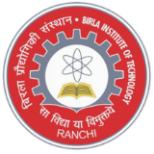
**COMPREHENSIVE SOCIAL DISTANCING SYSTEM**

A Thesis

***SUBMITTED TO***

**BIRLA INSTITUTE OF TECHNOLOGY, MESRA**

**RANCHI, INDIA**

** IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE OF**

**BACHELOR OF ENGINEERING (CS)**

BY

**NIRANJANAA MOHANBABU BE/50024/17**

**SHANIA FERNANDES BE/50010/17**

**ANAAM SAYEDA FATMA BE/50011/17**

UNDER THE GUIDANCE

** Mr. PRABAL DUTTA**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**

**WALJAT COLLEGE OF APPLIED SCIENCE**

**MUSCAT, OMAN**

**DECLARATION**

I/ We certify that

a. The work has not been submitted to any other College/Institute/University for any degree.

b. I have followed the guidelines provided by the college in writing project report.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the college.

d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the project report and giving their details in the references.

e. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Date: 16/12/2020

NIRANJANAA MOHANBABU BE/50024/17

SHANIA FERNANDES BE/50010/17

ANAAM SAYEDA FATMA BE/50011/17

|  |  |  |
| --- | --- | --- |
| **C:\Users\student\Desktop\bit.png** | **WALJAT COLLEGE OF APPLIED SCIENCES**  **In academic partnership with**  **BIRLA INSTITUTE OF TECHNOLOGY** |  |

**CERTIFICATE**

This is to certify that the project entitled “**COMPREHENSIVE SOCIAL DISTANCING SYSTEM**” submitted by Niranjanaa Mohanbabu, Shania Fernandes and Anaam Sayeda Fatma is an original work and indebtedness to other works/publications has been duly acknowledged at places. It has not been submitted to any other University or Institution for any other degree or diploma to the best of my knowledge and belief.





## ACKNOWLEDGEMENT

We owe the development of our project to God Almighty and our professors who helped and supported us during the making of the project. Our deepest thanks to our guide **Mr. Prabal Dutta** for extending his guidance & support to this project. We would also like to thank **Ms. Rajeev Ranjan** for his advice and unending support which was useful to us in the project. We would also like to thank our Institution and our faculty members without whom this project would have been a distant reality. We also extend our heartfelt thanks to our family and well-wishers. It would be incomplete on our part if we do not thank our parent college, **Birla Institute of Technology, Mesra, Ranchi, India** for giving us, ﬁnal year students, the opportunity to put all our acquired skills together in developing our ﬁnal year projects. The presence of Project as a separate unit in our syllabus, has not only allowed us to apply what we have learnt, but has also embedded in us the zeal and passion to be creative and learn more.

NIRANJANAA MOHANBABU BE/50024/17

SHANIA FERNANDES BE/50010/17

ANAAM SAYEDA FATMA BE/50011/17

**ABSTRACT**

With the ongoing pandemic, it is important to have advanced systems and services in place to mitigate risk. For public safety, health authorities have recommended face masks to control the spread of COVID-19. The system we have built uses Artificial Network to recognize a user not wearing a mask by using CCTV camera with quite good accuracy. The person is denied access to places where it is strictly advised to wear a face mask by screen panels installed at entrances sending a pop-up warning message. After the access is denied, the authorities will get an alert email in real time with the photo of the person being attached. This kind of system can be used at airports, Hospitals, Offices and public places. We have used Python, OpenCV, Keras with TensorFlow as a backend for building the

Deep Learning Convolution Neural Network in this project. This system also detects the temperature and counts the number of people, thus making it all in one system with no additional expensive system required.

#### CONTENTS

#### PAGE NUMBER

COVER PAGE

DECLARATION I

CERTIFICATE II

ACKNOWLEDGEMENT III

[ABSTRACT IV](#_TOC_250005)

[CONTENTS V](#_TOC_250004)

[LIST OF FIGURES VI](#_TOC_250003)

1. INTRODUCTION 1

2. REVIEW OF LITERATURE 3

3. PROBLEM FORMULATION 5

4. SYSTEM REQUIREMENTS 6

4.1 HARDWARE DESCRIPTION 6

4.2 BLOCK DIAGRAM 11

4.3 SOFTWARE DESCRIPTION 12

4.3.1 SOURCE CODE 13

5. METHODOLGY 23

6. RESULT 25

7. CONCLUSION AND FUTURE SCOPE 26

8. REFERENCES 28

# LIST OF FIGURES

1.1 CIRCUIT DIAGRAM OF AUTONOMOUS FIRE FIGHTING ROBOT

4.1.1 NIVIDIA 3080A

4.1.2 INFRARED SENSOR

4.1.3 CCTV INFRASTRUCTURE

4.1.4 RASPBERRY PI MODULO V2

4.2 FLOWCHART OF CNNs

4.3 BLOCK DIAGRAM FOR SOCIAL DISTANCING SYSTEM

**CHAPTER 1**

**INTRODUCTION**

The year 2020 has been such a roller coaster ride for all of us. A new disease, called as the CORONA VIRUS has been spreading profusely through the world. Many lives have changed as the situation had worsened as time passed. Lockdowns were established and people were forced to sit in their homes so that they are protected from the virus and outgoing are limited. The psychological effects of pandemics after a few weeks of isolation can cause lasting anxiety as we are living through the first global pandemic in the digital age, where the internet has made it possible to withdraw physically from the outside world. The corona virus disease (COVID-19) has affected virtually every industry in one way or another causing losses leading the downfall in the economy of several countries.

Amid the global crisis, new demand has emerged in the market, and that is of face mask detection. It is one such technology capable of detecting a face with a mask and verifying that person’s identity. It incorporates an AI-based pattern recognition system that uses biometric data of individuals. It extracts facial features and classifies them in different categories. Besides, it can also identify people without masks by generating an alarm or a notification to notify security or officials. They can see who has not covered faces with masks through software, mobile app, device, or a website.

**CHAPTER 2**

**LITERATURE REVIEW**

**CHAPTER 3**

**PROBLEM FORMULATION**

#### Problem Statement:

The deaths due to corona virus have been on a rise ever since the man has discovered it. Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment.  Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness. The best way to prevent and slow down transmission is to be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face. The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it’s important that you also practice respiratory etiquette

*3.1 General objective*:

1. Be able to detect if people are wearing masks while entering the public places. Record their temperatures for future references.
2. To make the work easier for all the employees who do manual work of taking temperature for every individual who visit public places.

#### Speciﬁc objectives:

1. Detecting the customers who do not wear mask and alert the authorities by using

2. Using infrared sensor to record temperatures of the current customers in the place.

3. Be able to train the classifier to detect faces wearing masks or not and keep in count the amount of people in the public places.

**CHAPTER 4**

**SYSTEM REQUIRMENTS**

**4.1 HARDWARE DESCRIPTION**

**4.1.1**  Nvidia GEFORCE RTX

The GeForce RTX 3080 is an enthusiast-class graphics card by NVIDIA, launched in September 2020. Built on the 8 nm process, and based on the GA102 graphics processor, in its GA102-200-KD-A1 variant, the card supports DirectX 12 Ultimate. This ensures that all modern games will run on GeForce RTX 3080. The GA102 graphics processor is a large chip with a die area of 628 mm² and 28,300 million transistors. The GeForce RTX 3080 features 8704 shading units, 272 texture mapping units, and 96 ROPs. Also included are 272 tensor cores which help improve the speed of machine learning applications. The card also has 68 raytracing acceleration cores. NVIDIA has paired 10 GB GDDR6X memory with the GeForce RTX 3080, which are connected using a 320-bit memory interface. The GPU is operating at a frequency of 1440 MHz, which can be boosted up to 1710 MHz, memory is running at 1188 MHz (19 Gbps effective).

****

*Figure 4.1.1 Nvidea Geforce RTX 3080*

**4.1.2 INFRARED THERMOMETER SENSOR**

The MLX90614 is an infrared thermometer for non-contact temperature measurements. Both the IR sensitive thermopile detector chip and the signal conditioning ASIC are integrated in the same TO-39 can. This sensor is used to measure the temperature even at a large distance as it has higher accuracy. Integrated into the MLX90614 are a low noise amplifier, 17-bit ADC and powerful DSP unit thus achieving high accuracy and resolution of the thermometer.

Figure 4.1.2 INFRARED THERMOMETER

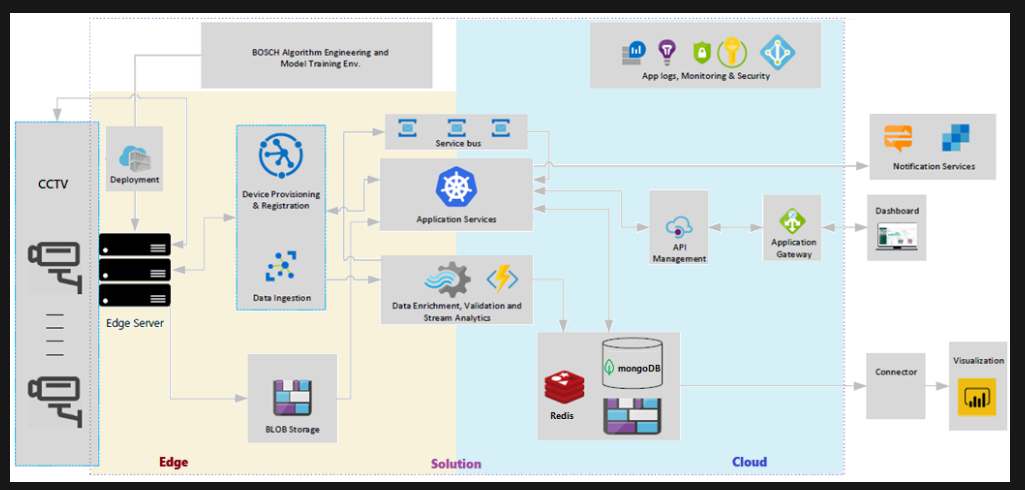
**4.1.3 CCTV Infrastructure**

 CCTVs send video data to Internet of Things (IoT) Edge servers. Edge computing handles device registration, provisioning, and data ingestion.

 The Bosch Algorithm Engineering and Model Training Environment uses custom vision analytics to continually retrain machine learning (ML) models, and directly updates edge servers.

 Edge servers send data to onboard and cloud stream analytics and blob storage. IoT Edge intelligent devices limit costs by preprocessing and sending only necessary data to the cloud.

 Stream analytics perform data enrichment and validation on both edge and cloud data.



**Figure 4.1.3 FLOW CHART**

**4.1.4 Raspberry Pi camera of module v2 is used for RGB(visual Spectrum) Camera**; The blue channel contains the visible spectrum image and the red channel contains the IR video thresholded at 30°C to show the person. A Haar Cascade Classifier is used for the detection of face and the eye to measure temperature as estimation is more accurate when estimated from the facial region.

**Fig4.1.4 Raspberry Pi Modulo v2**

**4.2. BLOCK DIAGRAM**

****

**Figure 4.2 Block diagram of Social Distancing System**

**4.3 SOFTWARE DESCRIPTION**

Let's start with what Convolutional Neural Networks really are. Like the way our brains identify objects when we see a picture, the goal is to get computers to recognize objects in the same manner. To a computer, an image is just another array of numbers. Each object has its own pattern and that is what the computer will use to identify an object in an image.

To explain convolutional neural networks in simple terms - Just as parents train their children to understand what a ball is or what food is, similarly, computers are also trained by showing a million images of the same object so that their ability to recognize that object increases with each sample.

**4.3.1 SOURCE CODE**

# Configuration related preprocessing step before mounting the drive

!apt-get install -y -qq software-properties-common python-software-properties module-init-tools

!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null

!apt-get update -qq 2>&1 > /dev/null

!apt-get -y install -qq google-drive-ocamlfuse fuse

from google.colab import auth

auth.authenticate\_user()

from oauth2client.client import GoogleCredentials

creds = GoogleCredentials.get\_application\_default()

import getpass

!google-drive-ocamlfuse -headless -id={creds.client\_id} -secret={creds.client\_secret} < /dev/null 2>&1 | grep URL

vcode = getpass.getpass()

!echo {vcode} | google-drive-ocamlfuse -headless -id={creds.client\_id} -secret={creds.client\_secret}

#Mount the google drive

from google.colab import drive

drive.mount('/content/drive')

"""## Data Pre-processing"""

# Import Necessary Libraries

import os

# Set Directory path for Dataset

os.chdir("/content/drive/My Drive/Youtube\_Projects/Face\_Mask\_Detection\_Alert\_System")

Dataset='Dataset'

Data\_Dir=os.listdir(Dataset)

print(Data\_Dir)

# Import necessary libraries

import cv2

import numpy as np

from tensorflow.keras.utils import to\_categorical

from sklearn.preprocessing import LabelBinarizer

from sklearn.model\_selection import train\_test\_split

img\_rows, img\_cols = 112, 112

images = []

labels = []

for category in Data\_Dir:

folder\_path = os.path.join(Dataset, category)

for img in os.listdir(folder\_path):

img\_path = os.path.join(folder\_path, img)

img=cv2.imread(img\_path)

try:

#Coverting the image into gray scale

grayscale\_img=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

#resizing the gray scaled image into size 56x56 in order to keep size of the images consistent

resized\_img=cv2.resize(grayscale\_img,(img\_rows, img\_cols))

images.append(resized\_img)

labels.append(category)

# Exception Handling in case any error occurs

except Exception as e:

print('Exception:',e)

images=np.array(images)/255.0

images=np.reshape(images,(images.shape[0],img\_rows, img\_cols,1))

# Perform one hot encoding on the labels since the label are in textual form

lb = LabelBinarizer()

labels = lb.fit\_transform(labels)

labels = to\_categorical(labels)

labels = np.array(labels)

(train\_X, test\_X, train\_y, test\_y) = train\_test\_split(images, labels, test\_size=0.25,

random\_state=0)

"""## Build Covolutional Neural Network (CNN) Classification Model"""

# Import Necessary Keras Libraries

from keras.models import Sequential

from keras.layers import Dense,Activation,Flatten,Dropout

from keras.layers import Conv2D,MaxPooling2D

# Define model paramters

num\_classes = 2

batch\_size = 32

# Build CNN model using Sequential API

model=Sequential()

#First layer group containing Convolution, Relu and MaxPooling layers

model.add(Conv2D(64,(3,3),input\_shape=(img\_rows, img\_cols, 1)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2,2)))

#Second layer group containing Convolution, Relu and MaxPooling layers

model.add(Conv2D(128,(3,3)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2,2)))

#Flatten and Dropout Layer to stack the output convolutions above as well as cater overfitting

model.add(Flatten())

model.add(Dropout(0.5))

# Softmax Classifier

model.add(Dense(64,activation='relu'))

model.add(Dense(num\_classes,activation='softmax'))

print(model.summary())

"""## Plot the Model"""

# Plot the model

from keras.utils.vis\_utils import plot\_model

plot\_model(model, to\_file='face\_mask\_detection\_architecture.png')

"""## Train the Model"""

from keras.optimizers import Adam

epochs = 50

model.compile(loss = 'categorical\_crossentropy',

optimizer = Adam(lr=0.001),

metrics = ['accuracy'])

fitted\_model = model.fit(

train\_X,

train\_y,

epochs = epochs,

validation\_split=0.25)

"""## Plot the Training Loss & Accuracy"""

from matplotlib import pyplot as plt

# Plot Training and Validation Loss

plt.plot(fitted\_model.history['loss'],'r',label='training loss')

plt.plot(fitted\_model.history['val\_loss'],label='validation loss')

plt.xlabel('Number of Epochs')

plt.ylabel('Loss Value')

plt.legend()

plt.show()

# Plot Training and Validation Accuracy

plt.plot(fitted\_model.history['accuracy'],'r',label='training accuracy')

plt.plot(fitted\_model.history['val\_accuracy'],label='validation accuracy')

plt.xlabel('Number of Epochs')

plt.ylabel('Accuracy Value')

plt.legend()

plt.show()

"""## Save or Serialize the Model"""

# Save or Serialize the model with the name face\_mask\_detection\_alert\_system

model.save('face\_mask\_detection\_alert\_system.h5')

# Import necessary linbraries

from keras.models import load\_model

import cv2

import numpy as np

import tkinter

from tkinter import messagebox

import smtplib

# Initialize Tkinter

root = tkinter.Tk()

root.withdraw()

#Load trained deep learning model

model = load\_model('face\_mask\_detection\_alert\_system.h5')

#Classifier to detect face

face\_det\_classifier=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

# Capture Video

vid\_source=cv2.VideoCapture(0)

# Dictionaries containing details of Wearing Mask and Color of rectangle around face. If wearing mask then color would be

# green and if not wearing mask then color of rectangle around face would be red

text\_dict={0:'Mask ON',1:'No Mask'}

rect\_color\_dict={0:(0,255,0),1:(0,0,255)}

SUBJECT = "Subject"

TEXT = "One Visitor violated Face Mask Policy. See in the camera to recognize user. A Person has been detected without a face mask in the Hotel Lobby Area 9. Please Alert the authorities."

# While Loop to continuously detect camera feed

while(True):

ret, img = vid\_source.read()

grayscale\_img = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_det\_classifier.detectMultiScale(grayscale\_img,1.3,5)

for (x,y,w,h) in faces:

face\_img = grayscale\_img[y:y+w,x:x+w]

resized\_img = cv2.resize(face\_img,(56,56))

normalized\_img = resized\_img/255.0

reshaped\_img = np.reshape(normalized\_img,(1,56,56,1))

result=model.predict(reshaped\_img)

label=np.argmax(result,axis=1)[0]

cv2.rectangle(img,(x,y),(x+w,y+h),rect\_color\_dict[label],2)

cv2.rectangle(img,(x,y-40),(x+w,y),rect\_color\_dict[label],-1)

cv2.putText(img, text\_dict[label], (x, y-10),cv2.FONT\_HERSHEY\_SIMPLEX,0.8,(0,0,0),2)

# If label = 1 then it means wearing No Mask and 0 means wearing Mask

if (label == 1):

# Throw a Warning Message to tell user to wear a mask if not wearing one. This will stay

#open and No Access will be given He/She wears the mask

messagebox.showwarning("Warning","Access Denied. Please wear a Face Mask")

# Send an email to the administrator if access denied/user not wearing face mask

message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)

mail = smtplib.SMTP('smtp.gmail.com', 587)

mail.ehlo()

mail.starttls()

mail.login('Developer.TAU@gmail.com','Hello@001@1234')

mail.sendmail('Developer.TAU@gmail.com','Developer.TAU@gmail.com',message)

mail.close

else:

pass

break

cv2.imshow('LIVE Video Feed',img)

key=cv2.waitKey(1)

if(key==27):

break

cv2.destroyAllWindows()

source.release()

# In[ ]:

**CHAPTER 5**

**METHODOLOGY**

This intelligent system was built in three major steps. First, we need to apply face detection to compute the bounding box location of the face in the image. Facial landmarks are applied allowing the system to localise the eyes, nose, mouth etc. Next, the neural network is designed and trained to detect correct placement of the mask on face. Only after this, entry is allowed.

Training: This focuses on loading the face mask detection dataset from disk, using Keras/TensorFlow to train the model on this dataset, and then serializing the face mask detector to disk.

Deployment: After the face mask detector is trained, following tasks of loading the mask detector performing face detection, and then classifying each face as with mask or without it.

These processes are elaborated upon in the following text.

A. Preparing Dataset;

To create the dataset of faces wearing face masks in this project so, following steps have been performed. First, apply face detection to compute the bounding box location of the face in the image. Apply the facial landmarks, allowing the system to localize the eyes, nose, mouth, etc. Get advantage of deep learning Technology. The mask will be automatically applied to the faces that the mask is based on the facial landmarks to resize, rotate and replace automatically. Repeat this process for all the input images, thereby creating artificial mask dataset.

B. Designing Neural Network

In this section, we will describe building the convolution neural network as well as training and serializing it. We choose to use tensorflow as our backend engine. To begin, we import required Keras libraries such as, but not limited to, Sequential, Flatten, Dropout and Conv2D. Sequential represents a linear stack of layers. It allows us to build a model layer by layer. It is the simpler version of Functional.We define the two required classes for our need:- with mask and without mask. We set our batch size to 32; this is the number of images required to be fed before the neural network updates its internal parameters.

C. Face Capture and Detection

In this section, we will be detecting Face ROI [region of interest] from live video feed and whether mask is worn or not, as well as write Python code for alerting authorities with pop-up window. We start by importing required Keras framework libraries such as numpy and cv2 to work with webcam and images. Tkinter is used to create warning pop-up and smtplib is used to send the alert email to authorities. Tkroot widget is created first, and hidden with withdraw function.



**Figure 5.1 Flow chart of comprehensive social distancing system**

**CHAPTER 6**

**RESULT**

We used OpenCV, tensor flow, keras and CNN to detect whether people were wearing face masks or not. The models were tested with images and real-time video streams. The accuracy of the model is achieved and, the optimization of the model is a continuous process and we are building a highly accurate solution by tuning the hyper parameters. This specific model could be used as a use case for edge analytics. Furthermore, the proposed method achieves state-of-the-art results on a public face mask dataset. By the development of face mask detection we can detect if the person is wearing a face mask and allow their entry would be of great help to the society.

Using thermal cameras and infrared sensors, human body temperature screening systems can monitor the body temperature of individuals at entry points like office buildings, retail outlets, construction sites, subway stations, etc. The intelligent temperature sensing hardware identifies an individual with elevated body temperature and gives a warning of temperature abnormality.

**CHAPTER 7**

**CONCLUSION AND FUTURE SCOPE**

**7.1 CONCLUSION**

As the technology are blooming with emerging trends the availability so we have novel face mask detector which can possibly contribute to public healthcare. In order to extract more robust features, we utilize transfer learning to adopt weights from a similar task face detection, which is trained on a very large dataset.

Technology for sure has matured and is slowly settling into the lives of people due to its usability and has made lives easier. Right from its invention to its accessibility, artificial intelligence is here not just to stay, but to grow and conquer.

By being able to know the capacity of number of people in the public places, it will be much easier to control the spread of the virus and thus protecting the public. We will be able to achieve maximum yield by using large datasets and safeguard the health of many individuals by recording using multiple sensors and being thorough in the set up.

**7.2 FUTURE SCOPE**

The technology behind social distancing system is changing day by day. Our model ensures the ease of using and comfort to be placed anywhere and can also be portable. This device can be updated by adding the following modiﬁcations.

1. Adding more sensors to make it more sophisticated using artificial intelligence to make it more accurate.

2. Using visual positioning system to livestream the place of entrances in public places.

3.Use of virtuals doors to keep the capacity control in check.

4.Use of wide range sensors and cameras to be able to record temperatures of many people at once.

**CHAPTER 8**

**REFERENCES**

[1] P. A. Rota, M. S. Oberste, S. S. Monroe, W. A. Nix, R. Campagnoli, J. P. Icenogle, S. Penaranda, B. Bankamp,K. Maher, M.-h. Chenet al., “Characterization of a novel coronavirus associated with severe acute respiratorysyndrome,”science, vol. 300, no. 5624, pp. 1394–1399, 2003.

[2] Z. A. Memish, A. I. Zumla, R. F. Al-Hakeem, A. A. Al-Rabeeah, and G. M. Stephens, “Family cluster of middleeast respiratory syndrome coronavirus infections,”New England Journal of Medicine, vol. 368, no. 26, pp.2487–2494, 2013.

[3] Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklöv, “The reproductive number of covid-19 is higher comparedto sars coronavirus,”Journal of travel medicine, 2020.

[4] Y. Fang, Y. Nie, and M. Penny, “Transmission dynamics of the covid-19 outbreak and effectiveness of governmentinterventions: A data-driven analysis,”Journal of medical virology, vol. 92, no. 6, pp. 645–659, 2020.

[5] N. H. Leung, D. K. Chu, E. Y. Shiu, K.-H. Chan, J. J. McDevitt, B. J. Hau, H.-L. Yen, Y. Li, D. KM, J. Ipet al.,“Respiratory virus shedding in exhaled breath and efficacy of face masks.”

[6] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, “Rational use of face masks in the covid-19pandemic,”The Lancet Respiratory Medicine, 2020.

[7] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Peiet al., “Masked facerecognition dataset and application,”arXiv preprint arXiv:2003.09093, 2020.[10]Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,”IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[8] A. Kumar, A. Kaur, and M. Kumar, “Face detection techniques: a review,”Artificial Intelligence Review, vol. 52,no. 2, pp. 927–948, 2019.D.-H. Lee, K.-L. Chen, K.-H. Liou, C.-L. Liu, and J.-L. Liu, “Deep learning and control algorithms of direct perception for autonomous driving,2019

**URL**

**[1]** IRF44N , <https://components101.com/mosfets/irfz44n-datasheet-pinout-features>.

**[2]**1N5822 Schotkey diode, <https://components101.com/diodes/1n5822-schottky-diode>.

**[3]**1N4007diode, <https://www.theengineeringprojects.com/2017/07/introduction-to-1n4007.html>

**[4]** LM 2576 Buck convertor, <https://components101.com/ics/lm2576-buck-converter-pinout-equivalent-datasheet>.

### 