3 Preprocessing

What can preprocessing do?

Decrease training time and increase performance by applying image transformations to all images in this dataset.

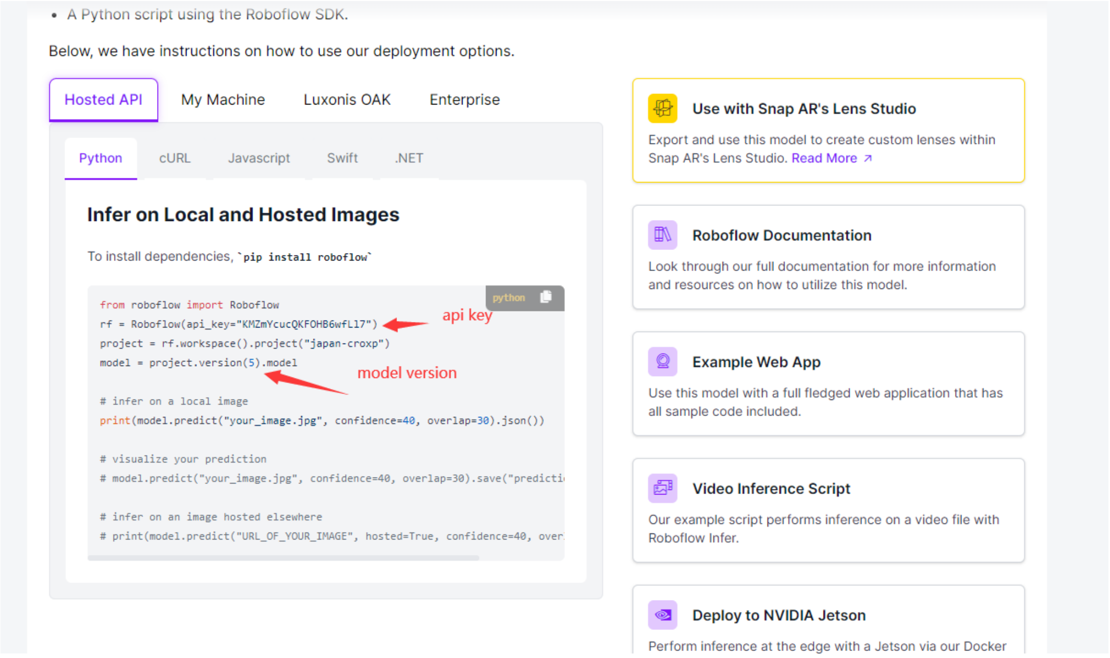
Auto-Orient — Edit ×

Resize
Stretch to 640×640 — Edit ×

Add Preprocessing Step

Continue

Follow the instructions on this page, and then you can train your model.



- A Python script using the Roboflow SDK.

Below, we have instructions on how to use our deployment options.

Hosted API | My Machine | Luxonis OAK | Enterprise

Python | cURL | Javascript | Swift | .NET

Infer on Local and Hosted Images

To install dependencies, `pip install roboflow`

```python
from roboflow import Roboflow
rf = Roboflow(api_key="KMZmYcucQKFOHB6wfLl7")
project = rf.workspace().project("japan-croxp")
model = project.version(5).model

# infer on a local image
print(model.predict("your_image.jpg", confidence=40, overlap=30).json())

# visualize your prediction
# model.predict("your_image.jpg", confidence=40, overlap=30).save("predicti

# infer on an image hosted elsewhere
# print(model.predict("URL_OF_YOUR_IMAGE", hosted=True, confidence=40, over
```

Use with Snap AR's Lens Studio
Export and use this model to create custom lenses within Snap AR's Lens Studio. Read More ↗

Roboflow Documentation
Look through our full documentation for more information and resources on how to utilize this model.

Example Web App
Use this model with a full fledged web application that has all sample code included.

Video Inference Script
Our example script performs inference on a video file with Roboflow Infer.

Deploy to NVIDIA Jetson
Perform inference at the edge with a Jetson via our Docker

After completing the training, click 'deploy', and you can then invoke your model. You can modify the content in Word Segmentation.py based on the Python code they provide.

3: Clustering

OCR branch:

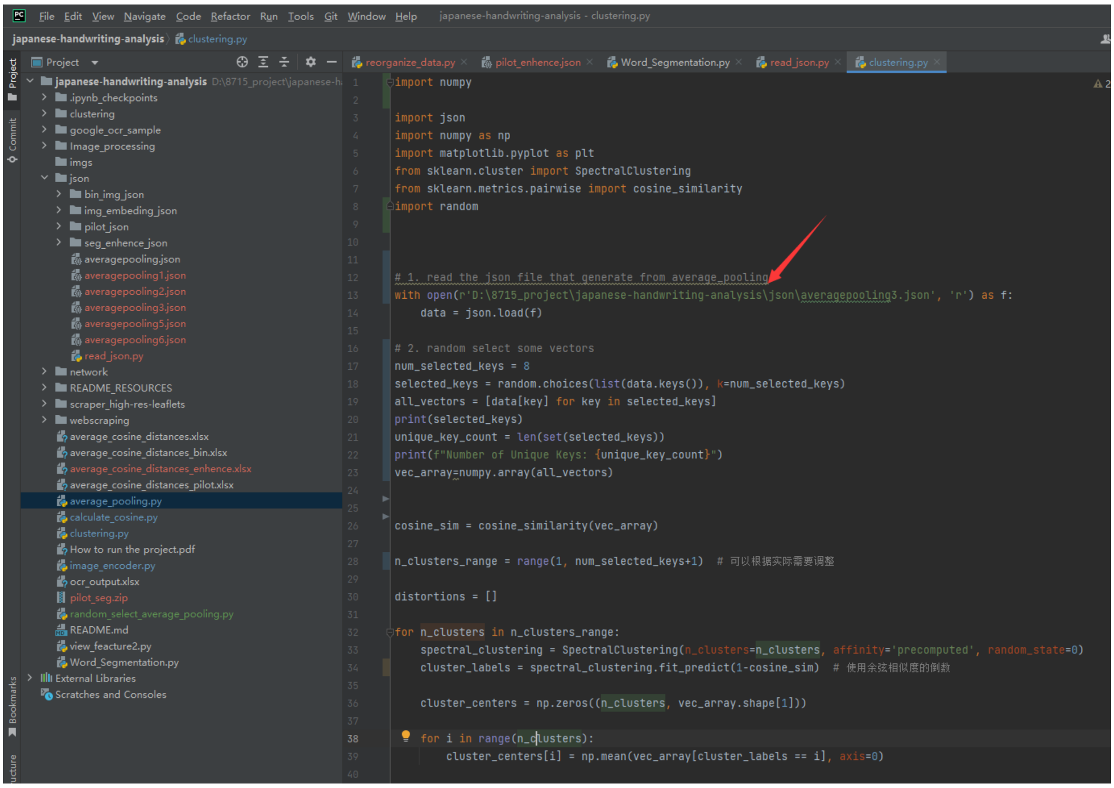
See the full plan in research report on confluence Report on OCR method progress, plan and constraints - International Collaboration on Handwriting - Confluence (atlassian.net)
Still lacking training and testing. Only finished finding OCR and use the model.
The result of seg_letter is in ocr_output.xlsx, if you want to use the model again. You can go to clustering/google_ocr.ipynb. Run every cell except last one, remember to change the input and output directory! The seg_letter is on google drive so probably download it to your computer and lead the path to it.
Ask Zhishang Bian if you have any questions.

Cluster with K-means:

This is work from semester 1 2023. New methods don't use it now. You probably don't need it
Open with Pycharm or Jupyter lab.
Jupyter lab: Run every cell and it will automatically download the library you need
Pycharm: Run main() and it will automatically download the library you need
Found it here:
https://gitlab.cecs.anu.edu.au/u7434576/japanese-handwriting-analysis/-/tree/main/clustering

Spectral clustering：

First,run the Word Segmentation.py,image_encoder.py,averagepooling.py



```
1   import numpy
2
3   import json
4   import numpy as np
5   import matplotlib.pyplot as plt
6   from sklearn.cluster import SpectralClustering
7   from sklearn.metrics.pairwise import cosine_similarity
8   import random
9
10
11
12  # 1. read the json file that generate from average_pooling
13  with open(r'D:\8715_project\japanese-handwriting-analysis\json\averagepooling3.json', 'r') as f:
14      data = json.load(f)
15
16  # 2. random select some vectors
17  num_selected_keys = 8
18  selected_keys = random.choices(list(data.keys()), k=num_selected_keys)
19  all_vectors = [data[key] for key in selected_keys]
20  print(selected_keys)
21  unique_key_count = len(set(selected_keys))
22  print(f"Number of Unique Keys: {unique_key_count}")
23  vec_array=numpy.array(all_vectors)
24
25
26  cosine_sim = cosine_similarity(vec_array)
27
28  n_clusters_range = range(1, num_selected_keys+1)  # 可以根据实际需要调整
29
30  distortions = []
31
32  for n_clusters in n_clusters_range:
33      spectral_clustering = SpectralClustering(n_clusters=n_clusters, affinity='precomputed', random_state=0)
34      cluster_labels = spectral_clustering.fit_predict(1-cosine_sim)  # 使用余弦相似度的倒数
35
36      cluster_centers = np.zeros((n_clusters, vec_array.shape[1]))
37
38      for i in range(n_clusters):
39          cluster_centers[i] = np.mean(vec_array[cluster_labels == i], axis=0)
40
```

Modify the path on line 13. You can generate the json file yourself or use the one in gitlab. Then you can run the program

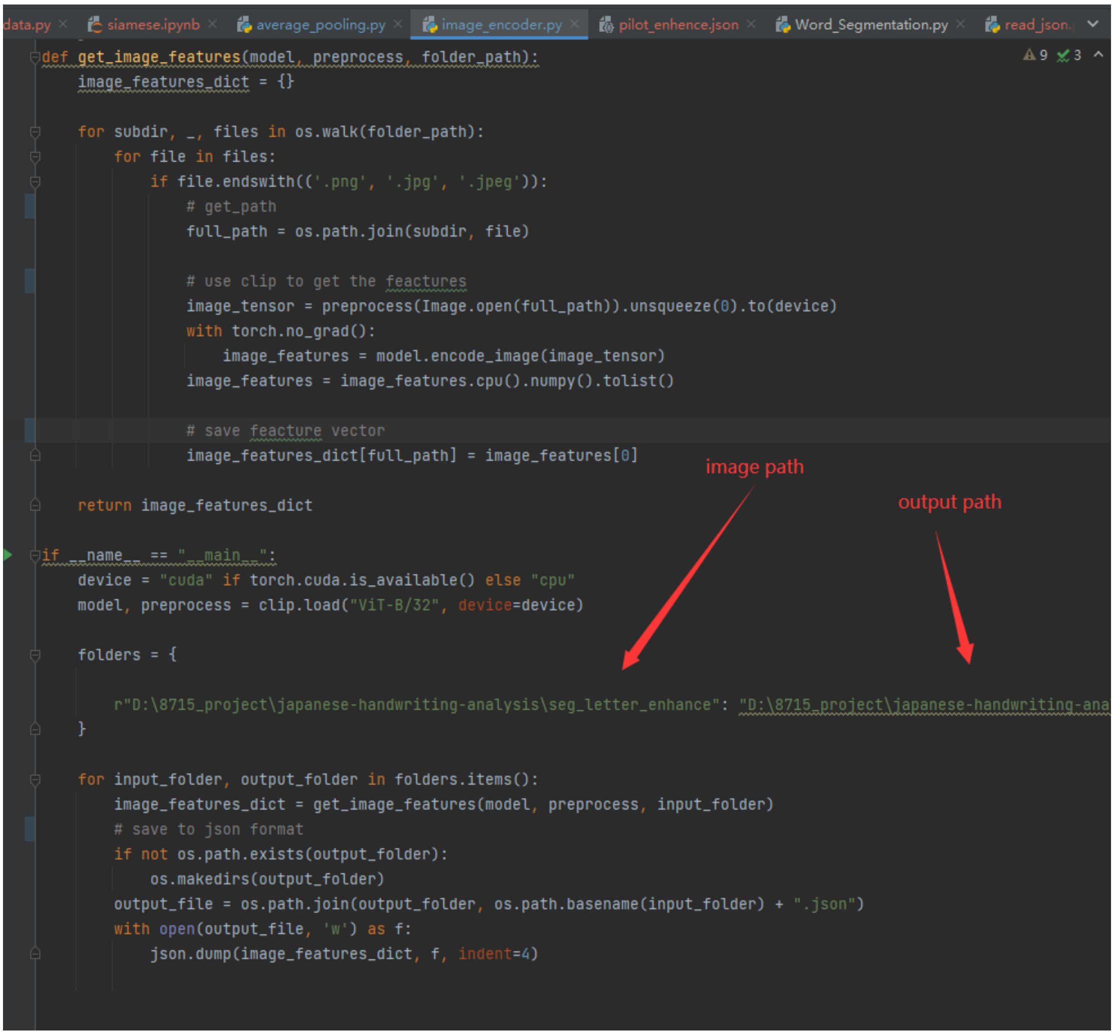
4：Image encoder：

Install the following package pytorch, pil,clip

The segamentaed image can be downloaded from here:

https://drive.google.com/drive/folders/1hecluxT4ad-6tQBP7q64SHWG3DRKsFKW

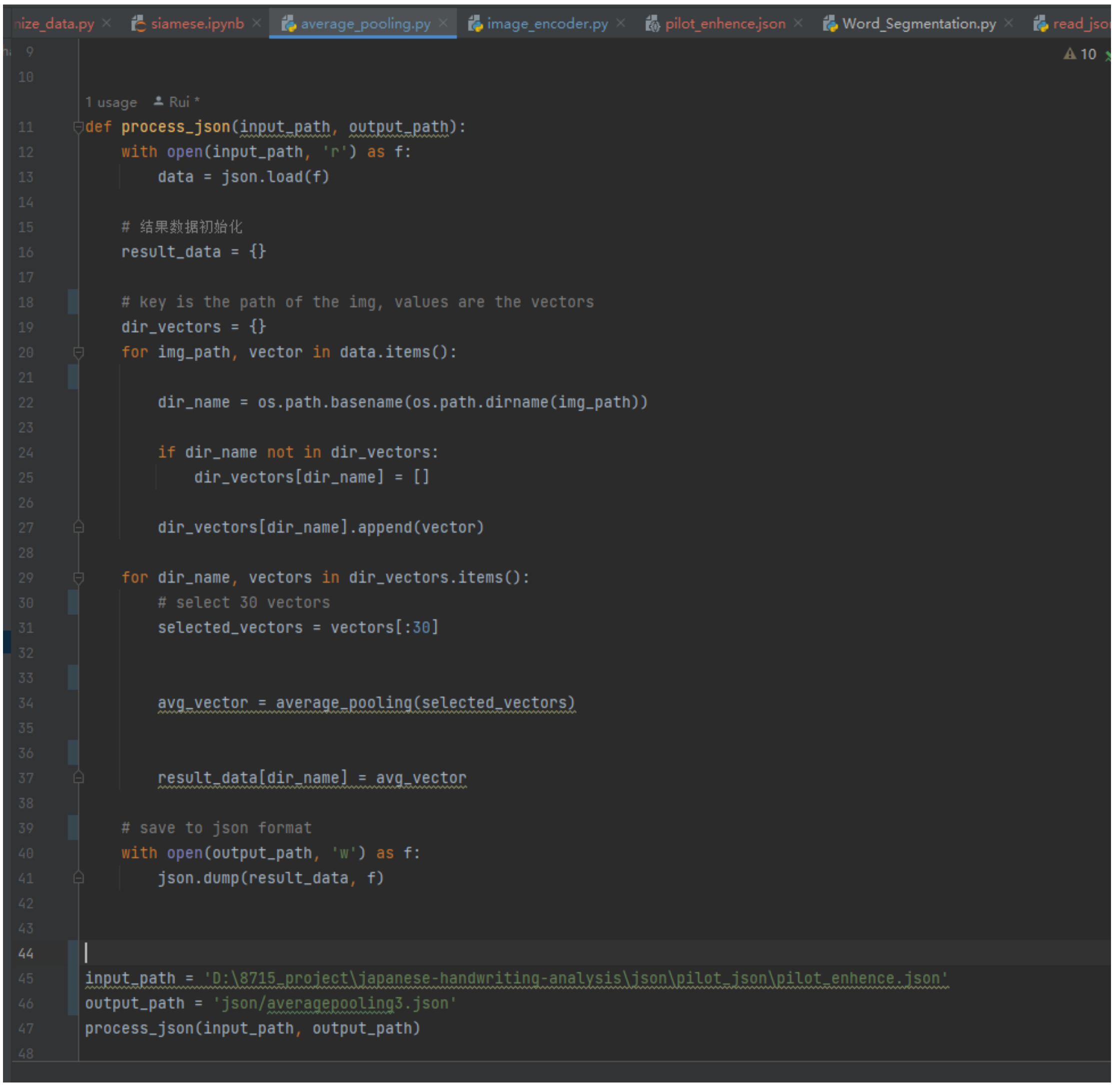
seg_letter.zip is the leaflets. Pilot_seg,zip is the test data. You can ou can unzip them in the root directory.



```python
def get_image_features(model, preprocess, folder_path):
    image_features_dict = {}

    for subdir, _, files in os.walk(folder_path):
        for file in files:
            if file.endswith(('.png', '.jpg', '.jpeg')):
                # get_path
                full_path = os.path.join(subdir, file)

                # use clip to get the feactures
                image_tensor = preprocess(Image.open(full_path)).unsqueeze(0).to(device)
                with torch.no_grad():
                    image_features = model.encode_image(image_tensor)
                image_features = image_features.cpu().numpy().tolist()

                # save feacture vector
                image_features_dict[full_path] = image_features[0]

    return image_features_dict

if __name__ == "__main__":
    device = "cuda" if torch.cuda.is_available() else "cpu"
    model, preprocess = clip.load("ViT-B/32", device=device)

    folders = {

        r"D:\8715_project\japanese-handwriting-analysis\seg_letter_enhance": "D:\8715_project\japanese-handwriting-ana
    }

    for input_folder, output_folder in folders.items():
        image_features_dict = get_image_features(model, preprocess, input_folder)
        # save to json format
        if not os.path.exists(output_folder):
            os.makedirs(output_folder)
        output_file = os.path.join(output_folder, os.path.basename(input_folder) + ".json")
        with open(output_file, 'w') as f:
            json.dump(image_features_dict, f, indent=4)
```

Modify the path on line 33. Then you can run it.

5: AveragePooling



```
 9
10
11  def process_json(input_path, output_path):
12      with open(input_path, 'r') as f:
13          data = json.load(f)
14
15      # 结果数据初始化
16      result_data = {}
17
18      # key is the path of the img, values are the vectors
19      dir_vectors = {}
20      for img_path, vector in data.items():
21
22          dir_name = os.path.basename(os.path.dirname(img_path))
23
24          if dir_name not in dir_vectors:
25              dir_vectors[dir_name] = []
26
27          dir_vectors[dir_name].append(vector)
28
29      for dir_name, vectors in dir_vectors.items():
30          # select 30 vectors
31          selected_vectors = vectors[:30]
32
33
34          avg_vector = average_pooling(selected_vectors)
35
36
37          result_data[dir_name] = avg_vector
38
39      # save to json format
40      with open(output_path, 'w') as f:
41          json.dump(result_data, f)
42
43
44
45  input_path = 'D:\8715_project\japanese-handwriting-analysis\json\pilot_json\pilot_enhence.json'
46  output_path = 'json/averagepooling3.json'
47  process_json(input_path, output_path)
48
```

Modify the path on line 45 and 46. Then run the file.