Face Mask Detection utilizing Tensorflow, OpenCV and Keras

Janavi Kartik Shah[1], Dhruv Jayprakash Shelke[2]

[1] Student, Dept. of Information Technology, St. Francis Institute of Technology, Mumbai, Maharashtra, India
[2]Student, Dept. of Electronics, Thakur College of Engineering and Technology, Mumbai, Maharashtra, India

---***---

Abstract - ***Face masks have been increasingly popular as a preventive precaution during dangerous illness outbreaks, which has boosted the demand for automated equipment that can spot people without masks in public areas. This project uses computer vision libraries and deep learning methods to present a Face Mask Detector. The detector uses OpenCV for real-time image preprocessing and visualization, and the TensorFlow framework with the Keras API for creating and training a Convolutional Neural Network. The main goal of this research is to create a reliable system that can properly tell who is wearing a face mask from who is not. The CNN is created with Keras, allowing for effective model definition and training. Real-time image acquisition from video streams or cameras is accomplished using OpenCV during the preprocessing stage, which is followed by scaling and normalization. The suggested system exemplifies how computer vision, deep learning, and real-time applications can work together. In order to effectively monitor mask compliance in congested areas, it provides a practical solution for public health and safety. The system is flexible and adaptable to many deployment settings thanks to the integration of TensorFlow, Keras, and OpenCV, which supports continuous efforts to reduce the spread of contagious diseases.***

***Key Words*:** *Covid-19, Machine Learning, Mask Detection, Convolutional Neural Network, TensorFlow*

1. INTRODUCTION

According to the World Health Organization's (WHO) Situation Report 205, the coronavirus illness 2019 (COVID-19) has infected more than 20 million people worldwide and caused more than 0.7 million fatalities. A wide range of symptoms, including minor signs and severe illness, are displayed by the condition, with respiratory problems like shortness of breath being a frequent occurrence. Elderly people who already have lung diseases are especially vulnerable to serious COVID-19 consequences, which puts them at greater risk.[1] Human coronaviruses like 229E, HKU1, OC43, and NL63 are well known to the general public. Other viruses like 2019-nCoV, SARS-CoV, and MERS-CoV initially infected animals before mutating into human-transmissible forms.[2] People who have respiratory issues run the risk of unintentionally exposing people nearby to infectious droplets, which could result in contact transmission. Additionally, the surroundings of an infected person can encourage contact transmission when virus-carrying droplets land on neighboring surfaces.[3]

To lessen the effects of COVID-19 and other respiratory viral diseases, it is imperative to wear a medical mask. Understanding when to wear masks for source control or COVID-19 transmission prevention is crucial for the general public. Utilizing a mask provides benefits such as lowering exposure risk during the "pre-symptomatic" stage and removing stigma associated with mask use for viral containment.[4] Medical masks and respirators for healthcare workers should be prioritized, according to the World Health Organization (WHO). Because of this, finding face masks has become increasingly important in today's global culture.

Face mask detection requires locating the facial area and then determining whether a mask is there. The general object detection challenges where object classes are determined have similarities to this difficulty. Face recognition, on the other hand, concentrates on identifying a particular category, namely faces, and has a variety of purposes, including autonomous driving, teaching, and spying. TensorFlow, Keras, and OpenCV are basic machine learning tools that the study uses to address this issue in an easy-to-follow manner.

The subsequent sections of this paper are structured as follows: Section II delves into relevant research on face mask detection, while Section III outlines the characteristics of the employed dataset. The specifics of the integrated packages for constructing the proposed model are elaborated in Section IV. Section V provides an outline of our approach, and experimental outcomes, along with analysis, are presented in Section VI. The paper concludes in Section VII, summarizing findings and outlining potential future endeavors.

2. DATASET

Three datasets were used for the experiment. Each dataset has 2994 pictures in it. the first dataset to show people with face masks. The second shows people who are not hiding their faces. The most recent dataset depicts people incorrectly donning masks. The bulk of the photos in Figure 1 have a frontal face position that centers on a single face that is consistently covered in a white mask.

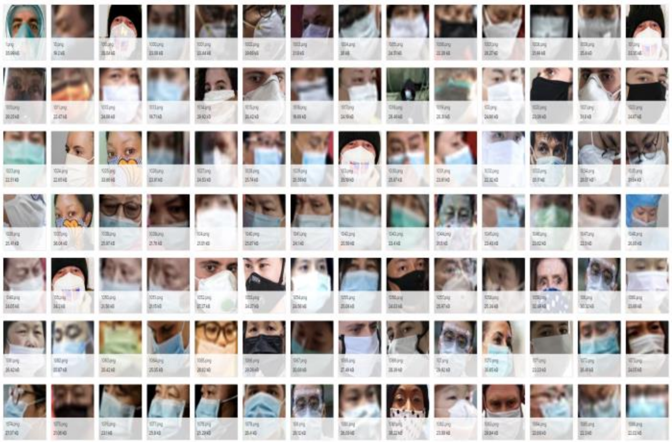


Fig. 1:. Dataset 1 sample

The 2994 photos in Dataset 2 clearly show faces without masks. Figure 2 displays many faces with various facial positions, including head twists, tilts, and slants. The photos also feature many mask designs, each with its own color scheme.

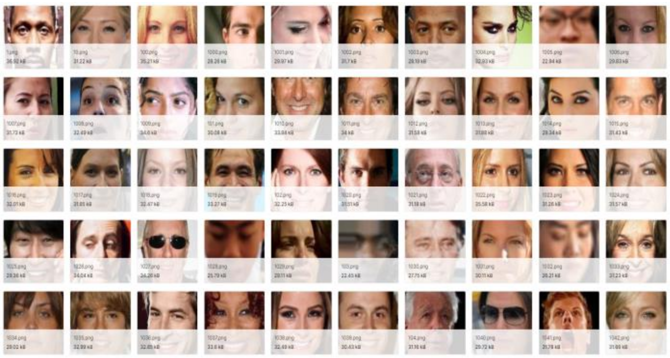


Fig. 2: Samples from Dataset 1 showing faces with no masks.

3. METHODOLOGY

Gathering and Preparing Data:

Assemble a dataset of pictures of people wearing and not wearing face masks. You can make your own datasets or use ones that are freely accessible. Create two subfolders in the dataset: one for photos with masks and the other for images without. To ensure uniform dimensions for the model input, resize and otherwise prepare the photos.

Data enhancement (optional but advised):

Apply data augmentation strategies like rotation, scaling, flipping, and brightness modifications using tools like ImageDataGenerator from Keras to broaden the diversity of your dataset and enhance model generalization.

Architecture and model selection:

As the foundation for your face mask detector, pick a suitable pre-trained deep learning model. ResNet, MobileNetV2, and VGG16 are popular options. Change the top layers of the selected model to correspond to the two classes ("with mask" and "without mask") in your problem. For fine-tuning, if necessary, add more layers.

Compilation of Models:

Compile the model using the proper optimizer (such as Adam), evaluation metric (such as accuracy), and loss function (often binary cross-entropy for binary classification).

Training:

Create training and validation sets from your dataset (for example, 80% training, 20% validation). To train your network on the training data, use the Keras model's fit function. Keep an eye on the validity accuracy to prevent overfitting.

Model assessment:

On a different test dataset that it hasn't seen during training or validation, gauge how well your trained model performs. To evaluate the model's performance, compute metrics like accuracy, precision, recall, and F1-score.

Deployment:

Prepare your trained Keras model for deployment by converting it to TensorFlow's SavedModel format. To record video frames from a webcam or other video source, use OpenCV. Preprocess the image for every frame, then send it into the algorithm for forecasting. Based on model predictions, create bounding boxes around any faces that were recognized and mark them as "with mask" or "without mask."

4. WORKING

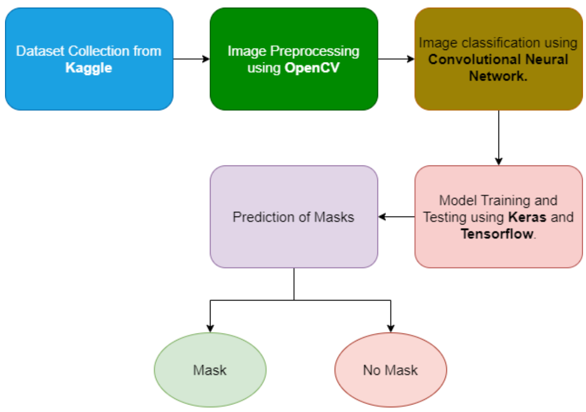


Fig. 3: Block Diagram for Face Mask Detection.

A dataset procured from the reputable platform Kaggle has been meticulously gathered, encompassing a diverse and comprehensive collection of data pertinent to the task at hand. This dataset serves as the foundation for the subsequent analytical processes. To ensure the quality and suitability of the dataset for analysis, a series of preparatory steps have been undertaken. The images contained within the dataset have undergone a meticulous processing phase utilizing the advanced OpenCV framework. This preprocessing phase is a critical aspect of the overall workflow, as it ensures that the subsequent analytical procedures are conducted on standardized and enhanced image data.

Subsequently, a sophisticated Convolutional Neural Network (CNN) architecture has been harnessed to undertake the pivotal task of image classification. CNN's ability to learn intricate features and patterns within images makes it particularly well-suited for discerning whether individuals are wearing masks or not. This stage is integral to the overall process, as the model's efficacy and accuracy hinge upon its capacity to effectively differentiate between masked and unmasked faces. The training and testing of the CNN model have been executed using industry-standard tools and libraries, specifically TensorFlow and Keras. These libraries provide a robust foundation for building, training, and evaluating machine learning models. The model's performance has been rigorously assessed through rigorous testing procedures to ensure its predictive capabilities align with the intended objectives. Upon successful training and validation, the resultant CNN model has evolved into a potent facial scanning system. This system is adept at examining images and making predictions regarding the presence or absence of facial masks. The outcomes of this predictive analysis culminate in a clear-cut categorization of the images into two distinct groups: those portraying individuals wearing masks and those portraying individuals without masks.

5. RESULTS

When the person removes the mask, a box that surrounds the person's face indicates whether or not they are wearing a mask. It recognizes the name of the individual who is not wearing a face mask if that person's face is registered in the database. This is shown in figure 5.1 mentioned below.

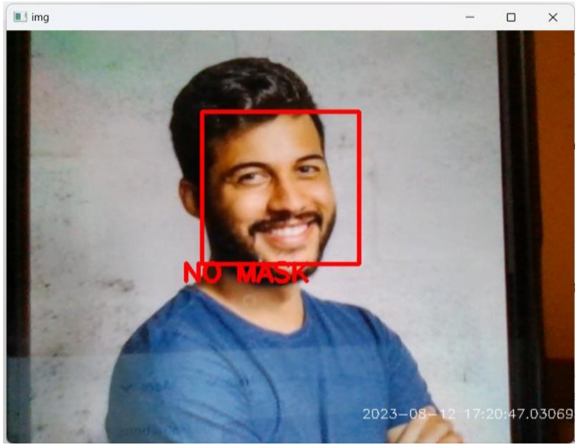


Fig 6: When the person doesn't wear a mask.

When a person wears a mask, the data model predicts that the person has a face mask and encloses the face with a green box, rather than a red one, to indicate the status of the mask being used in public spaces. This is mentioned in Figure 5.2 shown below.

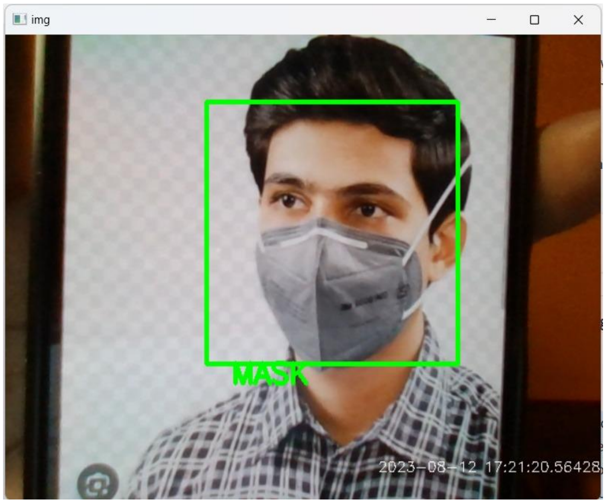


Fig. 5: A person wearing a mask.

The other important aspect of results is to check the accuracy of the system. These are compared over two datasets one with masks and one with no masks. The model is trained, validated, and tested using two datasets. According to dataset 1, the technique achieves accuracy of up to 95.77%. Fig. 5.3 shows how the cost of inaccuracy is reduced by this optimized precision. Dataset 2 is more adaptable than Dataset 1 because it has more faces in the frame and different sorts of faces with various angles. As a result, the model on dataset 2 achieves an accuracy of 94.58%, as shown in Fig. 5.4. MaxPooling is a key factor in this precision, among other things. It reduces the amount of parameters the model must learn while also giving the internal representation some basic translation invariance.

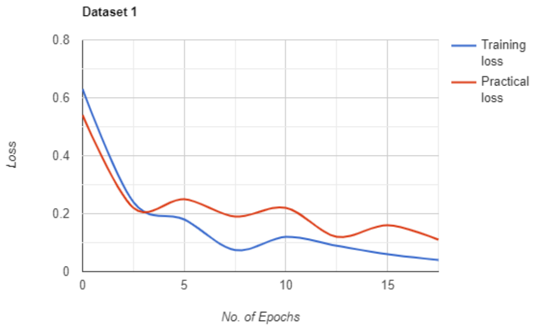
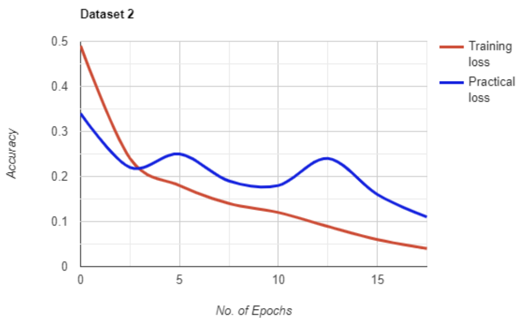


Fig 7: Epoch vs Loss for Dataset 1.

The input representation, which consists of a picture, is down-sampled via this sample-based discretization procedure to make it less dimensional. The optimal value for the number of neurons is 64, which is not an excessive number. The performance can be poorer with a lot more neurons and filters. The major part (face) of the image is filtered out to accurately and without overfitting detect the presence of the mask thanks to the optimum filter parameters and pool size.



.

Fig. 8: Epochs vs Loss for Dataset 2

6. CONCLUSIONS

In conclusion, this study presents a robust solution to the increasing demand for automated face mask detection in public settings, particularly during health crises. By employing a combination of computer vision libraries and deep learning techniques, the developed Face Mask Detector exhibits its efficacy in accurately identifying individuals wearing masks. Leveraging OpenCV for real-time image preprocessing and visualization, alongside TensorFlow and the Keras API for Convolutional Neural Network creation and training, the system showcases a cohesive integration of advanced technologies.

The practical significance of the research lies in its potential to enhance public health and safety through effective mask compliance monitoring. The synergy of computer vision, deep learning, and real-time applications is harnessed to provide a versatile solution adaptable to various deployment scenarios. By contributing to the mitigation of contagious diseases, the proposed system showcases the value of cutting-edge technologies in addressing real-world challenges, ultimately fostering responsible behavior and safety in shared environments.

REFERENCES

[1] Harish A., D. Kalyani, and R. Krishna Shri, “Face Mask Detection Using OpenCV,” CFP21ONG-ART; 978-0-7381-1183-4M.

[2] “Coronavirus Disease 2019 (COVID-19) – Symptoms”, Centers for Disease Control and Prevention, 2020. [Online]. Available:https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html. 2020.

[3] W.H.O., “Advice on the use of masks in the context of COVID-19: interim guidance”, 2020.

[4] M. Jiang, X. Fan and H. Yan, “RetinaMask: A Face Mask detector”, arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2005.03950. 2020.

[5] "YOLOv4: Optimal Speed and Accuracy of Object Detection" by Alexey Bochkovskiy, et al. (2020)

[6] "A Survey of Deep Learning-Based Face Mask Detection" by A. Dey, et al. (2021)

[7] "Deep Learning-Based Automatic Detection and Analysis System for COVID-19" by Apostolopoulos, Ioannis D., and Tzani, Aggeliki (2020)

[8] "A Cascade of Convolutional Neural Networks for Masked Face Detection" by Yang, Zhongyuan, and Luo, Zhongjie (2020)

[9] B. Suvarnamukhi and M. Seshashayee, “Big Data Concepts and Techniques in Data Processing”, International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 712-714, 2018. Available: 10.26438/ijcse/v6i10.712714.

[10] F. Hohman, M. Kahng, R. Pienta and D. H. Chau, “Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers,” in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 8, pp. 2674-2693, 1 Aug. 2019, doi: 10.1109/TVCG.2018.2843369.

[11] C. Kanan and G. Cottrell, “Color-to-Grayscale: Does the Method Matter in Image Recognition?”, PLoS ONE, vol. 7, no. 1, p. e29740, 2012. Available: 10.1371/journal.pone.0029740.

[12] Opencv-python-tutroals.readthedocs.io. 2020. Changing Colorspaces — Opencv-Python Tutorials 1 Documentation. [online] Available at:https://opencv-python-tutroals.readthedocs.io/en/latest/py tutorials/py imgproc/py colorspaces/py colorspaces.html. 2020.

[13] M. Hashemi, "Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation", Journal of Big Data, vol. 6, no. 1, 2019. Available: 10.1186/s40537-019-0263-7 . 2020.

[14] S. Ghosh, N. Das and M. Nasipuri, "Reshaping inputs for con-volutional neural network: Some common and uncommon methods", Pattern Recognition, vol. 93, pp. 79-94, 2019. Available: 10.1016/j.patcog.2019.04.009.

[15] R. Yamashita, M. Nishio, R. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology", Insights into Imaging, vol. 9, no. 4, pp. 611-629, 2018. Available: 10.1007/s13244-018-0639-9.

[16] "Guide to the Sequential model - Keras Documentation", Faroit.com, 2020. [Online]. Available: https://faroit.com/keras-docs/1.0.1/getting-started/sequential-model-guide/. 2020.

[17] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2020 Activation Functions: Comparison Of Trends In Practice And Research For Deep Learning. [online] arXiv.org. Available at: https://arxiv.org/abs/1811.03378. 2020.