Task A1

Using the lab examples, 68 features are extracted and scatter plot to see the viability of the algorithm. Then, all images are looped to extract the features, and label is fetched accordingly. However, worth to note not every image can be feature extracted. As it can be seen, 200 out of 5000 images cannot be feature extracted. This is due some picture even though it is positive image, some of them has different orientation and not front-faced, while others are blocked and shaded. The algorithm cannot detect the contour very vividly.

The data is then split to training and testing with 80:20 splitting. First, data is predicted using dummy classifier with most frequent strategy selected to provide baseline accuracy. Then, using two different models; SVC with RBF kernel & SVC with Linear kernel, data is trained and optimised using GridSearch. The parameter grids are C=[0.001, 0.01, 0.1, 10], gamma=[0.001, 0.01, 0.1, 1, 10] for RBF kernel while C=[0.01, 0.1, 10], gamma=[0.001, 0.01, 0.1, 1, 10] for Linear kernel.

With the GridSearch, the optimised hyper-parameters are identified which is C=10 and gamma=0.001 for RBF kernel while C=0.01, gamma=0.001 for Linear kernel. Both models’ learning curve are plotted to validate the model and prevent data overfit by observing the gap between training score and cross-validation score.

Finally, the confusion matrix along with other evaluation metrics such as precision, recall and f1 score is tabulated into a pandas dataframe to identify the best classifier among both. SVC classifier with Linear kernel, C=0.01 and gamma=0.001 is identified to be the best classifier.

Task A2

For task A2, author downloaded HaarCascade for face detection and smile detection from OpenCV. Then, some images are loaded and tested with the Haar classifier. Unfortunately, HaarCascade for smiling is questionable based on loaded images as author feels there are plenty false negative.

The task is then proceeded with feature extraction HOG (Histogram of Oriented Gradients) feature extraction as previous task. As mentioned before, 200 images failed to be detected. Regardless, same procedure as previous task is done except that author tested with reduced dimensions. As feature extraction produces 68 coordinates, author uses only 37 which consists of coordinates from jaw and mouth.

Both original 68 features point and reduced dimensions data is trained with two models; SVC with RBF kernel, SVC with Linear kernel. GridSearch is done to optimise the hyperparameter and found that for RBF kernel, C=0.1 and gamma=0.001 while for Linear kernel, C=0.01, gamma=0.001. With two different input sets and each with two model tested, learning curve graph for all four is plotted and comparison is made.

Last but not least, confusion matrix, precision score, recall score, and f1 score is tabulated in pandas dataframe. HaarCascade classifier is also introduced again to find the evaluation metrics and appended to the dataframe for comparison. It can be deduced that HaarCascade classifier has terrible recall score which prove author’s early hypothesis that HaarCascade has high false negative. From the result, it can also be seen reduced dimension produce relatively similar result as original 68 features. Hence, it is wise to use reduced dimension as this save training time as well as processing power. Ultimately, SVC classifier with Linear kernel, C=0.1 and gamma=0.001 with data dimension reduced is selected to be the best classifier.

Task B1

For the cartoon task, author test 68 feature extraction function at first. However, from 10000 images, 2185 images cannot be feature extracted. Approximately a fifth of data is lost which is quite huge. Author then display some images with or without the feature extraction and found out that even for some images with feature extraction coordinates, the coordinates are completely wrong. As for the case, author proceeded with using CNN instead.

With the CSV file, a pandas dataframe is created with filename and faceshape. This is vital in order to use ImageDataGenerator.flow\_from\_dataframe function from keras library. The data is split into training and testing with 75:25 before training data is split again into training and validation set using ImageDataGenerator function from keras library with batch size of 32.

CNN layer is then created with 5 layers along using Adam optimizers. CNN model summary is as follow. The data is trained using the model with epoch set to 25 initially. Loss history is then plotted to identify the training and validation errors as number of epochs increases. From the plot, it can be seen that from 15 epoch to 25, training and validation errors are barely changed.

The model is then evaluated by using testing data and confusion matrix is tabulated. The accuracy, recall, precision and f1 score are also printed to see the model’s performance.

Task B2

For Task B2, author uses HaarCascade Eyes to crop the eye rectangle. From the cropped image, author uses Hough Circles to find the eye and using bitwise\_and to crop only the circle. Then, using histogram, the RGB colour within the circle is detected and the highest is extracted using argmax function.

From this, all the images are loaded into a function to extract all the circles and labels for the eye colour. Using Hough Circles, 2305 images are unable to be detected as eyes out of 10000 images data. This might be due to some images wear glasses. However, there are still images without glasses and perfectly fine, still unable to be extracted using Hough Circles. Nevertheless, author still proceed to test this algorithm.

Data is standardizes using StandardScaler function from sklearn and split into train and test. Three classifiers are built; random forest classifier, k-neighbour classifier, SVC with Linear kernel classifier. All three of them produce below 70% accuracy which not satisfying to author. Author believe this low accuracy is due to the Hough Circles produced captured also the skin colour beneath or around the eye. The colour of the skin instead of eye is detected from argmax function in histogram.

Unsatisfied with this result, author uses CNN method with cropped image instead. Same model and number of layers is created as previous task except that image size is way smaller due to only eye image is taken instead of whole picture. Number of epochs is set to 25 as before. By observing the training and validation errors as epochs increase, author deduced that from 10th epoch onwards, training and validation errors are unchanged.

With result produced, data is then tabulated into confusion matrix by using test data. Accuracy, recall, precision and f1 score is also printed to check the model’s performance.