Is Face Recognition with Masks Possible?

Yaaseen Muhammad Saib[1], Sameerchand Pudaruth[2]
ICT Department, FoICDT
University of Mauritius
Reduit, Mauritius

***Abstract*—With the recent outbreak of the COVID-19 pandemic, wearing face masks has become extremely important to protect us, and to reduce the spread of the virus. This measure has made many existing face recognition systems ineffective as they were trained to work with unmasked faces. In this paper, several methods have been proposed for masked face recognition. Two pre-trained deep learning architectures (VGG16, and MobileNetV2) and the Histogram of Gradients (HOG) technique were used to extract the relevant features from face images of celebrities. A SoftMax layer and Support Vector Machines (SVM) were used for classification. Five scenarios were devised to assess the different models and approaches. With an accuracy of 96.8%, the best model was obtained with MobileNetV2 with a SoftMax layer on the dataset consisting of a mixture of masked and unmasked images. Three different types of masks were also used in this study. The mean accuracy was 91.35% when the same type of mask is used for training and testing. However, the accuracy dropped by an average of 5.6% when a different type of mask is used for training and testing. A contactless attendance system using the best masked face recognition model has also been implemented.**

Keywords—Face detection; face recognition; face mask; deep learning; VGG16; MobileNetV2; HOG

I. Introduction

There are several biometric systems available that can be used to secure access to data but in this work, the focus is on face recognition systems. Symanovich defined face recognition as the process of using the face of an individual from a photo or video to verify their identity [1]. Klosowski explained how this technology is being used around the world for many purposes such as unlocking mobile phones and laptops, monitoring people's physical access to restricted areas such as high-tech laboratories or even taking attendance in lectures [2]. Blokdyk explained the main processes in the face recognition system: face detection, feature extraction and classification [3]. The process of taking an image and locating the region that contains the face only is known as face detection. This region is then stored as a set of coordinates representing a bounding box around the detected faces. This is a very challenging task since faces in different images have many variations with regards to facial expressions, pose, degree of occlusions and lighting conditions [4].

The world is suffering from the outbreak of COVID-19, a contagious virus spreading from person to person [5]. People can become infected by getting into contact with an infected patient or by touching contaminated surfaces. Traditional systems such as passwords and fingerprints require contact with a surface and are therefore not secure when it comes to the transmission of the coronavirus while face recognition does not require any physical contact and can therefore be considered a safer approach in the current context. The United States Centers for Disease Control and Prevention has stated that the best way to prevent spreading of the virus is to avoid social contact and to wear a face mask [6]. However, face masks have made many existing face recognition systems fail. Face recognition systems usually use the geometry of the whole face including the nose and mouth, but this is now covered with a mask and therefore this makes the process more challenging. Furthermore, it is unsafe if the users have to remove their masks each time to verify their identity. Existing face recognition technologies have an accuracy rate of 97.7% for unmasked faces whereas for masked faces, the accuracy drops to 50% and sometimes the algorithm fails completely which makes the existing technology very inefficient. Furthermore, different mask shapes and colours also affect the accuracy of the face recognition systems [7].

The prime objective of this work is to develop a system to allow masked faces to be recognized with a high degree of accuracy. Several methods have been devised and tested to find the most suitable one for this problem. A classroom attendance system based on the best model was also implemented. This attendance system can also be used in different places. This paper proceeds as follows. Section 2 provides an overview of related works on masked face recognition. The methods, algorithms and datasets are described in Section 3. Implementation details are provided in Section 4. The results and their evaluation are discussed in Section 5. Section 6 concludes the paper.

II. Literature Review

In this section, we provide an overview of works that have been done on unmasked and masked face detection. Ejaz and Islam developed a masked face recognition system using transfer learning [8]. They used the AR and IIIT-Delhi Disguise Face Database datasets on which data augmentation was performed. MTCNN was used to detect and align masked faces. The face regions were cropped and resized to 160*160 images. Google FaceNet model combined with a deep CNN was used to extract features to be classified with SVM. For training and testing purposes, they used a ratio of 0.7 and 0.3 respectively. The system was tested with multiple scenarios and the average test accuracy obtained was 82.5%.

Wang et al. devised a method to improve the performance of face recognition systems by re-training them to recognize masked faces [9]. They proposed three datasets: MFDD, RMFRD and SMFRD. MFDD contains 25000 masked faces downloaded from the internet. RMFRD consists of 525 subjects with 5000 masked images and 90000 unmasked images. The last one was a software generated dataset. They developed a mask simulation software that adds virtual masks to faces. It performs face detection and alignment using the 68 face landmarks shape predictor. This allows retraining of existing face recognition systems to recognize masked images achieving an accuracy of 95%.

Hariri developed a system that performs masked face recognition whereby occluded regions of the face are discarded [10]. Firstly, face detection is performed on the image followed by face alignment. The image is then resized to 240*240 pixels. The image is cropped to keep the eye region. Features are then extracted using VGG16 and passed to the MLP classifier. RMFRD dataset was used to test the system. The highest accuracy obtained was 91.3%.

Anwar and Raychowdhury developed a system to convert existing face datasets to masked face datasets by using MaskTheFace, an open-source tool [11]. This dataset was then used to retrain the existing face recognition systems. Facenet face recognition was used to test the effectiveness of their masked dataset. After implementation, they reported an increase of approximately 38% in the true positive rate. The same was also achieved when tested using the real-world dataset MFR2. To train the program, they used a subset from the VGGFace2 dataset and applied the MaskTheFace tool to add virtual masks to the images. The accuracy achieved varied between 86% and 93%.

Li et al. implemented a system that focused mainly on the upper half face [12]. To extract features, ResNet50 was used to assign more expressive weights to the region of the eyes and lower weights to the occluded regions. Furthermore, they cropped the face at different levels to find the optimal cropping that would provide better results. They first discarded the bottom 50%, 30% and 10% of the image. They concluded that dropping the bottom 30% provided the best accuracy of 82.5%. Tests were performed on several datasets: AR Dataset, Extend Yela B Dataset and LFW dataset and a recognition rate between 81.4% and 92.6% was achieved.

Alyuz et al. devised a method to allow face recognition systems to work with partially occluded faces [13]. A technique called masked projection was used that analysed the face for occlusions and excluded them from the image. The occlusions are detected on a face by comparing them to a threshold value of distances on a non-occluded face. An alignment process is also done by comparing the centre of the image to one of an aligned face. Any necessary pose corrections are performed by the ICP algorithm. For training purposes, there is an independent matrix for each non-occluded region of the face that represents a subspace. The software is trained for each of the subspaces and when a face is to be recognized, the same process is applied and is then compared against corresponding subspaces of non-occluded regions. A recognition rate of 90% was achieved.

A system to recognize partially occluded faces with different poses was developed by Bagchi et al. [14]. Weighted median filters were applied to the dataset to remove noise. The faces are converted to data using the ICP algorithm. Occluded regions are detected from the face by comparison to a normalized face and information about those regions are discarded. The occluded regions detected are then restored to obtain a full face. This process is done by taking data from a normalized face and using the necessary regions. Lastly, feature extraction is performed, and the images are classified. The highest accuracy achieved was 91.3% .

Shepley developed a face recognition system using deep learning [15]. For face detection, DCNN was used which outperformed Haar Cascades and LBPs due to the large databases available. Face alignment was performed followed by extraction of features used to train a DCNN. To recognize unknown faces alignment and feature extraction is performed again. The encodings are then used for similarity comparison between the gallery faces and the face to be recognized. DeepFace, FaceNet and VGG-Face datasets were used to test the program and the recognition rate varied between 75% and 99%.

Parkhi et al. designed a system using deep learning to detect and recognize single or multiple faces from images and videos [16]. As for the model, CNN was used consisting of 11 blocks each having a linear operator and max pooling layers. The last 3 layers consisted of filters to match the size of the data. Data for 2500 male and 2500 female was collected to train and test the program. To evaluate the system, LFW and YTF datasets were used and a recognition rate of 96.0% was achieved.

Ge et al. developed a face recognition system to detect masked and occluded faces [17]. 35806 masked faces and 30811 unmasked images were downloaded from the internet. To detect faces, two pre-trained CNNs were combined for the extraction of features from input images which were then converted to a similarity-based descriptor by making use of the LLE algorithm and a dictionary that contains data of masked faces and synthesized normal faces. This allows facial landmarks from occluded regions to be recovered. An improvement of 15.6% was achieved on state-of-the-art at that time. Chowdary et al. developed a system that performs face mask detection to identify individuals who were not wearing a mask with a very high accuracy [18]. Image augmentation was performed on the SMFD dataset to increase the size of the training data. DNN was used for the image classification process. The Inception-v3 deep learning architecture was used to enhance the performance of the neural network.

Rekha and Chethan developed a face recognition system to take attendance automatically using live video [19]. Viola and Jones algorithm was used for face detection. The face region is cropped, and a correlation technique is used to recognize the face by comparing it to trained images. Finally, the attendance registry is updated for the recognized faces. Several tests were performed with different scenarios and the average face recognition rate achieved was 90%. Varadhrajan et al. also designed a face recognition system to take attendance [20]. The faces in the image are detected and cropped separately. For recognition, the eigenvalue method was used. An accuracy of 93% and 87% was achieved for face detection and recognition, respectively.

A large number of works on masked and unmasked face detection and recognition have been reviewed. While the majority of works has been done on unmasked faces, there are also a number of works that had been done on masked faces and on faces with different types of occlusion. There has also been a gradual and consistent increase in the accuracy of these systems.

III. Methods

The main objective of this study is to perform face recognition on masked faces. In this section, a solution has been proposed to overcome the main challenge of performing face recognition on masked faces. After acquiring the dataset, hybrid sampling is used to bring equality among all the classes in the dataset. Face detection is performed on the dataset to keep the face region only and discard any unnecessary information. This new dataset contains unmasked faces. Several versions of this dataset are created. Machine learning and deep learning algorithms are then used for extracting the relevant features before recognition is performed. The model is evaluated using standard performance measures. This set of steps is shown graphically in Fig. 1.

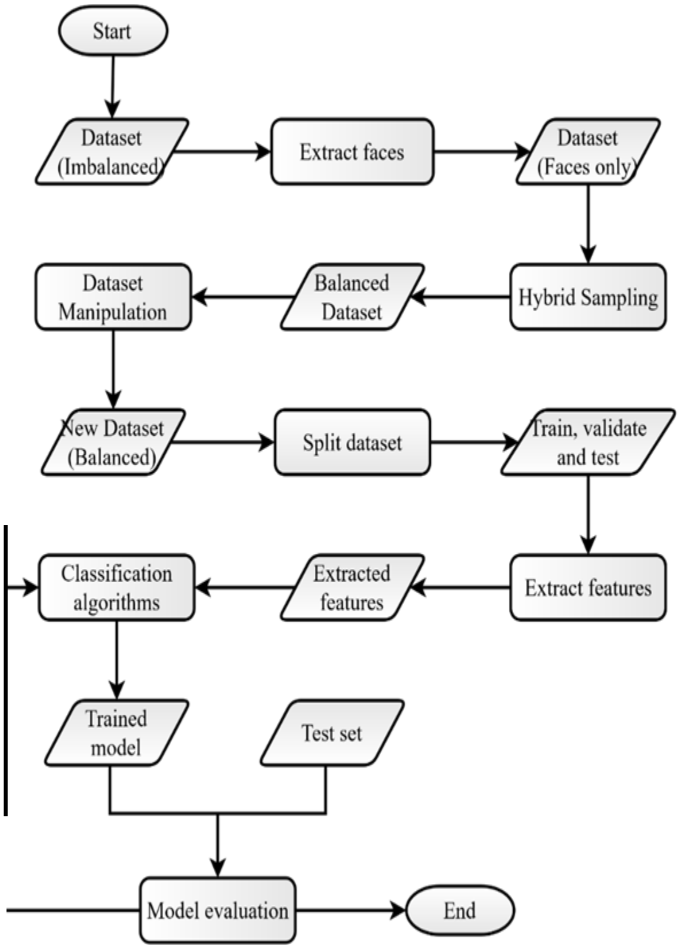


Fig. 1. Face Recognition with Masks.

Fig. 2. Sample Images from the Celebrity Dataset.

The Pins Face Recognition dataset was used in this study [21]. This dataset consists of 17,534 faces of 105 celebrities collected from Pinterest. The images are cropped to keep the face region only. There is an average of 150 unmasked images for each person. The images were taken in slightly different poses and different lighting conditions. This dataset was augmented with more celebrity images from the internet. Our final dataset consists of 170 persons with an average of 150 images per subject. Sample images from this unmasked celebrity dataset are shown in Fig. 2. For subjects with more than 150 images, undersampling was done by discarding the extra images and for subjects with less than 150 images, oversampling was done by adding slightly processed versions of existing images.

Four different variations of the original dataset were created. In the first one, a virtual mask is applied to all the images in the dataset using MaskTheFace [11], as shown in Fig. 3. In the second scenario as shown in Fig. 4, another variation of the dataset containing both masked and unmasked faces were created. In the third scenario, the images are cropped to keep the upper half of the face only i.e. the eyes and forehead regions only as shown in Fig. 5.

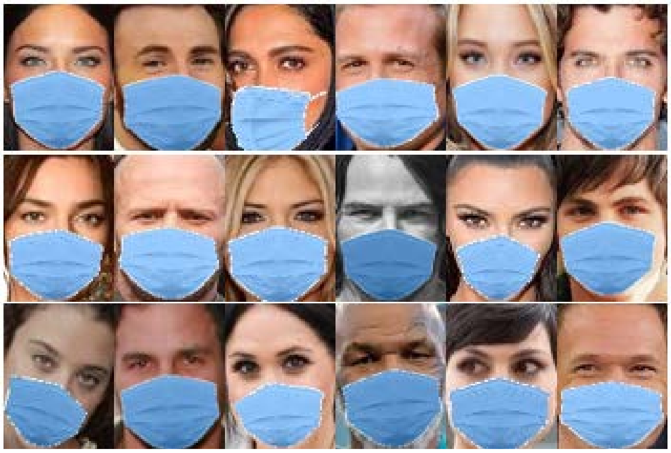


Fig. 3. Masked Faces.

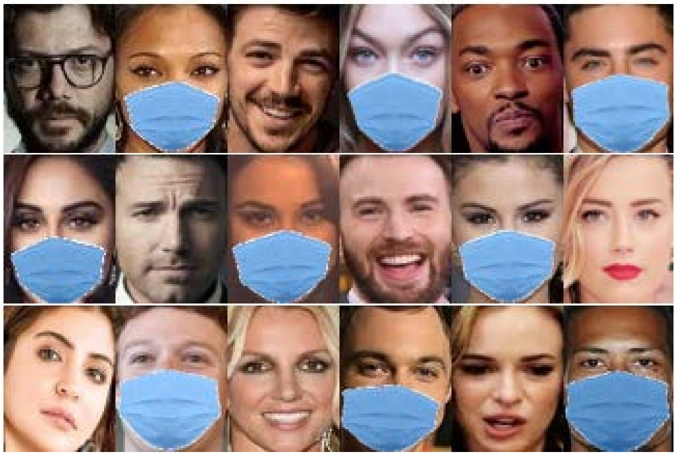


Fig. 4. Mixture of Masked and Unmasked Faces.

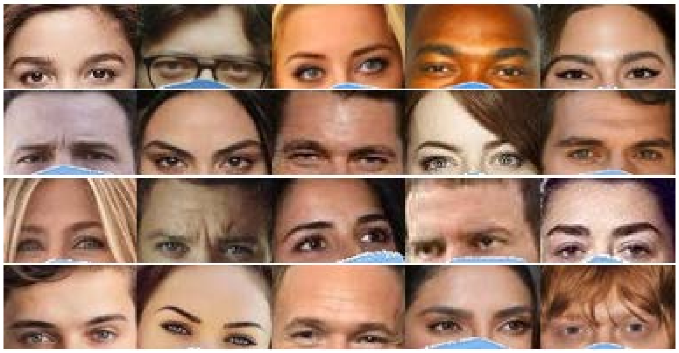


Fig. 5. Eyes and Forehead Regions Only.

In the fourth scenario, firstly face detection is performed to keep the face region only. Then for each face image, the bottom half is replaced with the bottom half face of another person. The same bottom half face is used for all images as shown in Fig. 6.

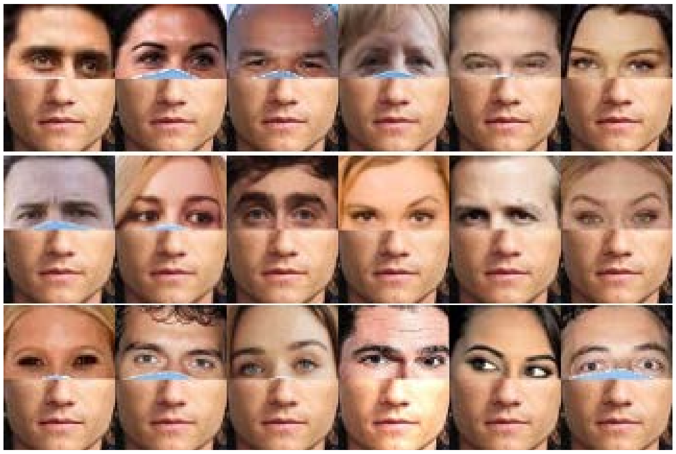


Fig. 6. Face Bottom Half Face.

In this section, we have described all the different steps in the face recognition system. We also described the dataset that we have used and how it was manipulated to produce four other datasets. The next section will provide implementation details to shed more light on how the system was developed.

IV. Implementation

This section aims to describe the different components of the system, the hardware and software requirements and the additional tools and facilities that are required to find the best masked face recognition system and to implement the attendance system. The libraries and the tools used for the development of the system are shown in Table I.

TABLE I. TOOLS

Tools	**Description**
OpenCV	Open-source library for image processing and machine learning [22].
NumPy	Powerful library used to handle matrix and multi-dimensional arrays. Also used for scientific mathematical operations [23].
Tensorflow	Performs rapid numerical calculations and allows the development of machine learning and deep learning models [24].
Keras	API used for deep learning and multiple back end deep learning is also supported [25].
Scikit-Learn	Support for a variety of unsupervised and supervised learning algorithms [26].
Face_recognition	Performs face detection, encoding and recognition [27].
Pillow	Powerful library written in python offering a wide range of image processing techniques [28].
Notebook PC	Windows OS, Intel i7 CPU, 8G RAM, NVIDIA GeForce GTX850M
Google Colab Pro	Training, validating, and testing the deep learning models

Firstly, a pre-trained deep learning model such as VGG16 and MobileNetV2 is loaded using cv2.dnn.readNet (modelFile, configFile) from the OpenCV library. The image is loaded using cv2.imread and then passed through a blob that performs pre-processing and normalization tasks using cv2.dnn.blobFromImage (image, scalefactor = 1.0, size, mean) where *image* is the input image, *scalefactor* is the value through which the image will be scaled, *size* is the dimensions of the image and *mean* is the mean RGB value of the pixels. The blob is then passed through the network to obtain the relevant blobs using net.setInput(blob). For each blob, a probability is calculated and if it is less than a specified value, the blob is ignored. If the probability is higher or greater than the specified values, it is considered to form part of the face region. Face detection is performed on all images in the original dataset. Feature extraction is performed using HOG and deep learning (VGG16 and MobileNetV2) architectures. For classification, a SoftMax layer and SVM have been used. For SVM, a linear kernel was used. All the eight scenarios are shown in Table II.

TABLE II. SUMMARY OF MODELS

Feature Extraction	Trainable Layer	Classifier
MobileNetV2	True	SoftMax
	False	SoftMax
VGG16	True	SoftMax
	False	SoftMax
VGG16	True	SVM
	False	SVM
HOG	-	SoftMax
	-	SVM

The face recognition system has been used to implement an attendance system. This system consists of five modules: Register Student, Train Model, Modules, Take Attendance and Exit, as shown in Fig. 7.

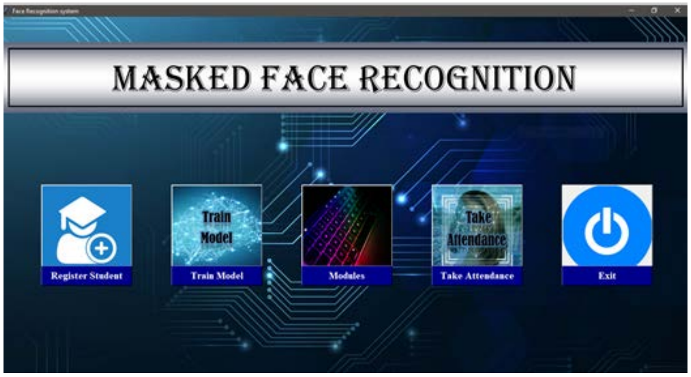


Fig. 7. Homepage of the Attendance System.

The Student Register module is used to register a student either by uploading an image or capturing one image using a webcam, as shown in Fig. 8.

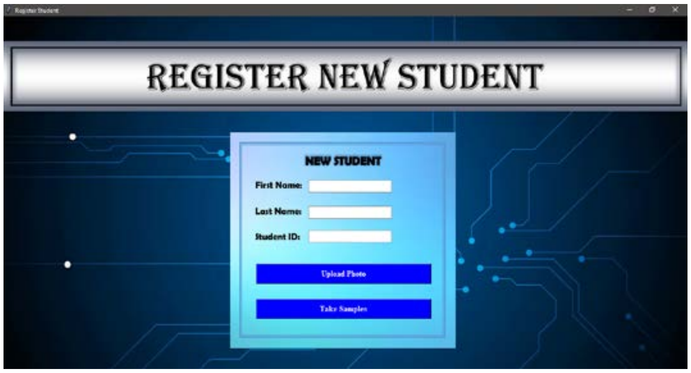
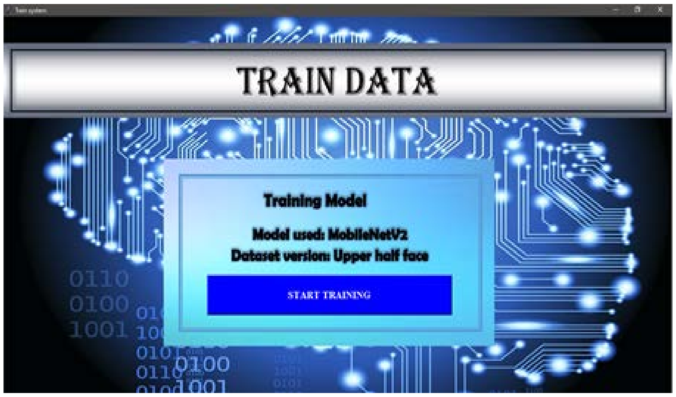


Fig. 8. Register Student Module.

The Train Model module is used to train the model based on the dataset, as shown in Fig. 9. The Modules management feature allows a user to add or delete courses for attendance purposes, as shown in Fig. 10. And the Take Attendance module is used to record attendance by uploading images or capturing one using a webcam, as shown in Fig. 11.

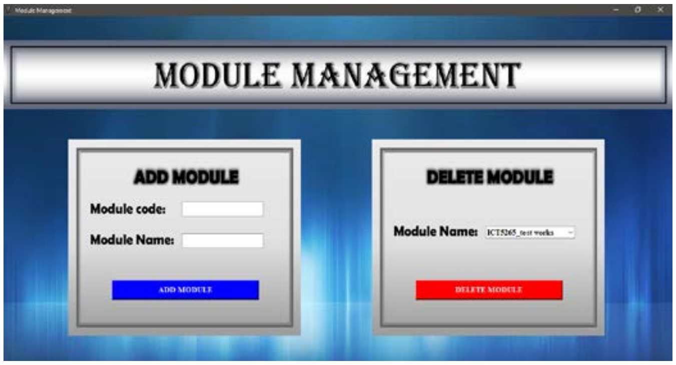


Training Model

Model used: MobileNetV2

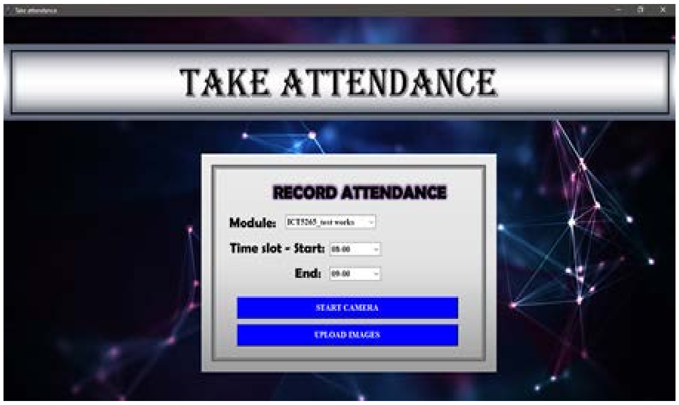
Dataset version: Upper half face

Fig. 9. Train Model Module.



Module Name: ICT5265_test works

Fig. 10. Module Management.



Module: ICT5265_test works

Time slot - Start: 08:00

End: 09:00

Fig. 11. Take Attendance Module.

V. RESULTS AND EVALUATION

In this section, all the models implemented are tested and evaluated with the five types of datasets to find the limitations of the models and ultimately determine the most suited model and type of dataset. The face recognition API is also tested and evaluated for applicable datasets. This section is divided into five parts, one for each type of dataset. The performance of each model is evaluated using classification accuracy. The dataset consists of 10,200 images, 170 subjects each having 60 images. Before training each model, the dataset was split into 2 sets, 0.9 for training and validation and 0.1 for testing. The training set was further split into two more sets, 0.8 for training and 0.2 for validation.

A. Unmasked Faces

After testing all the models with the unmasked dataset, the accuracy values obtained can be observed in Table III. The model performs better when features are extracted using transfer learning having trainable layers. The two best accuracy scores obtained were 93.82% and 95.59% with the VGG16 and MobileNetV2 models, respectively. Both the scores were obtained when the layers in the two models were set to trainable. For VGG16, there is a decrease of 3.06% in the accuracy score when the layers of the model were set to non-trainable. Furthermore, it can be observed that the accuracy decreases by 12.65% when HOG was used for feature extraction compared to the MobileNetV2 model. The SVM classifier results in a lower accuracy score compared to a SoftMax classification layer.

TABLE III. UNMASKED FACES

Model	Trainable Layer	Accuracy
VGG16 + SoftMax	True	0.9382
	False	0.9039
VGG16 + SVM	True	0.8833
	False	0.8833
MobileNetV2	True	0.9559
	False	0.7676
HOG + SoftMax	-	0.8294
HOG + SVM	-	0.8456

B. Masked Faces

All the accuracy scores obtained when testing all the models with the masked dataset are recorded in Table IV. The two best accuracy scores of 93.24% and 94.12% were obtained when VGG16 and MobileNetV2 were used. MobileNetV2 performed slightly better than VGG16. When HOG was used with and without SVM, an accuracy of 77.75% and 78.14% were obtained, respectively. When the SVM classifier was used to classify features obtained from the VGG16 model, the accuracy obtained was 85.25%, 7.99% less than the accuracy obtained when NN was used to classify the same features.

TABLE IV. MASKED FACES

Model	Trainable Layer	Accuracy
VGG16 + SoftMax	True	0.9324
	False	0.8480
VGG16 + SVM	True	0.8525
	False	0.8525
MobileNetV2	True	0.9412
	False	0.7745
HOG + SoftMax	-	0.7814
HOG + SVM	-	0.7775

C. Unmasked Faces and Masked Faces

The accuracy scores obtained with the different models when tested on the dataset consisting of an equal number of masked and unmasked faces are recorded in Table V. It can be observed that the two best accuracies obtained are 91.37% and 96.76% for the VGG16 and MobileNetV2 models, respectively. When the trainable layer in MobileNetV2 model is set to true, the accuracy drops to 79.51%. This is a significant difference of 17.25%. When the features were extracted using HOG and classified using SVM and SoftMax, the accuracies obtained are 90.83% and 85.10%, respectively. We observe that mixing the masked and unmasked dataset leads to a slightly higher accuracy than when using only the masked or unmasked datasets separately.

TABLE V. UNMASKED FACES AND MASKED FACES

Model	Trainable Layer	Accuracy
VGG16 + SoftMax	True	0.9137
	False	0.8814
VGG16 + SVM	True	0.9074
	False	0.9074
MobileNetV2	True	0.9676
	False	0.7951
HOG + SoftMax	-	0.8510
HOG + SVM	-	0.9083

D. Upper Half Face Only

The dataset consisting of the upper half face only is tested with all the models built and the accuracies are shown in Table VI. The two best accuracies achieved are 95.29% and 89.61% with the MobileNetV2 and VGG16 models, respectively. When the layers in the MobileNetV2 model were not set to trainable, the accuracy dropped to 75.49% which is 19.80% less than when the model has its layers set to trainable. The lowest accuracy of 71.08% was obtained when features were extracted using HOG and classified with a SoftMax layer. To conclude, the highest accuracy achieved with this dataset is 95.29% with the MobileNetV2 model whose layers were set to trainable during training. With this dataset, the model had fewer features to extract and classify compared to the unmasked dataset and the accuracy achieved is lower by only 0.30%.

TABLE VI. UPPER HALF FACE ONLY

Model	Trainable Layer	Accuracy
VGG16 + SoftMax	True	0.8961
	False	0.8206
VGG16 + SVM	True	0.8137
	False	0.8152
MobileNetV2	True	0.9529
	False	0.7549
HOG + SoftMax	-	0.7108
HOG + SVM	-	0.7456

E. Upper Half Face and Fake Lower Half Face

The dataset consisting of the original upper half face with a fake lower half face added to cover the masked region was tested with all the models and the accuracies achieved were recorded in Table VII. The best accuracy achieved is 93.33% and 93.04% with VGG16 and MobileNetV2, respectively, when both have their trainable layer set to true. For the VGG16 model, when the layers are set to non-trainable the accuracy dropped to 82.65, which is 10.68% less than when the trainable layer is set to true. When HOG is used, the accuracy dropped even further to 68.92%.

TABLE VII. UPPER HALF FACE AND FAKE LOWER HALF FACE

Model	Trainable Layer	Accuracy
VGG16 + SoftMax	True	0.9333
	False	0.8265
VGG16 + SVM	True	0.8294
	False	0.8294
MobileNetV2	True	0.9304
	False	0.6480
HOG + SoftMax	-	0.6892
HOG + SVM	-	0.7569

HOG feature extraction consistently resulted in lower performance and accuracy because it is a standard feature extractor, and it applies the same procedures to any given image. It determines the number of edges and their orientations region by region of the image and forms a collection of histograms of pixel orientations. When transfer learning such as VGG16 and MobileNetV2 are used, they extract features that are more specific and complex depending on the data on which they are training. Ultimately, they obtain the optimal feature space to achieve better performance [29].

A higher accuracy was achieved for all types of datasets when the layers of the pre-trained model were set to trainable since the features to be extracted from a face are more specific. Using pre-trained weights on face datasets does not give the best results since the model is trained on ImageNet dataset which is completely different and hence the need to retrain the model to optimize the feature space completely and adapt it specifically to this dataset. However, it takes more time to train the program since all the layers have to be updated but it achieves better performance [30].

In general, the SoftMax classifiers performed better than the SVM classifier. The average accuracy obtained with the SVM classifier was 83.23%. Luca explained why SoftMax generally performs better than SVM [31]. It can be observed that the MobileNetV2 model is robust as it consistently achieves the highest accuracy for each dataset except for the last dataset. The second most robust is the VGG16 model which has the second-best accuracy for each dataset. From Table V, it can be deduced that the dataset consisting of both masked and unmasked faces yielded the highest accuracy.

F. Mask Type

Different mask types are applied to faces in the testing set to observe whether the type of mask used affects the performance of the system. The different tests performed, and the results obtained are shown in Table VIII.

TABLE VIII. IMPACT OF MASK TYPE ON ACCURACY

Type of mask: Training	Type of mask: Testing	Accuracy
Surgical	Surgical	0.9412
	N95	0.8498
	Cloth	0.8756
N95	N95	0.9562
	Surgical	0.8726
	Cloth	0.9294
Cloth	Cloth	0.9531
	Surgical	0.9184
	N95	0.9194

G. Comparison with Existing Works

The dataset consisting of masked and unmasked faces yielded the highest accuracy. However, both the training and testing data had similar mask types. When tested with different types of masks, the accuracy decreases by approximately 5.6%. With an accuracy of 99.2% on unmasked faces, the Python face recognition API (face_recognition) outperforms all the implemented models. However, it cannot process half faces or masked faces, and therefore we added a fake bottom unmasked half face to the images. By doing this, we were able to make the API work and achieved an accuracy of 87.56% for masked face recognition.

Ejaz and Islam used CNN for feature extraction and SVM for feature classification and achieved an accuracy of 82.5% [8]. Wang et al. proposed a method whereby all existing face recognition systems have to be retrained by adding virtual masks to the faces in the existing datasets [9]. However, in this work, we saw that the accuracy drops by an average of 5.6% when face masks are used. Hariri developed a system whereby only the upper half face is used and the accuracy achieved was 91.3% using the RMFRD dataset which consists of 525 subjects [10]. Our proposed system was tested with the same type of dataset but consisting of only 170 subjects and the highest accuracy achieved was 95.3%. Hariri had used a VGG16 model while our best system requires MobileNetV2 [10]. The system developed by Anwar and Raychowdhury was tested using a dataset containing 42 images per person while our model was tested with 60 images per person [11]. They used the Inception-ResNet v1 architecture. The system was evaluated only on a dataset of masked images. This limits the system to perform well only with masked faces and is less effective with unmasked images. Our system performs equally well on both masked and unmasked faces.

VI. Conclusion

The COVID-19 pandemic has imposed the wearing of face masks in public places as well as in workplaces. This has created some difficulties for systems that were not trained to handle masked faces. Moreover, the wide varieties of face masks that are available make the face detection and recognition even more difficult. The objective of this work was to find out whether it is possible to recognise masked faces with a high degree of accuracy. Thus, five different variations of a celebrity dataset were created. Several feature extraction methods such as HOG and pre-trained deep learning models were used. The final classifications were made using a SoftMax function and SVM. The dataset consisting of the upper half face only may be deemed to be the more suitable one for practical applications since it has a reasonably high accuracy of 95.29% and it recognizes both masked and unmasked faces. Moreover, the type of masks used does not affect this system since the bottom half of the face is not taken into consideration. This system was further used to implement an attendance system. This face recognition system can be enhanced so that it can distinguish between real and fake faces in real-time. The attendance system can also be further developed so that it generates attendance reports automatically and send them to the required personnel via an email messaging system.

References

[1] Symanovich, S. (2019, February 8). How does facial recognition work? Retrieved May 28, 2021, from Norton: https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html.

[2] Klosowski, T. (2020, July 15). Facial Recognition Is Everywhere. Here’s What We Can Do About It. Retrieved May 28, 2021, from New York Times: https://www.nytimes.com/wirecutter/blog/how-facial-recognition-works/.

[3] Blokdyk, G. (2018). Facial Recognition A Complete Guide. 5STARCooks.

[4] Dubey A. K. and Jain, V. (2019). A review of face recognition methods using deep learning network. Journal of Information and Optimization Sciences, vol. 40, pp. 547-558. https://doi.org/10.1080/02522667.2019.1582875.

[5] Sauer, L. (2021, May 19). What Is Coronavirus? Retrieved May 28, 2021, from John Hopkins Medicine: https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus.

[6] CDC. (2021, March 8). How to Protect Yourself & Others. Retrieved May 28, 2021, from Centers for Disease Control and prevention: https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html.

[7] NIST. (2020, July 27). NIST Launches Studies into Masks’ Effect on Face Recognition Software. Retrieved October 15, 2020, from NIST: https://www.nist.gov/news-events/news/2020/07/nist-launches-studies-masks-effect-face-recognition-software.

[8] Ejaz, S., & Islam, R. (2019). Masked Face Recognition Using Convolutional Neural Network. In: Proceedings of the International Conference on Sustainable Technologies for Industry 4.0 (STI), 24-25 December, Dhaka, Bangladesh. https://doi.org/10.1109/STI47673.2019.9068044.

[9] Wang, Z. et al. (2020). Masked Face Recognition Dataset and Application. https://arxiv.org/abs/2003.09093.

[10] Hariri, W. (2021). Efficient Masked Face Recognition Method during the COVID-19 Pandemic. https://arxiv.org/abs/2105.03026.

[11] Anwar, A., & Raychowdhury, A. (2020). Masked Face Recognition for Secure Authentication. https://arxiv.org/abs/2008.11104.

[12] Li, Y., Guo, K., Lu, Y., & Liu, L. (2021). Cropping and attention based approach for masked face recognition. Applied Intelligence, vol. 51, pp. 3012-3025. https://doi.org/10.1007/s10489-020-02100-9.

[13] Alyuz, N., Gokberk, B., & Akarun, L. (2013). 3-D Face Recognition Under Occlusion Using Masked Projection. IEEE Transactions on Information Forensics and Security, vol. 8, no. 5, pp. 789-802.

[14] Bagchi, P., Bhattacharjee, D., & Nasipuri, M. (2014). Robust 3D face recognition in presence of pose and partial occlusions or missing parts. International Journal in Foundations of Computer Science & Technology, vol. 4, no. 4, pp. 21-35.

[15] Shepley, A. J. (2019). Deep Learning for Face Recognition: A Critical Analysis. https://arxiv.org/abs/1907.12739.

[16] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, Proceedings of the British Machine Vision Conference (BMVC), pp. 41.1-41.12. BMVA Press, September 2015.

[17] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting Masked Faces in the Wild with LLE-CNNs. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July, Honolulu, USA.

[18] Chowdary, J., Punn, N. S., Sonbhadra, S. K., & Agarwal, S. (2020). Face Mask Detection using Transfer Learning of InceptionV3. In: Bellatreche L., Goyal V., Fujita H., Mondal A., Reddy P.K. (eds) Big Data Analytics. BDA 2020. Lecture Notes in Computer Science, vol 12581. Springer, Cham. https://doi.org/10.1007/978-3-030-66665-1_6.

[19] Rekha, A. L., & Chethan, H. K. (2014). Automated attendance system using face recognition through video surveillance. International Journal For Technological Research In Engineering, vol. 1, no. 11, pp. 1327-1330.

[20] Varadharajan, E., Dharani, R., Jeevitha, S., Kavinmathi, B., & Hemalatha, S. (2016). Automatic Attendance management system using face detection. In: Proceedings of the Online International Conference on Green Engineering and Technologies (IC-GET), 19 November, Coimbatore, India. https://doi.org/10.1109/GET.2016.7916753.

[21] Burak. (2020). Pins Face Recognition. Retrieved October 30, 2020, from Kaggle: https://www.kaggle.com/hereisburak/pins-face-recognition.

[22] OpenCV. (2021). Open Source Computer Vision. Retrieved June 24, 2021, from OpenCV: https://docs.opencv.org/3.4/.

[23] NumPy. (2021, June 22). NumPy v1.21 Manual. Retrieved June 24, 2021, from NumPy: https://numpy.org/doc/stable/.

[24] TensorFlow. (2021). API DOcumentation. Retrieved June 24, 2021, from TensorFlow: https://www.tensorflow.org/api_docs.

[25] Keras. (2020). About Keras. Retrieved June 24, 2021, from Keras: https://keras.io/about/.

[26] Scikit-learn. (2021). Machine Learning in Python. Retrieved June 24, 2021, from Scikit learn: https://scikit-learn.org/stable/.

[27] Face_recognition. (2020, February 20). face-recognition 1.3.0. Retrieved June 24, 2021, from PyPi: https://pypi.org/project/face-recognition/.

[28] Pillow. (2021). Pillow. Retrieved June 24, 2021, from Pillow: https://pillow.readthedocs.io/en/stable/.

[29] Kibria, S. B., & Hasan, M. S. (2017). An Analysis of Feature Extraction and Classification Algorithms for Dangerous Object Detection. In: Proceedings of the 2nd International Conference on Electrical & Electronic Engineering (ICEEE), 27-29 December, Rajshahi, Bangladesh. https://doi.org/10.1109/CEEE.2017.8412846.

[30] Sagar, R. (2019, May 25). What Does Freezing A Layer Mean And How Does It Help In Fine Tuning Neural Networks. Retrieved May 23, 2021, from AnalyticsIndiaMag: https://analyticsindiamag.com/what-does-freezing-a-layer-mean-and-how-does-it-help-in-fine-tuning-neural-networks/.

[31] Luca, G. D. (2020, September 9). SVM vs Neural Network. Retrieved May 23, 2021 from Baeldung: https://www.baeldung.com/cs/svm-vs-neural-network.