Facial Recognition Report

Team Members:

- LENG Dydamian
- LEU Chankunvath
- SAN Piseth
- LIM Sok Heang

1. Introduction

The objective of this project is to create a facial recognition model from a given dataset. The model will be trained to accurately identify and classify faces based on certain features. This can have a wide range of applications, including security, entertainment, and marketing. By accurately identifying faces, we can improve security measures, personalize user experiences, and gain valuable insights into consumer behaviour.

2. Data Collection + Data Processing

- Explanation of the data collection process
    - Download the lfw_funneled data set
    - Extract the file
    - Create a python script to threshold the data into a new folder
        - Loop through each folder
        - For each subfolder that contain more than 20 images, it will be appended into the new subfolder
        - New subfolder will be named ‘threshold_dataset’
        - Names will remain the same, which these will be modified as labels for the model to predict as identifiers
- Description of any data pre-processing steps taken
    - Define a function call preprocess_and_crop_image that will utilize a face_classifier (facial detection) from OpenCV library and take in an image
        - Convert the image to a grayscale if it is in colour
        - Detect the face, crop the image, and resize the image to a 64x64 size
        - Normalize the cropped face by 255.0
        - If no face is detected, the same process will still proceed
    - Declare a variable that will contain the face classifier from the OpenCV library called face_classifier
- Discussion of any challenges faced during data collection and processing
    - Some image faces are not detected by the face_classifier (facial detection)
    - The image was resized to 128x128 which was difficult to apply feature extraction and apply other actions on the data set
    - Some images are cropped incorrectly and detected correctly causing the data to be unable to use

3. Feature Extraction

- Explanation of the feature extraction process
    - Shuffle the data

    - Split the dataset
    - Normalizing the dataset
    - Apply PCA
- Discussion of any feature engineering techniques use:
    - Shuffling the Data:
        - Reason: By shuffling data, we ensures that the model sees a diverse set of examples during training, preventing it from learning patterns specific to the order of the data.
    - Splitting the Dataset:
        - Reason: By dividing the dataset into training and validation sets, we can train the model on one portion and validate its performance on another. This helps assess how well the model generalizes to unseen data, enabling better evaluation and tuning.
    - Normalizing the Dataset:
        - Reason: We had to ensure that all input features have similar scales. In the context of image data, normalization often involves scaling pixel values to a specific range (e.g., [0, 1] or [-1, 1]). This can improve the convergence of optimization algorithms and make the model less sensitive to the scale of input features.
    - Applying PCA (Principal Component Analysis):
        - Reason: We chose PCA because PCA is a technique for dimensionality reduction which in the context of image data, PCA helps in capturing the most important information while reducing the number of features. This can lead to a more computationally efficient model, faster training times, and a reduction in the risk of overfitting, especially if the original data has a high degree of redundancy.

4. Model Training

- Explanation of the model selection process
    - SVM Model Classifier: gain better performance than other classifiers such as Random Forest.
    - Works well in high-dimensional spaces, which is beneficial when working with image data or data with a large number of features
    - Less prone to overfitting in high-dimensional spaces
- Description of the model architecture and its parameters
    - C = 1
    - gamma = 0.01
    - kernel = ‘rbf’
    - class_weight = ‘balanced’
- Discussion of any algorithms or techniques used for model training
    - Data Augmentation
        - Apply the skimage library to transform the images into different variations
        - Random Rotation
        - Random Horizontal flip with 50% probability
        - Random Zoom
        - Random Sheared
        - Random Contrast
        - Random Brightness factor
        - Random Gaussian blur
    - SMOTE
        - Reason why using SMOTE

        - Imbalance data: have imbalanced class distribution, meaning some classes have fewer example than others. This led to model biased toward the majority class.
        - Improved generalization: the model becomes more capable of recognizing and distinguish minority class instances
        - Effects on Model Accuracy: increase the number of instances of the minority class, making the dataset more balanced, allowing the model to predict much more accurately
    - Applying standard scaling
        - This is a method for standardizing features by removing the mean and scaling to unit variance
        - Ensures all features contributed equally to the model, preventing features with larger scales from dominating the leading process.

5. Conclusions

During our project, our model performed well and was able to predict 491 out of 756 data point during our testing, resulting in an overall accuracy of approximately 70%. However, upon further analysis of the model's performance, we discovered that the precision was 73%, recall was 71%, and f1-score was 71%.

Moreover, we gained insights from the data and model, which helped us improve our process. We found that out of the initial dataset, only 61 individual data points contained enough photos that we could use. This meant that we had to use augmentation to increase the dataset. However, we had to be careful as overfitting could occur if we were not cautious. Overall, these insights helped us to identify the flaws in our process and improve our model's performance.

Based on the analysis, it is recommended to improve the process by first splitting the data and then performing pre-processing to isolate the test and train datasets. Additionally, more knowledge and familiarity with the systems may help in setting and fine-tuning the augmentation. These improvements can lead to better outcomes in future work.