**Title of the project**

To select the best Facial Recognition algorithm between Fisherface, Eigenface and LBPH algorithms and use it to detect different human emotions or expressions.

**Problem Statement**

To successfully detect and recognize faces of individuals based on various photograph samples of each individual person, and to experimentally prove that the Eigenface algorithm works better as compared to the Fisheface algorithm by comparing the ROC curve(Receiver Operating Characteristics), when using the current training set.

The best Facial Recognition algorithm is then used to detect various human emotions and expressions.

**Summary**

Facial Recognition is one of the most secure modes of authorization which uses samples of photographs or videos to authorize a person's identity. But Facial Recognition has its own weaknesses or drawbacks owing to various criteria like difference in number of image samples, clarity, brightness of the photograph, etc and hence a lower accuracy rate compared to other modes of authentication like Fingerprint Recognition, which has its own uniqueness and greater performance.

The aim of this project work is to select the best algorithm which provides the best performance and greater accuracy and use it to verify people by comparing the facial recognition result with our database.

We implement the ROC (Receiver Operating Characteristics), namely FMR(false match rate) where an unknown face or an imposter is wrongly recognized, and FNMR (false non-match rate) when a genuine person is not recognized and use these Receiver Operating Characteristics to find out the better algorithm between the Eigenface and Fisherface algorithms.

We find the results obtained by Eigenfaces algorithm and plot the values of FMR and FNMR which uses PCA( Principal Component Analysis, where we find a subset of uncorrelated variables that holds most of the information and forms the basis of Eigenfaces algorithm which utilizes Eigenvectors for Face Recognition. This prevents the analysis of each and every pixel and thereby reduces the time consumption and unnecessary work. We also find the results obtained by Fisherface algorithm and plot values of FMR and FNMR.

**Workflow**

1. **Creating the Dataset**

1.Created a converter program which converts any image format into a jpeg format which is easier to work with within OpenCV.

2.Collected nearly 300 samples of 15 people in different illuminations and poses from the Extended Yale Face Database.

3.Subdivided this 300 samples into almost equal number of training and test data for each person.

4.Used Haar Cascades to detect the face of a person.

5. Ran the Haar Cascade face detector across all the training images and created a dataset of the training images which consists of only the facial region (our Region of Interest) and saved all the training images in the ‘dataset’ folder.

6. The training images in the dataset folder are of the form User.ID.SAMPLENUMBER,

Where,

ID= used to uniquely identify a person and is used during recognition

SAMPLENUMBER = denotes the index of the image which belongs to a particular person.

**2. Training the Recognizer**

1. We used EigenFace and FisherFace recognizers provided by OpenCV to train all the training images present within the dataset folder.

2.We first retrieve all the images present in the dataset folder and convert it into a grayscale image.

3.We then convert each of these images into a numpy array and store this numpy array within an array.

4.We then obtain the ID of all the images and store it in an array.

5. We return both the ID array and Image array back to the calling function.

6.We then use the recognizer we previously created to train each image with its corresponding ID.

7.We store the training data within a folder called recognizer in a yml format for future use.

**3. Creating the detector**

1.We convert each of the images present within the test folder of each user into a jpeg format.

2.We choose a test image of any person we want.

3. We convert this chosen image into a grayscale image.

4. We run the Haar Cascade detector to detect the face of the person.

5.We then retrieve the Region of Interest of the Image .

6. We resize this image to match the image size of the training images.

7.We apply the Adaptive Thresholding technique on the selected test image.

8.We then use the recognizer to predict the ID of the image.

9.We then make suitable calculations.

10.We plot a graph of the observations using the Matplotlib library in Python.

**Preliminary Study:**

1.Brushed up the concepts of python object oriented programming.

2.Learnt to use third party python modules and integrate it within our project.

3.Searched and downloaded some online courses, tutorials and documentations on OpenCV, feature and face detection.

4. Downloaded the required Haar Cascade Classifiers required for feature detection.

5.Downloaded relevant e-books to assist in theory related to the program written.

6. Downloaded and summarized various IEEE transaction papers related to PCA,LDA and LBPH and learnt how each of the different Facial Recognition method work.

7. Searched and downloaded a comprehensive Facial Database called Yale Face Database of more than 30 people with 50 samples each in different lighting conditions and expressions.

8. Converted all the images from .pgm format to .jpg format.

9. Collected a set of 250 images and divided it into training data and test data images.

9.Identified the Region of Interest in each of the training images using Haar Cascade Classifiers and saved it in a separate Directory.

**Literature Survey**

IEEE papers referred to:

1. **Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System.**

Related Work:

A biometric based attendance system was created by making use of Receiver Operating characteristics (ROC) as the main case study. Purpose of this research was to get the bets Facial Recognition Algorithm between Eigenface and Fisherface Facial Recognition algorithms.

This study showed that for the currently available training dataset, Eigenface algorithm performed better than the Fisherface algorithm. The Eigenface showed an accuracy of 70%-90% when used with the Attendance System.

The result of this study was used in conjunction with other multi-modal biometrics.

1. **Face Recognition Using Eigenface with Naive Bayes**

Related Work:

This paper aims to remove the main drawback associated with Eigenface recognition algorithm, that is, lack of good accuracy.

Naïve Bayes classifier was used on the result of Eigenface feature extraction to predict the face. A normalized z-score was also added to the returned confidence in order to improve the accuracy by a large margin.

This study showed that, prediction with normal Eigenface returned an accuracy of 70%, whereas prediction with z-score of Naïve Bayes classification returned an accuracy of 89.5%.

1. **Real-Time Eye State Detection System Using Haar Cascade Classifier and Circular Hough Transform**

Related Work:

This paper aims to explain the system of eye state detection and to comprehensively distinguish different eye states (identify whether eyes are opened or closed).

The focal point of this research is to make use of Haar Cascade Classifiers to first detect the person’s face and then go on to detect the eyes. This system ensures that the eyes are detected properly and its state is recognized.

This method showcased an accuracy of 95.86% on the current training set which consisted of image of both opened and closed eyes.

1. **Accurate Wild Animal Recognition Using PCA, LDA and LBPH**

Related work:

The main aim of this study is to showcase the performance of the well-known face recognition algorithms which include Linear Discriminant Analysis(LDA) , Principal Component Analysis (PCA) and Local Binary Pattern Histogram (LBPH) on a dataset which consists of high-quality images of wild animals.

This paper throws light on the advantages and drawbacks of these popular algorithms by a comparative study which takes into account the accuracy, time taken and efficiency by making use of two different sets of experiments.

This study showed that PCA performed better than LDA and LBPH for a sufficiently large training set, whereas LBPH performed reasonably better than PCA and LDA for comparatively smaller training set.

1. **Face Detection and Facial Expression Recognition System**

Related Work:

This paper demonstrates the construction and working of an Automatic Face and Expression Recognition System.

There are 3 proposed stages:

1. Face Detection: Here the users face is detected and computations on skin color is made using YCbCr technique and lighting compensation is calculated.
2. Feature Extraction: The result of above detection is used in order to extract relevant features of the face like eyes, nose and mouth using Active Appearance Model (AAM).
3. Facial Expression Recogniton: Uses simple Euclidean distance method to find out the expression of the user. The Euclidean distance between different feature points like opening of lips is taken into consideration and this Euclidean distance is compared with the training and test images to predict the facial expression.

This model returned an accuracy of 90 to 95%.

1. **A Survey on Comparison of Face Recognition Algorithms**

Related Work:

This paper demonstrates the working of different Face Detection and Recognition Algorithms like Fisherface, Eigenface and LBPH algorithms in different hardware environments like the Intel processors and ARM processors.

This study took into account various categories like RAM, clock speed and computing power of the computer.

This study showed that the Fisherface algorithm was faster compared to other algorithms whereas the LBPH fared better in terms of accuracy but consumed more resources from the processor.

1. **Automatic Facial Expression Recognition Using Features of Salient Facial Patches**

Related Work:

This paper shows that extraction of discriminative features from salient facial patches plays a pivotal role in effective facial expression recognition. The accurate detection of facial landmarks improves the positioning of the salient patches on facial images. A novel framework is proposed for expression recognition by using important features of selected facial patches. A few noticeable facial patches depending on the position of facial landmarks are extracted which are active during an emotion. These active patches are further processed to obtain the salient patches which contain discriminative features for classification of each pair of expressions. One-on-one classification method is adopted using these features.

In addition, an automated learning-free facial landmark detection technique has been shown which achieves almost similar performances as that of other landmark detection methods and also requires comparatively less execution time. The proposed method is found to perform consistently in different image resolutions and therefore, provides a comprehensive solution for expression recognition in low resolution or bad quality images.

**Functional Requirements:**

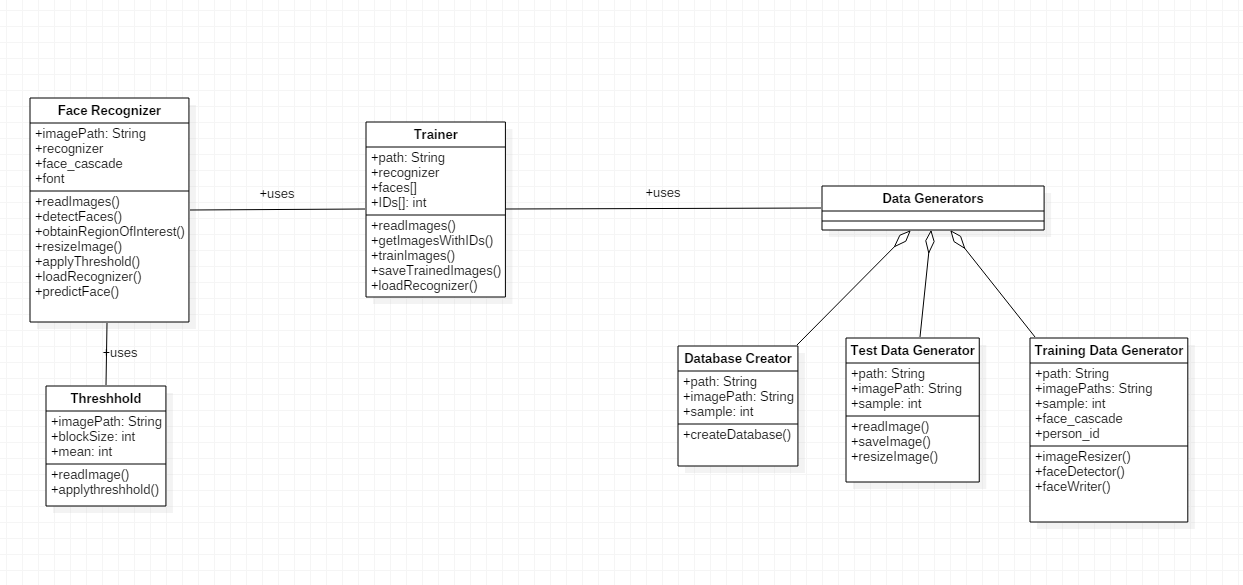
1. The program should be able to accurately detect the face of the person.
2. The program should be able to properly read and process the training data.
3. The program should be able to properly load and implement Eigenface, Fisherface and Local Binary Pattern Histogram.
4. The program should be able to generate the training data file corresponding to the algorithm and properly store the correlation between Facial Images Array and its corresponding IDs array.
5. The program should be able to efficiently store the training data images in the required format, resolution and naming convention ( User.person ID.Sample Number).
6. The program should be able to efficiently apply Binary Thresholding and Adaptive Gaussian Thresholding so as to accommodate images of various light intensities.
7. The program should be able to recognize the faces of different individuals based on Test Data images with some confidence.
8. The program should be able to detect and recognize various human emotions such as :
9. Anger
10. Sadness
11. Crying
12. Disgust
13. Happiness

**Non-functional requirements**

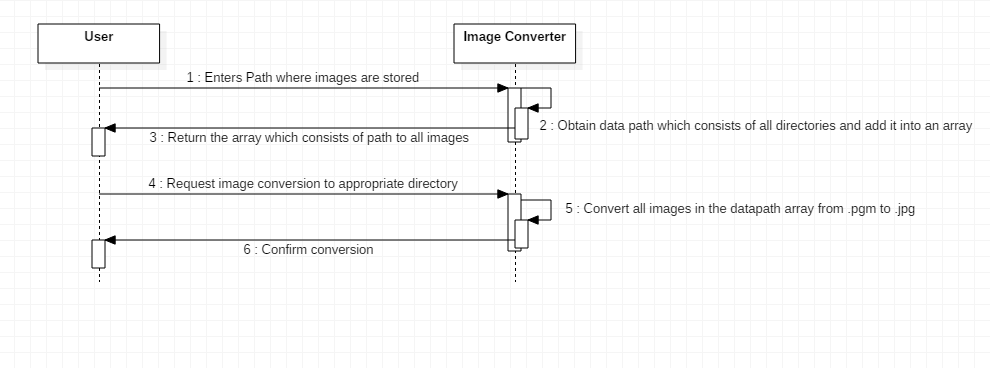
1. The program should be able to make use of Haar Cascades efficiently to detect only the faces of the person and not any other facial organ.
2. The program should be able to handle sufficiently large number of images for both training and test images.
3. The program should be able to run smoothly during the dataset creation process and output proper data.
4. The image size, dimensions, aspect ratios and resolutions need to be same for both training images as well as test images.
5. The Trainer needs to be able to properly map each of the Training Image faces with the appropriate IDs and store it in a yml file.
6. The Recognizer needs to be able to handle very large trained data in the form of yml files and properly predict the person ID for the input Test Image.
7. There needs to be at least 10 different training image for 10 different persons with difference in:
8. Lighting
9. Distance from Camera
10. Face angle
11. There needs to be at least 10 different training examples for each of the above mentioned human emotions and expressions for each person.

**High Level Design**

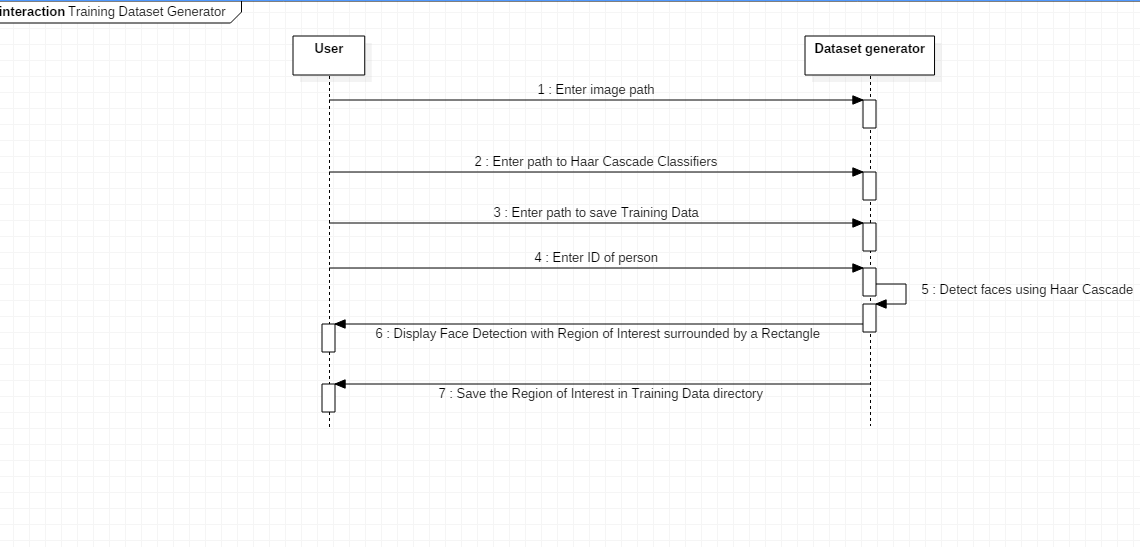
**Class Diagram:**



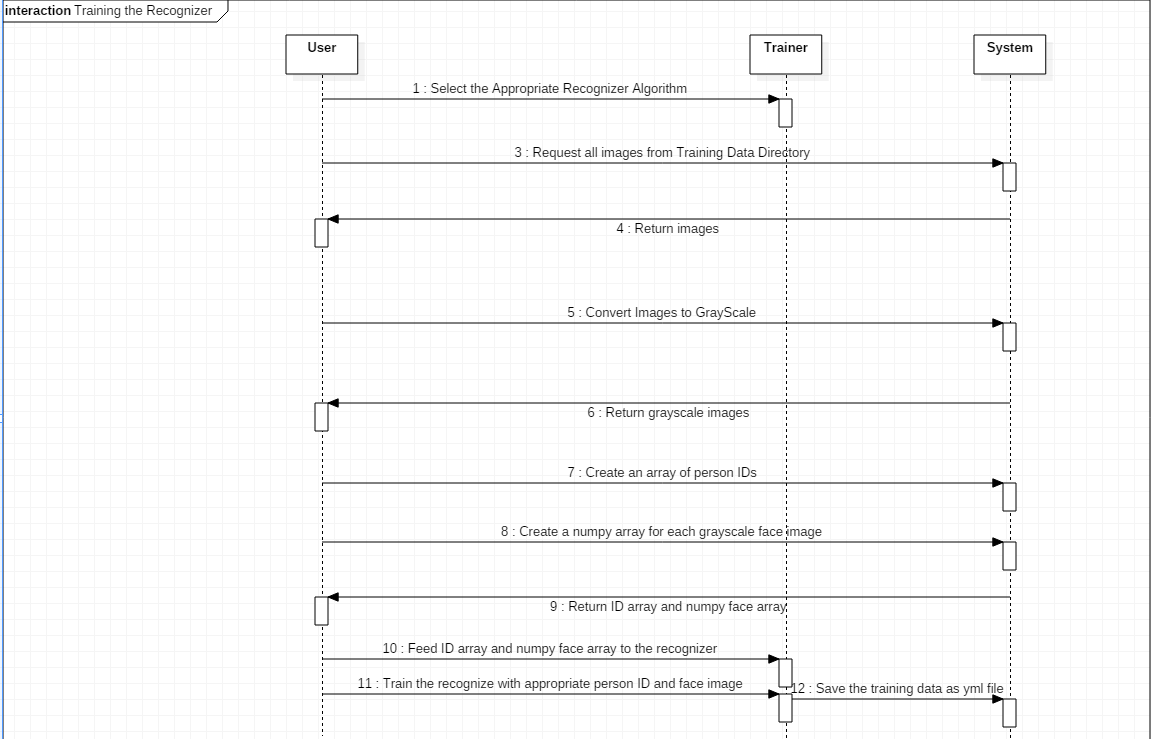
**Sequence Diagram for Image Converter:**

****

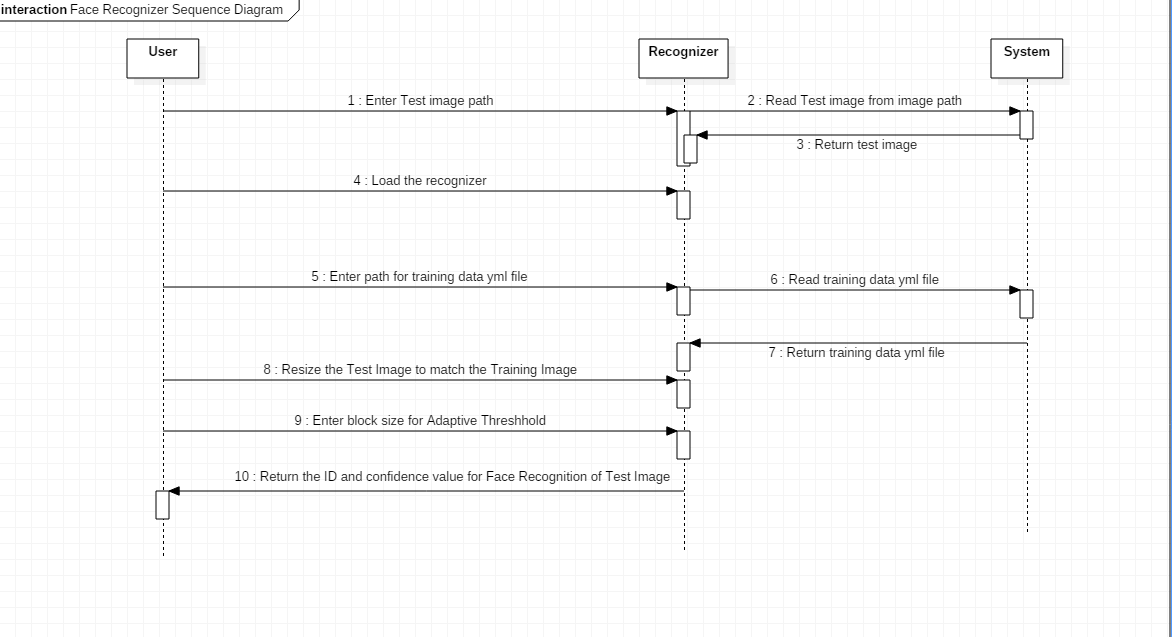
**2.Sequence Diagram for Dataset Generator**

****

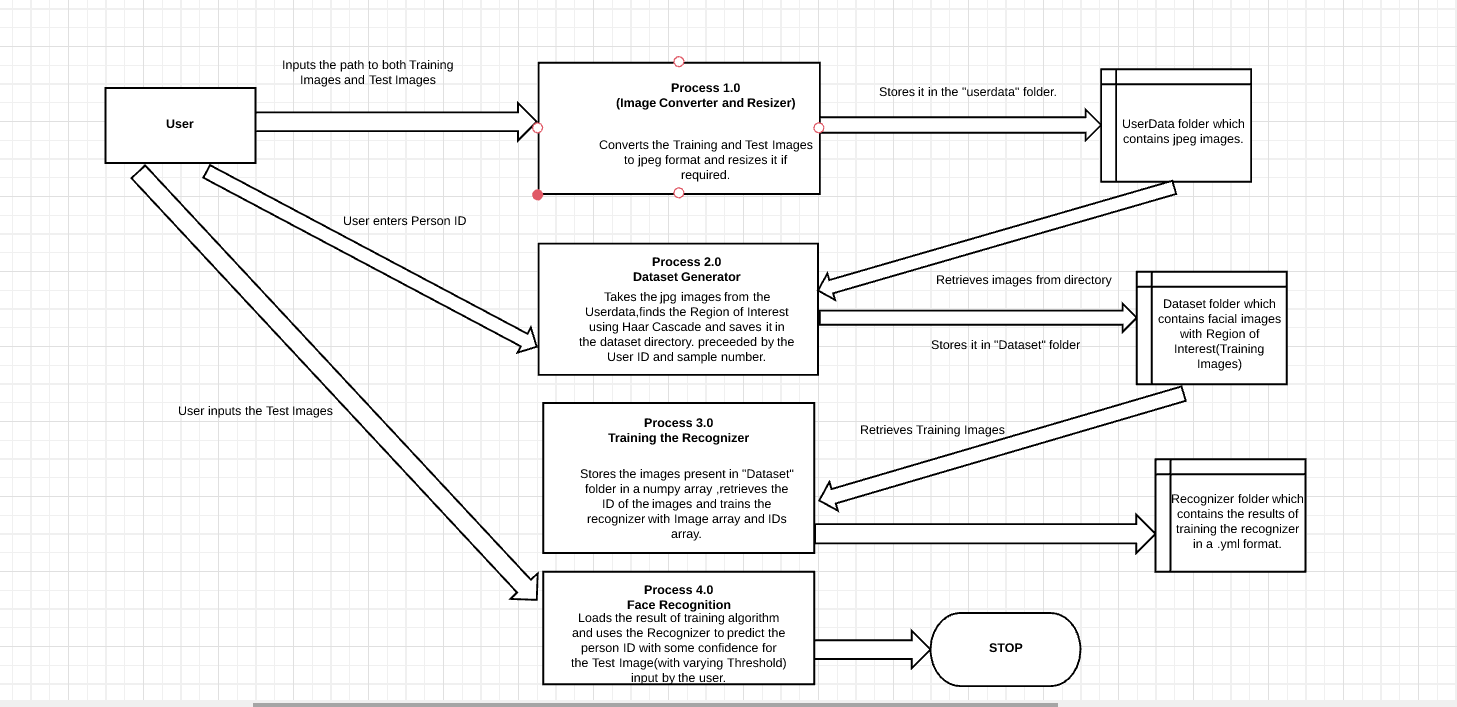
**3.Sequence Diagram for Training the Recognizer**

****

**4,Sequence Diagram for Facial Recognition**

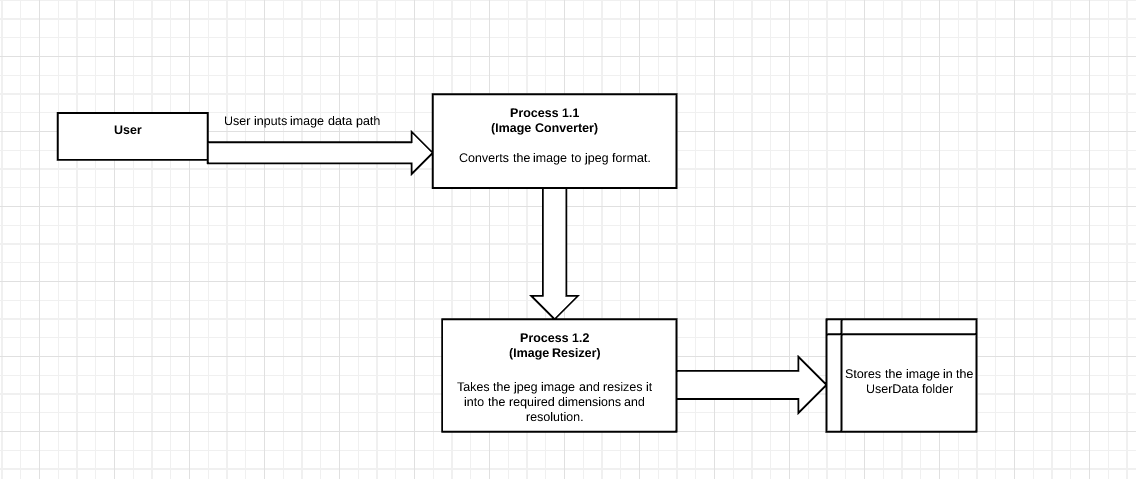
****

**Dataflow diagram Level 0:**

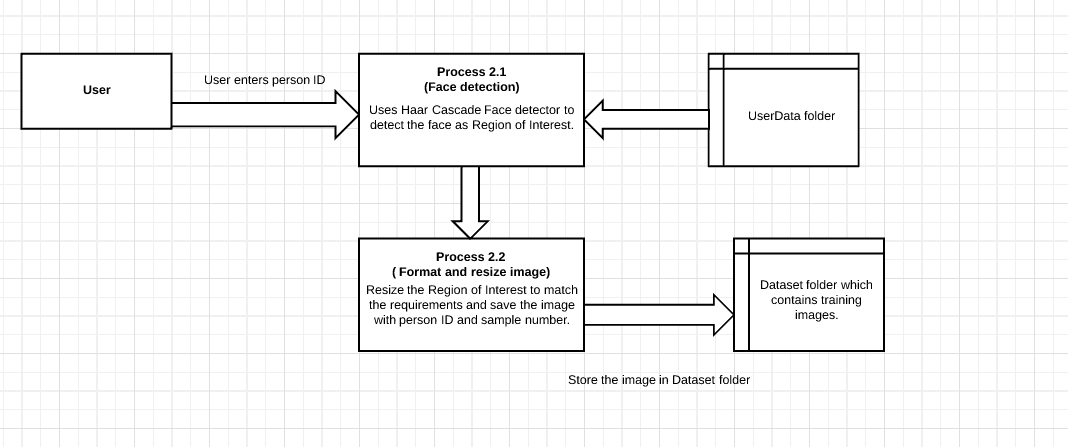
****

**Dataflow Diagram Level 1:**

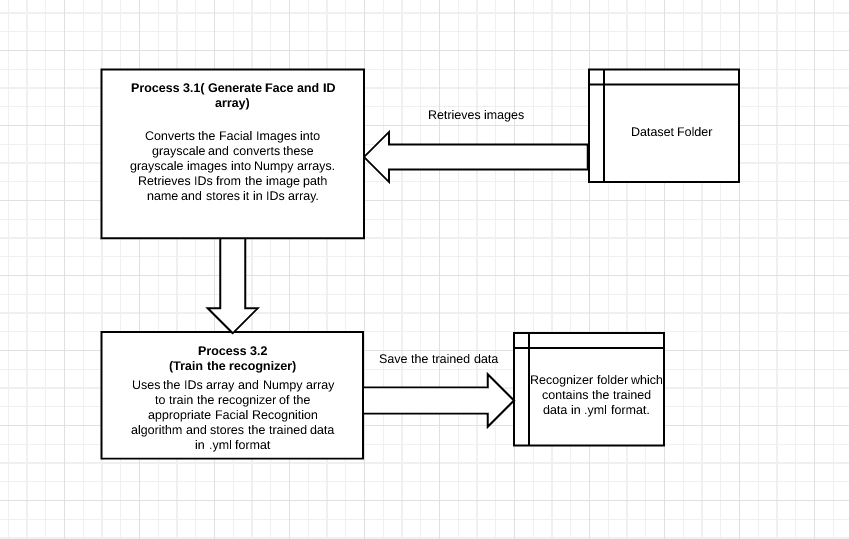
**Dataflow diagram for Image Processor:**

****

**Dataflow Diagram for Dataset Generator:**

****

**Dataflow Diagram for Training the Recognizer:**

****

**References:**

1. A. R. S. Siswanto, A. S. Nugroho and M. Galinium, "Implementation of face recognition algorithm for biometrics based time attendance system," 2014 International Conference on ICT For Smart Society (ICISS), Bandung, 2014, pp. 149-154.
2. E. B. Putranto, P. A. Situmorang and A. S. Girsang, "Face recognition using eigenface with naive Bayes," 2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS), Yogyakarta, 2016, pp. 1-4.
3. N. L. Fitriyani, C. K. Yang and M. Syafrudin, "Real-time eye state detection system using haar cascade classifier and circular hough transform," 2016 IEEE 5th Global Conference on Consumer Electronics, Kyoto, 2016, pp. 1-3.  
   doi: 10.1109/GCCE.2016.7800424
4. P. Kamencay, T. Trnovszky, M. Benco, R. Hudec, P. Sykora and A. Satnik, "Accurate wild animal recognition using PCA, LDA and LBPH," 2016 ELEKTRO, Strbske Pleso, 2016, pp. 62-67.  
   doi: 10.1109/ELEKTRO.2016.7512036
5. A. S. Dhavalikar and R. K. Kulkarni, "Face detection and facial expression recognition system," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2014, pp. 1-7.  
   doi: 10.1109/ECS.2014.6892834
6. A. Özdil and M. M. Özbilen, "A survey on comparison of face recognition algorithms," 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), Astana, 2014, pp. 1-3.  
   doi: 10.1109/ICAICT.2014.7035956
7. S. L. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," in IEEE Transactions on Affective Computing, vol. 6, no. 1, pp. 1-12, Jan.-March 1 2015.  
   doi: 10.1109/TAFFC.2014.2386334