INM460 Coursework : Face Recognition

Abin Abraham

1. Introduction

Face recognition is the process by which a computer identifies an individual from a digital image or video. It has massive number of applications across domains like security and surveillance as well as social media. Even in the recent COVID-19 epidemic, face recognition technology has been extensively used in high precision real time surveillance systems to identify and trace people who are should be under self-quarantine. The global face recognition market is expected to grow to USD 10 billion by 2025 [1].

Through this coursework, we have developed a face recognition system using MATLAB to detect and identify students, using a database of known face images and videos of the cohort in the computer vision elective. This is an object detection i.e. classification and localization problem. Overall, there are two tasks - Face Recognition and Creative alteration of faces, which is delivered via one single function RecogniseFace. RecogniseFace will take the image – group or individual, feature extractor type, classifier type and creative mode as inputs and return a matrix with all identified faces with their x and y coordinates of the middle point of the detected faces. In the first section we give an overview of the methods used in the implementation which is described in the second section. The third section deals with the analysis and discussion of results.

2. Dataset

The data was downloaded from Moodle. The original data source consists of 2 datasets:

- Individuals - Portrait images and videos of 48 individuals holding an A4 sheet with a label between 1-78 where each student is assigned a unique label.
- Group - Landscape images of different groups of students from the class.

There are students in the group images who were not part of the individual dataset. This implies that there will be individuals in a group image who have not been assigned a unique label as part of the first dataset. There are also students in the individual images who were not part of the group images. All images are in .jpg format and videos are in .mp4 format were captured using an iPhone. All individual images have been taken in a similar setting with student in front of a white background and have good lighting. All group images were captured at class in a wide room. There are huge variations among the photo angle, student posture and lighting conditions as students are in different places in the class.

3. Overview of methods

In this section we will briefly describe the methods used for the implementation along with their key algorithmic components. The best performing methods HOG-SVM and 'Pre-Trained CNN' has more details.

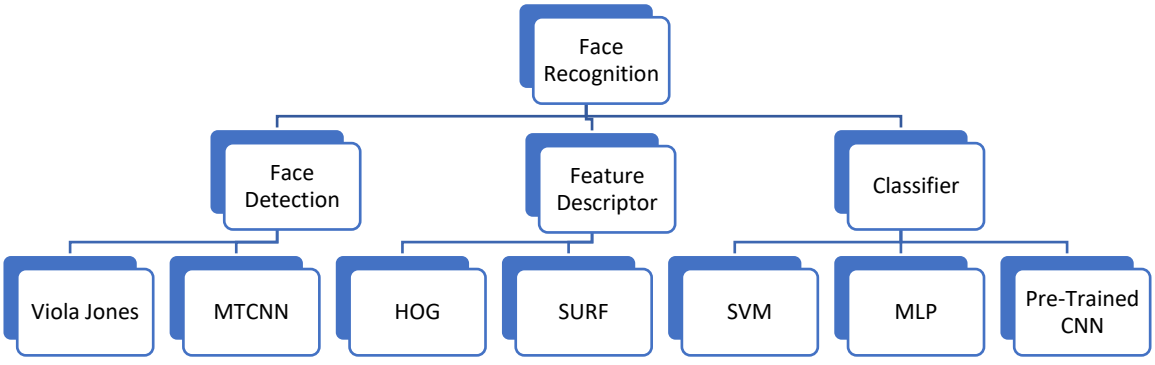


Figure 1 Overview of Methods

3.1 Face Detection

We will experiment with two state of the art approaches - Viola Jones object detector and Multi Tasked Cascaded CNN (MTCNN) for face detection. Viola Jones object detector [2] is widely used for object detection and implements Haar like features which are used to detect face features – edge features, line features and four sides features on an integral image. Features like eyebrows and nose are key features which are looked for. Later an integral image is created from the haar like features. Adaboost is used to select the best features and subsequently a cascaded ensemble of weak classifiers is used for classification of being a face or not. It has been widely used in real time face detection systems.

Multi Tasked Cascaded CNN [3] includes a three-stage multi-task deep convolutional networks. Firstly, candidate windows are produced through a shallow CNN called fast Proposal Network (P-Net). After that, refine these candidates are refined to remove non-faces in the next stage through a Refinement Network (R-Net). In the third stage, The Output Network (O-Net) produces final bounding box and facial landmarks position. Face alignment plays a key role in this process.

3.2 Feature Descriptors

Feature descriptor extraction is a key step in computer vision systems. The objective of this step is to identify and extract key features from the face which will be used for the model training and classification steps. There are multiple techniques like HOG, SIFT, SURF, ORB for key point detection and extraction. We will be using 2 methods in this coursework :

3.2.1 HOG

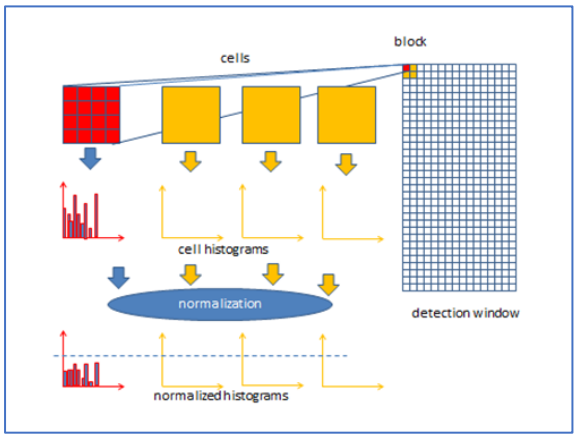


Figure 2 Histogram of Gradients [4]

Histogram of Oriented Gradients (HOG) extracts the intensity gradient (magnitude) and edge orientation(direction) for localized regions of an image and then generates a histogram for each of the regions. HOG is robust to rotation and scaling and gradient orientation is robust to change of absolute intensity values.

The processing steps are [4]:

i. An image is divided into blocks of 16X16 pixels where there is 50% overlap between each adjacent block.
ii. Each block is further divided into 8X8 pixel regions.
iii. Within each 8X8 pixel region gradient magnitude & gradient orientation is quantized.
iv. A histogram of oriented gradients is created into 9 bins.
v. Within each of the blocks, the histograms for the constituent 8X8 pixel regions are normalized so that lighting variation can be nullified. The histograms are joined to form a single matrix representing the histogram for the image block.
vi. The combined histograms for each block together form the HOG feature descriptors for the image.

3.2.2 SURF

Speeded up Robust Features (SURF) uses Haar wavelets responses locally around an interest-point and is known for fast computation. It has primarily two steps feature extraction and description. The first stage is the feature extraction. SURF is partly inspired by the scale-

invariant feature transform (SIFT) descriptor. We will use the bag of visual words for the feature description. The strongest features from each class are chosen to form the bag of features containing the vocabulary of visual words. The strongest features are chosen based on a predefined percentage parameter inputted to the system. These strongest features are then reduced through quantization of feature space using K-means clustering. For each image in the training data, SURF features are extracted and then quantized to the obtained K-means (the visual words). Then a histogram of visual word occurrences that represent that image is encoded. The histograms of the training data are used to train a classifier

3.3 Classification

Based on the features extracted by the feature descriptor the classifiers will classify each face into 1 among the classes between 1-78 based on the labels in the individual images. SVM and MLP are two classifiers which we will eb experimenting with. In addition, we will also use a pre-trained convolution neural network Alexnet. CNN employs automatic feature extraction and classification singularly and thus does not need a separate feature extraction step.

3.3.1 Support Vector Machine

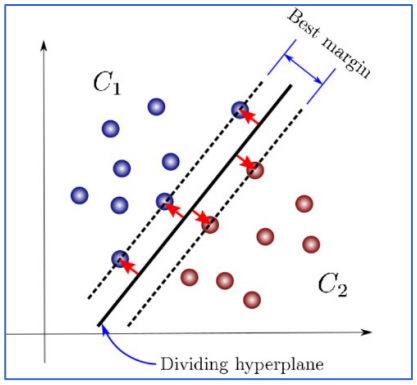


Figure 3: Multi-layer Perceptron (Source: Towards Data Science[13])

A support vector machine (SVM) is a supervised learning algorithm primarily used for binary classification tasks that determines an optimum hyperplane to split the data. The kernel is a mathematical transformation of the data into a higher dimension space so that the optimum hyperplane can be determined. This step can be computationally very expensive if not for the kernel trick which achieves this by using the dot products between the features. A linear kernel is used for linear classification and a nonlinear- polynomial or gaussian is used for nonlinear classification. Here C1 and C2 are the 2 binary classes. SVM tries to build a hyperplane with maximum margin that separates the two classes

3.3.2 Multi-Layer Perceptron

A multilayer perceptron(MLP) or artificial neural network is a supervised learning algorithm. It has an input layer, one or more hidden layers and an output layer. Each layer in the multilayer perceptron except the input layer, has weights, bias and a non-linear activation unit. The activation function maps the product of the weights and inputs and their sum with the biases, as the output of the layer, which is the subsequent input for the next layer. Depending on the activation layer at the output layer the MLP can be used for regression and classification.
The weights and biases are learned by a process known as back propagation Each run is called an epoch and the training process is continued for multiple epochs till the error converges to a minimum.

3.3.3 Transfer Learning using CNN

Deep learning combines both the feature extraction and classification phases. Convolutional neural networks are the most effective deep learning algorithm used for image classification. With transfer learning, instead of starting the learning process from scratch, we start from patterns that have been learned when solving a different problem and transfer the network to the new problem. Pre-trained models are used for transfer learning. For this coursework we will use AlexNet. from MATLAB trained on ImageNet [5]. Fine-tuning a network with transfer learning is usually much faster and easier than training a network with randomly initialized weights from scratch. AlexNet is a convolutional neural network that is 8 layers deep. The key algorithmic components of Alexnet in sequence are:CONV1, MAX POOL1, NORM1 ,CONV2, MAX POOL2, NORM2, CONV3, CONV4, CONV5, Max POOL3, FC6, FC7, FC8 [6] where

CONV refers to a convolution layer, POOL refers to a Overlapping Max Pooling Layer, NORM refers to local response normalization and FC refers to a Fully Connected (Dense) Layer.

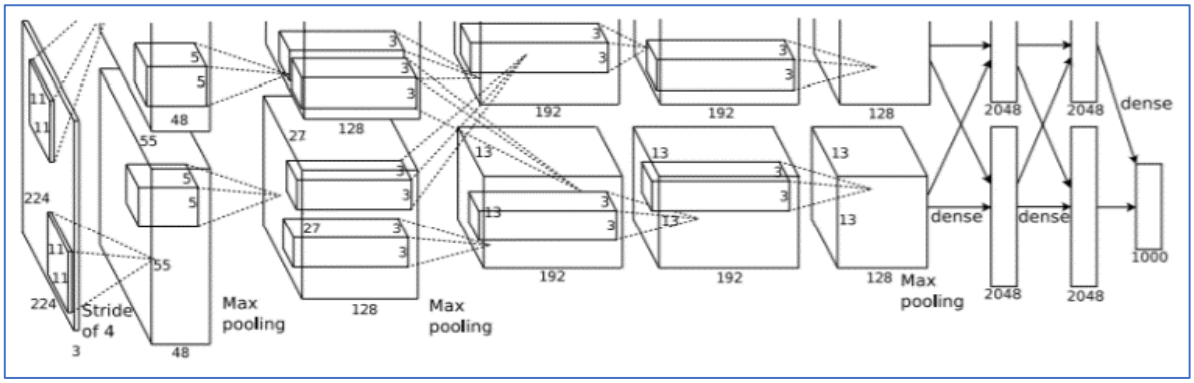


Figure 4: Alexnet architecture [13]

A key technique which was introduced in this architecture is the Rectified Linear Unit (ReLU) as an activation layer after each convolution and fully connected layer. Due to saturation (vanishing /exploding gradients problem in tanh function, the learning was slower. This is because, when the output of the sigmoid activation function is very close to 0 or 1, the gradient of these regions is almost 0, so that back propagation cannot continue to update some of the model parameters. Due to linear functionality in ReLU this problem is solved, and thus gradient descent is faster. Overlapping Max Pooling Layer has stride smaller the kernel size, thus there is overlap. Local response normalization encourages different feature maps to specialize by keeping only the most strongly activated neurons This process also prevent overfitting as per the authors [Source]. Thus Alexnet extracts features based on the spatial relationship via a sequence of convolutions and pooling layers and finally process them for classification with the fully connected layers.

3.4 Other Processing

3.4.1 Conversion to grayscale for Feature descriptors and Histogram Equalization

Many image processing and CV algorithms use grayscale images for input rather than colour images. This can be attributed to the separation of luminance, which is a more important to distinguish visual features from an image like edges, from chrominance which only adds more detail. Although features descriptors like HOG and SURF work on both greyscale and colour, it is more efficient to convert into greyscale images since it has only one channel compared to colour images. Features like brightness, contrast, edges, shapes and contours can be obtained without colour. Processing is also made much faster. CNN like Alexnet needs colour images and hence we cannot use greyscale images for it. Histogram equalization has been proven to improve face recognition accuracy due to improved face contrast [7].

3.4.2 Data Augmentation

Data augmentation is the process by which size and the quality of the training dataset is enhanced so that deep learning models can generalize the dataset better. It is a form of regularization to prevent overfitting. The classifier must ideally be invariant to wide variety of transformation like lumination, face orientation, etc. The training dataset may not represent all such cases and the data augmentation step will boost it. Images are high dimensional and thus transformations without corrupting the image can be easily simulated. The most common operations are blurring, translating the training images a few pixels in each direction , rotating the image by a small angle, horizontal flipping, altering the brightness. This will help to simulate conditions where the person is in different positions with tilted heads, poor lighting, blurred etc.

4. Implementation

Figure 4: Implementation Process Flow

4.1 Data Preparation

The data preparation steps can be summarized by the following steps. Detailed process flow diagram in (10. Appendix 1 – RecogniseFace Process Flow Diagram)

4.1.1 Folder creation and Image / Video Sorting

For supervised learning, it is important to have features and labels. We organized the images and videos into folders with their corresponding label names. Thus we structure our data directory with folder name 1 to 78 created using script. To classify initial training images to establish ground truth, we initially attempted to use an OCR method to identify the label based on the placard held by the student, but this was challenging and did not yield accurate results with junk read text or labels into being read specifically in case of images where the user held the placard slanted. Details are provided in the later section on OCR. After couple of attempts we had to resort to manual sorting. All images and videos from the original dataset were manually placed in the corresponding folder of their label. Similar exercise was done for the videos as well. Thus we have organized our original dataset in a format suitable for automated face detection, feature extraction and training which we will detail in the subsequent steps. Overall 40 images per person were extracted from images and video. This was done using automated scripts with final fine tuning manually where images which were too blurred were removed.

4.1.2 Face Detection and Extraction

We initially perform face extraction using the Viola Jones Object Detector FrontalFaceCART and FrontalFaceLBP varying the minimum size and threshold, but performance was sub optimal when this was used for final classification on test images. Later we switched to MTCNN MATLAB toolbox [9] as it is a more powerful face detector with ability to detect faces in different orientations. We iterate over each labelled folder created during folder creation and sorting and extract the faces from each image as well as 5-6 frames from each video. Faces were extracted from the training and validation images from both individual images and videos. Extracting faces from the training group images were done but labelling them was a cumbersome task as the labels had to be manually identified based on other images, which we only completed partially. Around 2-3 images per person was extracted from group image for training.

Folders for CNN classifier and Non-CNN Classifiers

As the pretrained models needed colour images, the extracted faces were copied to a folder for CNN classifier and resized to [227 227] to match the input size for Alexnet. For the non-CNN classifier which would have a Feature extraction process, these were converted to greyscale and size [80 80]

4.1.3 Data Augmentation

The following augmentations were performed on each set of images to have additional training data. Though there are a lot more options to augment, we restricted it to this so that the classifiers can better generalise and avoid learning noise.

1. Histogram equalization
2. Rotation (+10 degree and -10 degree)
3. Horizontal Flip
4. Blur
5. Resize to smallest image 50*50 (multiple experiments were run for resizing)

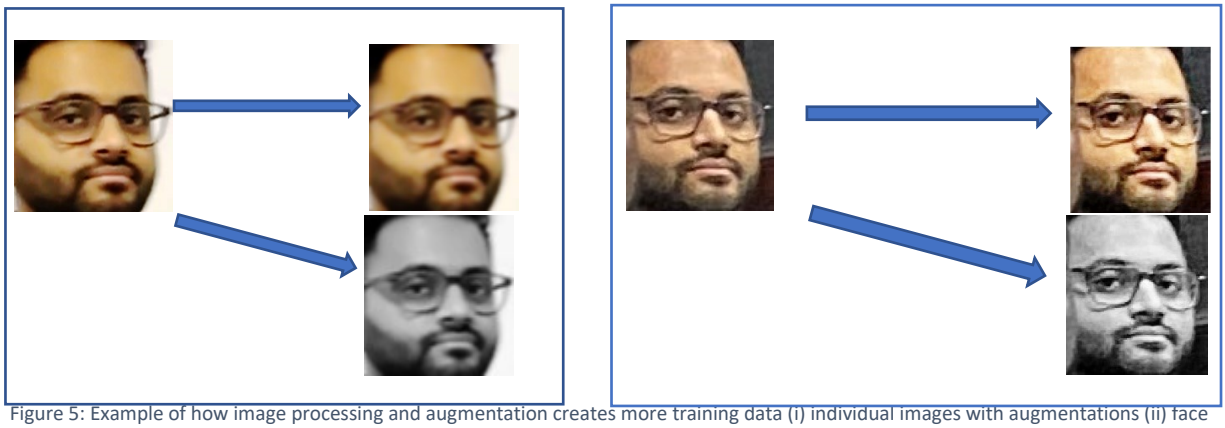


Figure 5: Example of how image processing and augmentation creates more training data (i) individual images with augmentations (ii) face extracted from group image after similar processing which will be done before testing

4.1.4 Data Split

The data is split into 80-20 proportion for training and validation using the MATLAB image datastore as we have a large dataset of more than 10000 images with around 210 images per person. Data is shuffled and randomized before splitting so that there is no bias. Care was taken to ensure that all the datasets are balanced so that there is no explicit bias for the classifier. Alternatively, a stratified split can be used.

4.2 Feature Extraction

4.2.1 HOG Extract

HOG features were extracted using extractHOGFeatures(I) function of MATLAB. The feature returns a 1-by-N vector, where N is the HOG feature length. Experiments were done changing the number of bins and cell size from the default values. We decreased the cell size [6 6] to improve small scale details. The number of bins were increased to 12 to encode more finer details. Each of these decisions were made after experimenting with different values and observing the overall validation accuracy.

4.2.2 SIFT using bag of visual words for each image

Codebook using K means clustering prepared using the bagOfFeatures(imds) function of MATLAB which extracts the SURF features The value of K which is used to quantize the features into visual vocabulary was selected by experimenting with values from 500-1000. The best validation accuracy was observed for K=790. Next step is to create the feature vectors using encode(bag,I) that represents a histogram of the visual word occurrence in an input image.

4.3 Face Classification

4.3.1 SVM

For SURF- SVM classifier, trained SVM classifier using trainImageCategoryClassifier combines the encoding and classification steps. For HOG- SVM, the features were encoded using encode functionality and later classifier was trained using fitcecoc. As the validation accuracy was relatively high 97%, did not further pursue hyperparameter tuning to prevent any overfitting possibility.

4.3.2 MLP

Trained MLP model using patternnet. Tuning of hyperparameters was performed for different network parameter – number of neurons per hidden layer and number of hidden layers in a grid search-based fashion and the best results were obtained for a single hidden layer with 150 neurons. Activation function is softmax since we are using patternet.

4.3.3 Transfer Learning

Since the last three layers of the pretrained network AlexNet (downloaded from MATLAB) is configured for 1000 classes, we replace these layers and fine tune it for the class images dataset. The number of classes are 48 which aligns with the number of individuals with labels Stochastic gradient descent is used for efficient training with min batch size of 32.

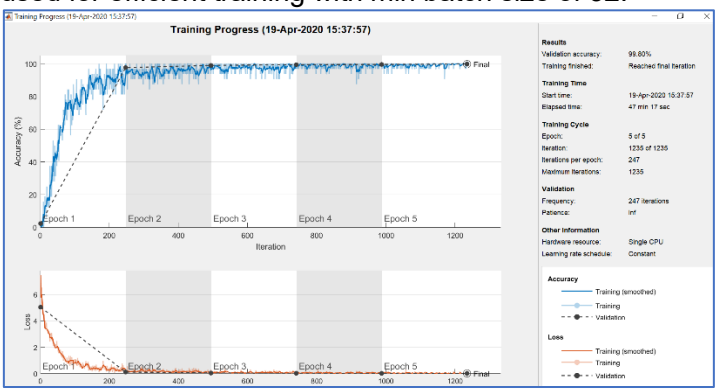


Figure 6: AlexNet Training/Validation accuracy and Loss

4.4 Creative Mode

In creative mode we have implemented 4 features which the user can enable with the values:

1. Superimposing content by putting sunglasses on every face
2. Superimposing content by putting a smiley on every face
3. Cartoon-ify all faces
4. Blur all faces
0. Creative mode disabled

Creative mode works perfectly for faces aligned straight to the camera but sometimes does not work when there are faces looking sideways. Superimposing sunglass depends on identification of eye, which in many images were challenging when the face was slanted.

5. OCR for Image Labelling

The initial approach was to segment the label placard from an image and then feed it to the OCR reader in MATLAB. This was attempted using MATLAB regionprops area and solidity parameters, but this did not give satisfactory results to generalize among multiple images with varying slant angles for the placard. Later the image segmenter tool in MATLAB was used, but again this was specific to an image and could not generalized as a function for other images. Later the MSER region- based approach provided in the MATLAB tutorial was followed [10]. This gave limited results, but we didn't finally use it for labelling.

6. Failed Approaches

6.1 Identify Unknown persons in group image

We had two approaches. The first approach was simple, using the post probabilities returned by all classifiers. A threshold value was determined , which on crossing, the label was considered valid. Setting the optimum threshold was challenging as this resulted in too many false positives for unknown persons. Experimenting with different values across classifiers, the best results were in CNN with minimal false positives and was retained. The thresholding for other classifiers were discarded as it was resulting in too many false positives for unknowns spoiling results for valid labelling.

The second approach was more complex

- Assign a label to a face if it has the highest post probability value, with the same label, among the group.

- For all the remaining images, take the next label based on the post probability values. Follow the same approach as in step 1 to avoid clashes.

All post probabilities below a threshold, to be assigned the label 'Unknown'+ Count. Where count started from 1 and increased with each new unknown person identified.

This logic in theory was more fool proof. Implementing this was challenging and resulted in errors, which could not be resolved in the timeframe and hence was discarded.

7. Results and Discussion

7.1 Individual Images

Given below is the training and validation accuracy for the individual images.

Feature Descriptor	Classifier	Training Accuracy	Validation Accuracy
HOG	SVM	100	97.07
HOG	MLP	99.53	96.87
SURF	SVM	99	97
SURF	MLP	98.98	95.46
NONE	CNN	100	99.8

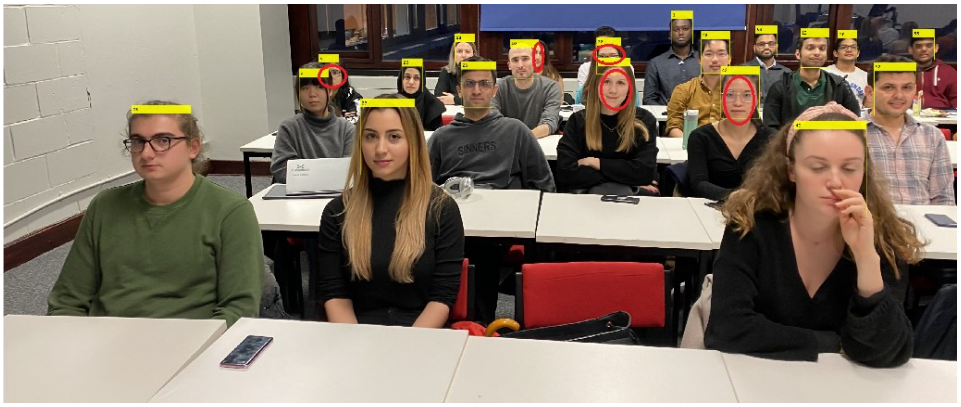
As we can see from the table, the best accuracy was obtained for CNN classifier using a pretrained AlexNet network. All the other classifiers had validation accuracy of more than 95.5% with reasonable drift from training accuracy suggesting no overfitting. Confusion matrix for validation images in CNN has been attached in the Appendix section (due to space constraints).

7.2 Group images

The performance for the classifiers on group images were relatively lesser accurate as there were 2 main issues to tackle:

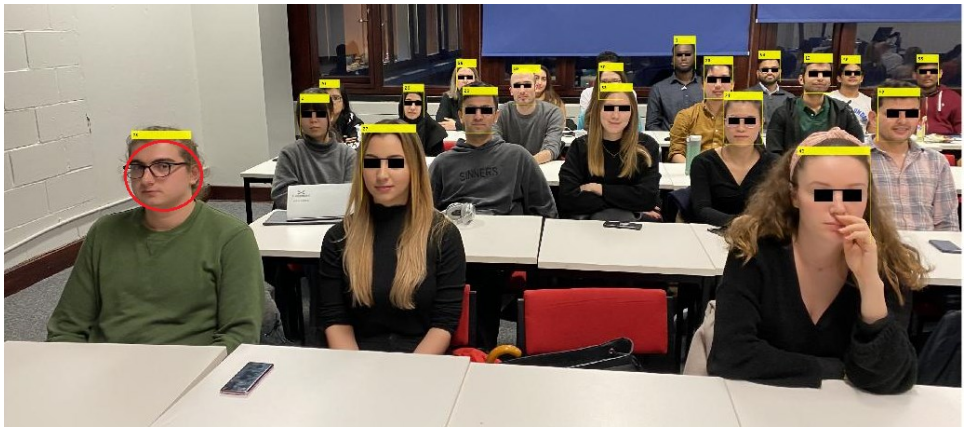
- Presence of individuals who did not have labels
- Different orientation for faces and lighting conditions

CNN and HOG-SVM combinations gave more than 70% accuracy across images (80%+ if unknown faces are not considered) while the other classifiers gave around 60% or more accuracy, depending on the images. Below is an example on the test image. Encircled in red are faces which were classified inaccurately (when unknown person labels were disabled). 2 of these were from people not in the training set. 2 of the other faces are occluded. 1 face which almost completely occluded was not detected by the face detector.

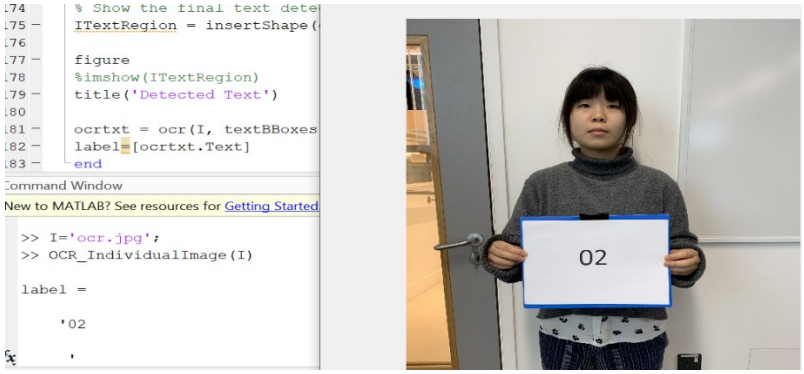


On enabling the unknown image labels (optional task) with threshold set, we can see that 2 people in the front row who were tagged with incorrect labels earlier are now tagged as ‘Unknown’ correctly. But 2 people who had an incorrect classification label earlier are now tagged as unknown incorrectly. In addition, one person in the last row which was correctly labelled earlier has now been tagged as unknown. Hence there is a trade off in determining the threshold value for post probability while determining the unknown people. The front row seems to be accurately identified across most images. As we progress to the back with occluded to slanted face images, the accuracy reduces.

Creative mode results were as expected for straight images but get inaccurate when the user had glasses or face was slanted, when it was not possible to detect eyes. An example below.



The OCR (optional task) results were mixed with accurate detections of the label only when the image was straight. Hence the performance is not robust.



8. Conclusion and Future work

We have thus successfully developed a face recognition system exploring multiple feature descriptors and classifiers. The performance on individual images was of high accuracy while that of test group images was very good but needs some improvement. This can be explained by the difference in data distribution between training + validation (individual images) and test images(group images). The best performance was observed for the pretrained CNN Alexnet and HOG-SVM combination. The success of CNN can be attributed to the automatic hierarchical learning of global and local features, in addition to the invariance to translation. Additional hyperparameter tuning may help to improve the accuracy a bit more. A key point referred to in face recognition papers is face alignment. Implementing this can help to improve the accuracy and can be considered a potential future work. Most face recognition systems are almost real time and hence performance speed and optimized code is key for adoption of systems. Algorithms like YOLO/ One shot learning are other potential steps to expand this project.

9. Reference

[1] https://www.globenewswire.com/news-release/2020/04/04/2011766/0/en/The-Global-Face-Recognition-Market-is-expected-to-grow-from-USD-3-546-56-Million-in-2019-to-USD-9-992-35-Million-by-the-end-of-2025-at-a-Compound-Annual-Growth-Rate-CAGR-of-18-84.html

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I.

[3] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, Oct. 2016.

[4] https://software.intel.com/en-us/ipp-dev-reference-histogram-of-oriented-gradients-hog-descriptor

[5] https://uk.mathworks.com/help/deeplearning/ref/alexnet.html

[6] http://cs231n.stanford.edu/slides/2019/

[7] https://towardsdatascience.com/increase-your-face-recognition-models-accuracy-by-improving-face-contrast-a3e71bb6b9fb

[8] K. Dharavath, G. Amarnath, F. A. Talukdar and R. H. Laskar, "Impact of image preprocessing on face recognition: A comparative analysis," 2014 International Conference on Communication and Signal Processing, Melmaruvathur, 2014, pp. 631-635.

[9] Justin Pinkney (2020). MTCNN Face Detection (https://www.github.com/matlab-deep-learning/mtcnn-face-detection), GitHub. Retrieved April 20, 2020. https://uk.mathworks.com/matlabcentral/fileexchange/73947-mtcnn-face-detection

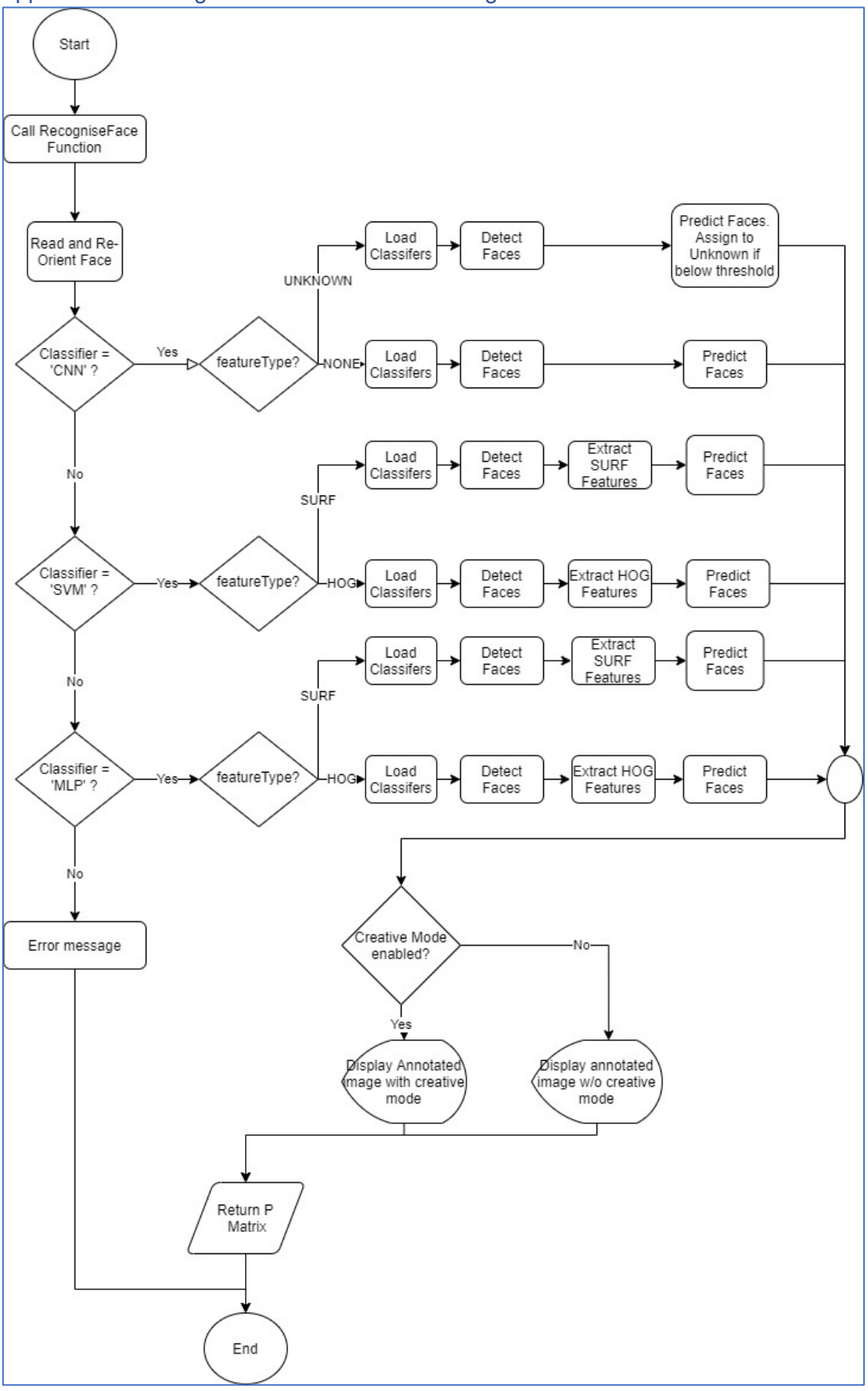
[10] https://uk.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html

[11] INM460 Computer Vision Tutorials and lecture slides

[12] MATLAB tutorials and forum

[13] A. Krizhevsky, I. Sutskever, and G. Hinton. “Imagenet classification with deep convolutional neural networks”. In NIPS, 2012

10. Appendix 1 – RecogniseFace Process Flow Diagram



11. Appendix 2- Confusion Matrix for CNN- AlexNet in Validation Dataset