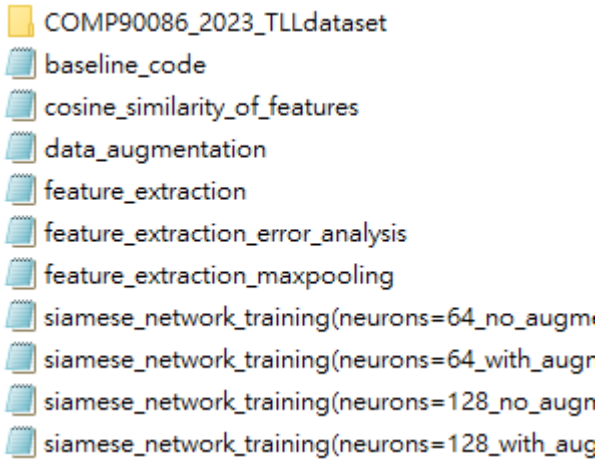
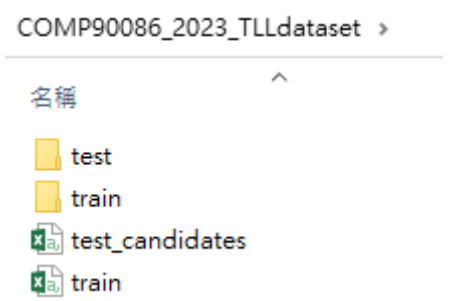
To reproduce the results, please place the code files (.ipynb) and the Numpy Array files (.npy) in the same directory as the folder "COMP90086_2023_TLLdataset". The "COMP90086_2023_TLLdataset" folder must contain the folders of train images and test images, along with two CSV files ("test_candidates.csv" and "train.csv"), as illustrated below.

Code Files and Numpy Array Files	COMP90086_2023_TLLdataset
	

baseline_code.ipynb

To calculate the cosine similarity between the pixels of two images, user can run "baseline_code.ipynb". For each of the left image in "test_candidates.csv", the process calculates the cosine similarity between the pixels of the left image and the pixels of the other 20 right images. The output file "baseline_results.csv" follows the submission format in Kaggle.

feature_extraction.ipynb & feature_extraction_maxpooling.ipynb

To execute feature extraction on the train images and test images, user can run "feature_extraction.ipynb" or "feature_extraction_maxpooling.ipynb".

- **"feature_extraction.ipynb":** Apply VGG16 to extract the features of the images without pooling.
- **"feature_extraction_maxpooling.ipynb":** Apply VGG16, ResNet, and DenseNet to extract the features of the images with max pooling.

The output are Numpy Array files (.npy) storing in the same directory as the code file.

cosine_similarity_of_features.ipynb

To calculate the cosine similarity between the extracted features of two images, user can run "cosine_similarity_of_features.ipynb". This process loads the feature extraction

results from previous section. Due to the large file size, the Numpy Array files from feature extraction using VGG16 without pooling are not provided; therefore, user need to execute the "feature_extraction.ipynb" code to generate the Numpy Array files before calculating the cosine similarity for the extracted features from VGG16 without pooling. The output files follow the submission format in Kaggle.

feature_extraction_maxpooling.ipynb

This notebook displays the example stated in the section of error analysis on feature extraction. The output image "example.png" illustrates the results in the report.

data_augmentation.ipynb

Before training the Siamese Networks, please run this code to implement data augmentation and create directories for storing the output files. The augmented images and a copy of the original left images in the train folder will be stored in the directory: ".\COMP90086_2023_TLLdataset\\train\\left_augmented". In addition, a new CSV file for training data ("train_augmented.csv") will be created in the same directory as "train.csv".

siamese_network_training(neurons=128_no_augmentation).ipynb
siamese_network_training(neurons=128_with_augmentation).ipynb
siamese_network_training(neurons=64_no_augmentation).ipynb
siamese_network_training(neurons=64_with_augmentation).ipynb

To train the Siamese Networks, user can run the codes of these 4 .ipynb files.
The differences of the codes are whether the training data include the augmented images (i.e. no_augmentation & with_augmentation) and whether the first layer of the Neural Networks within the Siamese Networks consist of 128 or 64 neurons (i.e. neurons=128 & neurons=64).
After training each model, the images of the training process and the test predictions will be output to the corresponding directory.
For running the codes in *no_augmentation*, the files will be stored in "L1", "L2".
For running the codes in *with_augmentation*, the files will be stored in "L1_ augmented", "L2_ augmented".
The L1 and L2 indicates the distance metric (i.e. Manhattan Distance or Euclidean Distance) that is employed in training the Siamese Network. The code would run through various parameter combinations and output the results to the respective folders. The CSV filenames of the predictions are in the form of "*Model_BatchSize_Layers*.csv". For instance, "vgg_32_2.csv" refers to applying the feature extraction results from

VGG16. The Neural Networks within the Siamese Networks consist of 2 layers, and are trained with a batch size of 32. The CSV files of the prediction results follow the submission format in Kaggle.

The prediction results of each training model is placed in the "Training Results" folder. Within the folder, two subfolders (i.e. "neurons = 128" & "neurons = 64") contains the initial ("neurons = 128") and the adjusted ("neurons = 64") attempt on training the models. Each folder would consist of the results regarding "L1", "L2", "L1_ augmented", "L2_ augmented", as stated above. The training graphs and the predictions by applying different parameter combinations can be found within these files.