FACE RECOGNITION VIA DEEP LEARNING

Jean Xavier, Tanmay Vikas Khachane

5567828,5570980

I. ABSTRACT

The aim of this project is to create a Face Recognition system via through deep learning methods that will help in recognizing the face of users based on the image dataset that is provided to it. The system will first be trained with a set of images which will be used as the default classifier for comparison.

The projects aim to create a system of high efficiency and accuracy, where an evaluation on the face properties such as the facial features, shape of the face would be applied through image processing technique. The software that will be used to test and simulate this system is MATLAB. Some of the techniques that will be used for training the classifier are the HOG features, ECOC classifier etc.

II. INTRODUCTION

Face Recognition is one of the many wonders that AI research has brought forward to the world. The face recognition is a biometric technology that allows identifying humans through digital image or video frame. The applications range from checking identies at international borders and searching mugshots in national crime databases to tagging faces in photos on social media websites. [1]

The performance of face recognition systems depends on the conditions under which the face images are taken. The biometric systems use distinguishing facial features like eyes, mouth, forehead and nose other spatial geometry features for discriminating one user from the other. [2] A typical face recognition system broadly follows the same procedures which are mainly

1) Loading the Image Files on the Database
2) Detection and Extraction of the Image
3) Feature Extraction and Representation
4) Classifier training
5) Face Recognition

a. LOAD THE IMAGE INFORMATION

The system begins by first collecting the images of different people and stores them onto the database of the system. Each individual on the database will have at least 7 different unique images of themselves. The quality of the images is not mattered for the system to detect the faces. Also with the increased data consumption of large quality images only medium set quality of images will only be added to the database.

The *imageSet* which is an in-built MATLAB function will used for grouping the images. The function returns an object for storing an image data set or a collection of image datasets. For this system the collected dataset will contain the faces of different people. The object contains image descriptions, location of the images and the number of images that are collected. The system will contain a face set of 110 people which will be fetched using the Image set function.

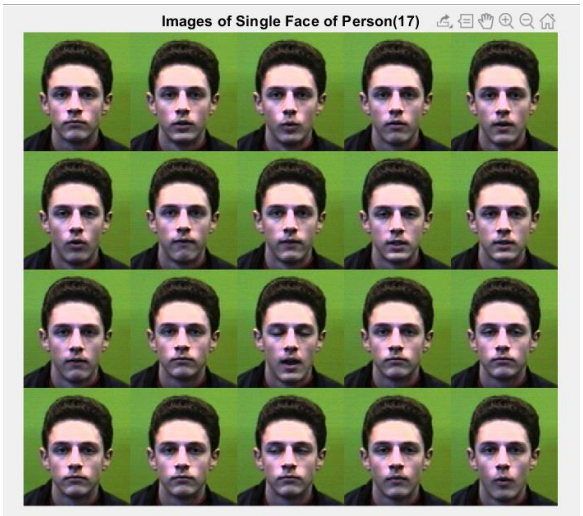

Figure 1 Image collection for a single person in the dataset

b. DETECTION AND EXTRACTION OF THE IMAGE

Face detection is an important step in face recognition that corresponds the localization of the face in a given image. Once the face is detected it is cropped for recognition or a rectangle box will be overlayered on the identified persons image. Although there exist several methods for face detection the performance was satisfactory until the method of Viola Jones. [2] The images below show the detected faces using the Viola Jones algorithm.

Figure 2 Face Detection using the Viola Jones Algorithm

However, through various test and research that were conducting while creating the Face recognition system it was proven that another face recognition method namely known as the HOG (Histogram of Oriented Gradients) face recognition had

given output increasingly higher accuracy than the Viola Jones Algorithm. Hence the feature extraction method that was selected to be used was set to the HOG method.

The HOGs are a feature descriptor that been successfully used for object and pedestrian detection, represented as a single value vector as opposed to a set of feature vectors where each represents a region of the image. The results of the image shown illustrates the output result when the HOG feature extraction method was used.

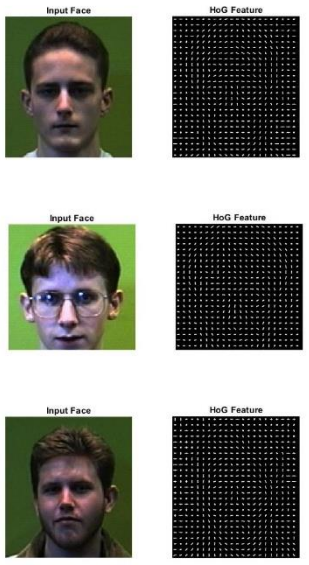

Figure 3 HOG Features extracted from the images

The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to the that of the edge orientation histograms, scale- invariant feature transformation but differs is it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. [3]

c. CLASSIFIER TRAINING

The HOG method is used for the feature extraction and the ECOC is used for classification. The ECOC (Error Correcting Output codes) basically converts multi-class classification problem into a binary classification problem with the help of various coding schemes accompanied by a linear like Support Vector Machine (SVM). The ECOC considers the problems of classification as a communication, ECOC address the multi class problem by dividing into several binary problems. [4]

In the Face recognition system the ECOC classifier is trained with the extracted features from the HOG. Given any input image the ECOC classifier works by extracting the features of the input image and then feeding those to all the binary learner. The Binary Learners using linear SVM, where one class is taken as positive and others are taken as negative and are separated using a hype-plane. Once the binary learner is trained, each learner produces a probability .The probability defines the output results from the classifier.

To use the ECOC classifier on MATLAB we use the function *fitecoc* that takes the input parameters of the training features and also the Response variable names. As seen on the code snippet below we create the *faceClassifier* variable that calls the fitecoc function with the training features extracted from the HOG method and the training label values which refers to identity of the person recognized.

```
faceClassifier = parfeval(fitcecoc(trainingFeatures,trainingLabel));
j = createJob(faceClassifier);

submit(j);
wait(j);
out = fetchOutputs(j);
```

d. MATLAB PARALLEL TOOLBOX

Since classification of large amounts of image data using a single core CPU unit would consume a lot of time. In order to conflict this a Parallel Toolbox on MATLAB will be integrated within the system to run the algorithm. The Parallel Computing Toolbox on MATLAB lets to solve computational and data-intensive problems using multicore processors, GPUs and computer clusters. High Level constructs such has parallel for-loops, parallelized numerical algorithms enable to parallelize MATLAB applications without CUDA or MPI programming. The CUDA is another parallel computing platform and application program interface created by Nvidia. It enables developers to perform asynchronous tasks.

III. CONFIGURATIONS

The face recognition system first requires the face set images of different people that needs to be used for further extractions. We use the *imageSet* function to group all the images from the face set into a single object.

```
% Takes image information from the Folder "Faces Data" groups all the
% images into one single object
Face_set_Database = imageSet('FacesData','recursive')
```

Once all the images are retrieved and saved on to the object, we split this data into sets. The first set will be used as the training set and the second set will be used for tests. The training and test set will be separated using the *partition* function on MATLAB.

```
%% We split the Facedatabase into two sets the Training Sets and Test Set
[trainingSet, testSet] = partition(Face_set_Database,[0.8 0.2]);
```

The next part of the algorithm of the code would be to extract the HOG features from the images. The HOG features will be extracted only from the training set images which we partitioned earlier. Inside the step of the extraction method we create a for loop that runs for the iterations for the number of images the training set contains. On each iteration the features are being extracted using the *extractHOGFeatures* function during each iteration the *featureCount* variable gets updated by 1. Another variable

personIndex is created that stores the facial feature count and the label of each individual person. Since they were 110 people's face set that were saved to the Face database. The *personIndex* will contain around 110 rows representing the values of each individual person.

```
%% To extract the HOG features from the Trainig set images
% and update the feature count and label names

training_Features = zeros(size(trainingSet,2)*trainingSet(1).Count,18144);
featureCount =1;
for i= 1:size(trainingSet,2)
    for j = 1:trainingSet(i).Count
        training_Features(featureCount,:) = extractHOGFeatures(read(trainingSet(i),j));
        trainingLabel(featureCount) = trainingSet(i).Description;
        featureCount = featureCount +1;
    end
    personIndex(i) = trainingSet(i).Description;
end
```

Figure 4 Extract HOG features from the Training set

Workspace

Name	Value
Face_set_Database	*1x111 imageSet*
featureCount	1793
i	111
j	16
personIndex	*1x111 cell*
testSet	*1x111 imageSet*
training_Features	*1792x18144 dou...*
trainingLabel	*1x1792 cell*
trainingSet	*1x111 imageSet*

From the figure illustrated above the following observations are made from values seen on the workspace of MATLAB. In total there were 110 images that were uploaded to the Face database. There were a total of 1792 observations made and out of this 18144 dimensionality labels were created for the training features.

The next step is to upload the images to the ECOC classifier has mentioned previously. To use the ECOC classifier we use *fitecoc* function which the input of the training features which we created previously, and the training label values. The execution part of this instance of the code consumes a lot of time to increase the efficiency of the system and to reduce the time taken for creating the classifier. We use the *parfeval* function which is a part of the Parallel Toolbox that helps in reducing the time taken for creating the Face Classifier.

```
faceClassifier = parfeval(fitcecoc(trainingFeatures,trainingLabel));
j = createJob(faceClassifier);

submit(j);
wait(j);
out = fetchOutputs(j);
```

A. FACE RECOGNITION SYSTEM BLOCK DIAGRAM

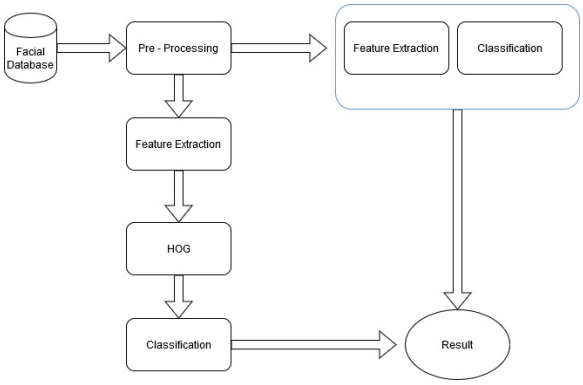

Figure 5: System Block Diagram

B. ALGORITHM FOR FACE RECOGNITION SYSTEM

1) Start
2) Load the image to the Face Dataset
3) Split the Face dataset into to two sets Training sets and Test tests
4) Using HOG feature extraction use the extraction method for Training sets
5) Training features and Training Label values are stored, and the feature count variables are updated
6) Execute the EOC classifier with the input parameters of the Training Features and the training Labels
7) Once the Classifier is successfully created it can then be tested
8) The query image can be used to identify the person from the Face Set gallery.
9) Results are shown in Figures MATLAB
10) End

IV. RESULTS

Once the classifier is successfully created we can then test the trained classifier by providing a query image. The query image can be any image that is included in the face dataset. The classifier compares the query image with all the trained images and displays the output picture of the identified person.

```
queryImage = read(testSet(person),1);
queryFeatures = extractHOGFeatures(queryImage);
personLabel = predict(faceClassifier,queryFeatures);
%Map back to training set to find identity
booleanIndex = strcmp(personLabel, personIndex);
integerIndex = find(booleanIndex);


subplot(1,2,1); imshow(queryImage); title('Query Face');
subplot(1,2,2); imshow(read(trainingSet(integerIndex),1)); title('Matched Class')
```

The prediction of the classifier is the *faceclassifier* variable the query image is an input image provided by the user. The query is also used for extraction using the HOG method. Once the extraction is done it then executes the *predict* function which takes the input parameter of the trained classifier which in this case is *faceClassifier* and the extracted features from the query image *queryFeatures.* If the query image matches to any of images on the database it shows the output of the identified image.

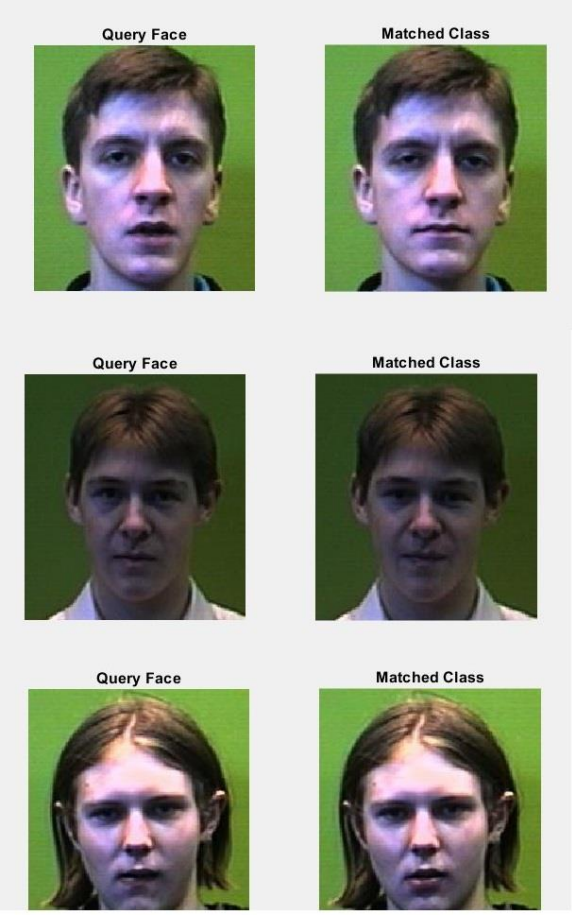

Figure 6 Matched face from the Classifier

The above images show the faces that have been recognized from the Trained Classifier that we created. The figure only shows the 3 identified persons but more many input images can be queried to search on the face database.

V. LIMITATION

One of the few limitation of the facial recognition are dealing with visual spectrum of these images many challenged such as illumination, pose, facial expressions and facial disguises play a big role in providing a false result from the classifier.

VI. CONCLUSION

The main objective of this project is detection of Faces using Image Processing Software. This detection is achieved successfully by using well integrated techniques such as HOG Image extraction , Viola Jones extract and finally creating an image classifier using the EOC method. The algorithm developed can be used by neural network programs to detect even larger number of images of faces using image processing and further developed to detects and faces as well as object detection.

VII. REFERENCES

[1] G.H.GivensaJ.R.BeveridgebP.J.PhillipscB.DraperbY. M.LuibD.Bolmed, "Introduction to face recognition and evaluation of algorithm performance," in *Computational Statistics & Data Analysis*, 2013.

[2] P. S. G. K. UmaraniJayaramana, "Recent development in face recognition," India, 2020.

[3] *. R. A. A. C Rahmad1, "Comparison of Viola-Jones Haar Cascade Classifier and Histogram of," in *IOP Conference Series: Materials Science and Engineering*.