

**Department of Artificial Intelligence & Machine Learning**

**Academic Year 2021-22(Odd)**

Mini Project Report

On

**“ Face mask detection ”**

**By**

**1NH20AI086** - Revuru Venkata Sesha Sai Jaswanth

**1NH20AI097** - Shripad Kathare

**1NH20CV016** - Aritroo Kumar Chowdhury

**Under the Guidance of**

Mr. Dinesh J

Assistant professor,

Dept. of Artificial Intelligence & Machine Learning,

New Horizon College of Engineering,

Bangalore-560103



**Department of Artificial Intelligence & Machine Learning**

CERTIFICATE

Certified that the Mini Project work entitled **“Face mask detection”** carried out by Revuru Venkata Sesha Sai Jaswanth(1NH20AI086) , Shripad Kathare (1NH20AI097) , Aritroo Kumar Chowdhury (1NH20CV016) . It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work.

**Prof. Dr. N V Uma Reddy**

Guide Head of Department

External Viva

Examiner Signature with date:

1.

2.

**ABSTRACT**

COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not.

After the new Coronavirus disease (COVID-19) case spread rapidly in Wuhan-China in December 2019, World Health Organization (WHO) confirmed that this is a dangerous virus which can be spreading from humans to humans through droplets and airborne. As for the prevention, wearing a face mask is essentials while going outside or meeting to others. However, some irresponsible people refuse to wear face mask with so many excuses. Moreover, developing the face mask detector is very crucial in this case. This paper aims to develop the face mask detector which is able to detect any kinds of face mask. In order to detect the face mask, a YOLO V4 deep learning has been chosen as the mask detection algorithm. The experimental results have been done in real-time application and the device has been installed at Politeknik Negeri Batam. From the experimental results, this device is able to detect the people who wear or do not wear the face mask accurately even if they are moving to various position.

**Table of Contents**

**Chapter Page No.**

1. **INTRODUCTION**  1
   1. Introduction
   2. Objectives
   3. Literature Survey
   4. Existing system
   5. Proposed system
2. **SYSTEM REQUIREMENTS**
   1. Hardware requirements
   2. Software requirements
3. **SYSTEM DESIGN**
   1. System architecture
   2. Algorithms/ Flow charts
4. **IMPLEMENTATION AND RESULTS**
   1. Psuedocode
   2. results
5. **CONCLUSION AND FUTURE ENHANCEMENT**

1.2

**CHAPTER-1 INTRODUCTION**

* 1. **Introduction**

According to the World Health Organization (WHO)’s official Situation Report – 205, coronavirus disease 2019 (COVID-19) has globally infected over 160 million people causing over 50 million deaths. Individuals with COVID-19 have had a wide scope of symptoms reported going from mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk. Some common human coronaviruses that infect public around the world are 229E, HKU1, OC43, and NL63. Before debilitating individuals, viruses like 2019-nCoV, SARS-CoV, and MERS-CoV infect animals and evolve to human coronaviruses. Persons having respiratory problems can expose anyone (who is in close contact with them) to infective beads. Surroundings of a tainted individual can cause contact transmission as droplets carrying virus may withal arrive on his adjacent surfaces.

To curb certain respiratory viral ailments, including COVID-19, wearing a clinical mask is very necessary. The public should be aware of whether to put on the mask for source control or aversion of COVID-19. Potential points of interest of the utilization of masks lie in reducing vulnerability of risk from a noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restraint the spread of virus. WHO stresses on prioritizing medical masks and respirators for health care assistants. Therefore, face mask detection has become a crucial task in present global society.

Face mask detection involves in detecting the location of the face and then determining whether it has a mask on it or not. The issue is proximately cognate to general object detection to detect the classes of objects. Face identification categorically deals with distinguishing a specific group of entities i.e. Face. It has numerous applications, such as autonomous driving, education, surveillance, and so on. This paper presents a simplified approach to serve the above purpose using the basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV and Scikit-Learn.

* 1. **Objectives**
* To Identify people who had covered the faces with mask properly.
* To stop the spreading of COVID-19 disease.
* To make sure people are following correct rules and precautions of COVID-19.
* To make life easier with the help of computer programming.
  1. **Literature Survey**
* Facial landmarks allow us to automatically infer the location of facial structures, including:  
  Eyes , Eyebrows , Nose , Mouth , Jawline  
  To use facial landmarks to build a dataset of faces wearing face masks, we need to first start with an image of a person *not* wearing a face mask
* From there, we apply face detection to compute the bounding box location of the face in the image
* Next, we need an image of a mask (with a transparent background) such as the one below
* This mask will be *automatically* applied to the face by using the facial landmarks (namely the points along the chin and nose) to compute *where* the mask will be placed
* The mask is then resized and rotated, placing it on the face
  1. **Existing system**

Some researchers have used the extraction of RGB color information to perform face mask recognition [27]. However, the article does not consider the case of non-standard wearing of masks, so the adaptability of the algorithm needs to be further improved. Combining YOLO-v2 and ResNet50, the authors in [28] realized face mask recognition whose backbone network is DarkNet-19. However, DarkNet-19 has been optimized by CSPDarkNet53. The ablation experiment in our paper shows that the CSP1\_X module produces better results than CSPDarkNet53. In [29], the authors pointed out that the combination of ResNet50 and SVM can realize face mask detection and its accuracy can reach up to 99.64%. However, the algorithm takes a lot of computational costs. Furthermore, the combination of SSD and MobileNetV2 for mask detection was proposed in paper [30], but its model structure is too complex and its performance is inferior to YOLO-v4. Only two categories are used in the papers mentioned in the above paragraph and the authors did not consider the influence of wearing masks irregularly on the algorithm. Therefore, the feature extraction ability and model practicability of these algorithms need to be improved. In this paper, based on improved YOLO-v4, face mask recognition is considered and three categories, face\_mask, face and WMI, are included. In addition, the feature extraction ability of this paper is improved by CSP1\_X, and CSP2\_X impels PANet to speed up the circulation of semantic features and strengthen feature fusion, thus improving the robustness of the model.

* 1. **Proposed work**

We propose a two-stage architecture for detecting masked and unmasked faces and localizing them.



**CHAPTER-2 SYSTEM REQUIREMENTS**

**2.1. Hardware requirements**

* Working WebCam
* 4 GB RAM and above
* 64-bit processor
* min I5 processor and adove

**2.2. Software requirements**

* python IDE
* **library used :** OpenCV , Tensorflow , MobileNet , Keras , Scipy ,

Imutis

**CHAPTER-3 SYSTEM DESIGN**

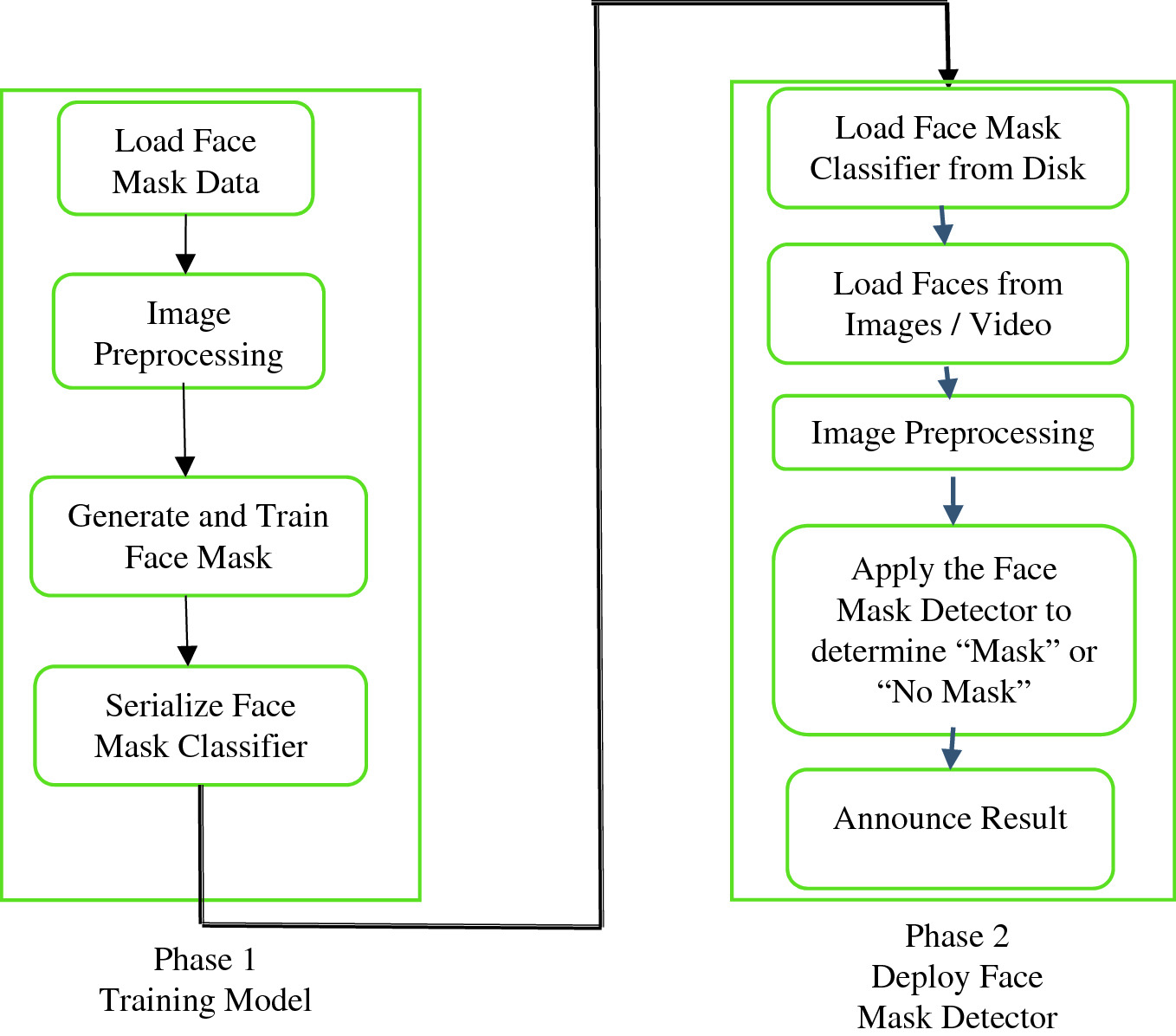
**3.1 System architecture**

It represents our architecture (input image taken from the dataset by (Larxel, 2020)). It consists of two major stages. The first stage of our architecture includes a Face Detector, which localizes multiple faces in images of varying

sizes and detects faces even in overlapping scenarios. The detected faces (regions of interest) extracted from this stage are then batched together and passed to the second

stage of our architecture, which is a CNN based Face Mask Classifier. The results from the second stage are decoded and the final output is the image with all the faces in the image correctly detected and classified as either masked or unmasked faces.

**3.2 Algorithms/ Flow charts**



**CHAPTER-4 IMPLEMENTATION AND RESULTS**

4.1 **Psuedocode**

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):

    (h, w) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),

        (104.0, 177.0, 123.0))

    faceNet.setInput(blob)

    detections = faceNet.forward()

    print(detections.shape)

    faces = []

    locs = []

    preds = []

    for i in range(0, detections.shape[2]):

        confidence = detections[0, 0, i, 2]

      if confidence > 0.5:

        box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])

            (startX, startY, endX, endY) = box.astype("int")

            (startX, startY) = (max(0, startX), max(0, startY))

            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

            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)

            faces.append(face)

            locs.append((startX, startY, endX, endY))

    if len(faces) > 0:

        faces = np.array(faces, dtype="float32")

        preds = maskNet.predict(faces, batch\_size=32)

    return (locs, preds)

prototxtPath = r"face\_detector\deploy.prototxt"

weightsPath = r"face\_detector\res10\_300x300\_ssd\_iter\_140000.caffemodel"

faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

maskNet = load\_model("mask\_detector.model")

print("[INFO] starting video stream...")

vs = VideoStream(src=0).start()

while True:

    frame = vs.read()

    frame = imutils.resize(frame, width=400)

    (locs, preds) = detect\_and\_predict\_mask(frame, faceNet, maskNet)

    for (box, pred) in zip(locs, preds):

        (startX, startY, endX, endY) = box

        (mask, withoutMask) = pred

        label = "Mask" if mask > withoutMask else "No Mask"

        color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) \* 100)

        cv2.putText(frame, label, (startX, startY - 10),

            cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)

        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

    cv2.imshow("Frame", frame)

    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):

        break

cv2.destroyAllWindows()

vs.stop()

4.1 results

**CHAPTER-5 CONCLUSION AND FUTURE ENHANCEMENT**

* 1. **Conclusion**

In this paper, a two-stage Face Mask Detector was presented. The first stage uses a pretrained retinaFace model for robust face detection, after comparing its performance with Dlib and MTCNN. An unbiased dataset of masked and unmasked faces was created. The second stage involved training three different lightweight Face Mask Classifier models on the created dataset and based on performance, the NASNetMobile based model was selected for classifying faces as masked or non-masked. Furthermore, Centroid Tracking was added to our algorithm, which helped improve its performance on videom streams. In times of the COVID-19 pandemic, with the world looking to return to normalcy and people resuming in-person work, this system can be easily deployed for automated monitoring of the use of face masks at workplaces, which will help make them safer.

* 1. **Future Enhancement**

There are a number of aspects we plan to work on shortly:

* Currently, the model gives 5 FPS inference speed on a CPU. In the future, we plan to improve this up to 15 FPS, making our solution deployable for CCTV cameras,without the need of a GPU.
* The use of Machine Learning in the field of mobile deployment is rising rapidly. Hence, we plan to port our models to their respective TensorFlow Lite versions.
* Our architecture can be made compatible with TensorFlow RunTime (TFRT), which will increase the inference performance on edge devices and make our models efficient on multithreading CPUs.
* Stage 1 and Stage 2 models can be easily replaced with improved models in the future, that would give better accuracy and lower latency.

**REFERENCES**

1. Alberca, G.G.F.; Fernandes, I.G.; Sato, M.N.; Alberca, R.W. What Is COVID-19? Front. Young Minds 2020, 8, 74. [CrossRef]
2. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 1998, 86, 2278–2324. [CrossRef]
3. Platt, J. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances in Kernel Methods. Support Vector Learn. 1998, 208.
4. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. Neural Inf. Process. Syst. 2012, 25. [CrossRef]
5. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. Comput. ENCE 2012, 3, 212–223.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 37, 1904–1916. [CrossRef] [PubMed]
7. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767
8. Bochkovskiy, A.; Wang, C.-Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]