INM460 Computer Vision Coursework

I. INTRODUCTION

This report overviews the methods and approaches used to build a face recognition function which recognizes faces from individual and group images of the students and critically evaluates the results obtained by face recognition function.

The structure of report is as follows. In first section of report, we describe raw data and explain the steps carried out to obtain data along with some pre-processing steps. After that, we briefly describe the techniques and algorithms used in face recognition task. Third section of report describes the implementation details of these algorithms and critically evaluates the obtained results. At last, we summarize the work done in few lines and suggest potential techniques that can help to improve the results in future.

II. RAW DATA

The dataset that is provided for our study consists of videos and images of students of class that have been taken individually and as a group. In individual images and videos, each person is holding A4 paper with specific identification number between 1 to 78. There are 8 images and videos for each person but few of them have only 6 images and videos. Group images have been taken in lecture theatre, so they have different scales (students sitting at the back appear smaller compare to students sitting in the front). Videos have duration of 6 to 7 seconds and have a record of students at different angles.

The images in dataset are in format “JPEG” while videos have “mp4” file format. The file formats of images and videos are easily recognised by python standard libraries; therefore, we don’t need any conversion. Labels for face recognition task are numbers between 1 to 78 holding by each student in individual images. But there are many students who took part in group images but did not appear in individual images, so they don’t have any label assign to them. Moreover, only 48 students took part in individual photo session and they have label assign to them between 1 to 78. One point to note is that inconsecutive numbers between 1 to 78 are assigned to 48 students.

Our main goal is to perform face recognition task which is supervised learning problem. So, our first step is to make folders for each label. We created 48 folders for individual captures. Each folder has videos and images related to that label. One folder is created for group captures which contains all group images and videos.

III. FACE RECOGNITION

The main objective of our course work is to create a face recognition function which takes image as input and returns a matrix in output. The first column of matrix is label of image whereas second and third columns are x and y coordinates of centre position of detected face respectively.

In this section, we will briefly discuss the methods and techniques we will use. After that we will address the implementation details of these techniques and combining them in order to achieve the final face recognition function. Lastly, we present the results of our face recognition function.

a) Methods and Techniques

We used SURF, SIFT and ORB for feature extraction. MLP, SVM, Logistic regression, resnet-50 and vgg16 are used for face classification. Three type of feature extractors SIRF, SURF and ORB are combined with classifiers SVM, MLP and Logistic regression. The subsequent section briefly describes the feature extractors and machine learning algorithms used for face recognition task.

1. Feature Extractors

Feature extraction is an integral part in object recognition task. Different techniques are used to extract features from image. In this section, we will briefly describe the feature extractions techniques which are used in our face recognition task.

A. Scale Invariant Feature Transform (SIFT)

Scale invariant feature transform features are invariant to scale and rotation along with they have ability to match different views of an object. The extracted features are highly distinctive and have capacity to match views in different illuminations, 3D viewpoints and additive noise. This property of SIFT makes it more robust to object recognition task. There are four stages to generate SIFT features.

- Detection of scale space extrema is the first stage which aims to find scale and orientation invariant points over all locations of an image by applying Difference of Gaussian function.
- After finding the location and scale of the key points, further improvement is required in order to get more accurate points. More precise scale space extrema are achieved by applying Taylor Expansion. At this stage, key points which have low contrast are rejected along with key points which are located at edges. Finally, we are left with strong interest points
- Next, invariance to image rotation is achieved by assigning orientation to each key point. We take neighbourhood around key point based on scale and then we compute gradient, magnitude and direction in that region. The gradient orientation of points inside region is used to form an orientation histogram which has 36 bins containing 360-degree orientation. Further, the orientation is calculated by considering the highest peak along with any peak above 80% of it. As a result, key-points are created which have same scale and location but different direction.
- As of now each key-point has location, scale and orientation. At this stage, we need to compute descriptor for each of them whereas ensuring to include invariance to other variations for instance different illuminations and 3D viewpoints. For this purpose, we consider 16×16 neighbourhood around each key-point which is further divided into 16 sub-blocks of 4×4 boxes. Next, an orientation histogram of 8 bins is formed for each sub-block. Thus, we have 128 bin values in total. At last, descriptor vector is obtained by flattening and concatenating the bin values (Lowe, 2004).

B. Speed-Up Robust Features (SURF)

SURF is a faster version of SIFT. In SIFT, difference of gaussian is used for finding scale space while SURF uses box filters for this purpose. Moreover, computing convolution with box filters by use of integral images is efficient and can be accomplished all together for various scales. Moreover, SURF performs detection based on Hessian matrix.

For orientation assignment in SURF, a circular neighbourhood of radius 6s (s is scale) is considered around key-point and Haar-wavelet responses are computed in both horizontal and vertical direction. We compute sum of all responses inside a sliding orientation window of 60 degrees to estimate dominant orientation.

In terms of descriptor extraction, a square region of size 20×20 is taken around interest points which is further partitioned into 4×4 square sub-regions. We compute horizontal and vertical Haar-wavelet responses for each sub-region. Finally, we create SURF feature descriptor vector $v = (\sum d_x, \sum d_y, \sum|d_x|, \sum|d_y|)$ which has 64 dimensions (Bay, Tuytelaars and Van Gool, 2006).

C. Oriented FAST and Rotated BRIEF (ORB)

ORB is a binary descriptor derived from a keypoint detector FAST (features from accelerated and segments tests) and BRIEF (binary robust independent elementary feature) descriptor. It is quicker as compared to SIFT and SURF.FAST keypoint detector is used to find keypoints by using intensity threshold among centre pixel along with pixels in circular region around centre. Further, the measure of corners is not given by FAST. For corner measure, Harris corner measures is applied, and top N points are chosen. Moreover, multiscale features are not formed by FAST. A scale pyramid of image is employed to get multi-scale features. For rotation invariance, moments are calculated through x and y in circular area of radius r around centre. After that, BRIEF descriptor is rotated corresponding to orientation of keypoints for feature description. By rotating BRIEF along keypoints, it loses its property of having large variance and mean near 0.5 along with being uncorrelated binary feature vectors. In order to tackle these issues, ORB operates a greedy search for all possible binary feature vectors in order to find the binary tests with high variance and mean close to o.5 with ensuring uncorrelatedness. Furthermore, multi probe Locality Sensitive Hashing (LSH) is used for descriptor matching (Rublee et al., 2011).

2. Machine Learning and Deep Learning Algorithms

This section briefly describes the functioning of machine learning and deep learning algorithms used for classifying the images in face recognition.

A. Support Vector Machine (SVM)

Support vector machine is widely used algorithm for binary classification task, and it can be extended for multi classification problem by using one-versus-one or one-versus-all approach. The main objective of support vector machine is to find optimal hyperplanes which can separate the data points by using maximum margin. The margin is the distance between closest data points of two classes. Figure displays the optimal hyperplanes for linearly separable data. Moreover, support vector machine has ability to separate the non-linear data by using different kernel functions. In the case of non-linear data, support vector machine maps data points to higher dimensional space, where, kernel function is used to separate the data points in higher dimensional space. Radial basis function or Gaussian kernel and polynomial kernel are commonly used non-linear kernels (Borah and Gupta, 2017).

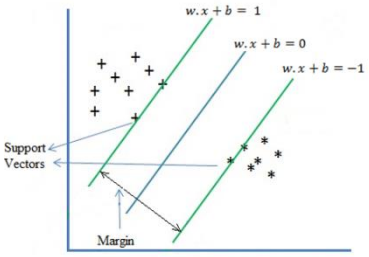


Figure 1: Optimal hyperplanes in case of linearly separable data *(Borah and Gupta, 2017)*.

B. Multi-Layer Perceptron (MLP)

Multi-layer perceptron which is also known as feed forward network has been used for both classification and regression problems. It consists of input layer, two or more hidden layers and output layer. There is unidirectional connection between nodes of each layers. Each node of one layer is connected to other nodes of layer by some weight and bias. The weight is crucial parameter in Neural network functioning and bias helps to find the best fit for given data. In MLP, output is obtained by passing weighted sum plus bias through function. This function is known as activation function which is applied to Hidden layers and output layer. Sigmoid, softmax, Rectifier linear unit function are few examples of activation functions. The use of multiple hidden layers and non-linear activation function enables MLP to deal with non-linear data efficiently. Backpropagation algorithm is used to train feed forward network and it consists of two stages. First, the input is passed through the network to get predictions and then we compute error which is minimized by backpropagating the derivative of cost function in order to update the weights. Stochastic gradient descent or batch gradient descent is used to adjust the weights (Zanaty, 2012). Figure 2 displays the architecture of MLP.

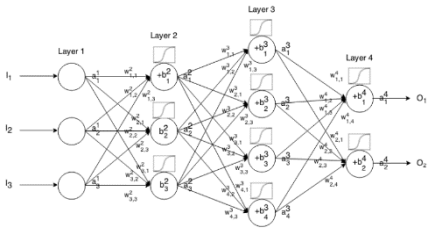


Figure 2: The structure of MLP

C. Convolutional Neural Network (CNN)

Convolutional neural network is a special kind of feed forward network and has been widely used for image classification task in computer vision. A CNN architecture is stack of convolutional layers, pooling layers and fully connected layers. In image classification task, input consists of an image and convolution is performed in order to get the features in image by using different filters to create feature maps. After that, Rectifier linear unit activation function is applied to convolution layers in order to remove linearity because image data is highly non-linear due to adjacent pixels, different colours, different elements in images but applying convolution might create feature maps which are linear thus we need to make them non-linear. Next, pooling layer is applied to each feature map which reduce the size and number of parameters which helps to avoid overfitting along with ensures spatial invariance in image. Max pooling and mean pooling are commonly used approaches for pooling. After pooling layer, we flatten them by taking each number row by row and placing into one long column which is passed through fully connected artificial neural network in order to get output. After getting the output, we compute error which is minimized by backpropagating error through the network. During training of CNN, weights are not the only parameters which are adjusted but we also adjust filters or feature detectors in order to get best feature maps. At last, we get a trained convolutional neural network which can recognize and classify the images (O'Shea and Nash, 2015).

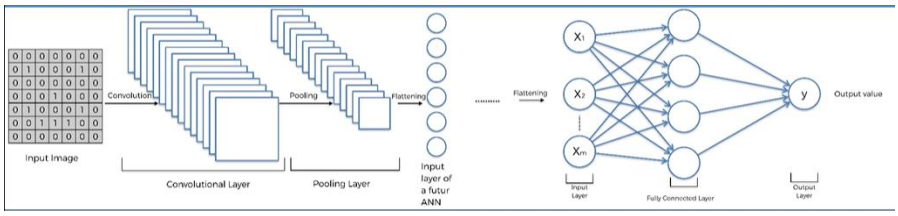


Figure 3: Convolutional neural network architecture

D. Multi-task Cascaded Convolutional neural network (MTCNN)

MTCNN is a deep learning approach for face detection and alignment. It can efficiently detect the faces and has higher performance compare to other traditional methods. There are three neural network cascades P-net, R-net and O-net in MTCNN. Initially, image pyramid is formed by scaling original image using different scales. After that, we feed image pyramid to P-net which is fully convolution network and generates candidate boxes which are adjusted by bounding box regression. Next, highly overlapped candidate boxes are merged by applying non-maximum suppression. Further, another CNN, R-net takes output of P-net as input and rejects the wrong candidate boxes and executes bounding box regression and employs non-maximum suppression merging. At last, O-net uses the output of R-net as input. It has kind of same function as R-net, but this stage gives out the final facial frame with five coordinates of facial landmarks. Each network in MTCNN has three outputs: the probability of having face bounding box, bounding box coordinates and facial landmarks coordinates (Zhang, Zhang, Li and Qiao, 2016).

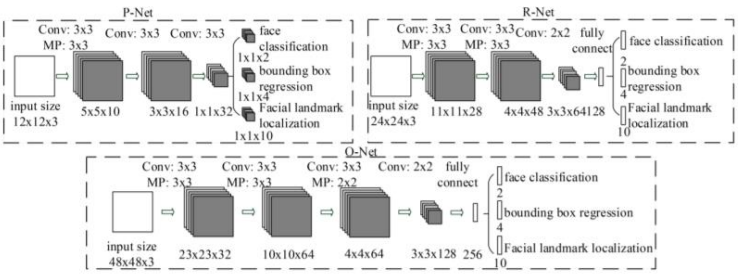


Figure 4: The Architecture of P-Net, R-net and O-net *(Zhang, Zhang, Li and Qiao, 2016).*

E. Logistic Regression (LR)

Logistic regression is a linear probabilistic model and has been widely used for binary classification problem. It gives good performance for linearly separable data. Logistic regression is used to find the probability of particular event occurring given features. A sigmoid function is used to find the probability of an example belonging to a particular class. The sigmoid function takes real numbers as input and gives output in range [0,1]. A threshold function can be used to convert probability into binary outcome. Maximum likelihood estimation is used to train coefficients of logistic regression which naturally arises cost function. The optimization algorithm such as gradient descent is used to minimize this cost function. In order to avoid model overfitting, Ridge and Lasso regularizations are commonly used which have hyperparameter C to be fine-tuned. Logistic regression can be extended to multi classification problem by using one versus rest (OVR) approach.

b) Implementation of Deep learning and Machine Learning Techniques

This section describes the implementation details of methods and techniques outline before to accomplish face recognition task. Face recognition task is divided into two sub-tasks: face detection and face classification.

I. Face Detection

Initially, we performed face detection using Haar cascades and Caffe model which are part of Open CV library. Both Haar cascade and Caffe model did not detect as many faces in group images. Then we decided to implement MTCNN which is a deep learning approach for face detection. MTCNN is a part of facenet-pytorch package and use GPU to perform detection which makes it more efficient. In terms of performance, MTCNN performed good for both individual and group images. Moreover, MTCNN detected more faces in group images as compared to Haar cascades and Caffe model.

II. Face Classification

Face classification is performed to predict labels for detected faces. We use machine learning and deep learning algorithms to perform classification. For classification, we need labelled dataset. After preparing dataset, we use three feature descriptors SIFT, SURF, and ORB to extract features which will use to train three classifiers Logistic regression, SVM and MLP. Furthermore, two convolutions neural network-based architecture (ResNet50 and vgg16) are also fine-tuned for face classification.

i. Data Preparation

We make use of face detection to prepare dataset for classification. As we have created folders for images and videos of each label, we iterate through each label folder in order to detect and extract faces from both individual images and frames extracted from videos (20 frames are extracted per video). As of now, faces are detected and extracted from group images and video frames (maximum 20 frames per video). Overall,

approximately 12,000 faces are extracted from group images. We need to label extracted group images manually which is not an easy task. As this a tedious task, we manually labelled 20 images for each class, and resorted to applying a creative augmentation strategy on individual images that would mimic group images.

The face images extracted from group images are significant to assess the performance of our classification model because group images have been taken in different condition compare to individual images. The augmentation strategy included applying random zoom, shear, horizontal flip, range of brightness levels, and rotation on individual face images. Initially, Gaussian blur and additive noise were part of the augmentation pipeline, however, they were discarded due to poor accuracies. In addition, we also resized the images to make them of size 224×224. Resizing the images will also help to train our model efficiently. Further, the problem of class imbalance is avoided. At last, we have approximately 17,400 images collectively for 48 classes.

Figure 5: Different type of data augmentation (Zooming, brightness, shear, horizontal flip)

ii. Training Machine learning and deep learning models

Prior to training machine learning models, feature extraction is performed. Visual- bag-of-words- approach was adopted to get features from images. First, various feature extractors (SURF, SIFT, ORB) are used to extract features from image. Then we created codebooks of features by using k means clustering with k =480.This k value has been derived by multiplying number of classes by 10. After that, vector quantization is applied to each image which maps each feature vector to the index of its nearest centroid. The number of occurrences of each visual word in image is counted in order to create histogram of features. Finally, bag of words features is used to train classifiers (MLP, SVM, LR).

We opted to perform grid search with 10-fold cross validation in order to find the optimal hyperparameters of the classifies. For SVM, different kernels (linear, rbf, polynomial) along with hyperparameters C ([0.001, 0.01, 0.1,1,10,100]) and gamma ([0.001, 0.01, 0.1,1,10]) are tried. Regarding MLP, number of hidden layers and number of hidden neurons are varied. Hyperparameter optimization has not been performed for logistic regression due to lack of time.

Deep learning models are more powerful for detection and recognition task. Convolutional neural network automatically extracts features from image that makes it more robust to image classification task. We implemented two pretrained CNN architectures namely Vgg16 and resnet50 to perform classification. The reason behind is that training of model created from scratch takes longer than training of pretrained models. Both models vgg16 and resnet50 are pretrained on ImageNet dataset of 1000 classes. There are two approaches to perform transfer learning: feature extraction and fine tuning. In fine tuning, we changed the final layer with the number of classes of our data along with a couple of fully connected convolutional layers of size 256 has been added. For training of model, it is ideal to use low learning rate in order to avoid changing the values of weights too much. In case of using pre-trained model as fixed feature extractor, we only train the weights of final layer and freeze the weights of all other layers.

We opted for both kind of transfer learning: fine tuning and feature extraction. The performance achieved by fine tuning of model is higher than of feature extraction. In terms of performance of vgg16 and resnet50, resnet50 has slightly higher performance than vgg16.

For both models, 20% data is chosen for validation set. All the input images are of size 224×224. The chosen optimization algorithm is Adam. We chose small learning rate of 0.0001 and batch size of 20 for training of model. Both models are trained for 10 epochs. Cross-entropy is used as cost function. 50% drop out is used to avoid overfitting. The number of classes in output layer has changed to 48.

III. Recognize Face Function

Group images were used to test our face recognition function. Our face recognition function applies face detection and extract face images which are pre-processed. Then it uses either deep learning model (resnet50 or vgg16) to classify face label or perform feature extraction using visual bag of words approach before using machine learning models (MLP, SVM, LR) to predict face label.

The recognize face function takes following arguments:

- Input image path
- **Classifier:** MLP, SVM, LR, Resnet50, Vgg16
- **Feature extractors:** SURF, SIFT, ORB

It gives a N×3 matrix P as output in which first column gives id of detected face and 2nd and 3rd columns give centre position of face (X, Y).

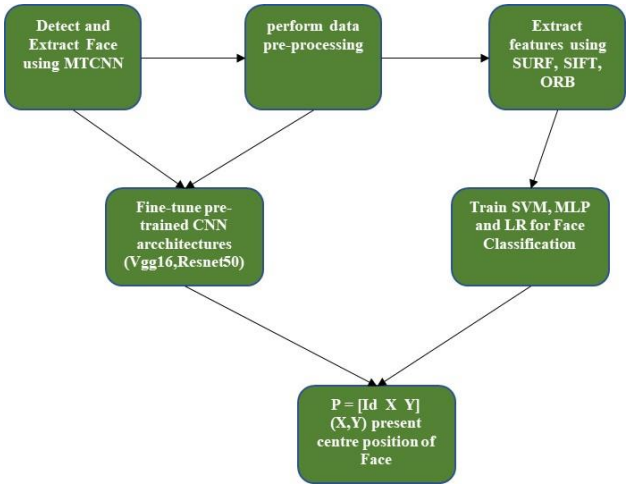


Figure 6: Workflow of face recognition function

c) Results

In this section, we will highlight the results achieved from previous work. Table 1 shows the validation accuracy of each model along with values of hyperparameters. We can analyse that Resnet-50 has the highest validation accuracy. Logistic regression gave the lowest accuracy which can be easily improved by performing grid search. Due to lack of time, we used default values for parameters in Logistic regression. SVM classifier gave better results with SIFT features while MLP performed better with SURF features. Vgg16 gave the second highest accuracy which is approximately 1% less than the accuracy of Resnet50. Overall, deep learning models performed better than machine learning models.

Table 1: Reported results for machine learning and deep learning models

Classifier		Kernel	C	No of hidden layers	No of hidden Neuron	Optimizer	Gamma	Validation accuracy
SVM	SURF	Radial basis function	1	N/A			10	94.05%
	SIFT							95.37%
	ORB							89.9%
Logistic regression	SURF	N/A					N/A	30.8%
	SIFT							38.25%
	ORB							33.7%
MLP	SURF		N/A	3	150	Adam		96.29%
	SIFT				100			95.3%
	ORB				50			92.59%
Resnet50	-			-	-	Adam		99.97%
Vgg16	-			-	-			98.53%

Moreover, face detector gave very accurate results. It was able to detect faces of students sitting at the back in group images. Group images are used to test face recognition function with our best classifier, Resnet-50. Figure 6 displays the result's after providing a few group images to the face recognition function. The faces enclosed by a red rectangle are misclassified by recognise function. The faces enclosed by a green rectangle indicate faces not detected by the function due to different facial positions not been recognised by the recognise function. Moreover, there are few students in group image who didn't appear in the individual photo session and have not been assigned an Id. The face recognition function gave them an ID of someone who s. Overall, the face recognition function made very accurate prediction with Resnet-50 classifier.

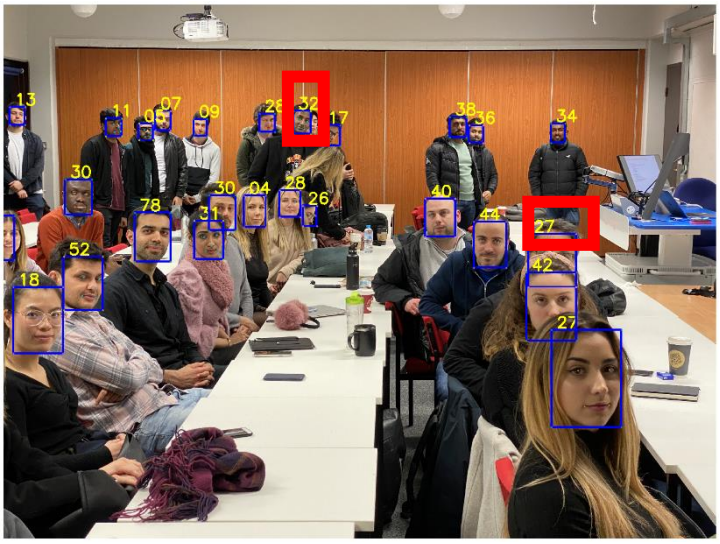


Figure 6: Correctly and incorrectly classified faces in group images by face recognition function with resnet-50 model Red rectangles shows misclassified faces and green rectangle depicts faces not detected by function

Figure 7: face recognition function performance with different modes (cartooning and sunglasses along with cigar)

We also tested the face recognition function with two creative modes: cartooning the image and sunglasses along with cigar on same group images to analyse the impact of creative mode on the performance of face recognition function. We examined that the recognition function was unable to make many correct predictions and assigned same label to many faces in group image for cartoon images. This effect can possibly be improved by introducing cartoon-related augmentation into the augmentation pipeline.

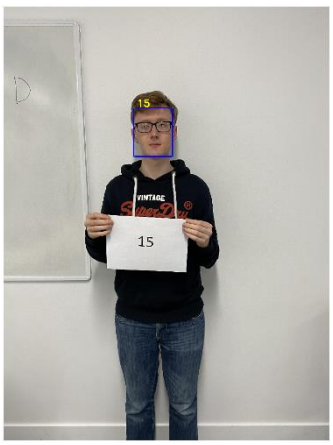


Figure 8: Testing recognition function on individual images

Figure 8 depicts the results achieved by testing recognition function with best classifier, Resnet-50 on individual images. It can be seen that the function detected correct ids for tested individual images. Further, confusion matrix is added in appendix section to visualize the performance of best classifier Resnet50.

d) Conclusion and Future Work

In this study, we explored various computer vision techniques for the purposes of facial recognition. Facial recognition is comprised of two steps: face detection and face classification. We used MTCNN to perform face detection. For the purpose of face classification three machine learning models SVM, MLP, LR with three feature descriptors SURF, SIFT and ORB are tried. In order to extract the features from images, visual- bag-of-words approach is used. Moreover, we also fine-tuned two pretrained convolutional neural network-based architectures vgg16 and resnet-50 to classify face labels. Both machine learning and deep learning models gave good results. But the highest performance is achieved with deep learning models particularly resnet-50. Finally, a face recognition function is created which takes an image path, classifier type, feature descriptor and creative mode type as input and returns a N×3 matrix which contains face Id, along with x and Y coordinates of centre position of detected faces if present.

Optical Character Recognition (OCR) was not used in this project to aid in labelling the images and was done manually, however as a future task we can possibly consider writing OCR functionality to automatically generate labels for the images. Furthermore, the process of labelling the facial images extracted from the group images and videos is very time consuming and tedious. In this project approximately 1000 group facial images were labelled from a possibility of 12000 images. To aid in this process, we can consider using face clustering techniques, density-based spatial clustering of applications with noise (DBSCAN) for small datasets or Chinese whispers clustering for large datasets. I believe both should be trailed out. Additionally, we did not consider introducing an ‘unknown’ category in the dataset, and this can easily be introduced. Essentially, the unknown category would contain images of people/students that have not been identified i.e. do not have an ID.

IV. References

Zhang, K., Zhang, Z., Li, Z. and Qiao, Y., 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10), pp.1499-1503.

O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.

Borah, P. and Gupta, D., 2017. Review: Support Vector Machines in Pattern Recognition. International Journal of Engineering and Technology, 9(3S), pp.43-48.

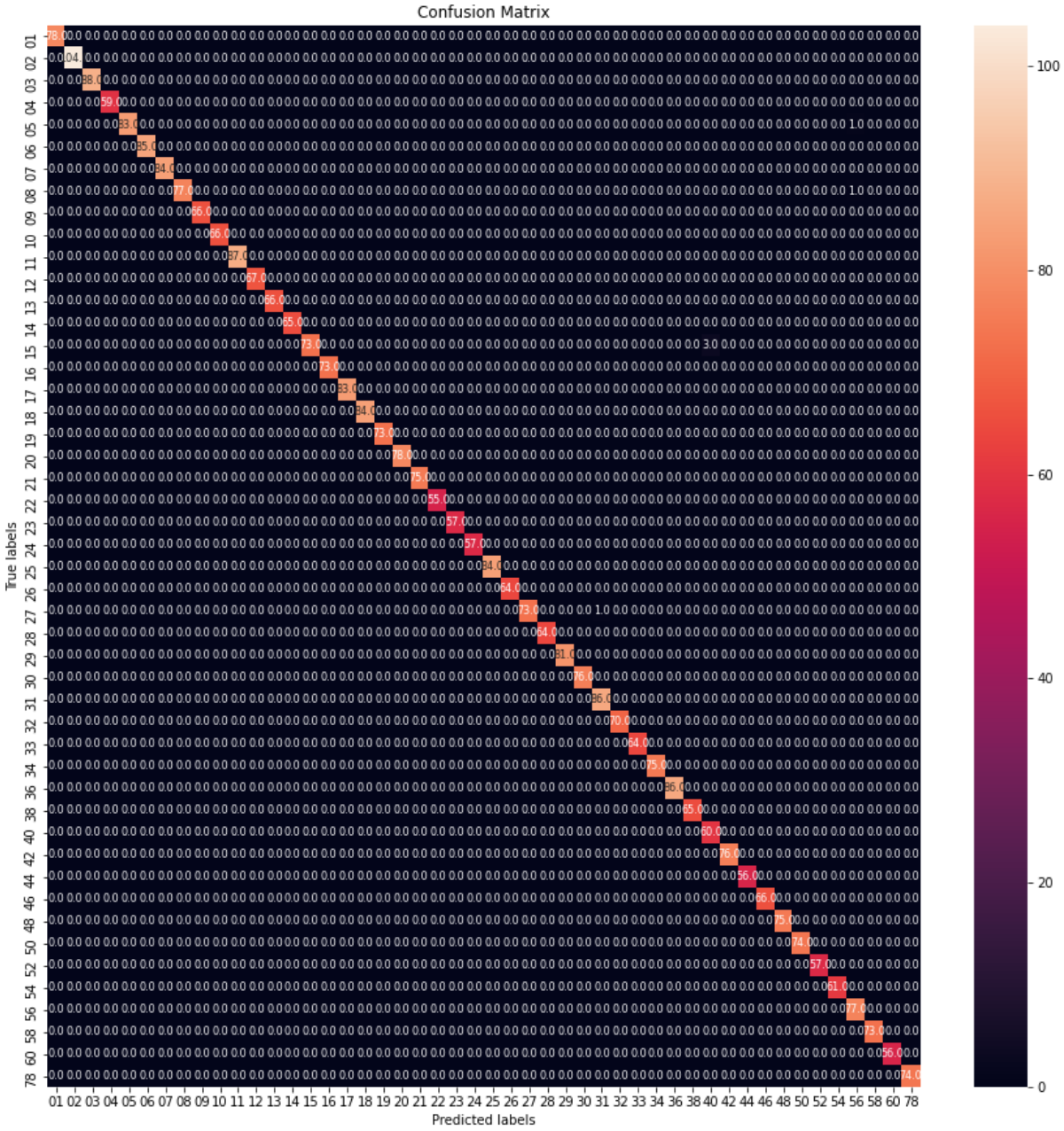
Zanaty, E. (2012). Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification. Egyptian Informatics Journal. 13. 177–183. 10.1016/j.eij.2012.08.002.

Lowe, D., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp.91-110.

Bay, Herbert & Tuytelaars, Tinne & Van Gool, Luc. (2006). SURF: Speeded up robust features. Computer Vision-ECCV 2006. 3951. 404-417. 10.1007/11744023_32.

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011. ORB: an efficient alternative to SIFT or SURF. Proceedings of the IEEE International Conference on Computer Vision. 2564-2571. 10.1109/ICCV.2011.6126544.

V. APPENDIX


Confusion Matrix
True labels
Predicted labels
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 36 38 40 42 44 46 48 50 52 54 56 58 60 78
100
80
60
40
20
0