

Detecting face mask using eigenfaces and vanilla neural networks

Raghav Sharma¹, Shridevi S. Krishnakumar², Abishek Seshan¹, Manan Rajotia¹

¹School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

²Centre for Advanced Data Science, Vellore Institute of Technology, Chennai, India

Article Info

Article history:

Received Jun 23, 2021

Revised May 17, 2022

Accepted Jun 9, 2022

Keywords:

Artificial intelligence

Eigenfaces

Face mask detection

Principal component analysis

Vanilla neural networks

ABSTRACT

Coronavirus has become one of the most deadly pandemics in 2021. Starting in 2019, this virus is now a significant medical issue all over the world. It is spreading extensively because of its modes of transmission. The virus spreads directly, indirectly, or through close contact with infected people. It is proclaimed that people should wear a mask in public areas as a counteraction measure, as it helps in suppressing transmission. A portion of the spaces, where the virus has broadly fanned out, is because of inappropriate wearing of facial cover. In crowded areas, keeping a check on facial masks manually is difficult. To automate this process, an effective and robust face mask detector is required. This paper discusses a hybrid approach using a machine learning technique called eigenfaces, along with vanilla neural networks. The accuracy was compared for three different values of principal components. The test accuracy achieved was 0.87 for 64 components, 0.987 for 512 components, and 0.989 for 1,000 components. Hence, this approach proved to be more promising and efficient than its counters.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Shridevi S. Krishnakumar

Centre for Advanced Data Science, Vellore Institute of Technology

Tamil Nadu 632014, Chennai, India

Email: shridevi.s@vit.ac.in

1. INTRODUCTION

Face mask detection is a subject of broad and current interest and has been undergoing intense study in this period, specifically, because of the spreading of Coronavirus at an alarming rate. After the World Health Organization (WHO) declared that wearing a mask can save a huge number of lives, every country in the world is enforcing this task, as it is of utmost importance. In every department, every field all over the world, be it educational, offices, corporates, public, private, medical institutions, etc. wearing a mask is necessary as it helps in controlling and limiting the transmission of the virus to a great extent. Face mask detection being a crucial task is quite challenging to perform with minimal to no human involvement. The two basic sub-principles, on which it is based, Recognition and Detection, each done separately are comparatively an easy task and have been implemented through various algorithms. There are different hybrid versions of YOLO [1], a model performing detection at superfast frame rates, convolution neural networks (CNN), Region-based CNN (R-CNN), Fast R-CNN, comparative studies done on them for different problem statements [2] and the list goes on (discussed in detail in the literature survey section). In this paper, an approach called eigenfaces [3], along with neural networks is presented as a complete model. This approach is a feature extraction approach and not a feature selection approach. This hybrid model has not been implemented for face mask detection yet. The reason behind selecting this approach is that eigenfaces gives the power of imagining how the algorithm works. The calculated eigenfaces can be plotted and studied, which helps in a much better understanding of the underlying principle.

This face mask detector is based on three major sub principles: i) face detection, ii) dimensionality reduction, and iii) mask detection. Each of these involves several steps, ranging from selecting and creating the dataset, to deciding the number of components of the eigenfaces model, followed by reducing the dimensionality, selecting the characteristics of the vanilla neural network, and then testing the mask detection model on various images.

2. LITERATURE SURVEY

Since the outbreak of the Coronavirus, there have been several studies on the mask detection topic. But, even before coronavirus disease (COVID-19), many studies have been conducted and papers have been published on this topic, using various deep learning, machine learning, artificial intelligence (AI), and image processing techniques. This section gives an overview of the related studies (both recent and former) that have been conducted on this topic.

Nieto-Rodríguez *et al.* [4] conducted a study before the outbreak of COVID-19 and therefore the dataset used was small as compared to the currently available datasets. The research was based on image processing techniques and was also extended to real-time systems. This system aims at detecting the presence or absence of a medical mask in the operating room and the objective was to trigger alarms only for official healthcare staff and doctors on duty who weren't wearing a surgical mask. Two filters were used in this research, one color filter for face detection and another color filter for mask detection; both of these were fed to the classifier. The recall was 95% for this system and a 5% false positive rate was observed. Also, since it has real-time image processing as mentioned earlier, the model was tested with various fps rates.

Naveen *et al.* [5] proposed a technique that uses features of the faces, which are caught locally and globally to recognize a mask and a genuine face. Local binary pattern (LBP) is a descriptor that has been extensively used for face recognition. The features extracted using binarized statistical image features (BSIF) and LBP focus near the eye and nose areas, to detect the presence of a mask. A Euclidean distance classifier fixed with a specific threshold is specified to categorize the image. The testing for this study was conducted on 3D mask database (3DMAD).

Ge *et al.* [6] proposed a model and dataset to recognize normal and masked faces. They presented an enormous dataset known as masked faces (MAFA), it had 35, 806 images of masked faces. The proposed model depended on convolutional neural organizations called locally linear embedding CNNs (LLE-CNNs), which comprises of three modules to be specific; a proposal, embedding, and testing. The average precision achieved in this study was 76.1%.

Ejaz *et al.* [7] used the principle of principal component analysis (PCA) on the Olivetti research laboratory (ORL) dataset. This dataset consists of a total of 500 images. The value of the components, which depends on the number of images used for the algorithm, is very less which leads to lesser accuracy in the case of masked faces. This paper concluded that PCA didn't perform well in the case of masked faces while it did for unmasked ones. The recognition accuracy in the case of masked faces dropped to around 70%.

Ud Din *et al.* [8] aimed to take an image with a mask, detect the mask, remove it, and produce an output image, completely reconstructed and without a mask. This study had two major modules, the initial stage had binary segmentation of the area covered by a mask, followed by the next stage which removes the mask and creates the impacted zone with very fine details while retaining the overall structure of the face. Generative adversarial network (GAN)-based network using two discriminators was used for this purpose. This model outperformed other existing models and algorithms used for this purpose namely gray-level co-occurrence matrix (GLCM), general component analysis (GCA), EdgeConnect, and maintaining and repairing GAN (MRGAN).

Ristea and Ionescu [9] conducted a study to detect if a person is or is not wearing a mask using just speech. This study also involved GANs. Various ResNet models were used for training, starting from the lowest number of layers as 18 and going up to 101 layers. These were then concatenated and fed into an support vector machine (SVM) classifier. The results were compared with various methods before and after performing augmentation. A boost of 0.9% was observed in the model after augmentation.

Jiang *et al.* [10] conducted a study that has ResNet models being used and also aimed at extending the model to MobileNet for integrating to hardware devices. A comparison was also done between the mentioned algorithms and the baseline model (the model to which the dataset originally belonged). The evaluation metrics used were precision and recall. The proposed strategy accomplished best-in-class results on a public face mask dataset, where they were about 2.3% and 1.5% higher than the baseline result in precision, and approximately 11.0% and 5.9% higher than baseline for recall. ResNet even outperformed MobileNet.

Qin and Li [11] made a model that uses different deep learning techniques for the extraction and classification steps. The method aimed at combining SRCNet which are super-resolution and classification

networks and “it quantifies a three-category classification problem based on unconstrained 2D facial images”. The proposed algorithm in this paper consisted of four major steps: first being pre-processing, then facial detection from the image and cropping out the detected face, image super-resolution, followed by facemask-wearing condition recognition. The dataset had 3,835 images and the SRCNet model achieved 98.7% accuracy, thus out-performing traditional approaches.

Lin *et al.* [12] introduced a segmented approach based upon mask R-CNN. Like other studies, ResNet was again used to extract features. Face detection dataset and benchmark (FDDB), annotated faces in the wild (AFW), and WIDER FACE benchmark datasets were used for training and testing. The model was compared to various existing methods like multi-scale based CNN (MSCNN), and contextual multi-scale region-based CNN (CMS-RCNN). Testing was done on the WIDER face dataset, the most challenging dataset available. Testing results were better than many methods for each easy, medium, and hard category subsets. Loey *et al.* [13] provided another deep learning approach was applied to the mask detection problem statement. In this study, for feature extraction, Resnet50, a Deep Learning method, was used again. It is a convolution neural network which is 50 layers deep. The detection part was done by the YOLOv2 model. The average precision achieved in this study was around 81% which outperformed some of the already existing models.

Nagrath *et al.* [14] used a “Single shot multibox detector” to detect a face and a “MobilenetV2 architecture framework” for classifying purposes. The advantage of using MobilenetV2 was that it is lightweight and can be extended and integrated into hardware devices also. SSDMNv2 contains single shot detector (SSD) and ResNet-10 was the backbone, ResNet-10 is a convolution neural network with 10 layers. The DNN based models used in this study were orientation invariant. This model had metrics like F1 score, accuracy, and FPS (frames per second), this out-performed various deep neural network (DNN) models to which it was compared.

Loey *et al.* [15] used two-step algorithms to detect masks; a hybrid learning model was proposed which included deep learning methods with classical machine learning methods. Feature extraction was done using Resnet50, which is a Deep Learning method as stated. It is a CNN which is 50 layers deep. After which, using various classic machine learning algorithms, like support vector machines (SVM), and decision trees (DT). the classification of masks was done. This model was trained and then tested on three different datasets and the results (accuracy metric) were astonishing, more than 99% for each of the datasets and even 100% accuracy for one of them.

Mercaldo and Santone [16] used a transfer learning approach to detect whether a person is wearing a mask or not with no human/manual involvement. The transfer learning model, as mentioned and in the title also, uses the MobileNetV2 model. This model works upon both images and videos. The dataset for this paper had approximately 4100 images and gave an accuracy of 0.98.

Chen *et al.* [17] proposed a detection system based on mobile phones. This approach used GLCMs and k-nearest neighbors (KNN) were used. On testing with validation datasets, this mobile system-based approach gave 82.87% accuracy. Suresh *et al.* [18] implemented a model used MobileNet trained on 3918 images and worked in real-time. This system added a feature of capturing faces without masks and sends them to higher authorities. Many techniques, working on videos and in real-time can be combined with various Video segmentation techniques [19].

Chowdary *et al.* [20] built a transfer learning process of InceptionV3. The fine-tuned model of InceptionV3 was trained on the simulated masked face dataset (SMFD) dataset. This process achieved an accuracy of 100% on the validation dataset. Vinh and Anh [21] utilized the Haar Cascade classifier along with the YOLOv3 algorithm used to detect the face and detect the mask, respectively. This was a real-time detector and gave 90.1% accuracy on experimentation.

3. PROPOSED METHOD

For mask detection, the initial step was to detect the face from the image and then crop it. Doing this, for a human is quite easy, but to do this automatically for a system was a relatively tough task. In this paper, the Haar-Cascade frontal face detection model was used to serve the purpose of face detection. This detected the face from an image, irrespective of the fact that the face was or was not wearing a mask. After the dataset was created, steps like preprocessing and data augmentation were performed. Two classes namely “mask” and “no mask” were created, after which the eigenfaces or the principal component analysis method was applied. The dataset was reduced drastically and the eigenfaces were computed. The computation involved many statistical steps ranging from building a covariance matrix to eigenvalues and eigenvectors, and in the end, retaining/extracting (might involve making new features) the features with the most variance. This modified dataset was then fed into the Vanilla or the artificial neural network (ANN) which had some dense layers, and dropout layers. with the specified number of components. Figure 1 demonstrates the methodology used.

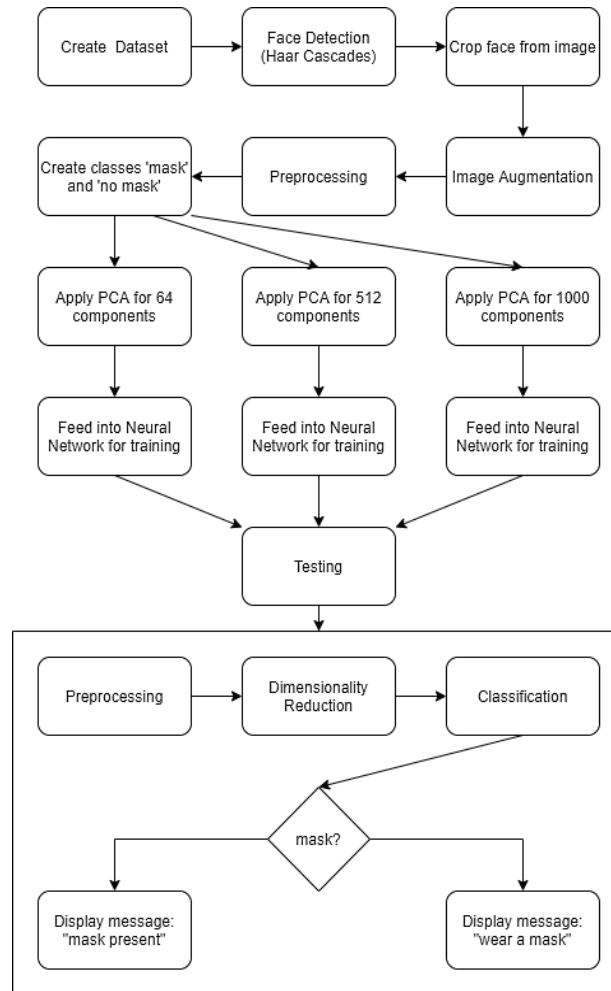


Figure 1. Stepwise methodology of mask detector

4. RESEARCH METHOD

This section aims to elaborate on the sub-sections that have been mentioned earlier. In a stepwise manner and articulate all the required algorithms for the face mask detection system. The section covers each aspect from scratch, from creating the dataset to finally testing the proposed model.

4.1. Create dataset

The images were acquired from various sources, mostly in .jpg format. The size of the images was not constant since they were from various sources. The images were divided into classes or folders. Namely, “mask” and “no mask” are the two specified classes. The aim was to create a dataset that will have more variance. Different types of images, from different angles, people of different genders, ethnicity, race, with varying facial features like beards, or having other coverings like spectacles, shades, caps, and different types of masks were included in the dataset. The dataset consisted of few present datasets along with the custom dataset built for this project. The overview of the dataset is shown in Figure 2.

Masked face recognition dataset (MFRD) in Figure 3 [22], this dataset contained images of people wearing face masks. Simulated Masked Face Dataset (SMFD) Figure 4, another popular dataset that had masks artificially added to the images. Beard-no beard dataset Figure 5, the images of this dataset, all belong to the “no mask” class, the purpose behind using this dataset was adding more variance in the dataset. Then there was a custom dataset (Figure 6) made with a webcam and had images of people with and without a mask. Each class had around 700 images and after performing data augmentation the dataset increases to 7,040 images for the “mask” class and 7173 for the “no mask” class. This made the total dataset 14,208 images.

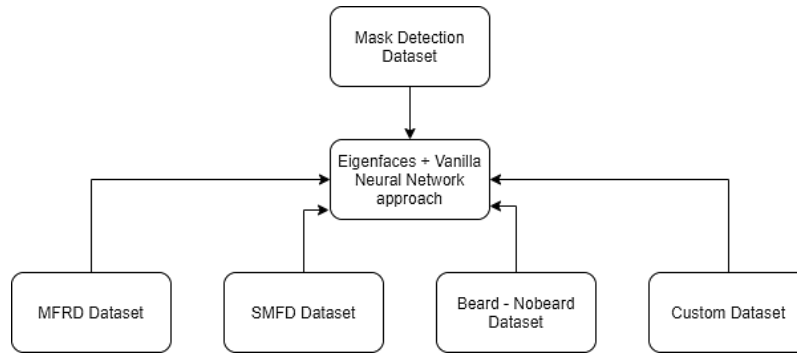


Figure 2. Dataset overview of mask detector



Figure 3. MFRD dataset



Figure 4. SMFD dataset



Figure 5. Beard-no beard dataset



Figure 6. Custom dataset

4.2. Face detection

After the images were acquired, properly segregated, and well defined into the two classes, the next step was to detect just the faces from the image. The reason why that is important is, in any image, there are areas, which do not serve any purpose and just add up to the complexity. Therefore, it is important to crop out just the face from an image because that is the exact and the only area required for this study. Therefore, to extract faces from the image, the Haar - Cascade Frontal Face Detection model was used. It is an object detection algorithm based on machine learning and is used to identify objects in an input be it an image, video, or real-time. Based on the concept of features proposed by Viola and Jones [23]. This study was then cited several times. Some of the works include using it for real-time problem statements, Ahmad *et al.* [24], studying/reviewing [25], and evaluation on different datasets [26]. A cascade function is trained from a lot of positive and negative images and is then used to detect objects in other images/video files. The algorithm has four stages: i) haar feature selection, ii) creating integral images, iii) Adaboost training, and iv) cascading classifiers.

It is a very well-known technique for detecting faces and body parts in an image but can be trained to identify almost any object. This algorithm worked well even for masked faces and detected them successfully with great accuracy. An image is fed to the Haar cascade classifier. A sliding window type method or specifically edge features, and line features. (Figure 7) is used to detect all faces in the given image, crop the faces and write them as separate images. If there are n faces in an image, the Haar cascade classifier produces n cropped images containing only the faces of the subjects, and this was done for the complete dataset. These images were then saved to their particular classes.

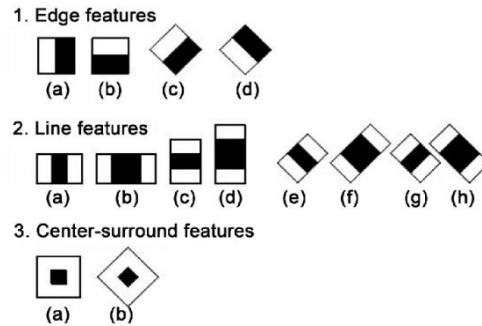


Figure 7. Haar features [27]

4.3. Data augmentation

This step was performed to bring variety to the dataset. Image Augmentation by Shorten and Khoshgoftaar [28] serves the purpose of adding variance and helps prevent the overfitting of the model by creating multiple varied images from the source image. Using the image data generator class from Keras library and some basic looping algorithms, about 10 images were created/ generated from 1 source image (Figure 8). As shown in Figure 9, the way these newly created images differ from the original image was that they were rotated, horizontally moved, vertically moved, resized, were out of focus, sharpened, and horizontally flipped.



Figure 8. Source images Figure 9. Images after performing image augmentation

4.4. Preprocessing

The next step was to standardize all the images and bringing them to a common general scale. This included reducing the dimensions by converting the image from red, green, and blue (RGB) to gray-scale. All images in both classes were resized from their original size to 150×150 . StandardScaler, a technique generally used in classification problems, was used for standardizing the images.

4.5. Calculating eigenfaces

After all the preprocessing steps are done, the next step was the most crucial step of the model. The eigenfaces approach which is based on principal component analysis (PCA) was used. This is an unsupervised learning approach that has functions like, data compression, dimensionality reduction, decreasing the required computation power which consequently helps in speeding up the algorithm. It is used for performing data analysis and for building predictive models just like the one being implemented in this study. The technique behind this principle is that the variance/major features/directions should be explained with a minimal number of features. For instance, there are N images in the dataset. By using orthogonal transformation these positively correlated N face images are expressed into K uncorrelated variables and these variables are known as eigenfaces. These eigenfaces are calculated from the covariance matrix after performing eigenvalue decomposition on it. All the images are converted into their vector forms and expressed in a new number of dimensions K where $K < N$ and hence dimensionality reduction is achieved.

A multivariate dataset like a set of images is a high dimensional data space, and after applying PCA the output is a lower-dimensional data space which can be called the shadow of the high dimensional data

space when viewed from a certain viewpoint, describes it in the best way, i.e. most features retained, they visually depict the major features of the dataset. Each facial image present in the dataset has been made up of proportions of all these K-selected features or eigenfaces. These proportions are the weights associated with each Eigenface for a particular face image. There are different proportions for every image based on the specifications of that image and the relation can be drawn as follows:

$$\text{Original image} = \frac{\text{Mean}}{\text{Average}} \text{face} + \text{weight1} * \text{eigenface1} + \text{weight2} * \text{eigenface2} + \dots + \text{weightk} * \text{eigenfacek}$$

The face image can be represented as a weight vector which denotes what proportion of some eigenface makes up the image. The transformation explained is defined in a manner that the first principal component shows the most dominant direction or features of the dataset. That means it is the most crucial eigenface (having maximum information retention and minimum noise) for representation of the dataset and with each succeeding component, the next most descriptive possible direction of features is calculated and which is also uncorrelated with the preceding components. Figure 10 shows some sample eigenfaces.

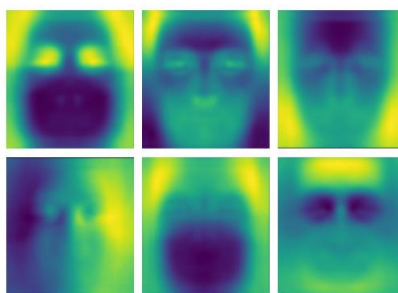


Figure 10. Sample eigenfaces

4.6. Steps for eigenfaces calculation

- The input image was converted to a face vector and this was performed for the complete dataset. For example, an image of $M \times M$ dimensions is converted to $M^2 \times 1$. This is the modified dimension of a single image. This can be imagined as a matrix where each column represents the modified face vector of a single image. The dimensions of this matrix are $M^2 \times N$, where N is the total number of images in the dataset.
- The average or the mean face was calculated from the matrix defined and subtracted from each face vector to get the normalized face vectors. The purpose behind doing this was to remove the redundant and unnecessary values and keep the values that had the highest variance.
- A covariance matrix was formed using the normalized face vectors matrix. Matrix multiplication was done on this matrix with its transpose to get the covariance matrix. $C = AA^T$, this resulted in a very large square matrix, which was made up of several eigenvectors.
- The covariance matrix was huge and had a lot of features to process. From the covariance matrix, the eigenvectors which had the maximum eigenvalues were selected, as these eigenvectors stored highly variant information. This was done by performing eigenvalue decomposition. K eigenvectors/eigenfaces also known as principal components were selected from the total number of eigenvectors. The value of K is always less than the number of images for performing dimensionality reduction.
- When the eigenfaces are calculated, they are calculated orthogonally, until they run out of dimensions to calculate. The first eigenface is the most feature abundant or retains the most features and as the algorithmic calculation proceeds, the amount of information stored in each eigenface keeps on decreasing and noise is added. Therefore, just K amount of eigenfaces are retained.
- The calculated eigenfaces could represent an image of a higher-dimensional space in a lower dimension space and retain the most information. Each image was represented as a weighted sum of the K eigenfaces. These weights were stored in a weight vector.
- For image i , the original image can be represented in terms of the weighted sum of the weight vector w_i and eigenfaces, with the sum of the average face found in the initial step.

4.7. Vanilla neural network and testing

The reduced dimensionality training set was fed into a vanilla neural network. The neural network had 2 dense layers. These layers are linear operations that connect each neuron in the previous layer to every neuron of the next layer. ‘Relu’, a non-linear activation function was used with them. Two dropout layers were added, each after the dense layers, to randomize the network. This is done by dropping arbitrary neurons to avoid overfitting. A final dense layer was added which provided the output in the shape of two classes. One represents ‘mask’ and the other represented ‘no mask’. ‘Softmax’ activation function was used with the final dense layer as it gives the output in form of probability for each class. This is the reason why Softmax is used in classification problems. The loss function used was categorical cross-entropy and the evaluation metric was accuracy. The model was compiled with 10 epochs, 20 batch sizes, and a learning rate of 0.001. Even while testing, the image had to be pre-processed before applying the eigenfaces (PCA) approach for reducing its dimensions. The resulting array showing the classification was a 1×2 dimension array, where $[1, 0]$ signifies the mask is there and $[0, 1]$ signifies the “no mask” class.

5. RESULTS AND DISCUSSION

The total images were 14,208 images and the dataset was divided into 0.8-0.2 train-test split. The training dataset now had 11,366 images while the testing dataset had 2,842 images. The number of PCA components that the model was worked upon was decided with consideration to the test set since the number of components K cannot be more than the number of images. Taking too many components will eliminate the essence of the algorithm, so K is taken out of a hyper-parameter 1,000. The results were compared with three different numbers of components, around 15% which is 64, around 50% which is 512, and taking 100% i.e. 1,000 components.

The dataset when expressed in terms of the number of components k (64, 512, and 1,000) reduced the dimensions drastically. Earlier, there were 150×150 , 22,500 pixels, 22,500 features/dimensions. Considering the training and testing dataset, the dimensions moved from (11,366, 22,500) and (2,842, 22,500) to (11,366, k) to (2,842, k) where $k=64, 512, 1,000$. Here each row represents an image vector expressed in terms of these dimensions (0 to $k-1$), or as explained earlier weighted eigenfaces and the sum of the mean face.

Figure 11 shows 3 sample images before applying PCA. While Figure 12 compares the eigenfaces expressed in terms of k eigenfaces where $k=64$ for row 1, $k=512$ for row 2, and $k=1000$ for row 3. As there is an increase in the number of components, the clarity of the image, or in this case Eigenface, also increases evidently. Table 1 and Figure 13 shows the result for each number of components in a tabular and statistical way respectively.



Figure 11. Sample images

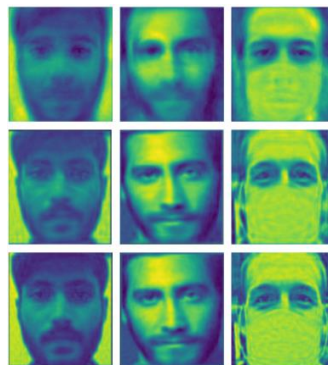


Figure 12. Qualitative analysis displaying eigenfaces for different components

Table 1. Final accuracy and loss for each value of k

Components	Training accuracy	Training loss	Testing accuracy	Testing loss
64	0.906	0.214	0.8797	0.303
512	0.985	0.069	0.9877	0.043
1000	0.993	0.039	0.989	0.064

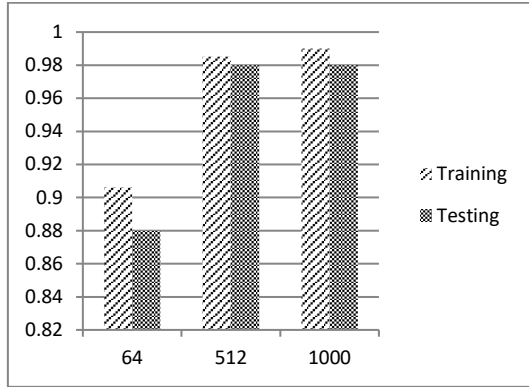


Figure 13. Bar chart for final accuracy

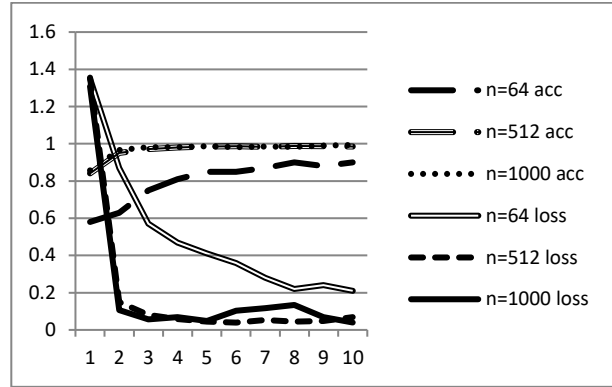


Figure 14. shows a line graph representing the accuracy and loss for each value of k and loss for each value of k

Figure 14 shows a line graph representing the accuracy and loss for each value of k. It shows how the accuracy and loss vary with each subsequent epoch. The reason loss shows more than 1 value initially isn't wrong as the metric used for loss was categorical cross-entropy (CCE) which is just a positive value (>0) and does not represent a fixed range. Compared to the existing works shown in a tabular format in Table 2, these models have been taken specifically as they had the same evaluation metric as the proposed model in this paper, which is, accuracy. The hybrid approach talked about in this paper outperforms most of the existing approaches and competes quite well with the remaining others.

Table 2. Comparing the proposed works with existing works in terms of accuracy.

Models	Accuracy Achieved
Hybrid (eigenfaces+VNN) k=64(our proposed work)	0.8797
Hybrid (eigenfaces+VNN) k=512(our proposed work)	0.9877
Hybrid (eigenfaces+VNN) k=1000(our proposed work)	0.989
Hybrid Transfer Learning SVM Classifier (Real Masked Face Dataset)	0.9964
Hybrid Transfer Learning SVM Classifier (Simulated Masked Face Dataset)	0.9949
Hybrid Transfer Learning SVM Classifier (Labeled Faces in Wild dataset)	1.00
InceptionV3	1.00
AlexNet	0.892
LeNet-5	0.846
Multi-granularity model	0.95
SRCNet YOLOv3	0.987
SSDMNV2	0.9264
Transfer Learning, MobileNetV2	0.98
YOLOv3	0.939

6. CONCLUSION

This paper was a study of a mask detection model based on eigenfaces and the vanilla neural network approach. Evidently, from the experimental results section, it can be concluded that as the number of components is increased, the model's performance improves and the reason for this is, that as the number of components is increased, more features are retained. When there is a drastic change in the number of components (from 64 to 512 components), the accuracy also changes drastically, but after a certain level, the change isn't quite evident (from 512 to 1000 components) as the principal components added later are not of the same importance as the ones added initially. The dataset has been changed and created several times, to add more variance to the data, also data augmentation techniques and dropout layers in models have been added to keep the variance and overfitting in control. Even after performing all these challenging techniques

the accuracy results achieved were satisfactory and hence this model can be used as an alternative approach to CNNs and other currently present techniques.





REFERENCES

- [1] Y. Wang and J. Zheng, "Real-time face detection based on YOLO," in *1st IEEE International Conference on Knowledge Innovation and Invention, ICKII 2018*, Jul. 2018, pp. 221–224, doi: 10.1109/ICKII.2018.8569109.
- [2] M. A. Anwer, S. M. Shareef, and A. M. Ali, "Accident vehicle types classification: A comparative study between different deep learning models," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1474–1484, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1474-1484.
- [3] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Jan. 1991, doi: 10.1162/jocn.1991.3.1.71.
- [4] A. Nieto-Rodriguez, M. Mucientes, and V. M. Brea, "System for medical mask detection in the operating room through facial attributes," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9117, pp. 138–145, 2015, doi: 10.1007/978-3-319-19390-8_16.
- [5] S. Naveen, R. Shihana Fathima, and R. S. Moni, "Face recognition and authentication using LBP and BSIF mask detection and elimination," in *2016 International Conference on Communication Systems and Networks, ComNet 2016*, Jul. 2017, pp. 99–102, doi: 10.1109/CSN.2016.7823994.
- [6] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with LLE-CNNs," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Jul. 2017, vol. 2017-January, pp. 426–434, doi: 10.1109/CVPR.2017.53.
- [7] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of principal component analysis on masked and non-masked face recognition," in *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019*, May 2019, pp. 1–5, doi: 10.1109/ICASERT.2019.8934543.
- [8] N. Ud Din, K. Javed, S. Bae, and J. Yi, "A novel GAN-based network for unmasking of masked face," *IEEE Access*, vol. 8, pp. 44276–44287, 2020, doi: 10.1109/ACCESS.2020.2977386.
- [9] N. C. Ristea and R. T. Ionescu, "Are you wearing a mask? Improving mask detection from speech using augmentation by cycle-consistent GANs," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Oct. 2020, vol. 2020-October, pp. 2102–2106, doi: 10.21437/Interspeech.2020-1329.
- [10] M. Jiang, X. Fan, and H. Yan, "RetinaMask: A Face Mask detector," *arXiv e-prints*, p. arXiv:2005.03950, 2020, [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2020arXiv200503950J>.
- [11] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–23, Sep. 2020, doi: 10.3390/s20185236.
- [12] K. Lin *et al.*, "Face detection and segmentation based on improved mask R-CNN," *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–11, May 2020, doi: 10.1155/2020/9242917.
- [13] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustainable Cities and Society*, vol. 65, p. 102600, Feb. 2021, doi: 10.1016/j.scs.2020.102600.
- [14] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustainable Cities and Society*, vol. 66, p. 102692, Mar. 2021, doi: 10.1016/j.scs.2020.102692.
- [15] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement: Journal of the International Measurement Confederation*, vol. 167, p. 108288, Jan. 2021, doi: 10.1016/j.measurement.2020.108288.
- [16] F. Mercaldo and A. Santone, "Transfer learning for mobile real-time face mask detection and localization," *Journal of the American Medical Informatics Association*, vol. 28, no. 7, pp. 1548–1554, Jul. 2021, doi: 10.1093/jamia/ocab052.
- [17] Y. Chen *et al.*, "Face mask assistant: Detection of face mask service stage based on mobile phone," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 11084–11093, May 2021, doi: 10.1109/JSEN.2021.3061178.
- [18] K. Suresh, M. B. Palangappa, and S. Bhuvan, "Face mask detection by using optimistic convolutional neural network," in *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, Jan. 2021, pp. 1084–1089, doi: 10.1109/ICICT50816.2021.9358653.
- [19] B. Somasundaram and S. Shridevi, "Video segmentation & retrieval," *IJRTE International Journal of Recent Technology and Engineering*, vol. 8, no. 1, pp. 1020–1024, 2019.
- [20] G. Jignesh Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face mask detection using transfer learning of inceptionV3," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12581 LNCS, 2020, pp. 81–90.
- [21] T. Q. Vinh and N. T. N. Anh, "Real-time face mask detector using YOLOv3 algorithm and haar cascade classifier," in *Proceedings - 2020 International Conference on Advanced Computing and Applications, ACOMP 2020*, Nov. 2020, pp. 146–149, doi: 10.1109/ACOMP50827.2020.00029.
- [22] Z. Wang *et al.*, "Masked face recognition dataset and application," *arXiv e-prints*, Mar. 2020, [Online]. Available: <http://arxiv.org/abs/2003.09093>.
- [23] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. I-511–I-518, doi: 10.1109/cvpr.2001.990517.
- [24] A. H. Ahmad *et al.*, "Real time face recognition of video surveillance system using haar cascade classifier," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1389–1399, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1389-1399.
- [25] K. Cen, "Study of Viola-Jones real time face detector," in *Ai2 (Allen Institute for Artificial Intelligence)*, vol. 9, no. 10, 2016, pp. 1–94.
- [26] R. Padilla, C. C. Filho, and M. Costa, "Evaluation of haar cascade classifiers designed for face detection," *Journal of WASET*, vol. 6, no. 4, pp. 323–326, 2012, [Online]. Available: <http://www.waset.org/publications/2910>.





- [27] M. A. H. Akhand, A. S. Akash, and A. S. Mollah, "Improvement of haar feature based face detection in OpenCV incorporating human skin color characteristic," *Journal of Computer Science Applications and Information Technology*, vol. 1, no. 1, pp. 1–8, Nov. 2016, doi: 10.15226/2474-9257/1/1/00106.
- [28] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

BIOGRAPHIES OF AUTHORS







Raghav Sharma     is a diligent individual, who completed his Bachelor of Technology (Computer Science Engineering) from Vellore Institute of Technology (VIT), Chennai, India. He is in his final year and is quite interested in Artificial Intelligence, Machine Learning, and development-related projects. He aims to contribute to these fields with his creative ideas and techniques. He is also passionate about learning and mastering new skills. He can be contacted at email: raghavsharma1999@gmail.com.







Prof. Shridevi S. Krishnakumar     is currently working as an Associate Professor in the Centre for Advanced Data Science at VIT University. She is a University rank holder and a medalist in her post-graduation and completed her Doctorate in Computer Science from MS University, Tirunelveli. Her research area of interest includes Semantic Artificial Intelligence, Web Mining, Machine Learning, and Deep Learning. She has published and presented many research papers in reputed International Journals and Conferences and has received best research awards for her works. She can be contacted at email: shridevi.s@vit.ac.in.



Abishek Seshan     is a skilled individual who completed his Bachelor of Technology in Computer Science Engineering from Vellore Institute of Technology (VIT), Chennai, India. He is passionate about Competitive Programming, Artificial Intelligence, Mathematics, and quizzes. He aims in contributing to the field of Machine Learning using his knowledge and technical skills. He can be contacted at email: abishekseshan54099@gmail.com.



Manan Rajotia     is a final year Computer Science Engineering student at VIT who is adept at juggling between development and administrative activities. He is a competitive programmer with a good understanding of Data Structures & Algorithms. A machine learning enthusiast and a highly motivated individual with good problem-solving skills, and exceptional leadership abilities. He can be contacted at email: manansharma.2604@gmail.com.