Real-Time Face Mask Detection with SVM and HOG Features

Utkarsh Kansal
Department of Computer Science, University of Southern California, Los Angeles, California 90089, USA
(Dated: April 30, 2021)

The coronavirus pandemic has changed our lives in many ways. Now, wear a mask has become a new normal for everyone. At places, wearing a mask has become mandatory for the safety of themselves as well as others. In this paper, a solution is presented to help businesses detect people wearing a mask or not automatically before entering their premises. An attempt has been made to extract HOG features and train a model with SVM to recognize people with a mask. This model is also compared with various other classifiers, including a state-of-the-art deep learning approach via transfer learning. It is found that the model performed better than other machine learning techniques and had similar performance with the MobileNetV2 deep learning model. A combination of Real-World Masked Face Dataset (RMFD) and Real-Time Medical Mask Detection dataset has been used to train, validate and test the model's performance. The SVM model trained on HOG features of RGB images achieved accuracy, F1-score, and AUC of 0.98, 0.98, and 0.995 on the test set, respectively.

Keywords: Machine Learning, Transfer Learning, SVM, HOG, Face Mask, COVID-19

I. INTRODUCTION

The new SARS-CoV-2 variant has spread across the globe, and many people have been infected by it till now. Millions of people lost their lives battling this new variant. Currently, more than 124 million people have been affected by this virus, and 2.7 million people succumbed to COVID-19 [1]. Scientists and doctors are working tirelessly to aid humanity in overcoming this ordeal. A lot of research is happening on many fronts to find either a vaccine or medicine for its cure. In the meantime, we must practice social distancing, wear a mask at all times when in public places, avoid crowds and poorly ventilated spaces [2].

The recent developments in Machine Learning and artificial intelligence have helped scientists and governments analyze the spread of COVID-19 and research and development for its treatment. Using ML technology, health workers can perform fast diagnosis and screening of chest X-ray scans which curbs the spread and is cost-effective. It is also used in contact tracing of the virus, which helps the government track down the hotspot locations. Moreover, ML is widely used in real-time forecasting and prediction of SARS-CoV-2 [3].

In this report, a mask face detection model based on Support Vector Machine (SVM) and Histogram of Oriented Gradients (HOG), a feature descriptor used to extract essential features of an image, is proposed. The model can detect people wearing a mask or not automatically, and it can be deployed in real-time to thwart the COVID-19 spread by restricting the entry of people in public places such as malls, restaurants, and bars. A comparison with other classic ML techniques and modern methods like transfer learning is also performed. A flowchart illustrating the workflow for building a face mask detection model is shown in FIG.1.

The organization for the rest of the paper is as follows: Section 2 discusses previous works that have been done related to face mask detection. Section 3 describes the dataset used. Section 4 is the proposed methodology to achieve the defined task. Section 5 reports the results of the classification task. Section 6 discusses the results and compares them with the output of other models using various performance metrics. Finally, the conclusions are drawn with possible future work for better performance on this classification task.

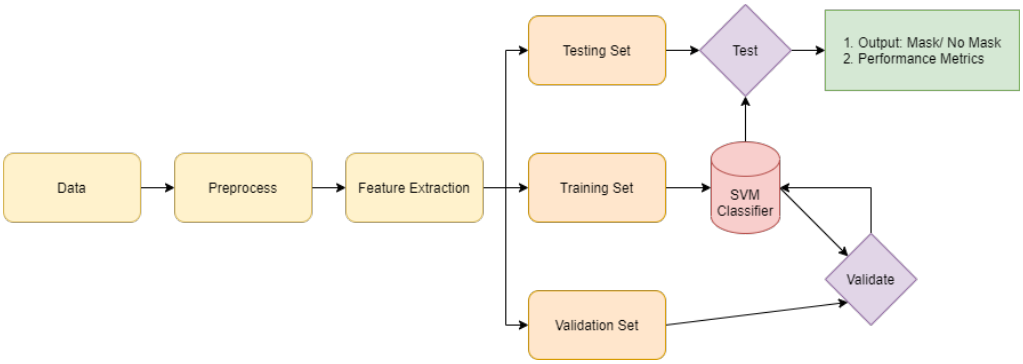

FIG. 1. Flowchart to Build a Face Mask Detection Model

II. RELATED WORKS

In [4], the authors present a hybrid model for face mask detection. They have a developed this model by combining deep transfer learning with machine learning models such as SVM, decision trees, and an ensemble consisting of KNN, Linear Regression, and Logistic Regression. They have used ResNet-50, which is 50 layers deep, for pre-processing to extract important features from the images and supplied this as an input to various machine learning models. The main motivation behind using simple machine learning models was to prevent overfitting. The authors used 3 datasets for their study. The first one contained 5000 images each of real people with and without masks. The second one had 1570 simulated images in which half of the people were wearing masks. A third dataset was formed by combing the 1st two datasets. Finally, the fourth dataset with 13,000 simulated images but of famous people wearing masks was exclusively used only for testing. Except for the fourth one, other datasets were divided into training (70%), validation (10%), and test (20%) sets. The performance metric used for comparing the various models were accuracy, precision, recall, and F1-Score. The models were trained separately on the three datasets and tested on all four later on. The results showed that the Decision tree performed the worst among all three classifiers and was slower to train than SVM. The ensembled model's performance was a tad bit better than SVM in terms of accuracy for all 4 datasets. When trained over the third dataset, the test accuracy of ensembled over the four datasets were 99.28%, 99.49%, 99.35%, and 100%, respectively. While SVM achieved 99.27%, 98.72%, 99.19%, and 100% respectively. However, the training time of ensembled was insanely high, compared to both SVM and Decision tree. Holistically judging the three, authors decided to select SVM as the best classifier in terms of performance and time consumption.

In [5], the authors made an attempt at facial recognition with people wearing masks using Principal Component Analysis (PCA). They used the ORL dataset, which had 10 images each of 40 different people. Also, they added 100 more images of their own with and without a face covering. First, they used the Viola-Jones face detection algorithm and cropped out the required facial area. However, this detection algorithm failed with images where people heavily covered their faces. Then they pre-processed these images and used PCA for important feature extraction. Eigenfaces were obtained as a result, which was used as an input to the Nearest Neighbour classification algorithm. The model used 300 images for training, and 4 experiments with 80,120,160, and 200 test images were conducted. Accuracy was only used as a performance metric. The results obtained were not so promising, having the best accuracy in experiment 1 with 80 test images of 96.25% for non-masked and 73.75% for masked images. While in experiment 4 with 200 test images, the accuracy dropped to 95.62% for non-masked and 68.75% for masked images.

III. DATASET

The combination of two datasets was used in the study with a total of 7531 masked and 7639 face images. FIG. 2 and FIG. 3 shows a sample of the dataset.

FIG. 2. Sample of Unmasked Images

FIG. 3. Sample of Masked Images

The first dataset used for the study is Real-World Masked Face Dataset (RMFD) [6]. The dataset consists of 2,118 masked face images and the rest 90,000 unmasked images of varied resolutions. As the dataset is highly imbalanced, the unmasked images are undersampled and brought to the same number of instances as masked images.

The second dataset used for the study is the Real-Time Medical Mask Detection dataset [7] that gathered mask and face images from various sources, including popular datasets [8] with 1376 images and Medical Mask Dataset by Mikolaj Witkowski present on Kaggle with 678 images. A total of 5413 masked and 5521 face images were used from this dataset.

IV. PROPOSED METHODOLOGY

The methodology is divided into two parts. In the first part, the procedure to build a face mask detection model using SVM is described. In the second part, the approach used to detect faces in real-time through the model built earlier is elucidated.

A. Training a Face Mask Detection Model

The image dataset is first preprocessed before training a classifier. All the images are resized and interpolated to the same resolution of 128x128. The pixel values are also normalized and brought between 0 and 1. Next, the dataset is passed through a Histogram of Oriented Gradients (HOG) Feature Descriptor [9] to extract only the most important information about the images. HOG technique was widely popular before deep learning took its place. In this feature engineering step, HOG looks only at the structure of the images and records the magnitude and direction of all the sharp edges. The minimalistic representation of images obtained through HOG is sufficient enough to differentiate between them. In the present study, horizontal and vertical gradients are calculated for every pixel in an image, and then a histogram of gradient is computed for each 8×8 block. These gradient values for each pixel over 8 x 8 blocks are quantized into 9 bins based on their orientation. Finally, the histogram is then normalized over a 24 x 24 block size to get the final feature vector. FIG.4 and FIG.5 illustrates the HOG features extracted for the dataset. These extracted features are split into three sets: train, validation, and test with ratios of 0.8, 0.1, and 0.1. A SVM classifier is trained and validated using the first two sets, and later its performance is tested on the test set. After tuning the parameters and hyperparameters of HOG and SVM functions, the final model is saved that can classify people with and without a mask. A flowchart showing the workflow to train the model is shown in FIG. 1.

B. Real-Time Face Mask Detection

First, a live video feed is captured through a camera and frame by frame analyzed. In each frame, faces are detected using the Viola-Jones object detection algorithm [10] with the help of Haar Cascades after converting it to grayscale. Each face is cropped out and resized into 128 x 128 and then normalized. After the cropped image of the face matches the input requirements of the model, HOG features are extracted from it. These HOG features are ingested in the model to know whether the face is covered with a mask or not. A bounding box is drawn around the face with a green outline if the mask is detected or a red outline if the mask is not present. Finally, this processed frame is shown to the user. The workflow for detecting face-mask in real-time is shown in FIG.6.

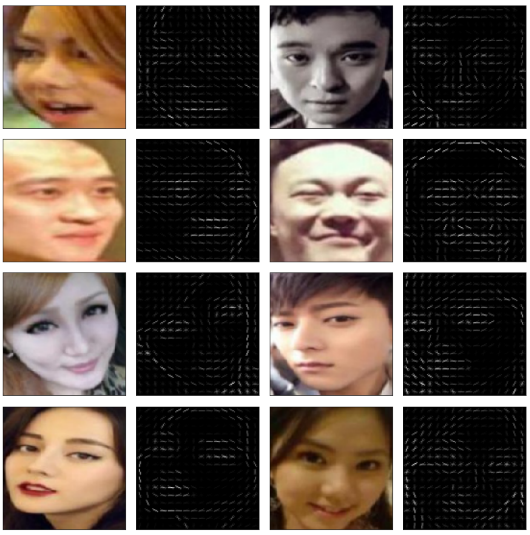

FIG. 4. HOG features of Unmasked Images

FIG. 5. HOG features of Masked Images

C. Evaluation Metrics

Various models were built to detect face mask, and their performance was evaluated based on the following metrics:

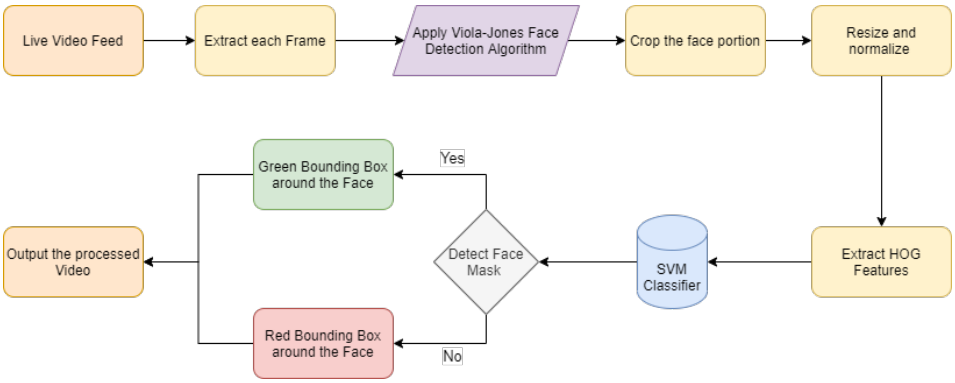

FIG. 6. Workflow to detect Face-Mask in Real Time

1. **Accuracy:** The model accuracy is calculated by adding the true positive and true negative and dividing it by the sum of true positive, true negative, false positive, and false negative. It is a measure of how accurate the model predictions are.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision:** Precision can be calculated by dividing the number of True Positives by the total of True Positives and False Positives. It is the ratio of correct predictions and total predictions made for the positive class.

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall:** Recall is also known as sensitivity. It can be calculated by dividing the number of true positives by the total number of true positives and false negatives. It is the ratio of correct predictions and the total number of actual instances of the positive class.

$$Recall = \frac{TP}{TP + FN}$$

4. **F1-Score:** F1 Score considers both the precision and recall and is their harmonic mean. It depicts the balance between recall and precision.

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall}$$

Where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

5. **AUC:** AUC (Area Under The Curve) is the measure of the ability of a classifier to distinguish between classes. Higher the AUC, the better the performance of the model in distinguishing between different classes.

V. RESULTS

In this section, experimental results obtained for face mask detection are illustrated. The dataset is divided into 3 parts: training (80%), validation (10%), and testing (10%). The training set is used to train the classifier. The validation set helped to tune the model's hyperparameters, and the test set is used to check the classifier's final performance. The SVM model trained on HOG features of the images gave a big boost in the performance. The confusion matrix, classification report, and the AUC-ROC curve are shown in FIG. 7, TABLE I, and FIG. 8 respectively. A comparison with different models is presented and discussed in the next section.

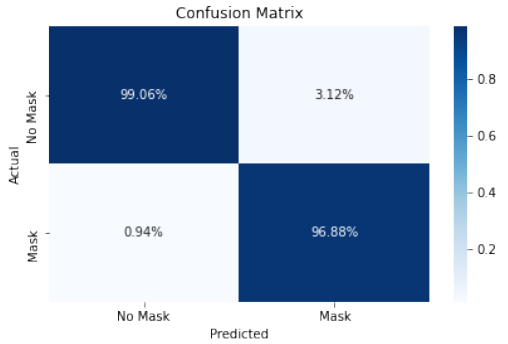

FIG. 7. Confusion Matrix

TABLE I. Classification Report

	Precision	Recall	F1-Score	Support
No Mask	0.99	0.97	0.98	764
Mask	0.97	0.99	0.98	753
accuracy			0.98	1517
macro avg	0.98	0.98	0.98	1517
weighted avg	0.98	0.98	0.98	1517

These figures and the table show that the classifier can predict people wearing a mask with precision and recall of 0.969 and 0.991, respectively. The F1-score obtained is 0.98, and high accuracy of 0.98 indicating that the

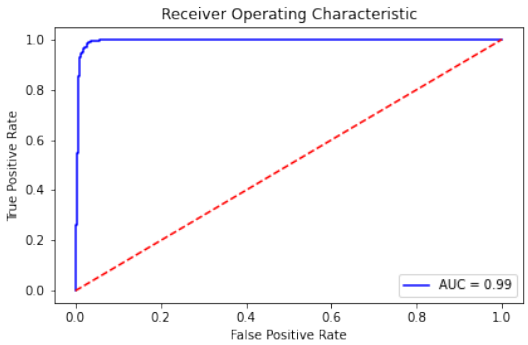

FIG. 8. AUC-ROC Curve

approach used gives robust results. The AUC score of 0.995 indicates that the model can correctly distinguish between people with and without masks. It is seen from the confusion matrix illustrated in FIG. 7 that the model predicted only 0.94% False-negative and 3.12% False positive. These statistical values suggest that the model did not falsely classify people wearing a mask as not wearing a mask (see Appendix A).

VI. DISCUSSION

This section compares the results of the proposed model with other deep learning and machine learning models. In addition, the limitations and future works are also discussed. It is found that the combination of SVM and HOG performed (accuracy) 4.7% and 2.9% better than vanilla SVM and Random Forest with HOG, respectively. The results demonstrated in this study by the proposed model match with state of the art deep learning method using MobileNetV2 architecture (see Appendix B). The difference in their performance was less than 2% in terms of accuracy. A simple feature engineering step of extracting HOG features before fitting any machine learning model improved the classifier's performance significantly. TABLE II and FIG. 9 shows a detailed comparison between various classifiers in terms of numerous performance metrics. It can be seen that every performance measure validates the fact that the proposed method works very well to accomplish the given task. However, during real-time detection, not just the classification performance matters but also the time it takes to make predictions is crucial. SVM fails in this miserably. As a benchmark, if we consider the time it takes for a classifier to make predictions on the test set, the proposed SVM + HOG model took 13 minutes 22 seconds while MobileNetV2 took 2.03 seconds. It is evident from this time duration that the proposed SVM model cannot be practically deployed. However, another model using Linear SVM that classifies on HOG features makes a prediction on test with 95.3% accuracy just takes 419 milliseconds. Thus making it almost 5 times faster than MobileNetV2 and a practical alternative to deploying in real-time.

TABLE II. Comparison between various Classifiers

Classifier	Precision	Recall	F1 Score	AUC	Accuracy
SVM + HOG	0.969	0.991	0.980	0.995	0.980
Linear SVM + HOG	0.955	0.950	0.952	0.989	0.953
Random Forest + HOG	0.924	0.983	0.952	0.989	0.951
SVM + Grayscale	0.930	0.936	0.933	0.981	0.933
SVM + HOG + Grayscale	0.963	0.992	0.977	0.994	0.977
Random Forest + HOG + Grayscale	0.923	0.983	0.952	0.989	0.951
MobileNetV2	0.999	0.996	0.997	1.000	0.997

The future work of the study would include using YOLO (You only look once) rather than Haar Cascades for improved real-time face detection. Moreover, better hyperparameter tuning might bring the performance of the proposed model to near perfection. Attention models can also be used for this classification task and are expected to more robust and outperform other models at the cost of higher model complexity.

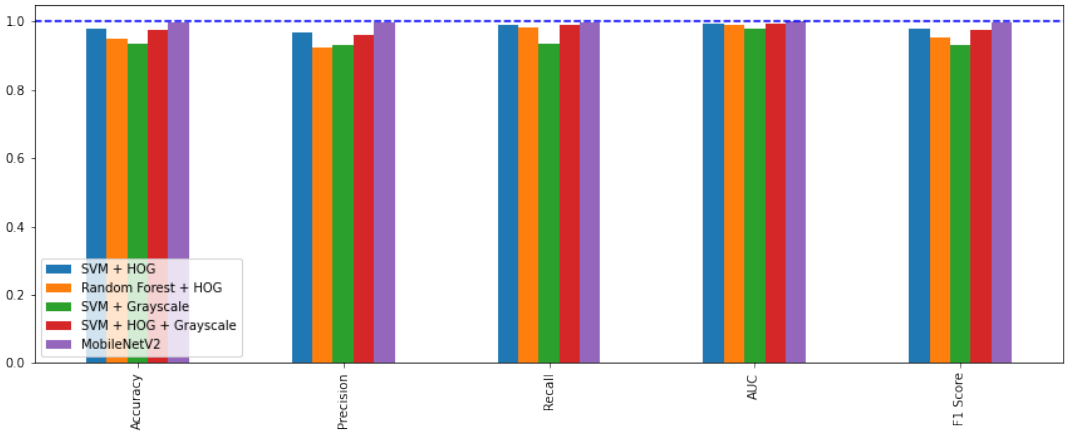

FIG. 9. Bar Plot Comparing Performance of Various Classifiers

VII. CONCLUSIONS

A supervised learning technique using SVM is applied to classify people wearing a mask or not. Important features are extracted from the images using HOG Feature Descriptor, which is used as an input for the model. This model performed with near-human perfection in distinguishing between the two classes. The AUC is 0.980 supports this claim that the SVM model works great in making the correct predictions. Accuracy and precision of 98% and 96.9% respectively give more evidence that the model is robust and not biased in predicting only 1 class. The model predicted 3.12% False positive, suggesting that only 3.12% of people are falsely classified as wearing a mask. At the same time, the model predicted 0.94% False-negative, which means that only 0.94% are falsely classified as not wearing a mask, suggesting that almost no one wearing a mask was misclassified. The performance is also on par with MobileNetV2, a state-of-the-art deep learning model trained using transfer learning technique with initial ImageNet weights. To deploy a system in real-time for detecting people wearing face-mask on the street, a Linear SVM model trained on HOG features is found to be most suitable as it takes the least amount of time in milliseconds while making predictions with an accuracy of 95.3%.

DATA AVAILABILITY

Data is available at:

- `https://drive.google.com/file/d/1UlOk6EtiaXTHylRUx2mySgvJX9ycoeBp/view`
- `https://github.com/TheSSJ2612/Real-Time-Medical-Mask-Detection/`

CODE AVAILABILITY

Code is available at `https://github.com/susano0/Face-Mask-Detection-using-SVM-HOG`

ACKNOWLEDGMENTS

I wish to express my deep sense of gratitude to Professor Marcin Abram for his valuable guidance and support. I would also like to extend my gratitude to TA Ninareh Mehrabi for her help during my work. Last but not least, I express my affection and gratitude to my parents for supporting and motivating me.

Appendix A: Sample Output

Face mask detection output using the Linear SVM classifier is shown in FIG. 10. It can be seen that Haar cascades detected 4 faces, and for each face, the SVM classifier correctly predicted whether the people are wearing a mask or not. The person wearing a mask has a green bounding box around his face, while people without a mask have a red bounding box.

FIG. 10. Face-Mask Detection Output

Appendix B: MobileNetV2 Modified Architecture

The top layers of the MobileNetV2 are removed, and 2 Dense layers are added to the original architecture. Initial weights used are from ImageNet, and all the parameters are later fine-tuned by training for 30 epochs in batches of 64. Adam optimizer with Binary cross-entropy loss function was used for training the model. The model summary and accuracy-loss graphs are shown in FIG. 11 and FIG. 12, respectively.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
keras_layer_10 (KerasLayer)  (None, 1280)              2257984
_________________________________________________________________
dense_20 (Dense)             (None, 64)                81984
_________________________________________________________________
dense_21 (Dense)             (None, 1)                 65
=================================================================
Total params: 2,340,033
Trainable params: 2,305,921
Non-trainable params: 34,112
_________________________________________________________________
```

FIG. 11. MobileNetV2 Model Summary

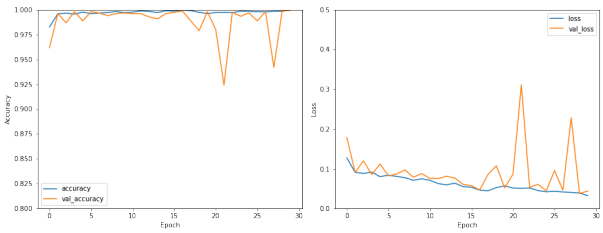

FIG. 12. Accuracy and Loss Graph

[1] Coronavirus (covid-19).

[2] How to protect yourself others.

[3] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review, Chaos, Solitons Fractals , 110059 (2020).

[4] 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 **167**, 108288 (2021).

[5] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, Implementation of principal component analysis on masked and non-masked face recognition, in *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)* (IEEE, 2019) pp. 1–5.

[6] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, *et al.*, Masked face recognition dataset and application, arXiv preprint arXiv:2003.09093 (2020).

[7] 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 **66**, 102692 (2021).

[8] P. Bhandary, prajnasb/observations.

[9] H. S. Dadi and G. M. Pillutla, Improved face recognition rate using hog features and svm classifier, IOSR Journal of Electronics and Communication Engineering **11**, 34 (2016).

[10] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Vol. 1 (IEEE, 2001) pp. I–I.