Face mask detection through convolutional neural network.

Sree Lakshmi Hiranmayee Kadali
Vishnu Kiran Dabbara

Computer Science Engineering
Blekinge Institute of Technology
Karlskrona
Sweden
2021

in cooperation with

Abstract

With shutting down of schools, lost jobs, and humanitarian crisis, Covid 19 has left a scar in everyone's life. With the need to elevate capitalism, health care, education, there is a fiscal need to get back to the new normal. Our real-time mask recognition system can help in detecting face masks in public places like schools, airports, hospitals, where there is a great need for measures like a face mask and social distancing [1]. The idea of a face mask detector uses the approach of Image recognition and AI which detects the person who is wearing face masks properly and warns if the face mask is half worn or not at all used. A subjective model can be done using Open-CV & Keras following a real-time application. The prime highlight of this idea is to monitor public spaces without the need for human supervision.[1]

While there are so many open-source models present in the market which uses object detection algorithm, it has so many pros and cons. Through research work, we have concluded convolutional neural network model has so many benefits compared to the traditional object detection algorithm.[2]

Our objective model is one such model which efficiently captures multiple people in real-time live applications. It ensures communal health without any human supervision.

We finally conclude that this subjective model works on the concept of deep learning through tensor flow with an architecture of CNN network. We can say this is a hybrid model of deep learning and machine learning model. Here we also researched various CNN networks to consider and conclude the best CNN network for best efficiency[3]. The resulting model showed almost 99.2% efficiency in real-time application.[4]

Keywords:

Convolutional Neural Networks, Deep learning, Machine learning

Acknowledgements

We are grateful to our mentor Prof. Wlodek J. Kulesza for helping us for the preparation of this project. His teaching and support helped us submit our work on time This assignment could not be completed without the help of our parents. They sincerely believed in our work and helped us to give ourselves time for the making of this thesis.

We would like to thank every person who stayed as a source of inspiration to us.

Sree lakshmi Hiranmayee Kadali
Vishnu Kiran Dabbara

Contents

List of figures

List of tables

List of acronyms

Acronym	**Unfolding**
CNN	Convolutional Neural Network
AI	Artificial Intelligence
ML	Machine learning
IOT	Internet of things

1 Chapter: Introduction

The need for face masks hugely rose with the Covid 19 epidemic all around the world. Before the covid 19 pandemic, masks are most generally used by doctors and very few people who protect their health from air pollution. The scenario went a great turn with the covid 19 epidemic virus. World health organization (WHO) in their latest 209th report declared that the SARS COV2(coronavirus) has almost affected 6 billion people around the world.[5] The rapid spreading of COVID-19 has enforced WHO to declare it as a global pandemic. Scientists proofed that wearing face masks helps on impending the spread of this deadly virus. The virus spreads through close contact in crowded areas. [6]

With the situation getting out of control, WHO suggested a few preventive measures to reduce the spread of the virus. The absence of antiviral medication and limited medical resources, lead to following measures like using face masks and social distancing. However, in highly populated countries like India, it is quite difficult to monitor people in large groups. The monitoring process to detect a person is wearing a face mask or not is an exhausting job. This is the main motivation for our project. The main aim of this project is to provide the best solution for monitoring people without the help of human supervision.[6]

Science and technology have given rise to much extraordinary application in the brim of the coronavirus epidemic. Many branches like AI, ML, IOT helped in fighting covid 19 in many ways.ML has helped researchers and clinicians to evaluate vast quantities of covid 19 data for predicting warning mechanisms for vulnerable age groups. IoT has been helpful in many real-time applications like automatic hand sanitization. The provision of healthcare needs funding from all the new emerging technologies for a better life. To mandate the use of face masks, it is important to devise some technique that helps in monitoring people in crowded areas with face masks.[7]

Here we introduce a real-time mask face detection model that is based on machine learning and deep learning. The proposed model can be integrated with surveillance cameras in crowded areas like schools, malls, hospitals. This helps by allowing the detection of people who are wearing masks or not wearing face masks. The model is an integration between deep transfer learning and machine learning techniques with OpenCV, tensor flow, and Keras. The main backbone of the architecture is the CNN model. We introduced a comparison between different CNN models to find the most suitable network that achieved the highest accuracy and consumed the least time in the process of training and object detection.[2]

2 Chapter: Survey of related work

When it comes to face mask detection, various researchers and analysts mainly focused on Grayscale face images. While some identification models focused on building a pattern, some others used AdaBoost, which was an excellent classifier for training purposes. Then came the viola-jones detector which was a breakthrough in face mask detection and real-time face detection got possible. Although numerous researchers have worked hard in finding the algorithms for face mask detection, there exists a real difference between “detection of face mask” and “detecting face under the mask[8]

Our research worked to detect masks over the face. Many contradictions around the object detection algorithm have led us to choose CNN model architecture. While every open-source model, uses an object detection algorithm, it results in low accuracy and poor real-time performance in application. The traditional object detection algorithm focuses on many parameters like uneven face scale, illumination, face density, etc. so basically, it fails to work in dim and dull light. Thus, we started searching for a new alternative model which could detect face masks as well as helps in detecting the facial features. This eliminates almost all the problems encountered with traditional object detection as well as machine learning approaches.[8]

In recent years, face mask detection models based on CNN have been widely developed and brought into use. In Convolutional Neural Network-based classification, face mask detector models train directly from the user's input dataset. This is then applied using deep transfer learning. Cascade CNN, came up in the year 2007 which helps in features aggregation of faces in the face detection model. In further research works, it was found out that the ROI method is used for fine-tuning the image dataset[9]

The CNN model uses Annotations are provided for present faces in these datasets. the datasets are divided into “dataset with mask” and “dataset without facemask”. The faces compared to earlier ones. Large datasets are needed for making better training and testing data to perform real-world applications in a much simpler way. This calls for various deep learning algorithms which can read faces and mask straight from the data provided by the user. Extensive research on the proposed backbone of the object with three popular pre-trained models has been done. The three models namely resnet50, Mobile net, Alexnet are worked on and the accuracy has been compared. Resnet50 has been found optimal choice for real-time application with almost 98% accuracy.[9]

Chapter:
Problem statement, objectives and main contribution

This project is aimed to focus on the real-time application of face mask detection in public spaces, to enforce everyone to wear a mask, To keep communities safe by wearing the mask in public spaces with the help of a face mask detector system which can be embedded in public places like malls, hospitals, schools.

OBJECTIVES:

• To enforce the mandate for wearing masks in public places following the Covid-19 pandemic.
• To effectively provide a working model for accurate mask detection
• To utilize image processing approaches to identify the presence of a mask on the face.
• To develop an efficient computer vision-based system focused on the real-time automated monitoring of people to detect face masks in public places.
• The main highlight of the project is monitoring people without human supervision.

MAIN CONTRIBUTION

The main objective of this project is to take care of human health and control the spread of deadly viruses. The sudden attack of COVID-19 has taught us to protect ourselves and others by following measures like wearing a mask and sanitizing our hands each now and then. To avoid its spread mask is the most important preventive measure. So, to prevent people from skipping wearing masks this project has come to the light. To avoid manual detection and for better accuracy, this project has been introduced. This enables the people to remind them of wearing their masks and allows only when they wear their masks.

Implementation of this model in the real world will be very fruitful. There are several contributions this model can provide, e.g., a very fast and accurate COVID-19 face mask detection with lower cost, less human intervention, Automated face mask detection

3 Chapter: Solution

Our project follows different phases for building a real-time face mask detector model with machine learning and deep learning using Python, OpenCV, and TensorFlow/Keras.

3.1 Modeling

The proposed model focuses on identifying face masks on every individual person through an image or video stream with the help of Python, Keras, TensorFlow, OpenCV, and the sci-kit learn library. we have developed our model in two phases:

This proposed model focuses on identifying face masks in a person through an image or video stream with the help of Deep Learning and Machine Learning using Keras, TensorFlow, OpenCV, and the Scikit-Learn library.

We have designed our model in two phases:
1.Training (Training the model using TensorFlow & Keras): Here we will load our data set which includes two types "with mask" and "without a mask". the training dataset then is stored in the memory disk for testing.
2.Deployment (Loading the trained model and applying detector over images/live video stream): In this phase, the device is customized according to the pre-trained model. we then split the datasets into testing and training datasets and evaluate them accordingly.
The deployment phase focuses on the data augmentation methods like cropping, rotation, shearing, etc to deploy the trained dataset. This is widely is used in large neural networks.
The next step is to compile the subjective model and train it on the augmented data.

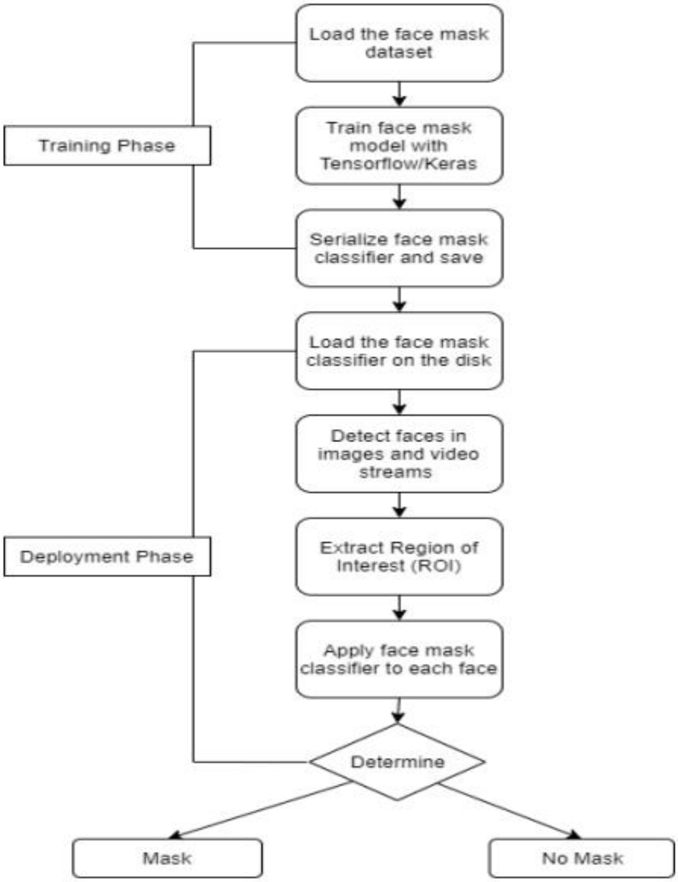


Figure 4-1. Flow diagram of creating an application and back end

In this proposed model, we see many concepts involved:

face mask detection is used achieved through a deep neural network called RESNET50 which is used for various categorical classification and object detection work. this CNN model architecture is easily executable in real-time surveillance cameras with less error rate, high accuracy, and less memory consumption.

we used ResNet 50 compared to two other different models, making it easy to embed in systems like raspberry pi, google Corel, nano, etc.

we used deep learning methodology, along with transfer learning. TensorFlow is a standalone open-source software library that is used for data flow and numerous computational tasks. TensorFlow greatly helps in machine learning applications.

Keras and TensorFlow are used for transfer learning in our project which automatically identifies face masks on a person.

OpenCV majorly focuses on feature analysis including features like image processing, video capture, and object detection, and face detection. it is widely used in real-time applications and detecting face masks on the face or facial features. we have used cascade classifications to execute infinite loops to detect faces through face masks.

Keras is an API for high-level neural networking. It follows strategies to develop APIs which are flexible and consistent. Keras reduces the major burden to the developer and provides clear and actionable error messages.

.

3.2 Implementation or Application

We have taken a total of 1365 images in our Face Mask Detection Dataset belonging to two labels i.e., with mask: 660 images and without mask:705 images.

WITHMASK:

Figure 4-2. Images in dataset with face mask

WITHOUT MASK:

Figure 4-3. Images in dataset without face mask

Training of the Dataset:

Training the dataset is the important step for making the model. Here generally we use convolutional networks, but our idea is to neglect CNN we use for image processing and introduce RESNET50 essentially what we do is usually after the image is processed as an array, we will send that into Mobile Net and then do the maximum pooling
Training report is about the loss, val_loss, accuracy, and val_accuracy. Depending on the activation function these parameters may have a chance of variation. For example, when we use the SoftMax activation function, val_loss and val_accuracy will start increasing simultaneously.
The validation set is used to evaluate the trained model. So, if the accuracy is increasing and val_accuracy is falling along with the increase in epoch, it means that the model is overfitting.
In the same way, the validation loss and loss are the losses of the validation data and the training data.[5]

4.4 MODEL COMPARISION:

We have compared almost three models during testing to find out the best accuracy. The top 1 error is presented graphically with the comparison of three models namely, Resnet50, Mobile net, Alexnet. The error rate in Alexnet is higher when compared to Resnet 50.
Now comparing the models depending on the inference time, all the models are given test images and inference time for detecting the face mask has been calculated. The mobile net takes more time to infer images, while Alexnet and Resnet take equal time to inference images.
Lastly comparing the memory usage of the models. The parameters can be obtained by generating a model summary with google collab for each model. the parameters obtained for Alexnet around the world in 28 million while Resnet and mobile have 3.5 and 25 million parameters respectively.

This finally concluded that Resnet 50 has the best accuracy, with less memory usage and less error rate. This is a boon for real-time surveillance applications.

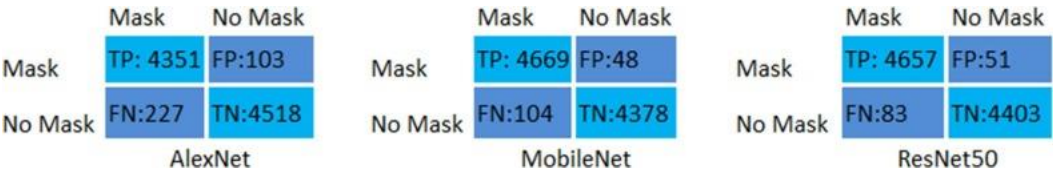


Figure 4-4. comparision within CNN models

4.5 Evaluation Table for trained model:

Table 3-1 evaluation table for resnet50 CNN model

	Precision	**Recall**	**F1-score**	**Support**
With mask	100	83	90	383
Without mask	85	100	92	384
Accuracy	99	98	91	767
Macro avg	92	91	91	767

As we can see the accuracy obtained by our trained face mask detector model is ~98%.

After training the model, the accuracy and the training loss with the epoch are plotted and saved in the directory mentioned. The model is saved in two forms one as the model file other as the .png file.

The below are the results of the training data:

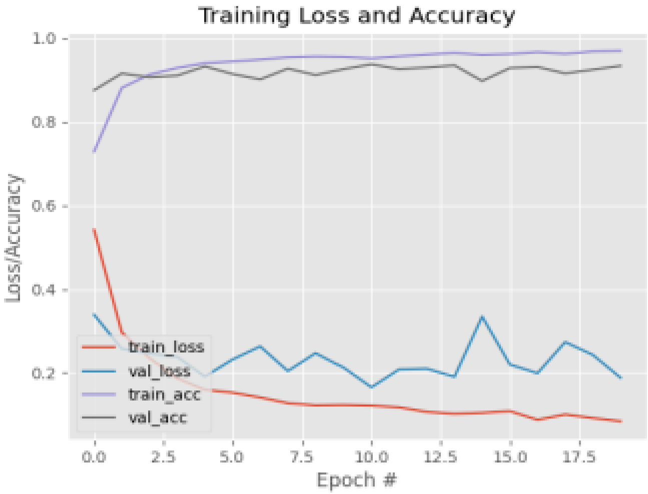


Figure 4-5. Accuracy of Resnet CNN model

Test outputs:

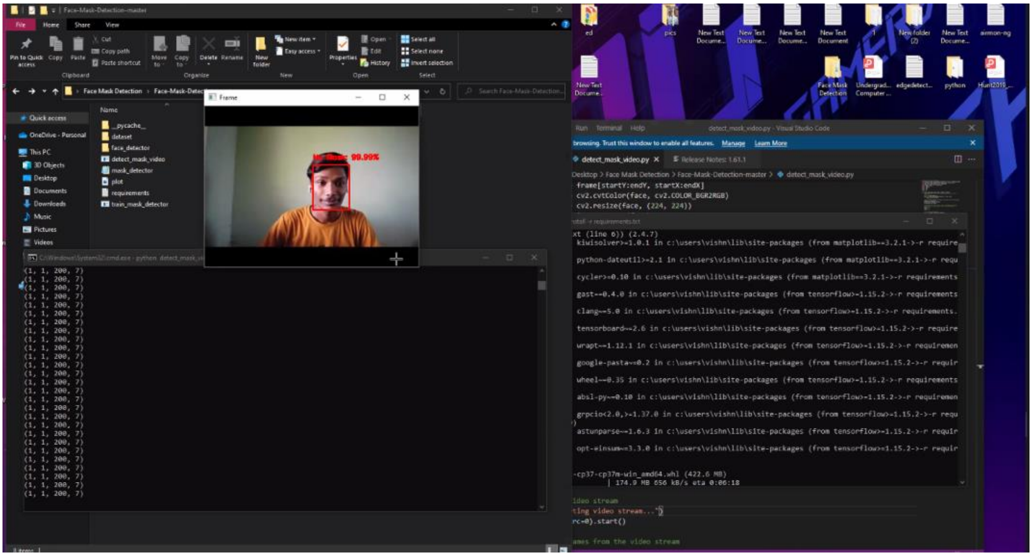


Figure 4-6. test output without mask

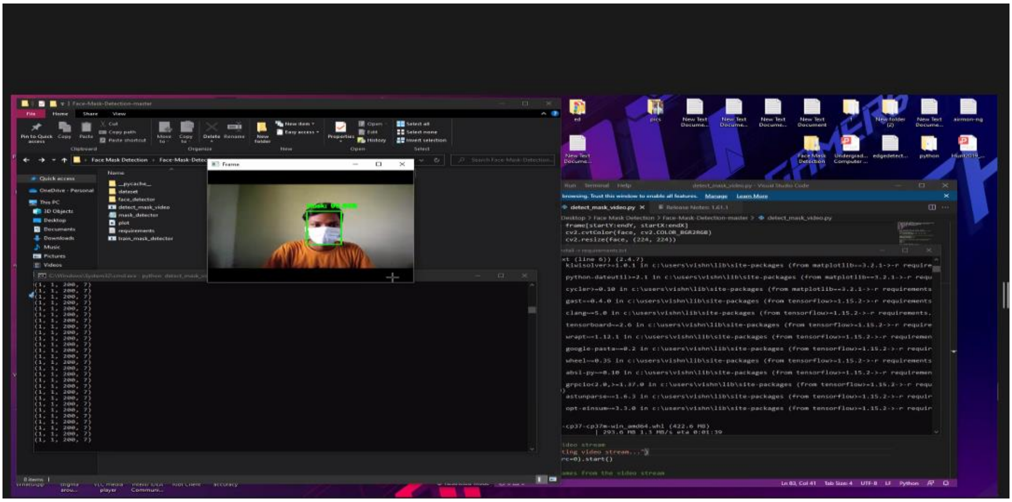


Figure 4-7. test output with mask

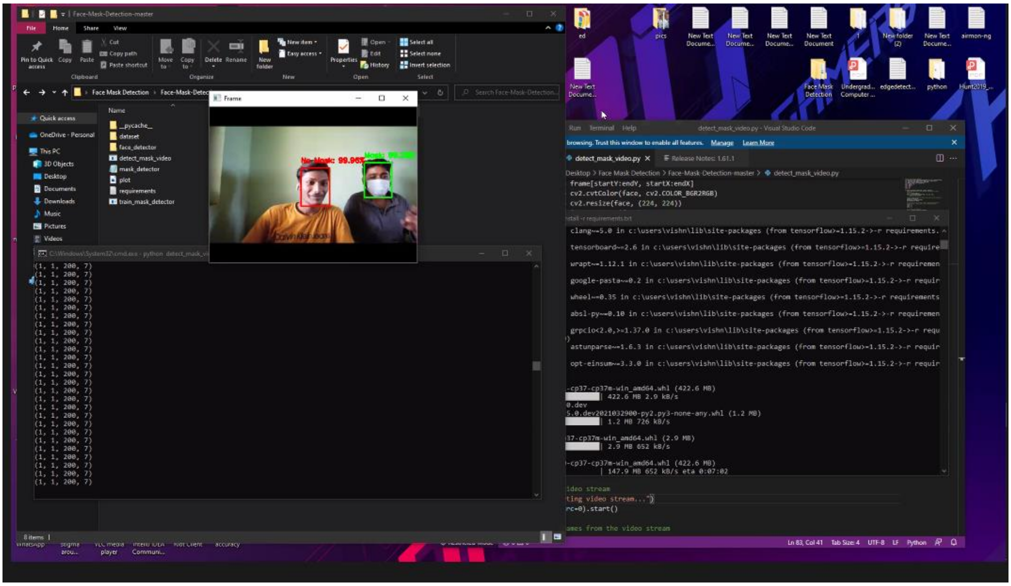


Figure 4-8. test output with and without mask

3.3 Validation or Verification

$$\text{Accuracy}=\frac{t_p+t_n}{(t_p+f_p+f_n+t_n)} \quad (3.1)$$

$$\text{Precision}=\frac{t_p}{(t_p+f_n)} \quad (3.2)$$

$$F1\ scores = \frac{2*(recall*precision)}{(recall+precision)} \quad (3.3)$$

$$\text{Recall}= (t_p)/((t_p + f_n)\ ) \ (3.4)$$

```
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       1.00      0.83      0.90       383
without_mask       0.85      1.00      0.92       384

    accuracy                           0.91       767
   macro avg       0.92      0.91      0.91       767
weighted avg       0.92      0.91      0.91       767

[INFO] saving mask detector model...
```

Figure 4-9. Accuracy of the cnn model

Where, t_p= true positive,

t_n=true negative,

f_p=false positive,

f_n=false negative

In true positive value means the values of images which were labelled true after prediction by model. True negative means where the result is true but after the prediction it shows false. False positives refer to the value of images which has false result before and after prediction. False negative refers to images which are first considered false but after prediction becomes true. The accuracy and precision of our model has been good to start with. Precision gave positive predicted values.
F1 Score: F1 scores are less than accurate measures as they are included in precision and recall values.

3.4 Chapter:

Conclusion and future work

In this model, we have developed a deep learning face detection model which helps in curtailing the spread of covid 19.
the model can be used surveillance cameras in crowded areas like malls, hospitals, schools, etc. the technique proposed handles different occlusions and dense situations. the approach not only helps in high accuracy but detects the mask with less inference time. furthermore, the subjective model uses deep transfer learning methodology which trains the data set with a huge number of face images and tests in real-time application. the resnet50 and ROI method helps in affine transformation is used along with OpenCV to detect the facial features also.
finally, the project model efficiently detects the mask on the face as well as the face under the mask. the accuracy of the model is almost 98-99% with great computational speed when in use.

The problem of wrong predictions has been removed successfully from the model because we train the model with various images in all kinds of variations and parameters which might affect the accuracy.

This face mask detector can be deployed in many areas like shopping malls, airports, and other heavy traffic places to monitor the public and to avoid the spread of the disease by checking who is following basic rules and who is not.

Finally, the face mask detector opens interesting future directions for researchers. This mask detection can be extended to detect facial landmarks for face mask detection purposes.

References

[1] M. Ciotti, M. Ciccozzi, A. Terrinoni, W.-C. Jiang, C.-B. Wang, and S. Bernardini, “The COVID-19 pandemic,” *Crit. Rev. Clin. Lab. Sci.*, vol. 57, no. 6, pp. 365–388, Aug. 2020, doi: 10.1080/10408363.2020.1783198.

[2] G. M. K, “COVID-19: Face Mask Detection using TensorFlow and OpenCV,” *Medium*, Jun. 09, 2020. https://towardsdatascience.com/covid-19-face-mask-detection-using-tensorflow-and-opencv-702dd833515b (accessed Nov. 08, 2021).

[3] “Survey of convolutional neural network--《Journal of Computer Applications》2016年09期.” https://en.cnki.com.cn/Article_en/CJFDTotal-JSJY201609029.htm (accessed Nov. 08, 2021).

[4] B. Batagelj, P. Peer, V. Štruc, and S. Dobrišek, “How to Correctly Detect Face-Masks for COVID-19 from Visual Information?,” *Appl. Sci.*, vol. 11, no. 5, Art. no. 5, Jan. 2021, doi: 10.3390/app11052070.

[5] S. E. Eikenberry *et al.*, “To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic,” *Infect. Dis. Model.*, vol. 5, pp. 293–308, Jan. 2020, doi: 10.1016/j.idm.2020.04.001.

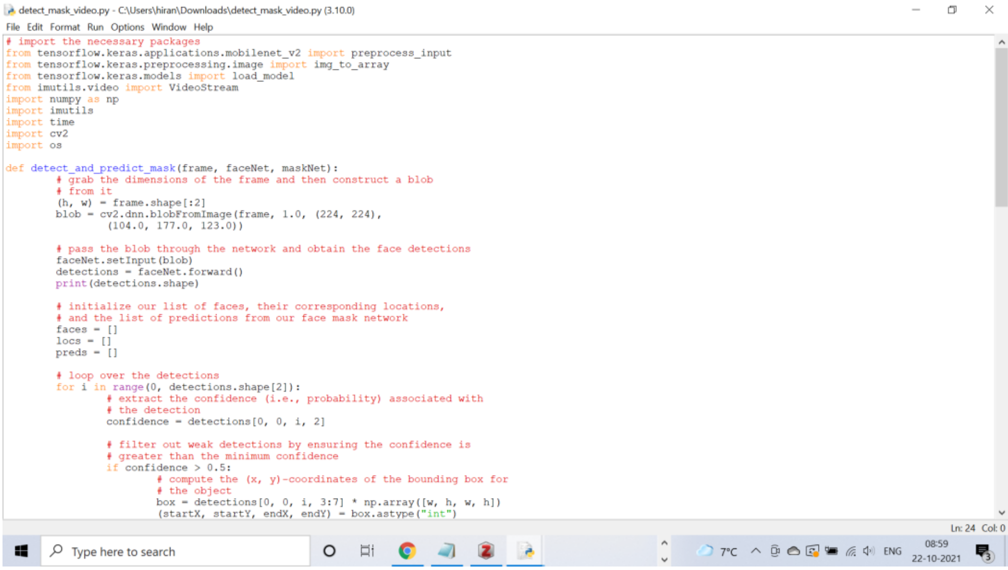
[6] L. R. G. Godoy *et al.*, “Facial protection for healthcare workers during pandemics: a scoping review,” *BMJ Glob. Health*, vol. 5, no. 5, p. e002553, May 2020, doi: 10.1136/bmjgh-2020-002553.

[7] “Robust Deep Learning Method to Detect Face Masks | Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture.” https://dl-acm-org.miman.bib.bth.se/doi/abs/10.1145/3421766.3421768 (accessed Nov. 08, 2021).

[8] 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*, vol. 167, p. 108288, Jan. 2021, doi: 10.1016/j.measurement.2020.108288.

[9] S. Sethi, M. Kathuria, and T. Kaushik, “Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread,” *J. Biomed. Inform.*, vol. 120, p. 103848, Aug. 2021, doi: 10.1016/j.jbi.2021.103848.

Appendix 1:
Code Snippet of the project



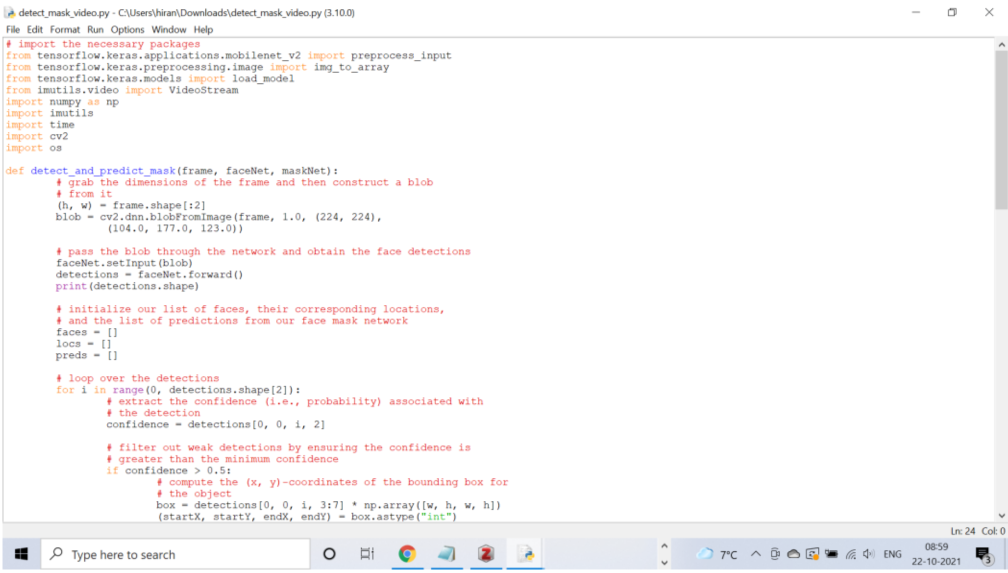
detect_mask_video.py - C:\Users\hiran\Downloads\detect_mask_video.py (3.10.0)

File Edit Format Run Options Window Help

```python
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
import os

def detect_and_predict_mask(frame, faceNet, maskNet):
	# grab the dimensions of the frame and then construct a blob
	# from it
	(h, w) = frame.shape[:2]
	blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
		(104.0, 177.0, 123.0))

	# pass the blob through the network and obtain the face detections
	faceNet.setInput(blob)
	detections = faceNet.forward()
	print(detections.shape)

	# initialize our list of faces, their corresponding locations,
	# and the list of predictions from our face mask network
	faces = []
	locs = []
	preds = []

	# loop over the detections
	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")
```

Ln: 24 Col: 0

Figure 5-1. code snippet 1



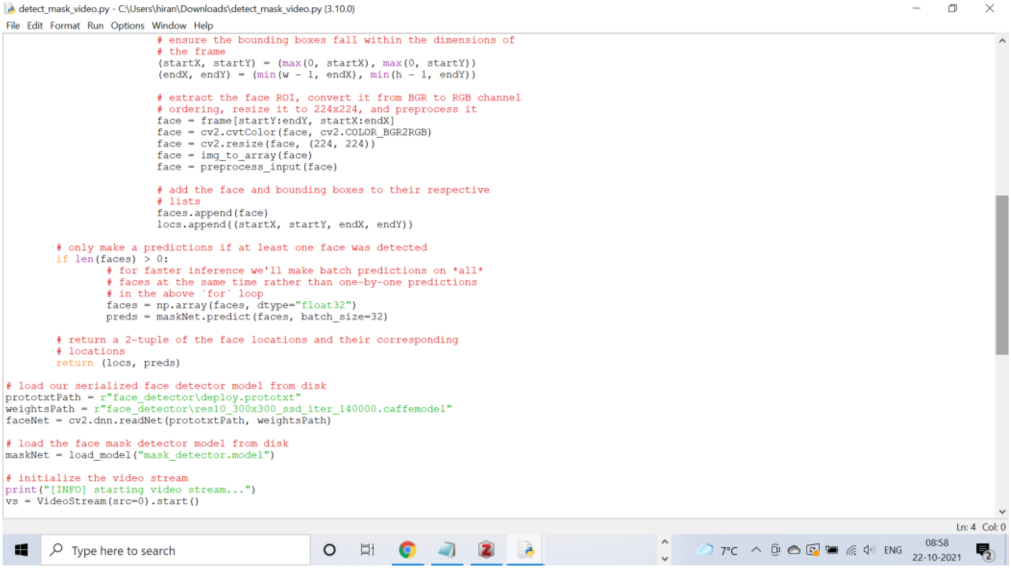
detect_mask_video.py - C:\Users\hiran\Downloads\detect_mask_video.py (3.10.0)

File Edit Format Run Options Window Help

```python
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
import os

def detect_and_predict_mask(frame, faceNet, maskNet):
	# grab the dimensions of the frame and then construct a blob
	# from it
	(h, w) = frame.shape[:2]
	blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
		(104.0, 177.0, 123.0))

	# pass the blob through the network and obtain the face detections
	faceNet.setInput(blob)
	detections = faceNet.forward()
	print(detections.shape)

	# initialize our list of faces, their corresponding locations,
	# and the list of predictions from our face mask network
	faces = []
	locs = []
	preds = []

	# loop over the detections
	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")
```

Ln: 24 Col: 0

Figure 5-2.code snippet 2



detect_mask_video.py - C:\Users\hiran\Downloads\detect_mask_video.py (3.10.0)

File Edit Format Run Options Window Help

```
				# 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)
				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()
```

Ln: 4 Col: 0

Type here to search

7°C ENG 08:58 22-10-2021

Figure 5-3. code snippet 3