International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072

Analysis of Various Machine Learning Classification Algorithms for COVID-19 Face Mask Identification

Gadige Vishal Sai[1], Panyala Harsha Vardhan Reddy[2], Sai Harsha Bandarupally[3], Chukka Nikhil[4]

[1]Software Engineer 1, NCR Corporation, Hyderabad, Telangana, India
[2]Associate Software Engineer, CGI Inc., Bangalore, Karnataka, India
[3]Full Stack Engineering Analyst, Accenture, Hyderabad, Telangana, India
[4]Application Developer, Accenture, Hyderabad, Telangana, India

---

Abstract - *Machine Learning is thoroughly implemented in a lot of applications to predict, forecast, analyze, and visualize data. During this tough COVID-19 pandemic, Machine Learning can play an enormous yet challenging role in containing this novel corona virus. Although there are vaccines available, there is no proper cure that can sweep out this virus completely. However, the chances of transmission of this virus get reduced if there is a proper usage of masks. Here, we use several machine learning (ML) classification algorithms like Logistic Regression, Random Forest Classification (RF), K-Nearest Neighbors (KNN) algorithm, and Support Vector Machines (SVM). Also, we use Adaboost to compare the performance of these algorithms and observed that the classification algorithm – Logistic Regression is powerful enough to classify whether a person is wearing a mask or not.*

Key Words: **COVID-19, Machine Learning, Classification Algorithms, Face-Mask identification & AdaBoost**

1.INTRODUCTION

Data Analysis is an important study for an organization to understand its problems related to the business. It also facilitates a way to explore data in many meaningful ways. For any research, data analysis plays a crucial role as it provides an explanation of various concepts, frameworks, and theories used. It is often said that "To prepare the right strategy at the right time, data analysis is used." One such scenario is the outbreak of the pandemic COVID-19. The novel Corona Virus COVID-19 has affected more than 210 countries and territories across the globe and is increasing at a rapid pace since then. At present, there is no complete cure other than maintaining proper health conditions by wearing a mask, using alcohol-based sanitizer, etc. and maintaining proper social distancing.

In this paper, we built several models, each based on a Machine Learning (ML) Classification Algorithm, applied on the face mask dataset provided by Prajna Bhandary. The dataset consists of 1376 images in total, where 690 images contain people with masks and the remaining 686 images contain people without face masks. Here, the models were built using Support Vector Machines (SVM), Random Forest Classification (RF), Logistic Regression, and K-Nearest Neighbors (K-NN) classification algorithms. In addition to this, we used Python's OpenCV module and other libraries used for classification. Also, we applied Adaboost to convert the weak classifiers to strong classifiers in order to increase the accuracy. Here, we built each model, created the confusion matrix, focused on the accuracy brought by them using the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) values and suggested a model with the best accuracy and other factors among the four models.

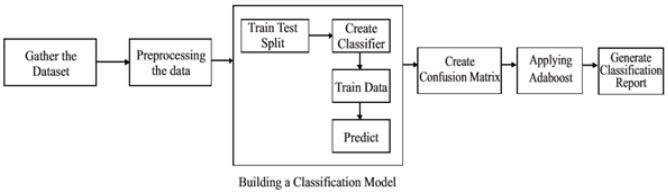

Fig -1: Dataflow diagram for the COVID-19 Face mask identification algorithm

2. RELATED WORK

The term Machine Learning can be coined as the scientific implementation of mathematical concepts by a machine, to develop computer programs that can access data and then use it to learn for themselves. Several applications are made using Machine Learning and recent researchers mainly focused on Computer Vision.

Related work includes an Automatic Face Detection System that can recognize a face, facial expression, head pose estimation, and also Human-Computer Interaction (HCI). It presents a comprehensive survey of various techniques explored for face detection in digital images [5].

Similar research includes a computer application that presents a new method of face recognized attendance management system using the Local Binary Pattern (LBP)

algorithm combined with advanced image processing techniques such as Histogram Equalization and image blending to improve accuracy [1].

Another research presents a method to generate accurate face segmentation masks from any arbitrary sized input image [2]. Similar researches include detecting multiple facial masks in a single frame.

3. METHODOLOGY

In this paper, we discussed and proposed the following method to identify whether a person is wearing a face mask or not.

1. Fetch the Dataset
2. Preprocessing the Data
3. Building a Classification Model
   - 3.1 Train Test Split
   - 3.2 Create Classifier
   - 3.3 Train Data
   - 3.4 Predict the Values
4. Create Confusion Matrix for the Model
5. Applying Adaboost
6. Generate Classification Report

Fig -2: Proposed Algorithm for Facemask identification

3.1. Fetching the Dataset

Here, to identify the presence of face mask or not, we used Face-Mask dataset provided by Prajna Bhandary. The dataset consists of 1376 images in total, where 690 images contain people with masks and the remaining 686 images contain people without wearing masks.

Fig -3: Dataset containing people wearing masks

Fig -4: Dataset containing people not wearing masks

3.2. Preprocessing the Data: Sample Code

Here, to preprocess the data, we used a common algorithm,

Begin:

1. ***for images in the dataset:***
   - ***1.1 Convert Image from RGB to Grayscale***
   - ***1.2 Resize the image to 110X110 pixels.***
2. ***Shuffle the data in order to increase the learning capability.***

End

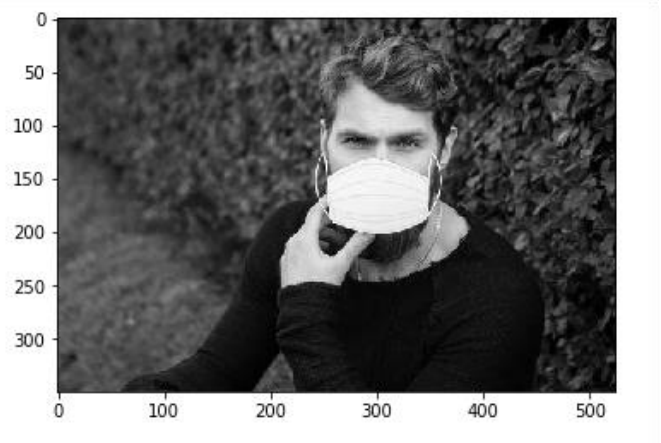

Fig -5: Sample image in the dataset

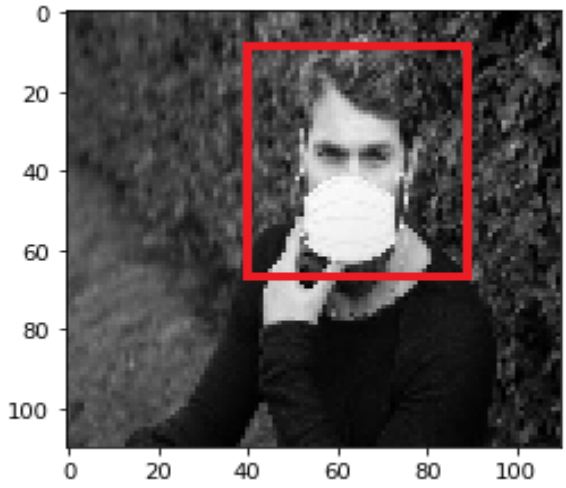

Fig -6: Image after preprocessing (Face identified)

3.3. Building a Classification model

The term classification refers to separating data into multiple classes. Each class represents a general physical entity (here, face with a mask, and without mask). Depending upon the need, the data, and the number of classes for classification, we select the appropriate classifier. Each classification model has an algorithm that has a set of parameters; based on the model we select values of the parameters, to generate new models with different accuracies, precision, support, and recall. With the help of grid search, we generate different models with different parameter values, and we choose a model with higher accuracy values.

In this paper, we mainly focus on building classification models using Logistic Regression, K-Nearest Neighbors (K-NN), Random Forest Classification (RF), and Support Vector Machines (SVM).

3.3.1. Support Vector Machine

SVM can be coined as a supervised learning model which abbreviates for Support Vector Machine. It contains associated algorithms that facilitate learning to analyze data, which is implemented for the purpose of prediction and classification analysis.

When trained using samples, each sample labelled with either one or other class category, the SVM builds a model which assigns new data to one or other category, by drawing a classifier. It is also known as Support Vector Networks which is a representation of data as data points in space, which are mapped so that the instances of different categories are separated using a classifier line which is as broad as possible. Whenever, further data (test data) is encountered, the values are also plotted on the space and are labelled accordingly.

Table -1: Table which shows the performance of SVM based on the parameters

	Large	Small
C	Low bias High variance	High bias Low variance
Gamma	High bias Low variance	Low bias High variance

3.3.2. Random Forest

Random Forest Classification (RF) is an ensemble learning technique used in prediction and classification analysis. The Random Forest or Random Decision Forests model can also be implemented in those tasks where we require multiple decision trees during training and producing the class label as output from the group of classifiers as per the classification problem. In general, random decision forests rectify the concept of over-fitting as observed in decision trees during its training.

Here, the accuracy of the model depends upon one important parameter, n_estimators which refers to the number of decision trees that should be implemented in a task.

3.3.3. Logistic Regression

Logistic Regression is a statistical technique that is used to create models that deal with the probability of a certain event or a class (here a face with a mask and a face without a

mask). When training data, trains the model; it maps the data on space and tries to separate the data points using a classifier line based on the training data. When new data (test data) is encountered, the data points from the test data are assigned a numerical value in the range 0 to 1, such that the sum of all the probabilities of the data is 1.

3.3.4. K-Nearest Neighbors

K-Nearest Neighbors is a machine learning, non-parametric technique mainly used in regression and classification analysis. In both regression and classification cases, the input data consists of the k nearest data points in the feature space. Here, the choice of the output depends on the application of K-NN in either regression or classification.

In the classification model, the K-NN algorithm considers the training data, plots the points on the feature space along with its class labels. When a k-value is given and when the test data is supplied, we classify a data point as a particular class when the majority of the k nearest points belong to a particular class label.

3.3.5. Train Test Split

The data that we obtained after preprocessing is shuffled data, i.e. it contains data of the images of both face with a mask and face without a mask. Then we divide the data as both training and test sets. Depending upon the ratio of the split, the learning capability of the model is changed. Generally, we can divide the data into train-test sets in the ratio of 60:40, 70:30, 75:25, 80:20, and so on. Here for our model, we considered the data to be segregated in the ratio of 80 and 20.

3.3.6. Create a classifier

A classifier is a specific algorithm or a function that is used to map the input training data to a specific class (here, face with a mask, and face without a mask). Based on the classifier, the test data will be classified into a class. Depending upon the model built, the subsequent library is fetched and then the corresponding classifier object is instantiated and later used for training purposes.

3.3.7. Training Data

Here, training data is 80% of the input data, that is fed on to the classifier to make sure that the classifier is trained and can predict accurately the new data (test data) with minimal error.

3.3.8. Predicting the values

Once the classifier is trained by the training data, the model then predicts the values of the remaining 20% test data by using the trained classifier.

3.4. Create Confusion Matrix for the model

After training and testing the model, the results of the prediction can be seen either by using visualization or using a confusion matrix. A confusion matrix has 4 elements, i.e. True Positives (True values predicted rightly), True Negatives (True values predicted wrongly), False Positives (False values predicted rightly), and False Negatives (False values predicted wrongly). Here, we always focus on reducing the values represented in the cells of True Negative and False Positive.

3.5. Applying Adaboost

AdaBoost is a boosting technique, used to create a highly précised prediction rule by combining several weak and unprecise rules. The concept of boosting is to consider the weak learners, make them strong to create a highly accurate prediction. The classifier models the relationship between the dependent training data (x) and the independent variables (y). Then by finding out the weak learners it iterates continuously till all the weak learners are trained sufficiently to provide correct results. The trained classifier is said to be efficient and reliable in its predictions based upon these three conditions:

1. Each classifier should be trained on “enough” training samples.
2. The classifier should provide good fit to the test samples by producing low training error.
3. The model should always be simple.

Initialization:

1. Given training data from the instance space $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{-1, +1\}$.
2. Initialize the distribution $D_1(i) = \frac{1}{m}$.

Algorithm:

for $t = 1, \ldots, T$: **do**

Train a weak learner $h_t : \mathcal{X} \rightarrow$ R using distribution D_t.

Determine weight α_t of h_t.

Update the distribution over the training set:

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.

end for

Final score:

$f(x) = \sum_{t=0}^{T} \alpha_t h_t(x)$ and $H(x) = sign(f(x))$

Fig -7: Adaboost algorithm

3.6. Generate Classification Report

In order, to calculate the aspect of predictions using any classification models, we construct the Classification report. The classification report is prepared by taking the values of TP, TN, FP, and FN generated by the confusion matrix. Each classification report shows the following metrics:

3.6.1. Precision

Precision: Prediction depicts, what percent of the prediction is correct. It is calculated using the formula,

Precision = (True Positive)/ (True Positive + False Positive)…….. Eq.(1)

3.6.2. Recall

Recall: Recall depicts, the amount of positive data points did you catch. It can also be defined as a classifier's ability to find all the positive points. It is calculated using the formula,

Recall = (True Positive)/ (True Positive + False Negative)…….. Eq.(2)

3.6.3. F1 – Score

F1-Score: F1-score depicts the percentage of the correct predictions. It can also be defined as the harmonic mean (HM) of both the recall and precision, such that it belongs to an interval (0.0 to 1.0). It is calculated using the formula,

F1-score = 2 * (Recall * Precision) / (Recall + Precision)…….. Eq.(3)

4. RESULTS AND DISCUSSIONS

In this classification problem, we successfully trained and built four models. The corresponding confusion matrices for these four models are as follows:

141	10
11	153

Fig -8: Confusion matrix (Error matrix) for Logistic Regression model after using Adaboost

84	7
21	135

Fig -9: Confusion matrix (Error matrix) for Random Forest model after using Adaboost

142	14
8	151

Fig -10: Confusion matrix (Error matrix) for K-Nearest Neighbors model after using Adaboost

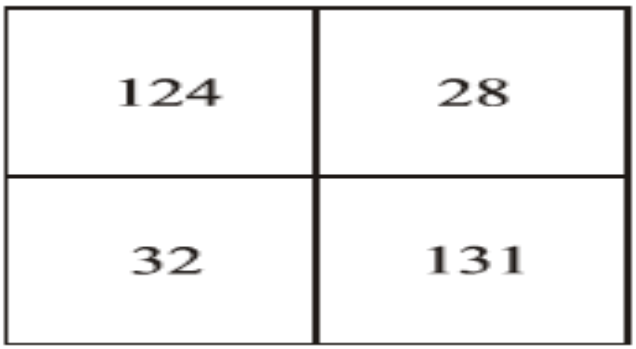

Fig -11: Confusion matrix (Error matrix) for SVM model after using Adaboost

Based on these confusion matrices, we calculated Precision, Recall, and F1 Score. By comparing these metrics, we tabulated the results as follows:

Table -2: Comparison of various classification models used

S.No.	Classification Algorithm	Precision	Recall	F1-Score
1.	Logistic Regression	0.9337	0.9276	0.9306
2.	Random Forest	0.9230	0.8000	0.8571
3.	K-Nearest Neighbours	0.9102	0.9466	0.9280
4.	Support Vector Machine	0.8157	0.7948	0.8051

Here, we tried to improve the accuracy of all these models by applying AdaBoost. However, AdaBoost does not show its effect on SVM because using SVM as a classifier for AdaBoost can look like going opposite to the benefit of the boosting principle since SVM is relatively a difficult classifier to train. After verifying the results, we can conclude that Logistic Regression turned out to be the best model for this Face-Mask Dataset. For further explanation, we also plotted a ROC curve for the model which is built using Logistic Regression.

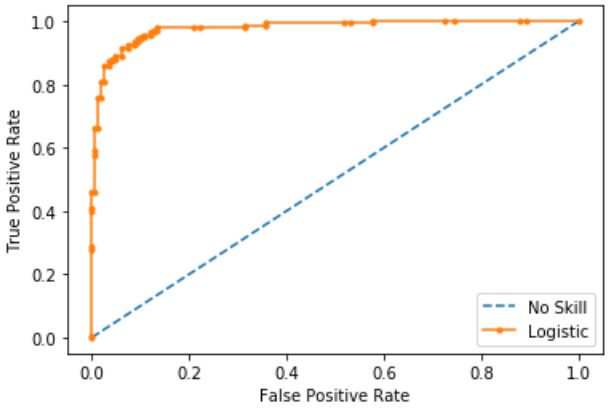

Fig -12: ROC Curve Logistic Regression

The Receiver Operating Characteristic (ROC) curve is considered as a performance measure for classification problems at multiple threshold changes. A probability curve, ROC, and measure or degree of separability tell us how highly a model is competent of distinguishing between classes. It is important to note that the range of ROC curves always lies between 0 and 1.

5. CONCLUSION AND FUTURE SCOPE

In this experimental paper, we successfully trained four classification models and observed that logistic regression turned out to be the best-proven model, considering the possible metrics. However, during this COVID-19 pandemic, it is suggested that wearing a mask primarily reduces the spreading of the coronavirus. Coronavirus is transmitted mainly through respiratory droplets and also affects the respiratory system. So, the widespread use of masks can help to contain the virus. Hence, this application can be deployed in crowded public places, where it can detect whether a person is wearing a mask or not. For example, this can be applied in ATM's, where the doors open automatically on identifying whether a person is wearing a mask or not. The future scope of this application would be creating a model using advanced Machine Learning concepts like Reinforcement Learning or using the implementation of self-learning neural networks.

REFERENCES

[1] Serign Modou Bah, Fang Ming, "An improved Face Recognition Algorithm and its application in attendance management system", published by Elsevier Inc. in December 2019.

[2] Toshan Meenpal, Ashutosh Balakrishnan, Amith Verma, "Face Mask Detection Using Semantic Segmentation", 2019 4th International Conference on Computing, Communications, and Security(ICCCS) on October 2019.

[3] S. Kumar, A. Negi, J. N. Singh, and H. Verma, "A deep learning for brain tumor mri images semantic

segmentation using fcn," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), Dec 2018, pp. 1–4.

[4] K. Li, G. Ding, and H. Wang, "L-fcn: A lightweight fully convolutional network for biomedical semantic segmentation," in 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec 2018, pp. 2363–2367.

[5] Ashu Kumar, Amandeep Kaur, Munish Kumar, "Face Detection Techniques: A Review", Article in Springers on Artificial Intelligence Review-July 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2015.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[10] T.-H.Kim,D.-C.Park,D.-M.Woo,T.Jeong,andS.-Y.Min,"Multi-class classifier-based adaboost algorithm," in Proceedings of the Second Sinoforeign-interchange Conference on Intelligent Science and Intelligent Data Engineering, ser. IScIDE'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 122–127.

[11] P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.

[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, July 2002.

[13] 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, Dec 2001, pp. I–I.

[14] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, "Towards real world human parsing: Multiple-human parsing in the wild," CoRR, vol. abs/1705.07206.