

Train Model Page Instructions

1. Install requirements.txt in Terminal:

`pip install -r .\requirements.txt`

```
Terminal: Local (2) × + ∨  
(venv) PS C:\Users\JPCLi\Desktop\Stanford\CEE 329\train app> pip install -r .\requirements.txt
```

And

`pip install "dash[diskcache]"`

2. Your work folder should include:

- a) main_train_app.py
- b) split_dataset.py train_model.py evaluate_model.py
- c) dataset (the raw dataset is downloaded from <https://data.mendeley.com/datasets/5y9wdsg2zt/1> and rename it into 'dataset', including Negative and Positive)

Name	Date modified	Type	Size
Negative	6/1/2022 10:24 PM	File folder	
Positive	6/1/2022 10:25 PM	File folder	

3. Run the file: main_train_app.py in Terminal

`python .\main_train_app.py`

```
Terminal: Local (2) × + ∨  
(venv) PS C:\Users\JPCLi\Desktop\Stanford\CEE 329\train app> python .\main_train_app.py
```

```
Terminal: Local (2) × + ∨  
(venv) PS C:\Users\JPCLi\Desktop\Stanford\CEE 329\train app> python .\main_train_app.py  
Dash is running on http://127.0.0.1:3001/  
  
* Serving Flask app 'main_train_app' (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: on
```

4. Open the link <http://127.0.0.1:3001/> in a browser

Split Data into Train and Test Sets

Generate Train and Test Data from Folder

Train and Evaluate the Model

Input the number of training epoch:

Train Model

Input the Number of Test Samples

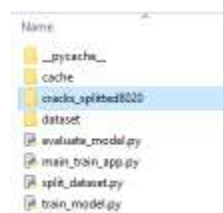
Evaluate Model in the test set

Check evaluate result

Evaluation Results

< >

- Click the **Generate Train and Test Data from Folder** button. Wait for several minutes. `CracksSplitted8020` will be generated.



Split Data into Train and Test Sets

Generate Train and Test Data from Folder

Splitting the dataset into train set (80%) and test set (20%)...
 Done!

Check the Train and Test Data

Train and Evaluate the Model

Input the number of training epoch:

Train Model

Input the Number of Test Samples

Evaluate Model in the test set

Check evaluate result

Evaluation Results

< >

- Click **Check the Train and Test Data**.


Split Data into Train and Test Sets

Generate Train and Test Data from Folder

Train Data

Cracks: 16000


00001.jpg



Test Data


Cracks: 4000

00005.jpg




Non-Cracks: 16000

00001.jpg



Non-Cracks: 4000

00006.jpg



< >

Train and Evaluate the Model

Input the number of training epoch:

1

Train Model

Input the Number of Test Samples

20

Evaluate Model in the test set

Check evaluate result

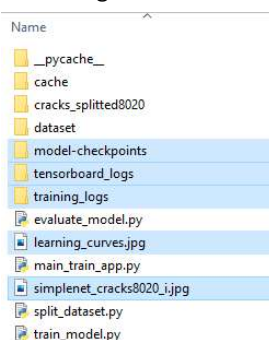
Evaluation Results

7. **Input the number of training epoch**, such as 5.

Estimate the training time: $\text{number of trainings} * (1 \sim 2) \text{ mins}$

To get a well-trained model, generally input >30 . Here we just take 5 as an example, which already can train a good model.

After the model is trained, 5 files will be generated.



Note:

- a) The following errors might arise. But it doesn't influence the workflow. Just ignore it.

Split Data into Train and Test

Generate Train and Test Data from Folder

Train and Evaluate the Model

Input the number of training epoch:
1

Train Model

the button has been clicked 1 times, the model is training...

Input the Number of Test Samples
20

Evaluate Model in the test set

Check evaluate result

Evaluation Results

Errors (1)

Callback error updating ..evaluate_done.children... 9:20:35 PM

- b) After model training is finished, 'Model training completed!' will show. The webpage might refresh itself. Just ignore it. All files related to the new model has been saved in your work directory. Go to the next step.

Train and Evaluate the Model

Input the number of training epoch:
5

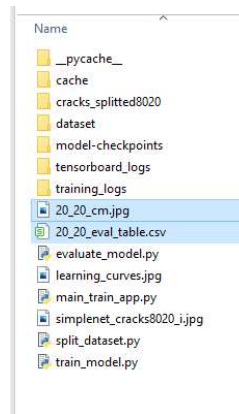
Train Model

the button has been clicked 1 times, the model is training...
Model training completed!

- c) Model training progress can be seen in your Terminal:

```
history = ModelFitGenerator(  
Epoch 1/5  
2022-06-02 17:35:02.021503: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384] Loaded cuD  
2022-06-02 17:35:06.457509: I tensorflow/stream_executor/cuda/cuda_blas.cc:1786] TensorFL  
125/125 [=====] - 215s 2s/step - loss: 0.2674 - accuracy: 0.8936  
Epoch 2/5  
125/125 [=====] - 53s 419ms/step - loss: 0.0498 - accuracy: 0.98  
Epoch 3/5  
125/125 [=====] - 50s 400ms/step - loss: 0.0442 - accuracy: 0.98  
Epoch 4/5  
125/125 [=====] - 50s 400ms/step - loss: 0.0400 - accuracy: 0.98  
Epoch 5/5  
125/125 [=====] - 54s 431ms/step - loss: 0.0406 - accuracy: 0.98
```

8. **Input the number of Test Samples**, such as 20 (max: 4000. There will be 4000 cracks images and non-cracks images in the test set). Click **Evaluate Model in the test set** button. After the evaluation is finished, 2 files will be generated:



Split Data into Train and Test Sets

Generate Train and Test Data from Folder

Train and Evaluate the Model

Input the number of training epoch:

Train Model

Input the Number of Test Samples

Evaluate Model in the test set

the button has been clicked 1 times, the model is evaluating...

Check evaluate result

Evaluation Results

Evaluation done: F1-score: 0.9749843652282676

Note:

The evaluation progress can also be seen in the Terminal:

```
2022-06-02 17:42:33.112744: I tensorflow/stream_executor/cuda/cuda_
1/1 [=====] - 15s 15s/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 20ms/step
1/1 [=====] - 0s 17ms/step
```

- Click **Check evaluation result**

There will be three figures created. The first one is the model structure. The second one is the model loss and accuracy during the training. The last figure shows the confusion matrix of the evaluation. In the table below, we can check the prediction result of each image.

Generate Train and Test Data from Folder

Splitting the dataset into train set (80%) and test set (20%)...

Done!

Check the Train and Test Data

Cracks: 16000



00001.jpg



Cracks: 4000

00004.jpg

This micrograph shows a section of normal colonic mucosa. The tissue is characterized by a regular arrangement of crypts and a well-defined lamina propria. There is no evidence of inflammation or neoplasia.

Input the number of training epoch:
5
Train Model

the button has been clicked 1 times, the model is training...
Model training completed!

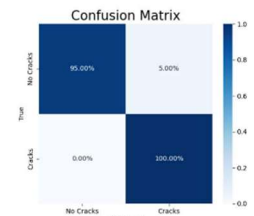
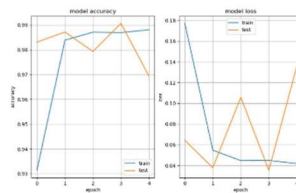
20


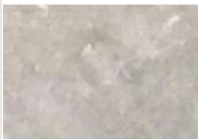
Evaluate Model in the test set

Check evaluate result

```

graph TD
    client_request[client_request] --> client2Client2[client2Client2]
    client2Client2 --> server_processing1[server_processing1_MatProcessing]
    server_processing1 --> client3_Client2[client3_Client2]
    client3_Client2 --> server_processing2[server_processing2_MatProcessing]
    server_processing2 --> Database_Plugin[Database_Plugin]
    Database_Plugin --> dataset_Client[dataset_Client]
    dataset_Client --> dataset3_Dataset[dataset3_Dataset]
  
```



	Inc	filename	Image	True Label	Predicted Label	Correct	Non-Cracks	Cracks
	f							
0	02617.jpg		Non Cracks	Non Cracks	Yes	100	0	
1	19113.jpg		Non Cracks	Non Cracks	Yes	100	0	

<<

<

1 / 4

>

>>