# Detecting Face Masks for Medical Professionals

Mushahid Khan
*Department of Computer Science*
*Ryerson University*
mushahid.khan@ryerson.ca

Sarah Sohana
*Department of Computer Science*
*Ryerson University*
sarah.sohana@ryerson.ca

Ahmed Aldaeni
*Department of Computer Science*
*Ryerson University*
ahmed.aldaeni@ryerson.ca

***Abstract*—The COVID-19 pandemic has caused a global health crisis. One of the effective protection methods is to practice social distancing. However, this is not always possible, especially for medical professionals. They have to save lives, wearing masks on a daily basis. For this project, we were interested in combining Convolutional Neural Networks(CNN) with classical machine learning algorithms to detect whether an individual is wearing a mask or not in real time. Our approach has two parts to it. The first part involves using CNN to do feature extraction. The second part involves using Support Vector Machine(SVM), K-nearest neighbor(KNN) and Decision Tree to do real-time classification of people wearing masks or not wearing masks. This method greatly improves the performance of the classical algorithms.**

***Index Terms*—Deep Learning; Convolutional Neural Network; Support Vector Machine; K-nearest neighbor; Decision Tree; COVID-19**

## I. Introduction

### A. Problem Description

COVID-19 virus is spreading very rapidly all around the world. It is affecting every industry and every day of life, which has caused a high infection and death rate. For the prevention of the spread of the virus, people have to practice social distancing. However, social distancing is not always possible especially in the case of medical professionals. Medical professionals deal with COVID-19 cases every day. They have to always protect themselves while they are saving lives, wearing masks. If there is a case where a doctor forgets to wear their mask, they could be in serious danger. In such a case, they should be notified immediately.

### B. Prior Work

Face mask detection, as an important research direction for computer vision, has been widely studied in recent years. In general, most of the focus has been on face construction and classification of whether someone is wearing a mask or not. For this project, our focus has been on identifying whether people are wearing or not wearing face masks to help reduce the transmission and spread of COVID-19. In [1], the authors presented a system for detecting face masks using Facemasknet deep learning network in real-time. The overall objective in [1] was to detect if an individual was wearing a mask or not in images and live video streams. The proposed system achieved 98.6% accuracy. [2] presented a hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. The model in [2] used deep transferring learning (ResNet50) as a feature extractor and classical machine learning algorithms to classify images of people with and without masks. The model in [2] achieved 99.49% accuracy.

## II. Overview

Machine learning solutions can greatly assist in the fight against COVID-19. Using classical machine learning algorithms with deep learning models can work great for detecting whether an individual is wearing a face mask or not. The proposed model here first trains the CNN algorithm to do feature extraction on images of people with and without masks. Once the features are extracted, it uses classical machine learning algorithms to detect whether a person is wearing a mask or not in real time. The classical algorithms used are SVM, KNN and Decision Tree.

## III. Problem Statement

Currently, the spread of the COVID-19 virus has caused a high infection and death rate. Since inception, there have been approximately 66 million reported cases. Of these 66 million, at least 1.5 million people have died. This global pandemic has meant that medical professionals are working around the clock to save lives while risking their own lives at the front lines. To prevent the spread of the virus, people need to wear face masks all the time, especially in the health sector. 40% of people with COVID-19 are asymptomatic but are potentially able to transmit the virus to others. So it is important to wear masks. Medical professionals always need to wear masks to not only protect themselves, but also to prevent the spread of the virus in hospitals. In the case of when a medical professional forgets to wear their mask, they should be notified immediately about it. This can greatly help save lives.

## IV. Dataset

For this project, the face mask image dataset from Kaggle was used [3]. This data set has 11,800 colored images of people with and without masks. All the images are not the same size. The dataset is balanced. This dataset was originally divided up into the following folders: Test, Validation and Train. We combined all the images into one folder, with two subfolders. The first subfolder consists all the images of people wearing masks and the second subfolder consists of images of people without masks. We did this so that we can do data preprocessing on all of them at once and then divide up the dataset into training, validation and test sets to maximize the efficiency as well as accuracy of the model. Few sample images from the dataset are shown in Fig. 1.


Fig. 1. Sample images from the dataset before preprocessing

## V. Methods and Models

### A. Data Preprocessing

Data preprocessing involves getting the data ready to be consumed by the machine learning algorithms. Before implementing the model, we had to do some image preprocessing. We started by converting all images into grayscale as colors are not an essential feature in detecting whether someone is wearing or not wearing a mask. Next, all the images were resized to be 100 x 100 pixels. This was done because each image in the dataset was of different size. Few sample images from the dataset after preprocessing are shown in Fig. 2.

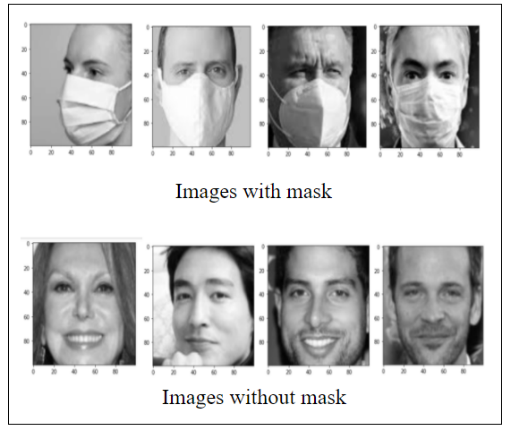


Fig. 2. Samples images from dataset after preprocessing

### B. Train, Validation and Test Sets

Once the images were preprocessed, they were split into training and test sets. 90% of the dataset was used to train the CNN model for feature extraction which was then used to train the classical machine learning algorithms. The remaining 10% was used to test the classical machine learning algorithms. Also, 20% of the training dataset was used as a validation set.

## VI. Methods and Models

The proposed model consists of two primary components. The first consists of using CNN to extract features of images of people with and without masks. The second consists of using the features extracted as inputs to train the following classical machine learning algorithms: SVM, KNN and Decision Tree. The classical machine learning algorithm that performs the best on features extracted from the test set will then be used to detect if an individual is wearing a mask or not in real time. This model in the training stage can be seen below in Fig. 3.

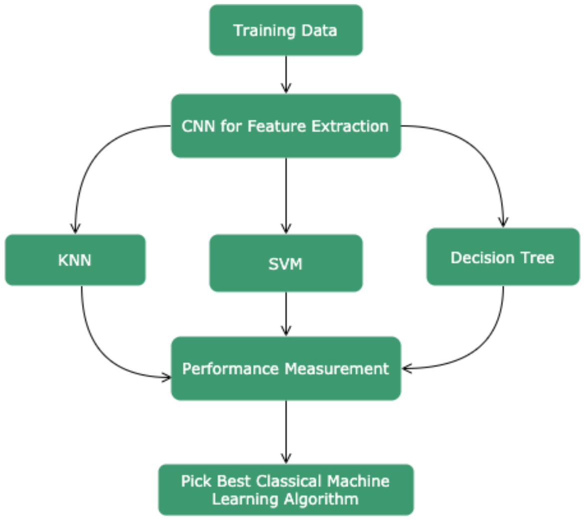


Fig. 3. The proposed model

When it comes to detecting in real time, OpenCV, which is a cascade classifier, will be used to detect the face. The area of interest, in this case being the face, will be passed as an image to CNN to have its features extracted. Then the features extracted will be passed into the best performing classical machine learning algorithm.

### A. *CNN for Feature Extraction*

CNN is a class of neural networks which is most commonly used for analyzing images. CNN can take in an image input and assign importance (weights and biases) to different aspects in the image as well as learn to differentiate between them. CNN's architecture is inspired by how the visual cortex is organized and is analogous to that of the connected patterns of neurons in a human's brain. For feature extraction, each input image is passed through a series of the following layers:

- Convolutional
- Pooling
- Rectified Linear Unit(ReLU)

The convolutional layer performs an operation called convolution on the input image. Convolution is a linear operation in which the set of weights are multiplied by the inputs. It puts images through convolutional filters or kernels. The dimensions of the output of the convolution layer are less than the input's. These filters activate certain features from the image.

Pooling layer reduces the spatial size of the input so that the number of parameters and computation in the network is reduced. Pooling also helps prevent overfitting as it provides an abstracted form of the representation. The ReLU layer changes all the negative values to 0 as shown in equation (1) below where x is the input to the ReLU layer. For this project, the CNN architecture can be seen in Fig. 4.

$$f(x) = max(0, x) \tag{1}$$

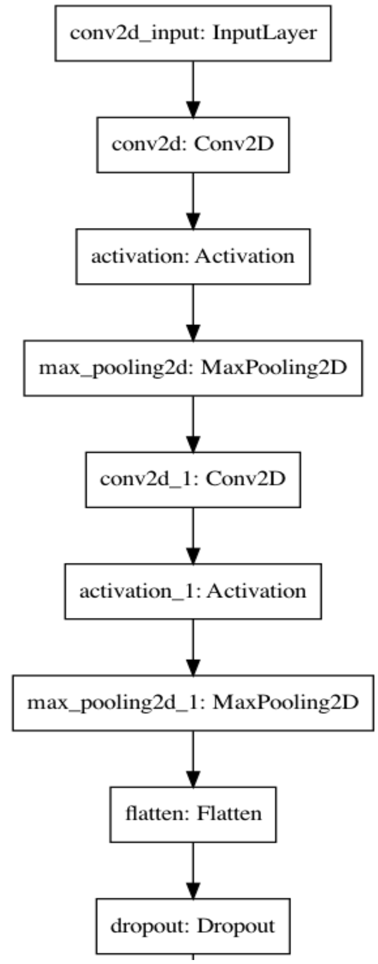


Fig. 4. CNN architecture used in this project

The input image, which is 100 x 100 pixels, is put through a convolutional layer of size 200 where each kernel is 3 x 3. Next, the output from the convolutional layer is passed into a ReLu layer and finally into a pooling layer of size 2 x 2. Then, the same thing is done again for the second time, but this time the convolutional layer has a size of 100. Then, the output from the second pooling layer is flattened into a vector. After being flattened, dropout is applied to it. Dropout layer works by randomly setting outgoing edges in the hidden layer to 0. This technique helps prevent a model from overfitting. Once dropout has been applied, the result is passed into a dense layer of reduced dimension. The output from the dense layer is the feature vector which we pass into the classical machine learning algorithms.

### B. SVM

SVM is a supervised machine learning algorithm which is used for classification. In this, data points are transformed into a higher dimension space using a technique called the kernel trick. Then, classification is performed by finding a hyper-plane which differentiates the two classes. For this project, we transformed the feature vector into 3 dimensions using the radial basis function kernel.

### C. KNN

KNN is a learning algorithm that looks at the labels of k number of data points which are closest to the target data point, using a distance formula such as euclidean distance, to make a prediction about the class for the target data point. Here, k is the hyperparameter and that needs to be chosen when the algorithm is getting used. For this project's dataset, using 9 neighbors worked best.

### D. Decision Tree

Decision tree is a classification model in the form of a tree structure. The data set is broken down into smaller subsets as the tree is incrementally developed.

### E. Performance Metrics Used

For evaluating the performance of the model, we used several performance metrics namely- Accuracy, Precision, Recall, F1 Score and false positive rate from Confusion matrix. Accuracy is the number of correctly classified images of people wearing masks over all the predictions made. Accuracy is a very good measure to evaluate performance if the data is balanced. Precision is the number of positive predictions that are correct over all the positive predictions. A positive prediction is a prediction which states that someone is wearing a mask. Recall is the measure of how many true positives get predicted out of all the positives in the data. F1 score is the harmonic mean between Precision and Recall. The higher the F1 score, the more accurate the model is. False positive rate (FPR) is a measure for how many negative cases get incorrectly identified as positive. In our project, it means the number of cases where people are not wearing masks but has been identified as wearing masks. The equations for Accuracy, Precision, Recall, F1 score and FPR are given below:

$$Accuracy = \frac{TP + TN}{(TP + FP) + (TN + FN)} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1Score = 2 * \frac{(Precision * Recall)}{Precision + Recall} \tag{5}$$

$$FPR = \frac{FP}{TN + FP} \tag{6}$$

TP is the count of True Positive samples, TN is the count of True Negative samples, FP is the count of False Positive samples, and FN is the count of False Negative samples. The TP, FP, TN and FN are represented in a grid-like structure called the Confusion matrix. This is used to describe the performance of a classification model on a set of test data for which the true values are known.

## VII. Results and Discussions

Fig. 7 presents the accuracy for the SVM, Decision tree and KNN given our test set. SVM and KNN had the highest accuracy of 0.989 whereas Decision tree had the accuracy of 0.983.

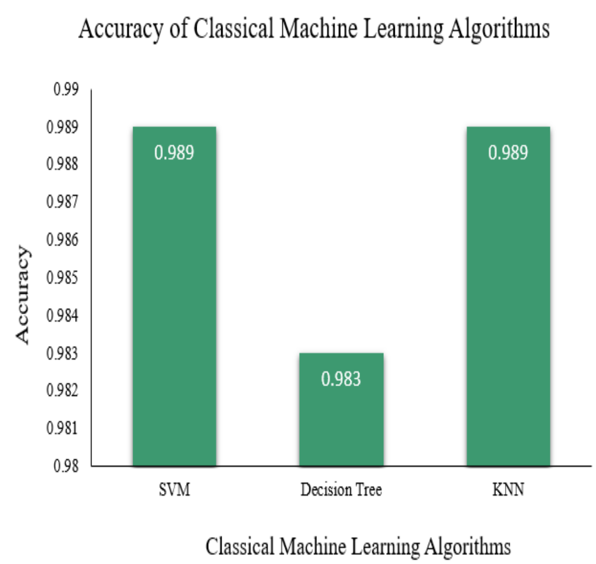


Fig. 5. Accuracy for classification algorithms

Fig. 8 illustrates the achieved Precision, Recall and F1 score for SVM, KNN and Decision Tree algorithms. SVM had the highest Precision (0.996), KNN had the second highest Precision score (0.994) and Decision tree had the lowest (0.990). For Recall, KNN had the highest value. F1 score seeks balance between Precision and Recall. Again, we observed that SVM and KNN had the highest F1 score (0.9891).

Confusion matrices are another useful insight into the performance of the classifiers. Fig. 9, 10 and 11

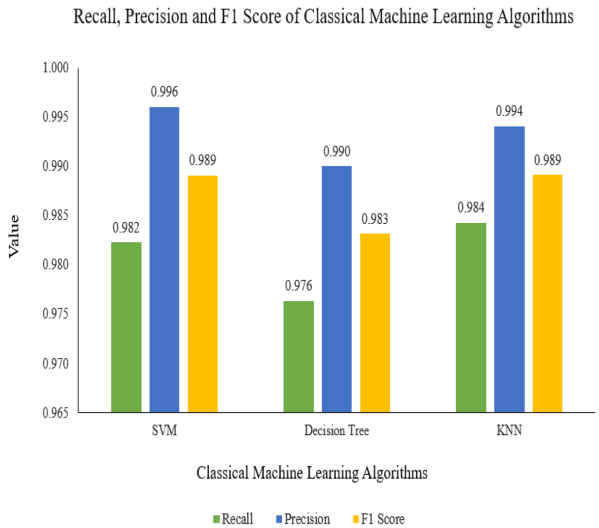


Fig. 6. Precision, Recall and F1 score for the classification algorithms

illustrate the confusion matrices for SVM, KNN and Decision tree respectively for the test data. In the Confusion matrix, 0 represents the label of wearing a mask and 1 represents the label of not wearing a mask. Using the Confusion matrices, we can calculate the false positive rates. False positive rate is important here because if the model misclassifies a person who is not wearing a mask as wearing a mask, this can cause serious problems. If a healthcare professional is not wearing a mask but the model predicts that they are wearing a mask, then the life of the healthcare professional is at risk. SVM had the false positive rate of 0.004, KNN had the rate of 0.006 and Decision tree had the highest rate of 0.010. We observed that SVM had the lowest false positive rate among the three classification models.

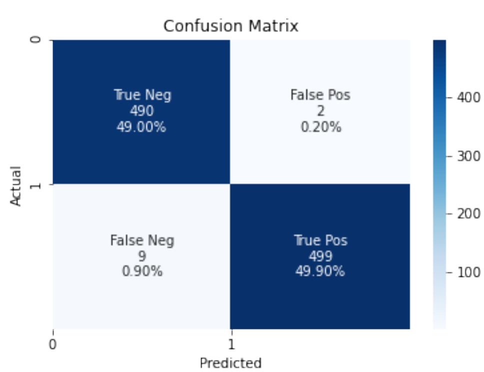


Fig. 7. Confusion Matrix for SVM

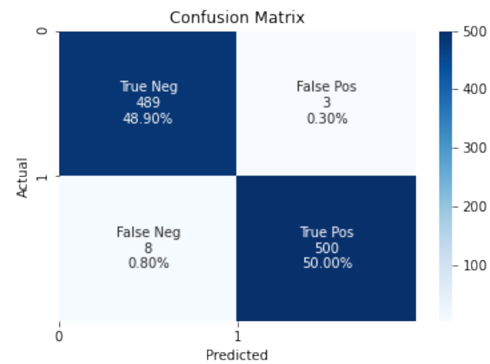


Fig. 8. Confusion Matrix for KNN

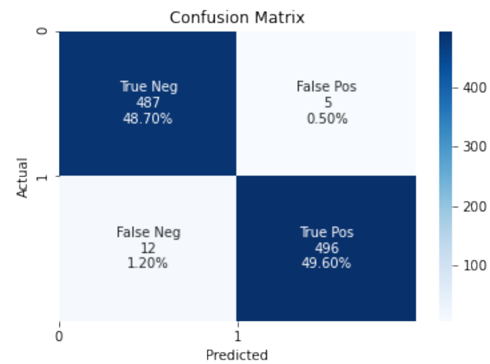


Fig. 9. Confusion Matrix for Decision Tree

Considering the Accuracy and F1 score, SVM and KNN performed best. Considering Precision, SVM had the highest score. Looking at the Confusion matrices and the false positive rates, SVM had the lowest false positive rate. Using the above evaluations, we chose SVM to do classification in real-time.

## VIII. Future Work

If we were to continue the project, we would like to add a cell phone notification system and a video surveillance aspect. The video surveillance will make sure anyone entering the premise has a mask on and will generate an alert if they are not wearing a mask. The cell phone notification system will send a text message alert to the individual, reminding them to wear a mask, if they are registered in the premise.

## IX. Implementation and Code

To get ideas for the project and its implementation, we looked at research papers and blogs. After looking through numerous sources, the research paper that gave us our initial idea of using CNN for feature extraction and classical machine learning algorithms for classification was [2].

This project was implemented in the Python language. For the implementation of CNN, Keras library's Sequential API from Tensorflow was used. Tensorflow is an open source library to help develop and train machine learning models. It particularly focuses on training deep neural networks. The Sequential API enables the creation of models layer by layer. For the classical machine learning algorithms, Scikit-Learn library was used. Scikit-Learn is an open source library for Python programming language which features various machine learning algorithms.The code for this project can be accessed at the following link -

https://github.com/sohanasarah/FaceMaskDetection

## REFERENCES


[1] M Inamdar, N Mehendale,"Real-Time Face Mask Identification Using Facemasknet Deep Learning Network," (July 29, 2020). Available at SSRN 3663305 (2020): https://ssrn.com/abstract=3663305 or http://dx.doi.org/10.2139/ssrn.3663305

[2] M Loey, G Manogaran, MHN Taha, NEM 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 (2020): 108288.

[3] "Face Mask 12K Images Dataset,"Accessed on: Dec 1, 2020.[Online].Available: https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset