README File

The code used to process and classify data for this study is contained within this folder. Instructions for running each section are listed below.

We have not included the dataset in the zip file as it requires more storage than is supported by Canvas. The link to the xray_pooled folder is: https://drive.google.com/drive/folders/1K_X7N9zYnu2fEWGVmL63t-pVWGtQqqw_?usp=sharing

Preprocessing

The preprocessing for this project is contained entirely in the process_images.py file which is in the “Preprocessing” folder. I do not recommend running this script as it does put a lot of strain on your local machine. If you choose to run this script, you can see the file’s docstrings for information on the parameters to each of the functions. The two functions that we used to build/analyze the dataset are under the “if _name_ ..” conditional at the bottom of the file.

Clustering Algorithms

To run KMeans and Agglomerative Clustering, run the notebooks titled kmeans.ipynb and AgglomerativeClustering.ipynb in Google Colab. They are in the “Code” folder under the “Clustering” folder. They use the data xray_dataset_pooled folder mounted from a shared Google Drive, so code for importing the data may need to be adjusted if the main folder is not uploaded to a shared drive.

MPP & kNN

The code for this portion is in the “MPP_kNN folder”.

To run the MPP code, make sure to first run the Matrix_Mean.py and input the dataset from the project. Additionally run the Covariance_Matrix.py and input the dataset in order to get the covariance matrices needed to run case 2 and case 3 for MPP. The Matrix_Mean.py and Covariance_Matrix.npy will output .npy files that will make running MPP much quicker and the covariance matrices would be used in the Mahalanobis distance function within the kNN.py script. For the MPP.py make sure to load the averages and the covariance matrices in order to get the output labels needed for getting classification accuracy. The same stays true for the kNN if you are going to use Mahalanobis distance as the distance metric for nearest neighbors. Like

MPP, the kNN.py script will output the prediction labels as a numpy array that you can then use to get accuracy scores.

CNN and BPNN

The CNN can be found in the “CNN.ipynb” file in the “CNN” folder and the BPNN can be found in the “BPNN.ipynb” file in the “BPNN” folder. The models must be ran in the same directory as the dataset, and the dataset folder must be named “xray_dataset_pooled”, like the original folder in the Google Drive.

I recommend running both of these files on Google Colab, a free service that hosts your Jupyter Notebook files. This allows you to run both of the models on GPUs; this capability is already built into the code.

Decision Tree, SVM, Random Forest, XGBoost

All of these algorithms can be found in the “Decision Tree/Random Forest/SVM/XGBoost.ipynb” file which is a Jupyter Notebook file.