Face Mask Detection from Live Footage

Indranil Banerjee[1] , Swarnali Das[1], Animesh Banerjee[1], Pallab Santra[1] and Suman Karmakar[1]

[1]NSHM Knowledge Campus Durgapur

Abstract. In our efforts to protect ourselves from the COVID-19 Pandemic, the use of face masks has become nearly ubiquitous. It has become increasingly crucial, especially in public settings like shopping malls, theaters, and parks, to monitor whether individuals in the crowd are adhering to mask-wearing guidelines. The development of an AI solution designed to identify whether a person is wearing a face mask and grant them access could bring significant societal benefits. This paper involves the creation of a straightforward face mask detection system using Convolutional Neural Networks (CNN), which is a deep learning technique. The CNN model is constructed using the TensorFlow framework and the OpenCV library, both well-suited for real-time applications. This version can serve as a foundation for the development of a comprehensive software application, enabling anyone to test it before its public release. The accuracy of this version exceeds 96 percent, and it can also be integrated with additional methods to further enhance accuracy levels.

Keywords: Convolutional Neural Networks, Deep learning, Face mask detection, Artificial Intelligence.

1. Introduction

The year 2020 brought about a series of remarkable events, with the emergence of the COVID-19 pandemic as the most profoundly life-altering occurrence of the year. This global health crisis has had a profound impact on people's well-being and livelihoods, leading to the enforcement of stringent measures to control its spread. From basic hygiene practices to advanced medical treatments, individuals have put forth significant efforts to protect their own health and contribute to the safety of society. Among the various types of personal protective equipment, face masks have become an essential item. People routinely wear face masks when leaving their homes, and authorities have enforced mask mandates in public places and group settings. To ensure compliance with this vital safety measure, a comprehensive strategy is required. One potential solution is the implementation of a face mask detection system. Face mask detection entails determining whether an individual is wearing a mask. This task involves two key components: face detection and mask detection. Face detection is a subset of object detection, with applications in fields such as security, biometrics, and law enforcement. Although several face detection systems exist and have been deployed worldwide, there is an ongoing need for optimization to combat the ongoing increase in

COVID-19 cases. In this paper, our goal is to create a face mask detector capable of distinguishing individuals wearing masks from those who are not. This report presents our proposed detector, which utilises SSD (Single Shot MultiBox Detector) for face detection and employs a neural network for mask detection. The algorithm's implementation covers images, videos, and live video streams. The following sections of this report offer a comprehensive organisational structure and further insights into the paper.

2. Previous Work

In the year 2020 [1-6], a series of remarkable events unfolded, with the COVID-19 pandemic emerging as the most profoundly life-altering occurrence since the year's inception. This global health crisis has had a profound impact on people's well-being and livelihoods, leading to the implementation of strict measures to curb its spread. From adhering to fundamental hygiene practices to advanced medical treatments, individuals have made extensive efforts to protect their own health and contribute to the safety of society. Among the array of personal protective equipment, face masks have become an indispensable item. People routinely wear face masks when leaving their homes, and authorities strictly enforce mask-wearing in public places and group settings. To ensure compliance with this essential safety measure, there is a need for the development of a comprehensive strategy. One potential solution is the implementation of a face mask detection system. Face mask detection involves determining whether an individual is wearing a mask. This task can be divided into two key components: face detection and mask detection. Face detection is a subset of object detection with applications in various fields, such as security, biometrics, and law enforcement. While several face detection systems have been developed and deployed globally, there is an ongoing need for optimization to combat further increases in COVID-19 cases.

In this paper, our objective is to create a face mask detector capable of distinguishing individuals wearing masks from those who are not. This report outlines our proposed detector, which leverages SSD (Single Shot MultiBox Detector) for face detection and employs a neural network for mask detection. The algorithm's implementation covers images, videos, and live video streams. The subsequent sections of this report provide a detailed organizational structure and further insights into the paper.

2.1. Design

The objective of this system is to categorize whether an individual is wearing a mask or not, using input from:

- Images
- Real-time streaming videos

The classification process for the images is carried out in two phases:

1st Phase

- In the initial phase, the face mask dataset is loaded onto the system.
- A variety of Supervised Learning classifiers are employed to construct a training model.

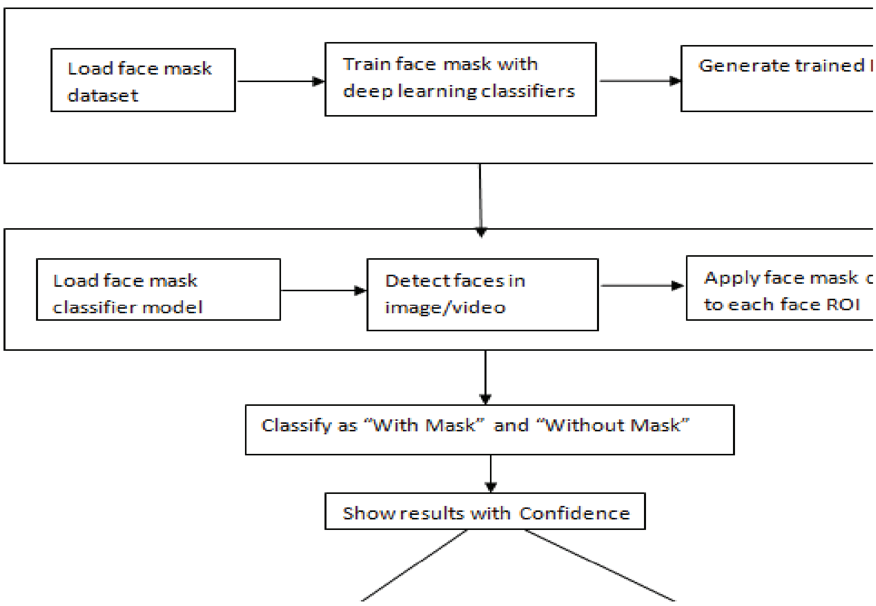


Fig. 1. The block diagram of the developed framework

2nd Phase:

Begin by opening the mask classifier model and loading it.

1. Proceed to scan the images or video stream for faces.
2. Apply the classifier to each region of interest (ROI) on the detected faces.
3. Categorise the images as either 'With Mask' or 'Without Mask.'

2.2 Implementation

This paper presents an intelligent framework designed for face mask detection. As the number of COVID-19 cases decreases, many workplaces are gradually reopening with both partial and full staffing, and educational institutions are also contemplating resuming their activities. Our system can be deployed at the entrances of a range of establishments, encompassing businesses, educational institutions, and both public and private offices. When the system identifies an individual without a face mask, it will produce an audible alert, prompting them to put on a mask. A visual representation of the developed framework can be seen in Figure 1.

A. Proposed Workflow:

We have opted to create a straightforward Convolutional Neural Network (CNN) model using TensorFlow and the Keras library, along with OpenCV, to ascertain whether an individual is wearing a face mask for their safety. We elaborate on all facets of our work below.

B. Deep Learning Architecture:

The deep learning architecture is devised to glean essential nonlinear features from the provided samples, which can subsequently be utilized to predict unseen samples.

C. Image Processing:

The system utilizes a Haar Cascade Classifier to capture input from a video camera. The images captured via the webcam undergo preprocessing before advancing to the subsequent stage. During preprocessing, the RGB color image is converted to grayscale, eliminating superfluous information unnecessary for face mask detection. Subsequently, the images are resized to a uniform (150x150) size, followed by normalization to ensure pixel values fall within the range of 0 to 1. This normalization expedites the learning process for the algorithm and facilitates the extraction of vital image features.

D. Dataset Collection:

To train our deep learning architecture, we compiled a dataset, with the success of the learning technique heavily dependent on CNN. The dataset, sourced from reference [10], exclusively comprises face images, totaling around 1,315 images. Among these, 658 images depict individuals wearing face masks, while 657 images portray individuals without face masks. For training purposes, 80% of the images from each category are employed, while the remaining images are reserved for testing. Figure 2 illustrates examples of images from the two distinct categories.

E. Architecture Development:

Our learning model is constructed using CNN, renowned for its prowess in image pattern recognition. The neural network must learn from data representing both classes. The architecture encompasses an input layer, multiple hidden layers, and an output layer. The hidden layers feature numerous convolution layers, each housing 100 kernels of dimensions 3x3. Max pooling layers, with a window size of 2x2, consolidate findings from the preceding convolution layer by selecting the maximum value within each 2x2 window. This reduces the spatial representation size and trims the number of parameters, streamlining network computations. The output from the convolution layers is flattened and converted into a 1-D array. Subsequently, a dropout layer and two dense layers follow. The dropout layer guards against overfitting by selectively deactivating units, while the dense layers contain neurons that grasp nonlinear features. The flattened outcome proceeds to the initial dense layer with 50 nodes and eventually to a second dense layer with two nodes corresponding to the two categories.

F. Alert Generation:

The primary purpose of our system is to identify individuals not wearing face masks. The learning architecture categorizes input images as either “mask” or “no mask.” If an individual is detected without a mask, a persistent beep alert is triggered until they wear one. When everyone is compliant with mask usage, they are shielded from the virus, rendering our system an effective tool in containing the spread of COVID-19.

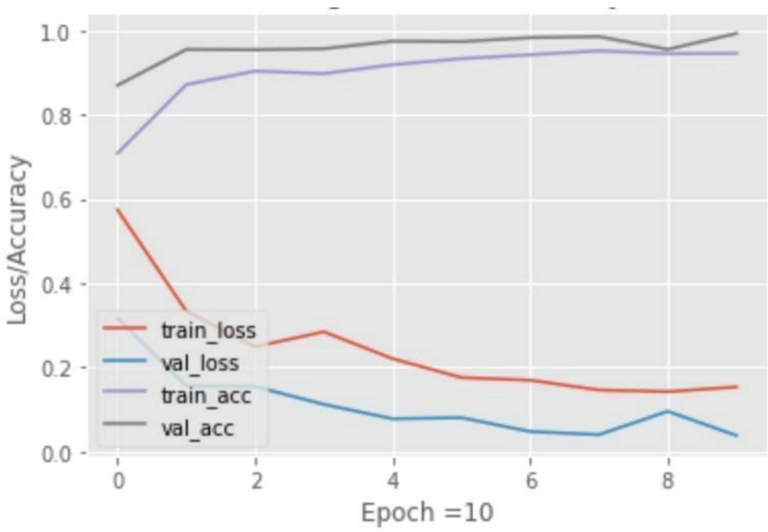


Fig. 2 Training and Loss Accuracy

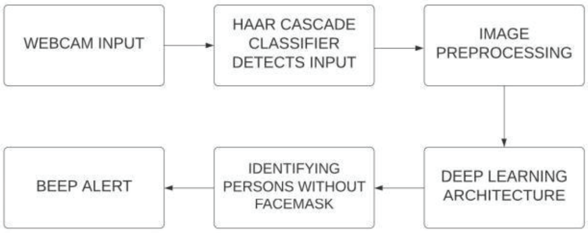


Fig. 3 Block Diagram of the proposed model

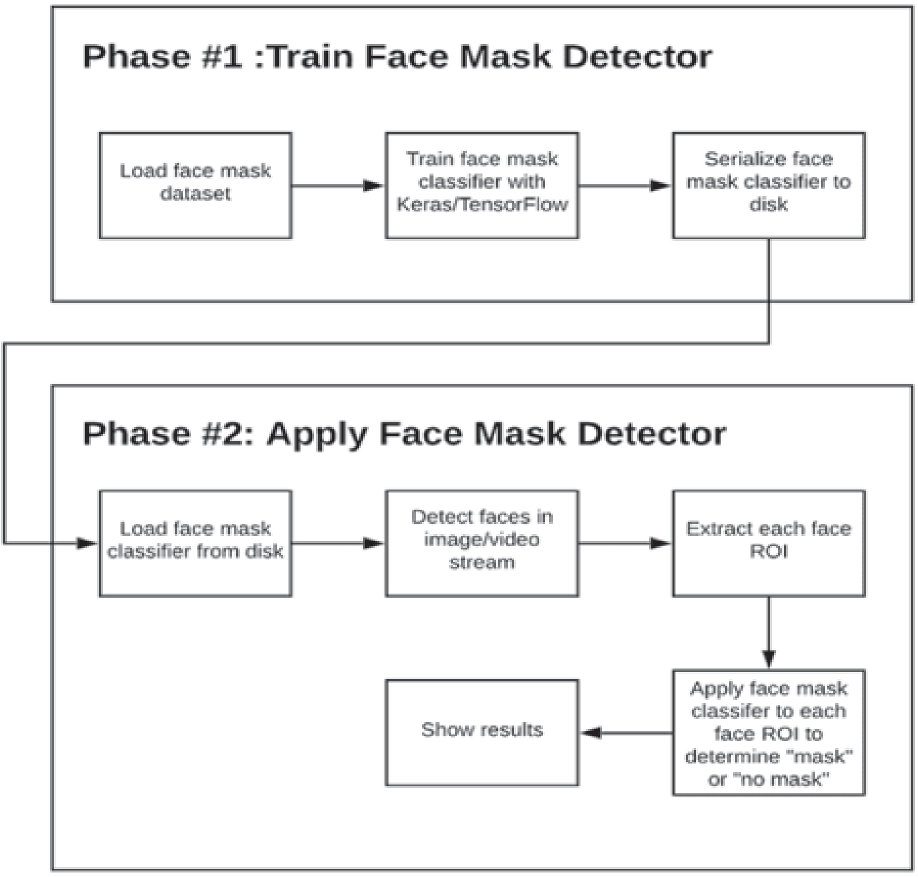


Fig.4. Methodology for face mask detection

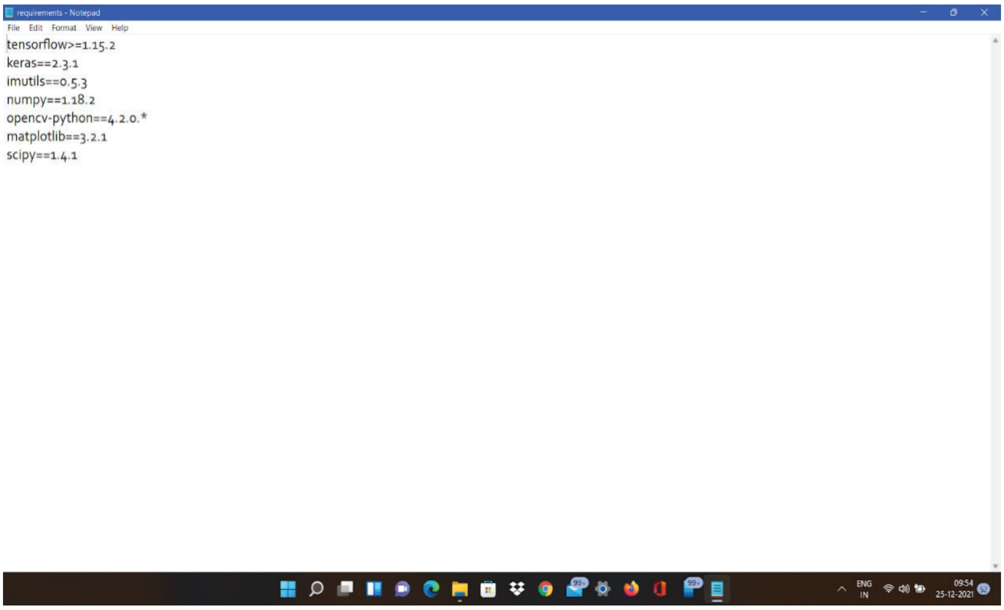
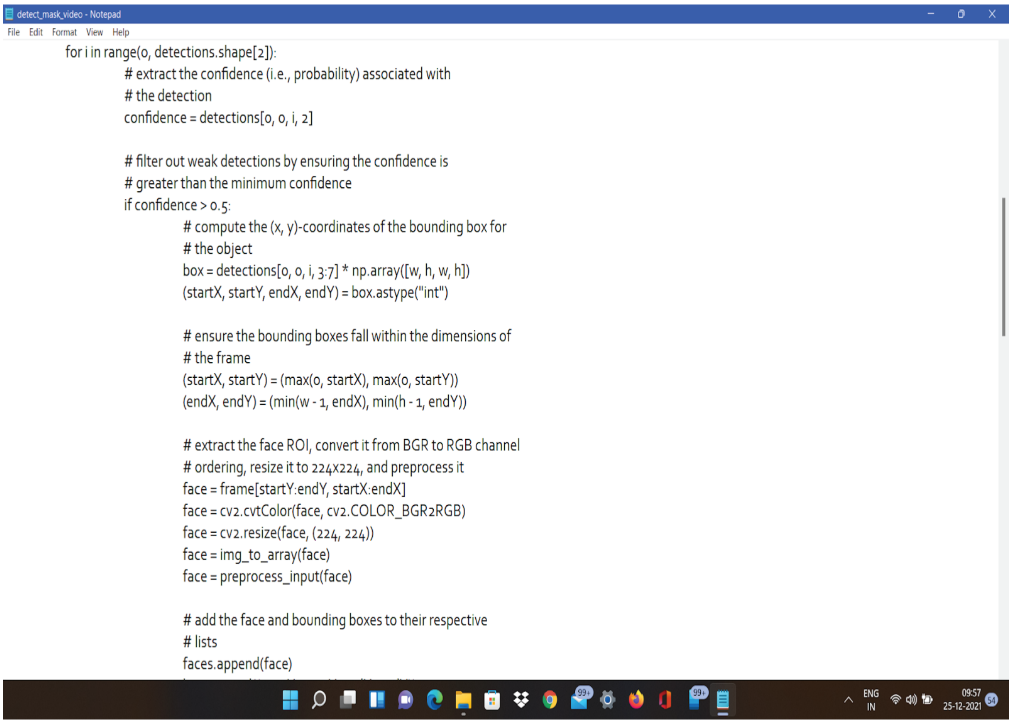
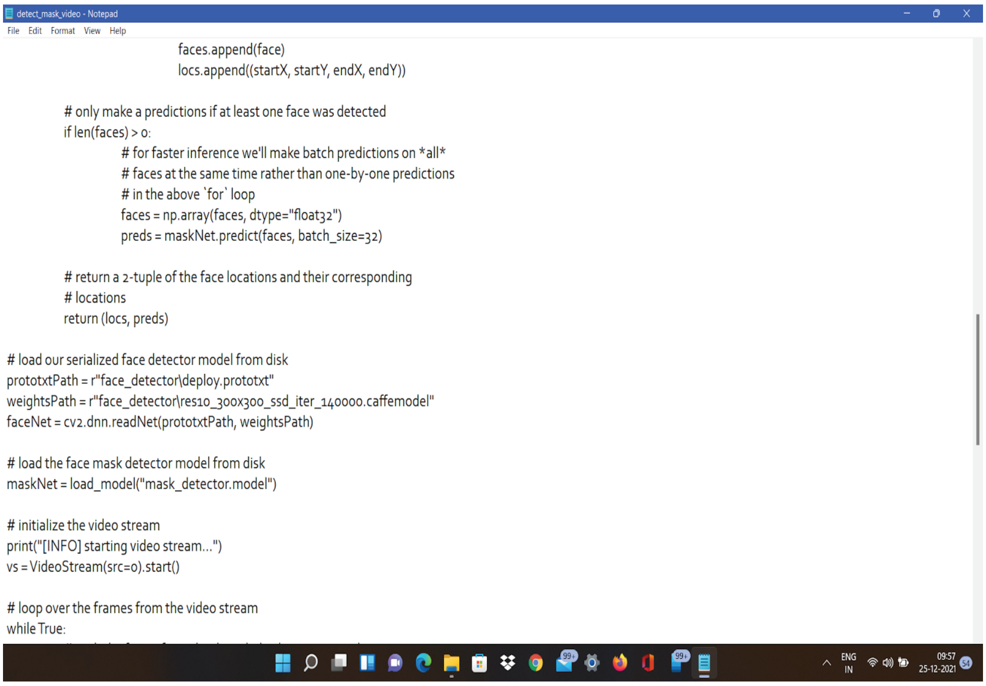


Fig.5. Output screen showing package details



```
for i in range(0, detections.shape[2]):
        # extract the confidence (i.e., probability) associated with
        # the detection
        confidence = detections[0, 0, i, 2]

        # filter out weak detections by ensuring the confidence is
        # greater than the minimum confidence
        if confidence > 0.5:
                # compute the (x, y)-coordinates of the bounding box for
                # the object
                box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                (startX, startY, endX, endY) = box.astype("int")

                # ensure the bounding boxes fall within the dimensions of
                # the frame
                (startX, startY) = (max(0, startX), max(0, startY))
                (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

                # extract the face ROI, convert it from BGR to RGB channel
                # ordering, resize it to 224x224, and preprocess it
                face = frame[startY:endY, startX:endX]
                face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
                face = cv2.resize(face, (224, 224))
                face = img_to_array(face)
                face = preprocess_input(face)

                # add the face and bounding boxes to their respective
                # lists
                faces.append(face)
```

Fig.6. Output screen showing code details



```
                faces.append(face)
                locs.append((startX, startY, endX, endY))

        # only make a predictions if at least one face was detected
        if len(faces) > 0:
                # for faster inference we'll make batch predictions on *all*
                # faces at the same time rather than one-by-one predictions
                # in the above `for` loop
                faces = np.array(faces, dtype="float32")
                preds = maskNet.predict(faces, batch_size=32)

        # return a 2-tuple of the face locations and their corresponding
        # locations
        return (locs, preds)

# load our serialized face detector model from disk
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk
maskNet = load_model("mask_detector.model")

# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
while True:
```

Fig.7. Output screen showing code details

```
detect_mask_video - Notepad
File  Edit  Format  View  Help
	# to have a maximum width of 400 pixels
	frame = vs.read()
	frame = imutils.resize(frame, width=400)

	# detect faces in the frame and determine if they are wearing a
	# face mask or not
	(locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

	# loop over the detected face locations and their corresponding
	# locations
	for (box, pred) in zip(locs, preds):
		# unpack the bounding box and predictions
		(startX, startY, endX, endY) = box
		(mask, withoutMask) = pred

		# determine the class label and color we'll use to draw
		# the bounding box and text
		label = "Mask" if mask > withoutMask else "No Mask"
		color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

		# include the probability in the label
		label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

		# display the label and bounding box rectangle on the output
		# frame
		cv2.putText(frame, label, (startX, startY - 10),
			cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
		cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
```

Fig.8. Output screen showing code details

```
detect_mask_video - Notepad
File  Edit  Format  View  Help
		# unpack the bounding box and predictions
		(startX, startY, endX, endY) = box
		(mask, withoutMask) = pred

		# determine the class label and color we'll use to draw
		# the bounding box and text
		label = "Mask" if mask > withoutMask else "No Mask"
		color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

		# include the probability in the label
		label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

		# display the label and bounding box rectangle on the output
		# frame
		cv2.putText(frame, label, (startX, startY - 10),
			cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
		cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

	# show the output frame
	cv2.imshow("Frame", frame)
	key = cv2.waitKey(1) & 0xFF

	# if the `q` key was pressed, break from the loop
	if key == ord("q"):
		break

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

Fig.9. Output screen showing code details

3. Result analysis

To maintain a balanced representation of the various classes, the dataset is divided into training and testing sets. The dataset consists of a total of 1315 samples, with 80% allocated for the training phase and 20% for testing. The developed architecture is trained for 10 epochs, as further training would lead to overfitting on the training data. Overfitting occurs when a model starts learning irrelevant patterns from the training samples, resulting in increased training accuracy but decreased test accuracy. You can observe the graphical representation of accuracy and loss in Fig. 2. The trained model achieved an accuracy of 95%, as depicted in Fig. 4. The methodology for face mask detection is illustrated in Fig. 4, while Fig. 5 to Fig. 9 provide details of the code implementation.

4. Conclusion

With the continuous advancement of technology and the emergence of new trends, there is a promising face mask detection system that holds the potential to significantly enhance public healthcare. This system has been meticulously trained on an extensive dataset and leverages various tools such as OpenCV, TensorFlow, Keras, and Convolutional Neural Networks (CNN) to determine whether individuals are wearing face masks. Through rigorous testing with both images and real-time video, the model has achieved a high level of accuracy. Ongoing optimization efforts involve fine-tuning hyperparameters to further enhance the model's performance. This particular model serves as a compelling use case for edge analytics and has the capacity to make a substantial contribution to society by detecting mask usage and granting entry accordingly.

In recent times, over fifty countries worldwide have mandated the use of face masks in public spaces, including settings such as supermarkets, public transportation, offices, and stores. Retail establishments often rely on software solutions to monitor the number of people entering their premises and assess the impact of digital displays and promotional screens. We have plans to enhance our Face Mask Detection tool and offer it as an open-source paper. Our software is adaptable and compatible with various camera types, including USB, IP, and CCTV cameras, enabling real-time detection of individuals without face masks. This live video feed detection can be seamlessly integrated into both web and desktop applications, enabling operators to receive real-time notifications. In cases where someone is not wearing a mask, the software can capture an image for future reference. Additionally, an alarm system can be implemented to alert the operator when someone without a mask enters the monitored area. The software also provides the option of connecting to entrance gates to ensure that only individuals wearing face masks are granted entry.

References

1. Sujath, R., Chatterjee, J.M. & Hassanien, A.E. A machine learning forecasting model for COVID-19 pandemic in India. Stoch Environ Res Risk Assess 34, 959–972 (2020). https://doi.org/10.1007/s00477-020-01827-8

2. Navares, Ricardo & Díaz, Julio & Linares, Cristina & Aznarte, José. (2018). Comparing ARIMA and computational intelligence methods to forecast daily hospital admissions due to circulatory and respiratory causes in Madrid. Stochastic Environmental Research and Risk Assessment. 32. 10.1007/s00477-018-1519-z.

3. Cui, Huijuan & Singh, Vijay. (2017). Application of minimum relative entropy theory for streamflow forecasting. Stochastic Environmental Research and Risk Assessment. 31. 10.1007/s00477-016-1306-7.

4. Mohammad Marufur Rahman, Md. Motaleb Hossen Manik and Md. Milon Islam, 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," Oct. 2020.

5. Toshan Meenpal, Ashutosh Balakrishnan and Amit Verma, 2019 4th International Conference on Computing, Communications and Security (ICCCS), "Facial Mask Detection using Semantic Segmentation," Oct. 2019.

6. Opitz, Michael, et al. "Grid loss: Detecting occluded faces." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. Springer International Publishing, 2016.