ABSTRACT

In December 2019, a new Corona Virus disease(COVID-19) cases spread swiftly in Wuhan, China. The World Health Organization (WHO) confirmed that it is a deadly virus that can be transmitted fr om person to person by droplets and air contamination. As of now, wearing a face mask is very nec essary for preventing infection. People should wear a face mask when going outside to meet others and keep a social distance to prevent the transmission of COVID-19. However, some people don't follow the instruction properly while visiting in public places like Airports, Shopping Malls, Stations, etc and it is not possible for humans to monitoring each an every people to see wheather they were wearing mask or not. Implementing and developing such models which can be used for detecting face mask in real time applications like CCTVs, Scanning Screens and many other appliancesis just a major ways to identify and solve such kind of social problems and it is also a minor contribution towards the prevention for spreading this pandemic which was going on in our day to day life.

This mini project aims to develop the face mask detector using Convolutional Neural Network and deep learning. For testing and training and testing we are using dataset of images which are containing both images like with mask and without mask and developed a model which can be used a real time face mask detector.

From the given dataset we are getting an good experimental results which can be used for face mask detection even people are moving to various positions.

Table Of Contents

CHAPTER - 1 INTRODUCTION

1.1. Introduction

The end of the year 2019 was not unfortunate for people at the end of 2019 the highly contagious disease (COVID-19) was first reported in Wuhan, and it has become a major health problem for people in China and abroad. the earth. The epidemic has had a devastating effect on communities and economies around the world, causing global health problems. It is a serious respiratory infection caused by Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV-2). There is a major evolution of Corona Virus and it is attacking different countries at different times and it is causing a huge loss to the global economy which can no longer be found in the next few years. Globally especially the third wave of COVID-19 (a variant of Omicron) has created a major health challenge worldwide. Many closures in various industries were created during the COVID-19 era and moreover many sectors were created and it was difficult to fix their projects and infrastructure construction was halted due to this epidemic and we still have no final solution. .

At present, the virus has the potential to spread rapidly and spread to various parts of the world. The WHO final figures show that 152,543452 cases were confirmed and 3198528 deaths as a result of the epidemic. According to the Centers for Disease Control Prevention (CDC) the infection with coronavirus is spread mainly through the respiratory tract produced when people breathe, talk, cough or sneeze with infection.

Therefore to prevent the rapid spread of infection with COVID-19 many solutions, such as closure and closure, are proposed by most international governments. However this ineffectiveness of COVID-19 management can also be assessed by game-theory scenarios beyond the public property game. In particular, some researchers have focused on the government's skepticism about setting serious but necessary measures to contain viruses (eg: Stay Home and Locks), as well as introducing masks when people go out of their essential needs. . The government is strictly following COVID measures that can reduce the spread of the virus. Some governments may also hope that the insecurity of the herd in retrieval and vaccination will allow them to avoid imposing these completely unusual measures and will lead to the end of the epidemic.

Many fields and domains were developed by COVID-19 but nevertheless, it has created a path for researchers in the field of Artificial Intelligence and Mechanization. We have seen many research topics, such as the creation of new ways to automatically detect COVID-19 measures and to detect masked and uninfected people. Considering that there were some errors in the results of the initial laboratory tests and your delays, the researchers focused on different approaches. Using advanced Artificial Intelligence (AI) techniques combined with the chest by radiological imaging can lead to a more accurate diagnosis of COVID-19, by analyzing chest X-ray images. This approach allows for the detection of patients with COVID-19 and may help to control the loss of specialist doctors in remote areas.

Prior to the COVID-19 era some people wore masks to protect themselves from air pollution, while some people wore face masks to hide their faces and emotions from others. Prevention against coronavirus is a mandatory measure, according to the WHO wearing a mask is an effective way to prevent at least 80% of respiratory infections. Apart from this the WHO also recommends body distance training to reduce the spread of the virus. Many organizations apply the rules of wearing a face mask to protect themselves. Personally checking that people entering the organization wearing a mask is a very difficult and controversial task. To solve this type of problem many propose an in-depth learning model, called the Mobile Net Mask based primarily on the Mobile Net V2 modules and Open CV, to prevent human-to-human coronavirus transmission and face masking or without a mask. A separate set of data containing masked and non-masked images is used to train and test the model.

Discovering a face mask involves finding a part of the face and finding out if it contains a mask or not. The problem is solved by using the usual opposition method to obtain object categories. Face identification works in stages and divides a particular business group, Face. It has many applications such as automatic driving, surveillance, and so on. This small project introduces a simplified way to achieve the above objectives using basic Machine Learning (ML) packages such as TensorFlow, Keras, Open CV, MobileNet V2 and Scikit-Learn.

1.2 Objectives

1. To develop an objection method that combines one stage and two stage detectors for accurately detect the object in real time from video streams with transfer learning at the backend.

2. To Stop the spreading of COVID-19.

3. To create an optimized model which can be easily deployable for embedded devices used for surveillance purposes.

4. To make life easier with the help of Computer Programming.

5. To make sure the COVID protocols are strictly followed by publics.

1.3 LITERATURE SURVEY

Several approaches are used for facial mask detection. For instance electromagnetic and radiometry techniques are used for facial masks detection. Deep Neural Network using machine learning are applied in face mask detection.

Face detection is a technique that has a wide range of applications, including face tracking, pose estimation, and motion detection. Face identification is a tough task since faces vary in size, shape, and colour from person to person, and they are not immutable.

There are two major issues in face detection. To begin with, there is a scarcity of datasets with both masked and unmasked faces, as well as the exclusion of face expression in the covered area.

The key aim in this project is to correctly detect the face from the image and then determine whether or not it has a mask. The proposed approach should be able to detect a face as well as a mask in motion in order to perform surveillance activities.

There are some points that need to be followed for the execution of this project:

- We have to detect the landmarks of the face including eyes, nose, mouth, jawline by using the facial landmarks method. To use this method we need to build a dataset of faces wearing face masks and another dataset which will contain faces not wearing a face mask.

- In the next step we have to apply face detection to compute the bounding box location of the face in the image.

- Next we need an image of a mask(with transparent background) and it will be automatically applied to the face by using facial landmarks to compute where the mask will be placed.

- Then the mask is resized and rotated, placing it on the face and applying colour on the bounding box of the face.

1.4 EXISTING SYSTEM

The major requirements for implementing this project is using python programming language along with deep learning and convolutional neural networks(CNN).

By using certain methods like Machine Learning, Computer Vison and also with python libraries the architecture consists of Mobile Net at the backbone, it can be used for high and low computation scenarios.

We are using four modules

1. Datasets Collecting: We collect no of dataset with face mask and without masks. The accuracy is dependent on the numbers of images that we collect.

2. Extracting Datasets: We can extract the datasets of mask and without mask by using the feature mobile net v2 to set a properly trained model that can be implemented in real time.

3. Model Training: The training of model would be done by using OpenCV and keras(Python libraries).

4. Facemask Detection: We can detect Pre processing image and also detect via live video stream. If a person wear mask it will permit that particular person or else it will give the buzzer to wear a mask and to prevent them from virus.

Advantages of implementing this project:

1. Manual monitoring is very difficult and hectic task for offices to check whether all people are using face masks or not during their working hours. So in our technique we are using webcam to detect peoples faces and to prevent them from virus transmission.

2. It is fast and have very high accuracy.

3. This system can be used in ATMs, banks, Railway stations and many other public gathering places to ensure proper practicing of COVID protocols.

1.5 PROPOSED SYSTEM

The proposed method contains a cascade classifier and pre trained CNN which consists of two 2D convolution layers connected to layers of dense neurons which is followed by some certain algorithm.

The proposed model is designed using python libraries namely Tensorflow, Keras and OpenCV. The model we used is MobileNetV2 of convolutional neural network(CNN). Transfer learning is the method of using MobileNetV2 and CNN. Transfer learning is using some pre trained model to train our present model and get the proper prediction which is very helpful for smooth and easy training. We have initialized the model learning rate, number of epochs and batch size.

We are using datasets of images with two class one is with mask and another is without mask to train the model. The dataset contains average number of 993 images of with mask and 1918 images of without mask.

Firstly we feed the dataset in the model and run the training program, which trains the model based on the given datasets. Then we are running the detection method which turns on the video stream, captures the frames continuously from the video stream with an anchor box which gives signals based on the person is wearing mask or not by using object detection process. This is passed through MobileNetV2 model layers which classifies the image as with or without mask if the person is wearing mask a green anchor box will be displayed and red if not wearing mask with the accuracy tagged on the top portion of the box.

The Face mask detection system uses AI technology to detect a person is wearing mask or nor and it can be connected with any surveillance system installed at any place. The authorities or admin can check the person through the system to conform their identity.

CHAPTER 2 SYSTEM REQUIREMENTS

a. Hardware Requirements

- Working Webcam
- 4GB of RAM and above
- 64-bit Processor
- Min I5 processor or above

b. Software Requirements

- Python programming language
- Python IDE : Pycharm, Jupyter notebook, Microsoft VS Code.
- Python Libraries : OpenCV, Tensorflow, MobileNetV2, Keras, Numpy, Sickitlearn, imutis, matplotlib.

CHAPTER 3 SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

The input image is taken from the dataset. It consists of two major stages, the first stage of our architecture includes a face detector which localizes multiple image of varying sizes and detects faces in overlapping scenarios.

The detected faces from the region of interest is extracted from this stage and then branched together and passed to the second stage.

In second stage of our architecture, which is a convolutional neural network 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 is correctly detected and classified as either masked or unmasked faces.

There are some several methods used here:

1. Data Visualization : Data visualisation is the process of translating abstract or raw data into meaningful representations through knowledge exchange and insight finding through encoding. This is a very useful technique to investigate a certain pattern in a dataset.

   The dataset is divided into two categories: photographs with face masks are labelled 'with mask,' and images without masks are labelled 'without mask.' The list of directories in the supplied path is then returned using the os.listdir method. ['with mask', 'without mask'] will be the variable names for the categories.

   We need to differentiate such categories for labelling the photographs by looping through them using any looping expression. The labels will then be set to [0,1], and the photos will be reshaped and converted.

2. Training Of Model : The training of the model will be done by using CNN and it has also became the most ascendant in miscellaneous computer vision tasks.
   This model will be built by using CNN architecture and training, testing and splitting the data will also be done by using convolutional neural network method(CNN).

3.2 ALGORITHMS/FLOWCHARTS

Algorithm : Face mask detection

Input: Dataset including faces with and without masks.

Output : Categorized image depicting the presence of face mask.

for each image in dataset do:

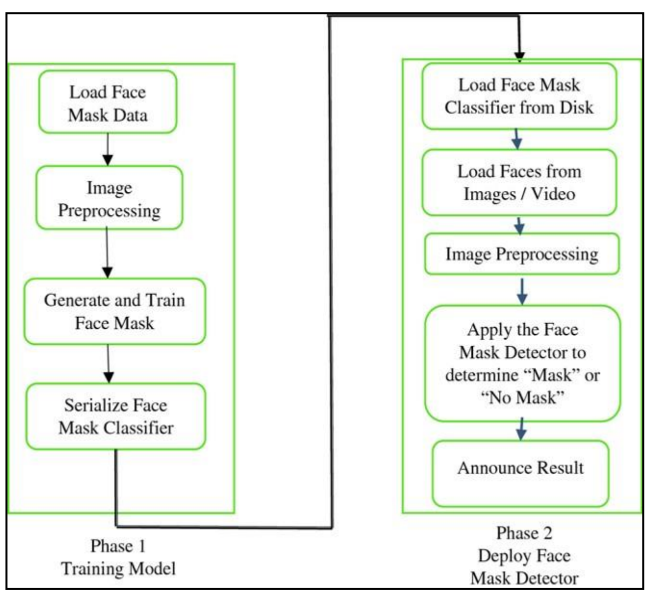
1. Divide the image into two groups and give each one a name
2. Convert the RGB image to Gray scale image.
3. Resize the grayscale image to a 100 × 100 pixel size.
4. Convert the image to a 4D array by normalizing.

For building CNN model do :

1. Add a 200-filter Convolutional layer.
2. Add a second 100-filter Convolutional layer.
3. Install Flatten layer on network network.
4. Add a dense layer of 64 neurons.
5. Add a dense final layer with 2 effects for 2 stages.

Split the data and train the model

Flowchart:



<u>**Process of training and Deploying the model**</u>

CHAPTER 4 Implementations and Results

4.1 PSEUDOCODE

<u>Pseudocode(For training Model)</u>

IMPORT PACKAGES:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os
```

INITIALIZE LEARNING RATE = 10e-4

```
INITIALIZE EPOCHS=20
INITIALIZE BS=32
  READ Directory=r"Specified path of dataset."

  READ Catagories=["Give folder name of dataset with mask and without mask."]
  Print Loading Images.....
  CREATE EMPTY LISTS data=[] and labels=[]
  LOOP catagories
  READ PATH=join(Directory,Category)
  LOOP Path
  JOIN img_path=join(path,img)

  STORE image=load_img(img_path,target_size=(224,224))

  Preprocess Input using necessary functions
  append image
  append cataegory

  ENCODE lb=LabelBinarizer()
  CONVERT labels=lb.fit_transform(labels)
          labels=lb.to_categorical(labels)
  CREATE data=np.array(data,dtype="float32")
  TRAIN AND TEST X and Y.
  GENERATE IMAGE DATA USING ImageDataGeneration()
  LOAD MOBILE NET v2
  LOAD baseModel = MobileNetV2(weights="imagenet", include_top=False,
          input_tensor=Input(shape=(224, 224, 3)))
  CONSTRUCT THE HEADMODEL USING SPECIFIC FUNCTIONS

  LOOP using for layer in baseModel.layers:

                  layer.trainable = False
  COMPILE THE MODEL
  print "[INFO] compiling model.  "
```

STORE opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
COMPILE model.compile(loss="binary_crossentropy", optimizer=opt,
    metrics=["accuracy"])
Print "Training head… "
TRAIN HEAD NETWORK

print "[INFO] evaluating network.. "

MAKE PREDICTIONS ON TESTING SET

For each and every images we need to find the index of the label with largest predicted probality
PREDICT predIdxs = np.argmax(predIdxs, axis=1)
SAVING THE MODEL

print "[INFO] saving mask detector model. "

SAVE model.save("mask_detector.model", save_format="h5")
PLOTTING THE RESULTS OF THE MODEL
INITIALIZE N = EPOCHS

USING numpy and matplotlib.pyplot packages plot all the necessary training and testing results.

Pseudocode(for detecting face mask in video stream)

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
FUNCTION detect_and_predict_mask(frame, faceNet, maskNet):
DETECT (h,w)=frame.shape[:2]

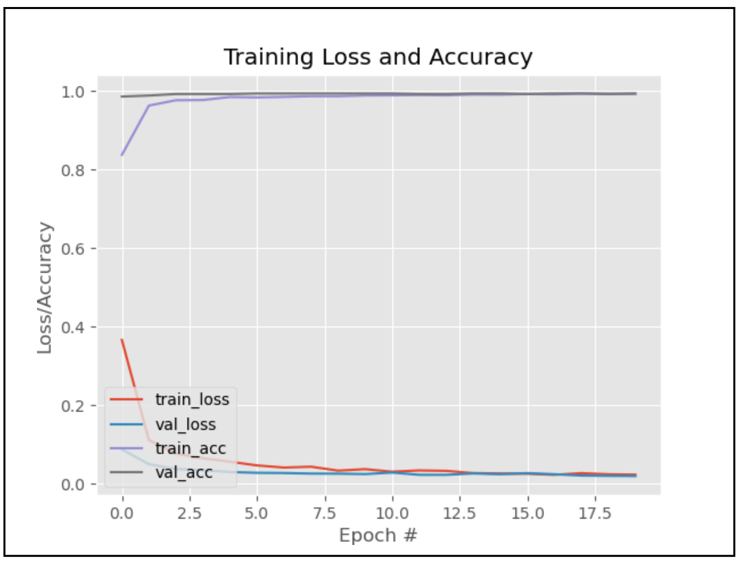
```
DETECT FRAME SIZE

INITIALIZE faces=[],locs=[],preds=[]
LOOP i in range(0, detections.shape[2]):
        confidence=detections[0,0,i,2]
        CONDITION confidience>0.5
        box = detect the box and resize it.
                        (startX, startY, endX, endY) = box.astype("int")
        SET DIMESIONS AND COLOUR OF THE FRAME BOX
        CONDITION len(faces) > 0:

                        faces = np.array(faces, dtype="float32")
                preds = maskNet.predict(faces, batch_size=32)
RETURN locs,preds

SET PATH prototxtPath = r"face_detector\deploy.prototxt"

SET PATH weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
READ faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
LOAD maskNet = load_model("mask_detector.model")
LOOP while : true
                LOOP OVER THE VEDIO STREAM FOR DETECTING MASK

OUTPUT cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF
CONDITION FOR BREAK if key == ord("q"):

                        break
CLEAN cv2.destroyAllWindows()
        vs.stop()
```

4.2 RESULTS

We tested the model in different cases, below mentioned table and the results of those cases with a number of 20 seasons and a set size 32 consisting of the total three conditions. We used average Pooling to photograph smooth image. This table shows the comparison results various parameters and conditions.

Model	Learning Rate	With mask distance	Without mask distance	Blur image quality	Multiple people capturing
1.	1e-4	161 cm	190 cm	Good	4 people
2.	1e-3	155 cm	187 cm	Average	3 people
3.	1e-2	146 cm	179 cm	Bad	3 people

According to the above results the first model is the best compared to all models. The structure of the best model from Our research is illustrated below. Indicates training loss, loss of validation, training accuracy and validation accuracy. In the amount of epoch against loss or accuracy. It seems from the structure that as the number of epochs increases training and verification accuracy increases with training as well verification accuracy decreases. And again confirmation of the accuracy is higher than the accuracy of the proven training that the model does not suffer from overuse.



Graph for Number of epochs Vs Loss or Accuracy

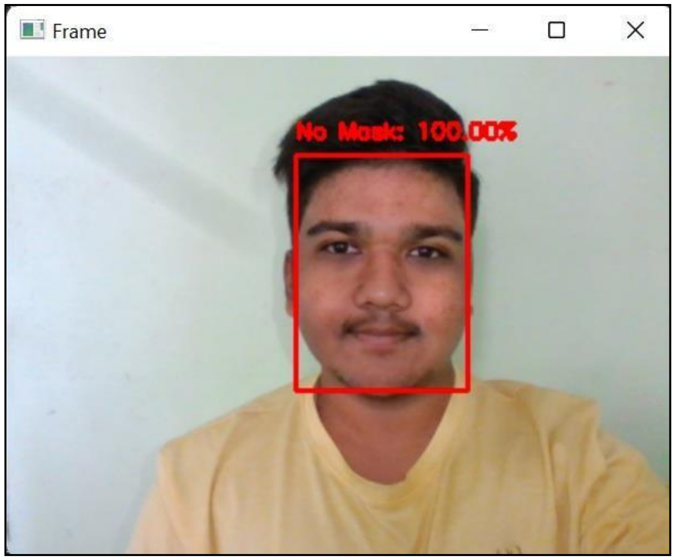
The current proposed model provides greater individual accuracy face with mask on and off. For many faces and offers very good accuracy. Works easily on just about any mobile device by turning on video streaming, without external computer hardware requirement. In addition we will work to improve accuracy in order to get the most face mask, to divide the face into three categories namely, With Masks, Without Masks.

AFTER TRAINING THE RESULTS OF THE TRAINED MODEL

```
Command Prompt
acy: 0.9935
Epoch 16/20
95/95 [==============================] - 56s 594ms/step - loss: 0.0275 - accuracy: 0.9898 - val_loss: 0.0290 - val_accur
acy: 0.9922
Epoch 17/20
95/95 [==============================] - 55s 583ms/step - loss: 0.0242 - accuracy: 0.9898 - val_loss: 0.0282 - val_accur
acy: 0.9935
Epoch 18/20
95/95 [==============================] - 56s 590ms/step - loss: 0.0253 - accuracy: 0.9931 - val_loss: 0.0254 - val_accur
acy: 0.9935
Epoch 19/20
95/95 [==============================] - 58s 610ms/step - loss: 0.0219 - accuracy: 0.9941 - val_loss: 0.0243 - val_accur
acy: 0.9935
Epoch 20/20
95/95 [==============================] - 57s 604ms/step - loss: 0.0235 - accuracy: 0.9927 - val_loss: 0.0236 - val_accur
acy: 0.9948
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       0.99      0.99      0.99       383
without_mask       0.99      0.99      0.99       384

    accuracy                           0.99       767
   macro avg       0.99      0.99      0.99       767
weighted avg       0.99      0.99      0.99       767

[INFO] saving mask detector model...

C:\Users\Aritroo\Downloads\Face-Mask-Detection-master\Face-Mask-Detection-master>
```

IMPLEMENTATION OF OUR MODEL IN VIDEO STREAM

FOR THE PERSON WITHOUT MASK:



FOR THE PERSON WHO IS WEARING A FACE MASK:

CHAPTER 5 CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

To reduce the spread of the COVID-19 epidemic, steps must be taken. Show us a facemask detector using Convolutional Neural Network(CNN) and convey learning strategies to neural organizations. To train, validate and test model, using database featuring 993 and 1918 concealed facial images photos of exposed faces. These photos were taken different assets such as Kaggle database and RMFD. The model is made in photos and live video transfers. To choose a basic model, we checked the ratings such as accuracy, precision, and memory and selection MobileNetV2 architecture with the best display we have 99% accuracy and 99% memory. It is more than that using MobileNetV2 makes it easy to present the model to be installed structures. This face mask finder can be set at many regions such as shopping malls, air terminals and other large traffic areas for general population testing and to prevent the spread of the virus by looking at who is following important rules and who does not follow.

5.2 FUTURE ENHANCHMENT

There are a few aspects we plan to work on soon:

- Currently, the model offers 5 FPS cutting speed per CPU. In the future, we plan to upgrade this to 15 FPS, making our solution workable for CCTV cameras, without the need for a GPU.

- The use of Learning Machine in the field of mobile messaging is increasing rapidly. Therefore, we plan to include our models in their versions of TensorFlow Lite.

- Our designs can be customized with TensorFlow RunTime (TFRT), which will increase the performance of the views on end-to-end devices and make our models work better on multi-stranded CPUs.

- Stage 1 and Phase 2 models can be easily replaced with future upgraded models, which can provide better accuracy and lower latency.

REFERENCE

1. https://ieeexplore.ieee.org/document/9342585
2. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 1998, 86, 2278–2324. [CrossRef]
3. Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, Yanhong Luo, "A Novel Image Classification Approach via Dense-MobileNet Models", Mobile Information Systems, vol. 2020,ArticleID 7602384, 8 pages, 2020. https://doi.org/10.11 55/2020/7602384
4. M. S. Ejaz and M. R. Islam, "Masked Face Recognition Using Convolutional Neural Network," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9068044
5. Changjin Li, Jian Cao, and Xing Zhang. 2020. Robust Deep Learning Method to Detect Face Masks. In *Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture* (*AIAM2020*). Association for Computing Machinery, New York, NY, USA, 74–77. DOI:https://doi.org/10.1145/3421766.3421768
6. 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.
7. https://www.ijert.org/covid-19-facemask-detection-with-deep-learning-and-computer-vision