INM460: Computer Vision – Facial Recognition

Hisho Rajanathan

Introduction

Computer vision is a subfield of artificial intelligence, where algorithms are trained to distinguish and process information from the visual world such as videos and images [1]. Facial recognition has been gaining a lot of traction in the industry. Japan had developed a computer vision system for the Olympics in 2020 which will be able to authorise individuals and allow them access to the games [2]. Recently in the UK, the Metropolitan Police have developed a facial recognition system to scan individual faces to identify if individuals are wanted for serious crimes or wanted by the Courts [3].

There have been various instances where an individual has been introduced to a number of people, however they have forgotten their names. This is the driving factor for developing a computer vision system to detect and identify individuals from a database of known student faces in the Computer Vision class. This paper develops a Facial Recognition MATLAB function to identify an individual in various test images.

Speeded-Up Robust Features (SURF) and Histogram of Oriented Gradient (HOG) features were used to extract features from the image and video dataset. These extracted features were training on Support Vector Machines (SVM) and Multilayer Layer Perceptron (MLP). In addition, a pre trained convolutional neural network (CNN), 'AlexNet', was used to train and classify the images from a database of known images.

Data

Data Collection

The dataset was created by taking pictures of a class of Computer Vision students as well as taking images of individual students at different angles. When the individual student photos were taken, the individual had to hold an A4 sheet of paper with printed digits, which is used to identify the student in various testing images. The data set consisted of 48 individual students which generated 48 class labels ranging from "01" to "78" non sequentially. On average, each person had six to eight pictures taken of themselves from different angles and six to eight videos. The pictures of the students in the class room were also taken, where a lot of the photos did not include all students. The individual pictures were taken in a different environment to the class photos and all images and videos were stored in RGB format. The individuals were manually placed into their respective folders labelled "01" to "78".

Data Pre-Processing

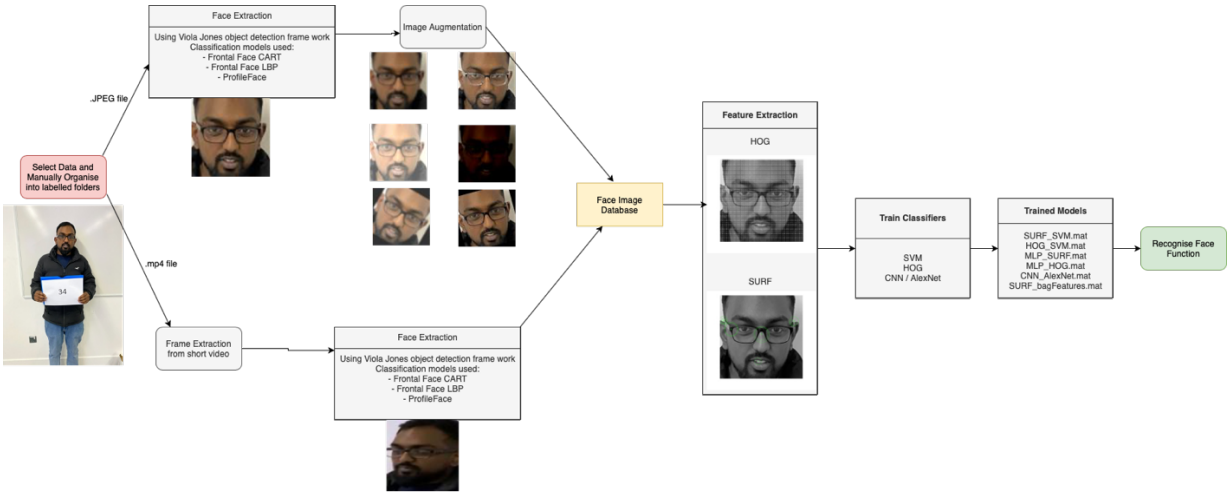

Figure 1 Flowchart showing Methodology

Frame Extraction

The raw data consisted of short videos of individuals, showing the individuals at different angles. Frames were extracted from the video and for the purpose of this study; we have set the number of frames to be extracted at 30. This will allow for sufficient additional images for the dataset, so each of the algorithms can be trained on. As the start and the end of the video consisted of dark or black frames, a brightness threshold was included during the frame extraction process, so these frames were not extracted.

Face Extraction

Viola Jones object detection framework was used to detect faces from the images and extracted frames from the videos. Three classification models that were used to detect the faces were "FrontalFaceCART", "FrontalFaceLBP" and "ProfileFace". FrontalFaceCART detects faces that are upright and facing forward and uses CART based classifiers; FrontalFaceLBP also detects faces that are upright and facing forward and uses LBP based classifiers; and lastly, ProfileFace detects upright profiles using Haar Features to encode face details. The reason for using three different classification methods is because the video was taken at different angles and so certain classification models could not detect the faces. *Vision.CascadeObjectDetector* was used to detect the faces with a minimum size of [100,100] as in the individual images the faces were roughly the size of 200x200. Keeping a minimum size of 100 x 100 will avoid false positive detections such as detecting a person's nose or ears. A merge threshold of 10 was used to supress false positive detections. Once a face is detected by the Viola Jones Object Detection Framework, a bounding box is drawn around the individual's face to show that the framework has detected a face. The bounding box has dimensions which can be used to identify the x and y coordinates of where the face is, as well as the width and height of the bounding box.

Data Augmentation

Data augmentation was applied to the whole dataset, because when training deep neural networks, it helps reduce overfitting and it helps with generalising the classifier [4]. Image augmentation was only applied on the original images per person instead of the extracted frames from the short videos. This was done for two reasons; the first being the extracted frames had blurry images and faces that were at different angles; the second due to computational reasons a large dataset was avoided. As well as using image augmentation to avoid overfitting, we were aiming to try and replicate the environment of the class photos with image augmentation. We were aiming to recreate individuals sitting in the class image such as where their faces are slightly tilted, to help the models perform accurately on the testing data. Furthermore, for individuals sitting at the back or in a distance from

where the lecturer took the photo from, the faces were slightly blurred; hence another data augmentation process was to blur the images. The augmentation was applied on the cropped faces after the face extraction step in the method.
The different types of augmentation applied were :

- Gaussian blur where standard deviation (k) is equal to 10 or 12 using the *imgaussfilt* function in MATLAB (a).
- Dilating image by creating a non-flat ball-shaped structuring element using the *imdilate* function in MATLAB (b).
- Increasing the brightness of an image by 100 units and applying gaussian blur with standard deviation of six (c).
- Decreasing the brightness of an image by 100 units and applying gaussian blur with standard deviation of six (d).
- Rotating an image 15° anticlockwise and applying gaussian blur with standard deviation six.
- Rotating an image by 30° anticlockwise (f)
- Rotating an image by 30° clockwise, increasing the brightness by 50 units and applying a gaussian blur with standard deviation eight (e).

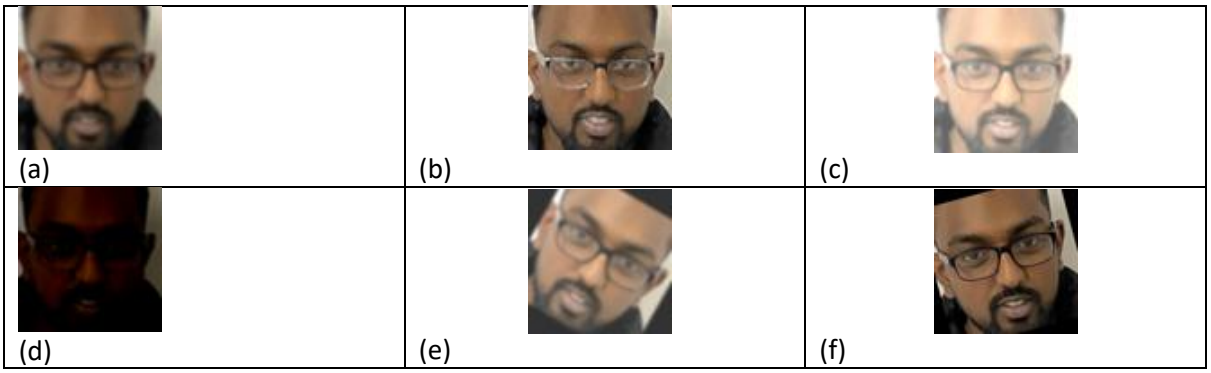

Figure 2 Different types of augmentation applied to cropped faces

Gaussian blur smooths images (blurs) which in turn removes the detail and noise from the image [5], hence will help with individuals who are near the back of the photo as their faces are blurred compared to individuals at the front of the photo. Dilation of an image gradually increases the boundaries of an image, which is similar to some of the faces near the back of the class image. Dilation closely replicates a person under direct lighting [6]. Rotation was applied to images to replicate slight tilts of the face in the group and individual images and on some of the rotations additional augmentation was included such as making the image darker or brighter with gaussian blur.

Feature Extraction

Feature extraction is a common practice to solve computer vision problems such as object and face detection. It is a type of dimensionality reduction that represents the most important parts of an image [7]. For the purpose of this study, Histogram of Oriented Gradients (HOG) and Speeded-up Robust Features (SURF) will be used as two methods of feature extraction. Prior to feature extraction all the faces were resized to [80 x80] to allow for faces at the back of the class image. These faces at the back of the class are generally a smaller size; hence to try and increase the accuracy for the models, the image size was reduced.

HOG Features

Histogram of Oriented Gradients is generally used to extract features from images where it mainly focuses on the structure or the shape of the image. HOG extraction is similar to edge detection as it is able to extract the gradient and orientation of the edges [8]. The default value in MATLAB of cell size (pixel size), 8x8, was used for the HOG Feature extraction, as seen in Figure 3. To extract HOG Features, the RGB image has to be converted to grayscale. HOG features are said to have outperformed existing feature extractions for human detection [9]. HOG is widely used in pedestrian detection.

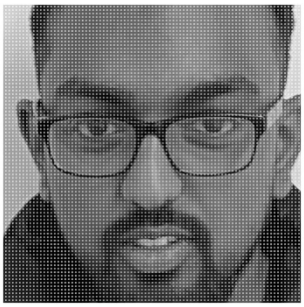

Figure 3 HOG Features extracted from a student's face using a cell size 8x8

SURF

Speeded-Up Robust Features (SURF) was a method proposed to improve Scale-Invariant feature transform (SIFT) descriptor for feature detection. SURF is seen to be much quicker and more robust compared to SIFT. This paper looks at extracting Features using the Bag of Words approach. The Bag of Word approach uses key points which are points that stand out in the image and extracts the features [10]. Using Figure 4 as an example, the key points for this individual is around the glasses and the teeth. The Bag of Words approach in MATLAB automatically converts the RGB images to gray scale in the function *bagOfFeatures*. Bag of Words is a technique that has been adapted for computer vision, originating from natural language processing. The bag of words feature extraction sues a grid strep of [8 8] and uses K – Means clustering to create group the key points together or more commonly known as “visual words”.

Figure 4 SURF Features extracted from a student's face

Classifiers

Three different classifiers have been used for to identify the faces in the test images. Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN) were used. Both the SVM and MLP use HOG and SURF features in the model, whereas CNN uses AlexNet which is a type of a CNN with eight deep layers.

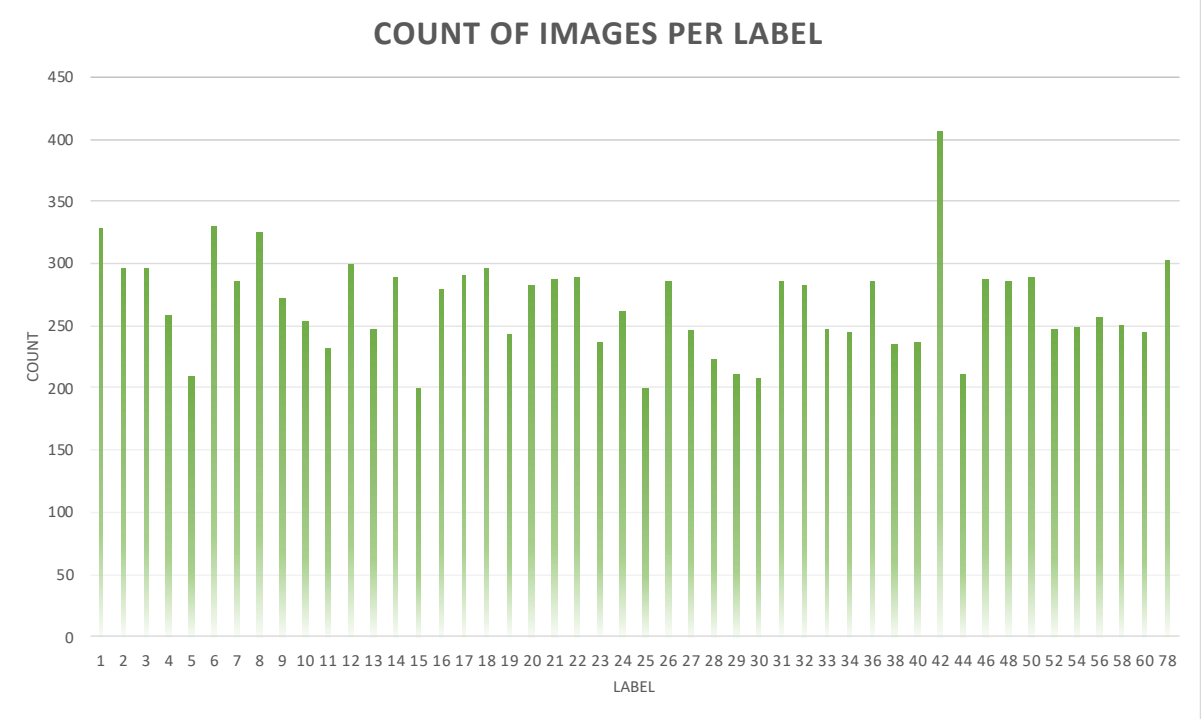

Figure 5 Number of images per label

Once face extraction is complete, not all labels have the same number of images. From Figure 5 we can see that label 42 had the most number of images saved after frame extraction and face detection, however Label 25 had the least. Label 25 had a hood on at the time of taking the pictures, which resulted in the Viola Jones object detection framework to perform poorly. The merge threshold and minimum size in the *Vision.CascadeObjectDetector* were varied as well as using different classification models, but no additional faces were detected. To tackle the issue of certain labels having a low number of faces detected, these images were copied multiple times. For this paper, we have set the minimum threshold as 200 images for each label. In addition, to combat the upper end, the datastore was limited to 200 images per label. This ensures that there is no class imbalance and would help with the accuracy rate .

Figure 6 Face extraction from Label 25

A 90:10 training test split was adopted to train the models to classify the individuals, which is generally seen as the most common training test split. Hyper parameter tuning was conducted on the validation dataset to identify the most optimal models to be used on the test (unseen) data. In total, 9,600 images were used, 8,640 for training and 960 for validation.

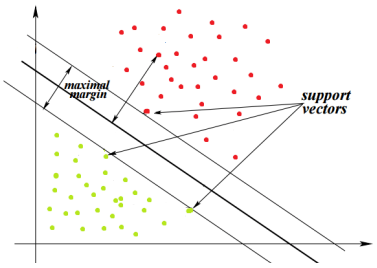

Figure 7 Example of SVM separating data points [11].

SVM

Support Vector Machine (SVM) is an algorithm that can be used for both classification and regression problems, but it is mainly used for classification problems. The SVM algorithm uses boundary points to separate the data points based on a classification or label. For this paper, the SVM algorithm tries to identify the faces for each label given. The aim of the SVM algorithm is to try and identify the optimal hyperplane as this can be used for predictions [11]. The optimal hyperplane is seen to have the largest margin as seen in Figure 7.

In MATLAB, to train the SVM model, the function *trainceoc* was used for both the HOG features model and SURF features model. An advantage of using the SVM model, is that computational time is low and it requires low memory. SVM also works well when there is a distinct separation between the labels, however for this task, SVM may not have been a suitable model. There are a number of individuals in the image dataset who look quite similar, so it might not be able to distinguish a clear separation between these individuals. The disadvantages of SVM include that it does not perform well with large datasets, which is this case for this dataset, which consist of 9,600 images. In addition SVM's do not perform well with noise, which occurs when there are similar faces and hence there will be misclassification of individuals.

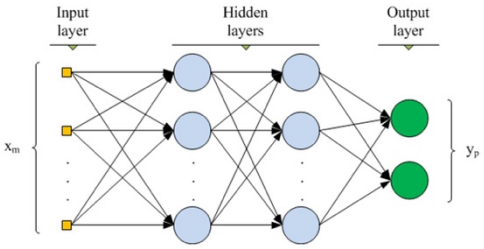

Figure 8 Example of a Multi-Layer Perceptron Network

MLP

Multi-Layer Perceptron (MLP) is a type of neural network. They are fully connected feed forward networks consisting of one or more layers of neurons where data is fed into the input layer and predictions are made on the output layer [12]. MLP are mainly used for classification problems where the inputs are assigned a class or a label. MLP's are generally used for pattern recognition, hence would be seen as a good model to use for image classification. A disadvantage for using MLP for computer vision problems is that as the number of dimensions increase, the model becomes very inefficient. The model was trained in MATLAB using the *patternet* function with a hidden layer size of 80 and training function Scaled Conjugate gradient backpropagation. Hyperparameter optimisation was performed to identify the highest accuracy for using both HOG features and SURF features.

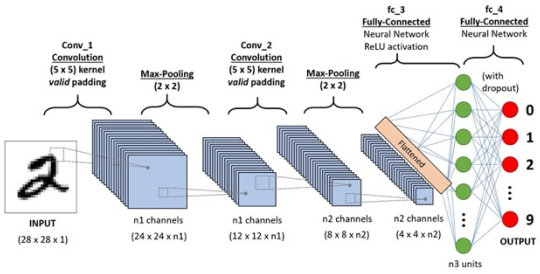

Figure 9 Example of CNN used to classify handwritten digits [13]

CNN

Convolutional Neural Networks (CNN) have been used more regularly in computer vision problems due to the advancements of the use of deep learning with in computer vision. Convolutional Neural Networks are able to take an image as the input and pass it through a number of layers such as a fully connected layer, SoftMax layer, drop out layer in order to classify the input image to its respective label. This project uses AlexNet which is a type of CNN.

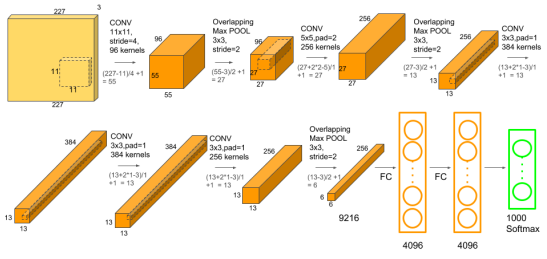

Figure 10 Structure of AlexNet [14]

AlexNet

AlexNet is a convolutional neural network that consists of eight deep layers, five convolutional layers and three fully connected layers [14]. AlexNet was developed by Alex Krizhevsky and competed in the ImageNet Large Scale Visual Recognition Challenge in September 2012. It was created as a solution for CNN's which struggled to perform well on high resolution images [15]. AlexNet has been trained on more than one million images and can classify various images into 1000 object categories [16].

A requirement for AlexNet is that input images should have a size of 227 x 227 x 3 that can be seen in Figure 11, which shows that the activation for the first layer is 227 x 227 x 3. The 3 determines that the image should be an RGB, in comparison to SVM and MLP which requires gray scale images.
Part of the process of using AlexNet the final three layers will need to be replaced by a fully connected layer, SoftMax layer and classification output layer.

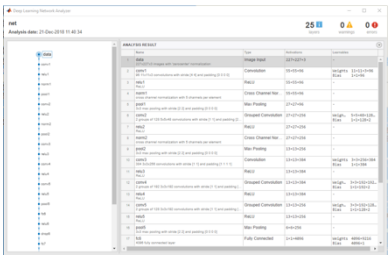

Figure 11 Visualisation of AlexNet network architecture [16]

Hyperparameter tuning was conducted on the AlexNet by varying different parameters to determine the optimal hyperparameters for the highest accuracy.

It was found that applying additional augmentation for the AlexNet, using the *imageDataAugmenter* caused the model to perform worse and predicted the correct labels for each individual. In addition, the number of epochs were reduced to 6 as after the first epoch the training and validation accuracy was very close to 100%. This shows signs of overfitting, hence a dropout Layer was included in the layers for the AlexNet, as a dropout layer is an effective regularisation method and helps prevent overfitting [17]. The optimal learning rate was found to be 0.0001, and validation frequency was 20.

Recognise Face

The requirement for this task is to create a recognise face function which returns a matrix P which describes the student(s) present in an RGB image. The P matrix is a N X 3 matrix, where by the first column is the ID of the individual, the second column represents the x co-ordinate of the centre of the face of the individual and the third column represents the y co-ordinate of the centre of the face of the individual. If no faces are detected the P matrix returned will be empty.

$$P = RecogniseFace(I, featureType, classifierType, creativeMode)$$

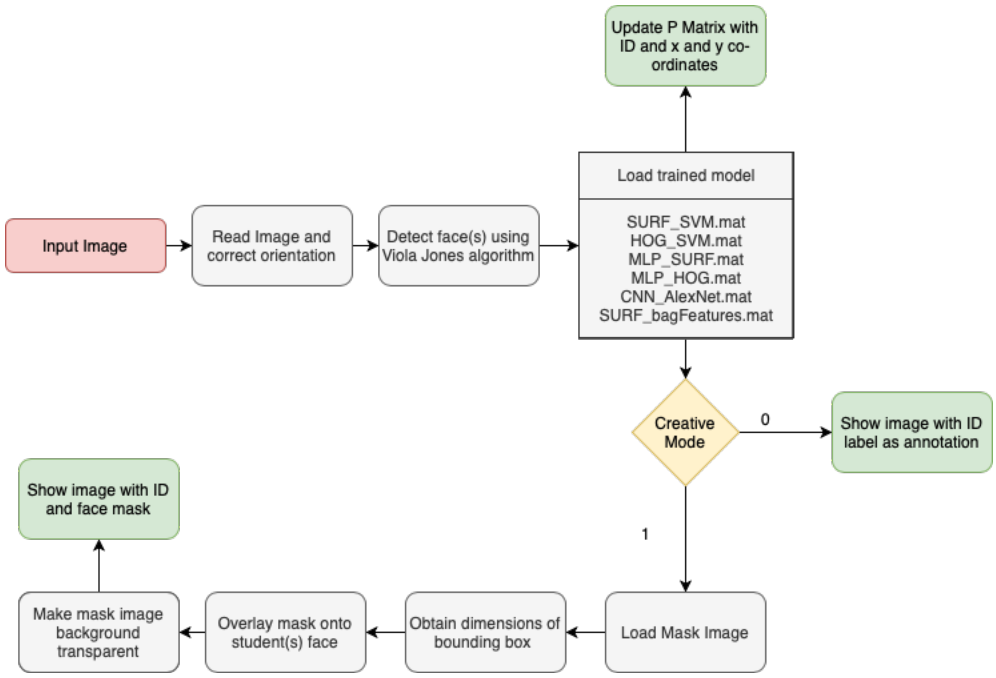

Figure 12 Flow diagram of Recognise Face Function

Results

Feature Type	Classifier Type	Validation Accuracy	Creative Mode
HOG	SVM	99.06%	1 or 0
SURF	SVM	92.36%	1 or 0
HOG	MLP	78.16%	1 or 0
SURF	MLP	76.21%	1 or 0
-	CNN	99.22%	1 or 0

Figure 13 Validation accuracy of different classifiers and feature types

Figure 13 shows how the different variables required can call the Recognise Face function. For the CNN/ AlexNet there are no feature types hence this is left blank. CNN is the best performing model on the validation set and this is also the case on testing images.

Feature Type	Classifier Type	Test Accuracy
HOG	SVM	74.29%
SURF	SVM	45.45%
HOG	MLP	72.90%
SURF	MLP	56.19%
-	CNN	81.25%

Figure 14 Accuracy of different classifiers and feature types

The accuracy of the different combination of feature types and classifier types were averaged across five group photos. It can be seen that CNN/AlexNet had the highest validation accuracy although the difference between SVM HOG and CNN was negligible. Hence, we would expect the testing accuracy for both models to be very similar. However, the CNN/AlexNet performed much better on these class photos compared to the other models. From Figure 15, we can see that the AlexNet performs much better on the right hand side image where all the individuals are facing straight on compared to the image on the left hand side, where a lot of the individuals' faces are at an angle. Even with different methods of augmentation, all the models struggled to correctly classify individuals who were at a distance from the lecturer who took the photo.

Figure 15 AlexNet model predicted individuals in class photo

In addition, the individual images that were fed into training the networks were taken in a different location to the class image, hence the different lighting and backgrounds have to be taken into consideration. It was surprising that initially the models, performed poorly when no augmentation was applied to the images. Once the different augmentation techniques were applied, as described above, the models started to predict the individuals correctly. Also, the faces at the back of the class

were a lot smaller than the ones at the front and they were blurry hence the models did not perform as well for these individuals.

As seen in Figure 15, there are a number of individuals in the left hand side that have the same label, such as the individuals which have the label '04'. They look very similar hence the model could not correctly identify the difference between the two. In addition, as these two individuals are near the back of the photo, their resolution is very poor, which is understandable to why the CNN/AlexNet model could not distinguish between the two.

All the models were able to predict nearly everyone correctly when testing unseen individual images, however this could be because there are a lot more individual images, hence the performance was better compared to the class photos.

Surprisingly, there was a large difference in accuracy when using SURF features compared to HOG features for the SVM model, with the SVM model using SURF features performing poorly. This was expected as HOG features generally perform better than SURF features for human detection [9].

Creative Mode

The Recognise Face function has another variable input of creativeMode, when set to 1 a mask will be superimposed on top of each student(s) face. The mask that will be superimposed is the 'Guy Fawkes mask'.

Figure 17 Superimpose Face Mask onto Student's Face

Figure 16 Superimpose Face Mask onto multiple Students face

The idea behind the creative mode, is to utilise the recognise face function to identify the faces in the picture and draw a bounding box around the face. The properties of the bounding box used are the x and y coordinates as well as the width and height of the bounding box. The mask has to be resized to the bounding box to successfully superimpose the mask onto the students' faces.

As the background of the face mask was black, the black background pixels were replaced by the student's face to successfully place the mask on the face.

Conclusion and Future Work

In this paper, we trained three different classifiers, SVM, MLP and CNN/AlexNet with different features, SURF and HOG in order to correctly identify faces in test images. There were many pitfalls of this project, such as there were many individuals who were in the class photos but did not have an

individual photo or vice versa. There were individuals who were incorrectly classified because of this and surprisingly there were some individuals who looked very similar, hence the models were not able to correctly distinguish between them. For future work, the individuals that did not have their individual photos taken, they could have been given a label of 'Unknown' in the class photo; unfortunately this was out of scope for this project.

In addition, the training time for the CNN/AlexNet was just over seven hours, due to completing this work on a laptop which did not have a NVIDIA GPU. If a machine with a GPU was used, the hyperparameters for the CNN/AlexNet model could have been finely tuned to improve the accuracy as well as reducing calculation time. In addition a larger dataset could be used if a GPU was used and this could also help improve the accuracy for each of the models

The initial step of sorting each individual into their respective labelled folder was very time consuming. Instead of doing a manual procedure, Optical Character Recognition (OCR) could have been applied. Due to stability and high accuracy errors the OCR can produce, this was not completed for this project.

There were only two feature extractions explored within this paper, it would be interesting to see how other feature extractions perform with different models such as Scale – Invariant Feature Transform (SIFT) or Local Binary Patterns (LBP). In addition other classifier types such as Random Forest, R – CNN, Fast R-CNN, Faster R – CNN and You Only Look Once (YOLO) can be looked into to see if the accuracy increases for face detection. These models are increasingly being used in object detection in the computer vision field.
Regarding the Creative Section of this project, the mask was overlaid on top of the face, however if the face was slightly tilted or at a different angle, the mask did not sit right on the face. A future work would be to try and use segmentation to correctly look at the angle of the face to make sure the mask looks correct on the face.

References

[1] Tarroni, G., 2020. *Computer Vision: Lecture 1*. [online] Moodle.city.ac.uk. Available at: <https://moodle.city.ac.uk/pluginfile.php/1863415/mod_resource/content/3/Lecture%2001.pdf>.
[2] Thalesgroup.com. n.d. *Facial Recognition In 2020 (7 Trends To Watch)*. [online] Available at: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>.
[3] Gayle, D., n.d. *Met Police Deploy Live Facial Recognition Technology*. [online] the Guardian. Available at: <https://www.theguardian.com/uk-news/2020/feb/11/met-police-deploy-live-facial-recognition-technology>.
[4] Perez, Luis & Wang, Jason. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning.
[5] Homepages.inf.ed.ac.uk. n.d. *Spatial Filters - Gaussian Smoothing*. [online] Available at: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>.
[6] Homepages.inf.ed.ac.uk. n.d. *Morphology - Dilation*. [online] Available at: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>.
[7] Uk.mathworks.com. n.d. *Feature Extraction*. [online] Available at: <https://uk.mathworks.com/discovery/feature-extraction.html>.
[8] Singh, A., 2019. *Feature Engineering For Images: Introduction To HOG Feature Descriptor*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>.
[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.
[10] Davida, B., 2018. *Bag Of Visual Words In A Nutshell*. [online] Medium. Available at: <https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb>.
[11] HackerEarth Blog. n.d. *Simple Tutorial On SVM And Parameter Tuning In Python And R | Hackerearth Blog*. [online] Available at: <https://www.hackerearth.com/blog/developers/simple-tutorial-svm-parameter-tuning-python-r/>.
[12] H. Boughrara, M. Chtourou and C. B. Amar, "MLP neural network based face recognition system using constructive training algorithm," 2012 International Conference on Multimedia Computing and Systems, Tangier, 2012, pp. 233-238.
[13] Saha, S., n.d. *A Comprehensive Guide To Convolutional Neural Networks — The ELI5 Way*. [online] Medium. Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
[14] Learnopencv.com. n.d. [online] Available at: <https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png>.
[15] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.
[16] Tarroni, G., 2020. *Lab 09: Object Detection And Instance Segmentation*. [online] Moodle.city.ac.uk. Available at: <https://moodle.city.ac.uk/pluginfile.php/1895298/mod_resource/content/1/Lab09.pdf>.
[17] Brownlee, J., n.d. *A Gentle Introduction To Dropout For Regularizing Deep Neural Networks*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>