CHAPTER 1:

**INTRODUCTION**

According to the World Health Organization (WHO)’s official Situation Report – 205, coronavirus disease 2019 (COVID-19) has globally infected over 20 million people causing over 0.7million deaths . Individuals with COVID19 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

**LITERATURE SURVEY**

* 1. **An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network :**

COVID-19 pandemic caused by novel coronavirus is continuously spreading until now all over the world. The impact of COVID-19 has been fallen on almost all sectors of development. The healthcare system is going through a crisis. Many precautionary measures have been taken to reduce the spread of this disease where wearing a mask is one of them. In this paper, we propose a system that restrict the growth of COVID-19 by finding out people who are not wearing any facial mask in a smart city network where all the public places are monitored with Closed-Circuit Television (CCTV) cameras. While a person without a mask is detected, the corresponding authority is informed through the city network. A deep learning architecture is trained on a dataset that consists of images of people with and without masks collected from various sources. The trained architecture achieved 98.7% accuracy on distinguishing people with and without a facial mask for previously unseen test data. It is hoped that our study would be a useful tool to reduce the spread of this communicable disease for many countries in the world.

* 1. **Masked Face Recognition Using Convolutional Neural Network :**

Recognition from faces is a popular and significant technology in recent years. Face alterations and the presence of different masks make it too much challenging. In the real-world, when a person is uncooperative with the systems such as in video surveillance then masking is further common scenarios. For these masks, current face recognition performance degrades. An abundant number of researches work has been performed for recognizing faces under different conditions like changing pose or illumination, degraded images, etc. Still, difficulties created by masks are usually disregarded. The primary concern to this work is about facial masks, and especially to enhance the recognition accuracy of different masked faces. A feasible approach has been proposed that consists of first detecting the facial regions. The occluded face detection problem has been approached using Multi-Task Cascaded Convolutional Neural Network (MTCNN). Then facial features extraction is performed using the Google Face Net embedding model.

#### 2.3 EXISTING SYSTEM

face detection problem has been approached using Multi-Task Cascaded Convolutional Neural Network (MTCNN). Then facial features extraction is performed using the Google Face Net embedding model.

1.This system is capable to train the dataset of both persons wearing masks andwithout wearing masks.

After training the model the system can predicting whether the person is wearing the mask or not wearing mask.

**CHAPTER 3:**

**METHODOLOGY :**

#### PROPOSED SYSTEM

1. This system is capable to train the dataset of both persons wearing masks and without wearing masks.
2. After training the model the system can predicting whether the person is wearing the mask or not .
3. It also can access the webcam and predict the result.

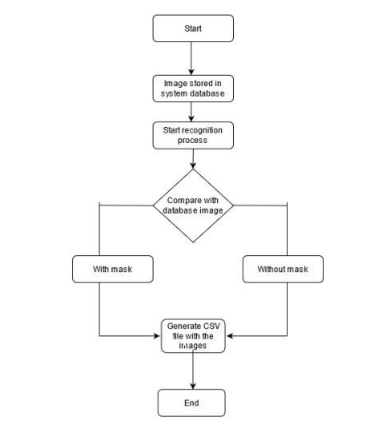


Fig 3.1: Flow chart diagram of face mask detection

#### 3.2 TENSORFLOW FRAMEWORK:

Tensor flow is an open-source software library.

Tensor flow was originally developed by researchers and engineers.

It is working on the Google Brain Team within Google’s Machine Intelligence research organization the purposes of conducting machine learning and deep neural networks research.

It is an opensource framework to run deep learning and other statistical and predictive analytics workloads.

It is a python library that supports many classification and regression algorithms and more generally deep learning.

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks.

It is a symbolic math library, and is also used for machine learning applications such as neural networks.

It is used for both research and production at Google, TensorFlow is Google Brain's second-generation system.

Version 1.0.0 was released on February 11, While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units).

Tensor Flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

The name Tensor Flow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

#### 3.3 OPENCV:

1. It is a cross-platform library using which we can develop real-time computer vision applications.
2. It mainly focuses on image processing, video capture and analysis including feature like face detection and object detection.
3. Currently Open CV supports a wide variety of programming languages like C++, Python,

Java etc. and is available on different platforms including Windows, Linux, OS X, Android, iOS etc.

1. Also, interfaces based on CUDA and OpenCL are also under active development for

high-speed GPU operations. Open CV-Python is the Python API of Open CV.

1. It combines the best qualities of Open CV C++ API and Python language.
2. OpenCV (Open-Source Computer Vision Library) is an opensource computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications

and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

1. The library has more than 2500optimized algorithms, which includes a comprehensive set of both classic and state-of -the-art computer vision and machine learning algorithms.
2. Algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

#### ARCHITECTURE OF NEURAL NETWORKS:

#### 3.4.1 FEED-FORWARD NETWORKS:

Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down. to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

#### 3.4.2 FEEDBACK NETWORKS:

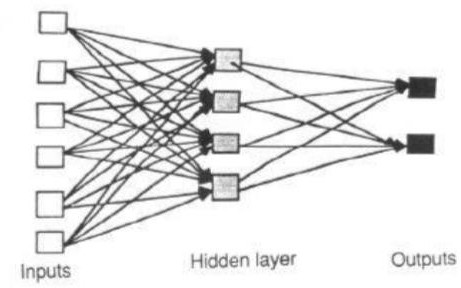
Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organization

Fig 3.2 Layers in NN

#### NETWORK LAYERS:

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of input units is connected to a layer of hidden units, which is connected to a layer of output units.

The activity of the input units represents the raw information that is fed into the network.

The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

Also distinguish single-layer and multi-layer architectures. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

#### Convolution Neural Network

A convolution neural network is a special architecture of artificial neural network proposed by yann lecun in 1988. One of the most popular uses of the architecture is image classification. CNNs have wide applications in image and video recognition, recommender systems and natural language processing. In this article, the example that this project will take is related to Computer Vision. However, the basic concept remains the same and can be applied to any other use-case!

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that we developed for neural networks still apply on CNNs. In more detail the image is passed through a series of convolution, nonlinear, pooling layers and fully connected layers, then generates the output.

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the visual cortex. CNNs use relatively little pre-processing compared to other image classification algorithms. CNN is a special kind of multi- layer NNs applied to 2-d arrays (usually images), based on spatially localized neural input. CNN Generate ‘patterns of patterns’ for pattern recognition.

Each layer combines patches from previous layers. Convolutional Networks are trainable multistage architectures composed of multiple stages Input and output of each stage are sets of arrays called feature maps. At output, each feature map represents a particular feature extracted at all locations on input. Each stage is composed of: a filter bank layer, a non-linearity layer, and a feature pooling layer. A ConvNet is composed of 1, 2 or 3 such 3-layer stages, followed by a classification module.

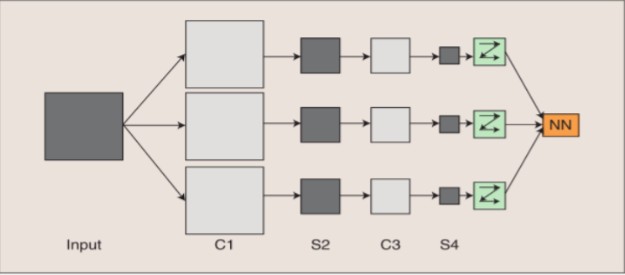


Fig 3.3 Convolution layer and pooled layers working process (Deep Learning Model)

Basic structure of CNN, where C1, C3 are convolution layers and S2, S4 are pooled/sampled layers.

Filter: A trainable filter (kernel) in filter bank connects input feature map to output feature map Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli.

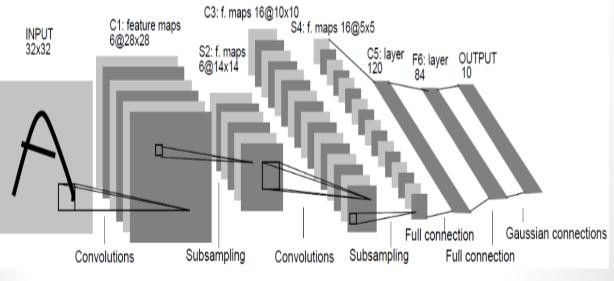


Fig 3.4: Convolution network principle

#### CONVOLUTIONAL LAYER

It is always first. The image (matrix with pixel values) is entered into it. Image that the reacting of the input matrix begins at the top left of image. Next the software selects the smaller matrix there, which is called a filter. Then the filter produces convolution that is moves along the input image. The filter task is to multiple its value by the original pixel values. All these multiplications are summed up and one number is obtained at the end. Since the filter has read the image only in the upper left corner it moves further by one unit right performing a similar operation. After passing the filter across all positions, a matrix is obtained, but smaller than a input matrix.

This operation, from a human perspective is analogous to identifying boundaries and simple Colors on the image. But in order to recognize the fish whole network is needed. The network will be -consists of several convolution layers mixed with nonlinear and pooling layers. Convolution is the first layer to extract features from an input image. Convolution features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

* An image matrix of dimension (h x w x d)
* A filter (fh x fw x d)
* Outputs a volume dimension (h-fh+1) x (w-fw+1) x1.

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below



Fig 3.5: Convolution with filter example

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called “Feature Map” as output shown in below

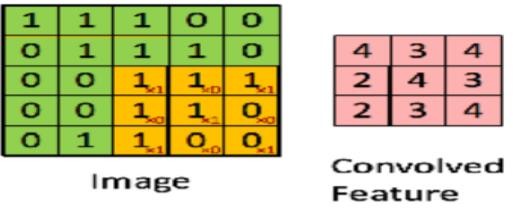


Fig3.6: Output of convolution Network

#### 3.7 THE NON-LINEAR LAYER:

It is adder after each convolution operation. It has the activation function, which brings nonlinear property, without this property a network would not be, sufficiently intense and will not be able to model the response variable.

#### 3.8 THE POOLING LAYER:

It follows the nonlinear layer. It works with width and height of the image and performs a down sampling operation on them. As a result image volume is reduced. This means that if some features already been identified in the previous convolution operation, then a detailed image is no longer needed for further processing and is compressed to less detailed pictures.

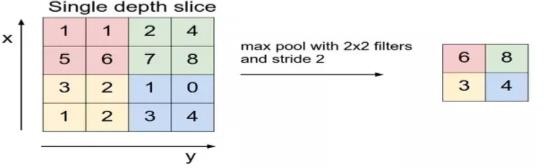


Fig3.7: Max Pooling Layer

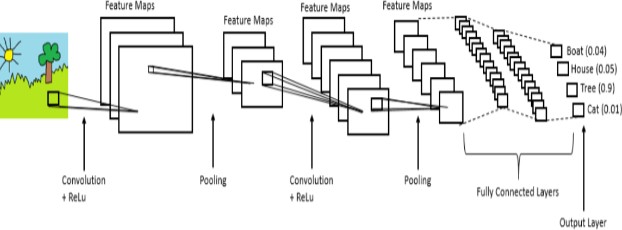


Fig 3.8: Overall structure of Convolution Neural Network (CNN)

#### FULLY CONNECTED LAYER:

After completion of series of convolution, non-linear and pooling layer, it’s necessary to attach a fully connected layer. This layer takes the output information from convolution network. Attaching a fully connected layer to the end of the network results in N dimensional vector, where N is the amount of classes from which the model selects the desired class.

#### CNN MODEL

1. This CNN model is built using the Tensorflow framework and the OpenCv library which is highly used for real-time applications.
2. This model can also be used to develop a full-fledged software to scan every person before they can enter the public gathering.

**CHAPTER 4**

**DESIGN**

#### UML Diagrams:

### CHAPTER 4 DESIGN

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as templated texts.

The kind of the diagram is defined by the primary graphical symbols shown on the diagram. For example, a diagram where the primary symbols in the contents area are classes is class diagram. A diagram which shows use cases and actors is use case diagram. A sequence diagram shows sequence of message exchanges between lifelines.

UML specification does not preclude mixing of different kinds of diagrams,

e.g. to combine structural and behavioral elements to show a state machine nested inside a use case. Consequently, the boundaries between the various kinds of diagrams are not strictly enforced. At the same time, some UML Tools do restrict set of available graphical elements which could be used when working on specific type of diagram.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams.

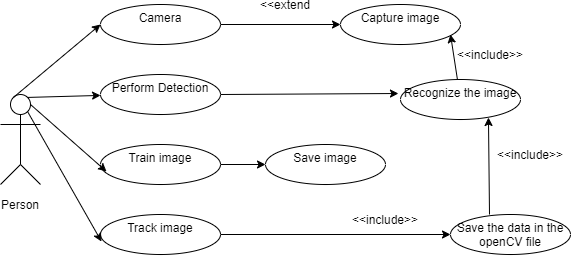
Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

#### Use Case Diagram

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

* + - * Scenarios in which your system or application interacts with people, organizations, or external systems.
      * Goals that your system or application helps those entities (known as actors) achieve.
      * The scope of your system

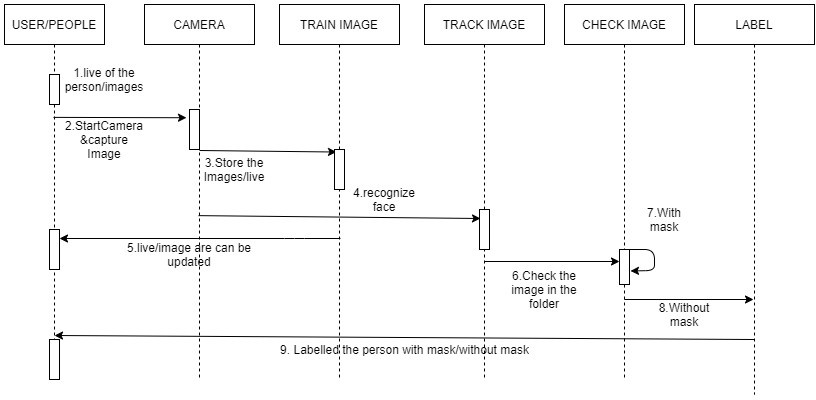


#### Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

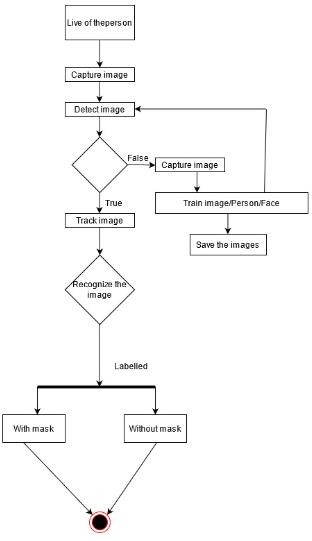
* + - * Represent the details of a UML use case.
      * Model the logic of a sophisticated procedure, function, or operation.
      * See how objects and components interact with each other to complete a process.
      * Plan and understand the detailed functionality of an existing or future scenario.



#### Activity Diagram

An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system.

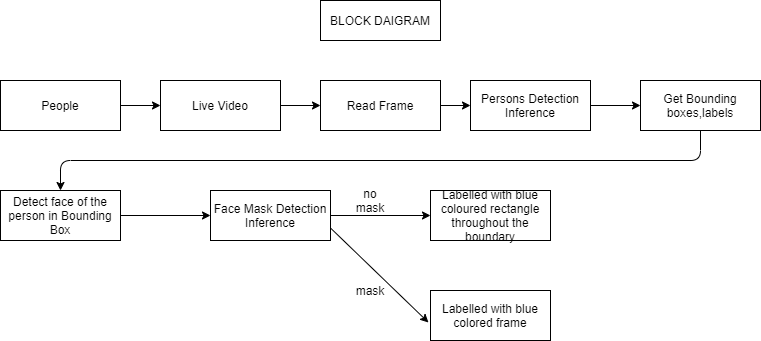
An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



#### BLOCK DIAGRAM

A block diagram is a graphical representation of a system – it provides a functional view of a system. Block diagrams give us a better understanding of a system’s functions and help create interconnections within it.

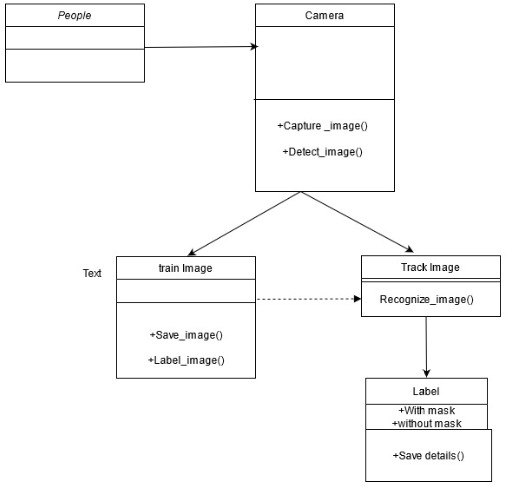
They are used to describe hardware and software systems as well as to represent processes.



#### CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application.

Class diagrams are the only diagrams which can be directly mapped with object- oriented languages and thus widely used at the time of construction.



#### DATA FLOW DIAGRAM

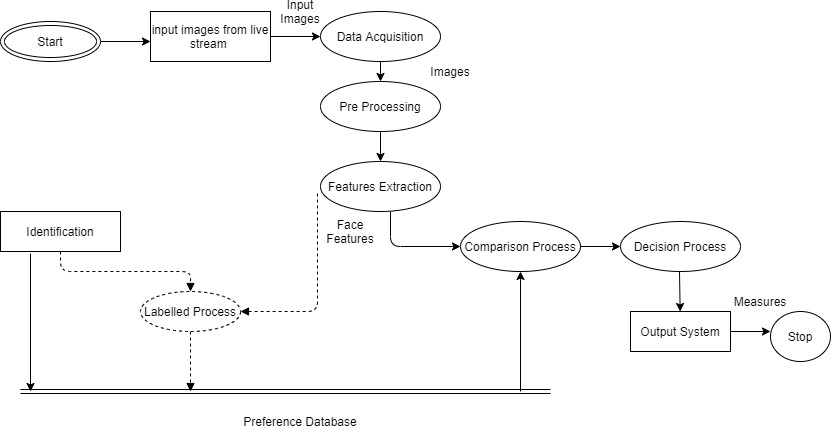
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

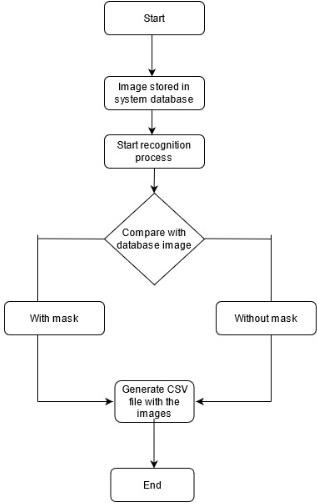
A graphical tool for defining and analyzing minute data with an active or automated system, including process, data stores, and system delays. Data Flow Data is a key and basic tool for the architecture of all other objects. Bubble-bubble or data flow graph is another name for DFD.

DFDs are a model of the proposed system. They should indicate the requirements on which the new system should be built in a clear and direct manner. This is used as a basis for building chart structure plans over time during the design process. The following is the Basic Notation for building DFDs:

1. **Dataflow:**Data flows in a certain direction from the source to the destination.
2. **Process**: People, processes, or technologies that use or produce(Transforming) Information No information about a body part.
3. **Source:** People, programs, organizations, and other things can be external data sources or locations.



#### FLOWCHART DIAGRAM



### CHAPTER 5 EXPERIMENT ANALYSIS

* 1. **MODULES**

1. Creating image datasets a data-loaders for train and test using the experiments folder split
   * Training Dataset: A dataset that we feed into our algorithm to train our model.
   * Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.
2. Training the model
3. Visualizing images

#### DATASET LINK

https://github.com/prajnasb/observations/tree/master/experiements/data

#### CODE IMPLEMENTATION

##### Creating image datasets an data loaders for train and test using the experiments folder split

# 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")

# 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()

# loop over the frames from the video stream

while True:

# grab the frame from the threaded video stream and resize it

# to have a maximum width of 400 pixels

frame = vs.read()

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

# detect faces in the frame and determine if they are wearing a

# face mask or not

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

# loop over the detected face locations and their corresponding

# locations

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

# unpack the bounding box and predictions

(startX, startY, endX, endY) = box

(mask, withoutMask) = pred

# determine the class label and color we'll use to draw

# the bounding box and text

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

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

# include the probability in the label

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

# display the label and bounding box rectangle on the output

# frame

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

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

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

# show the output frame

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop

if key == ord("q"):

break

# do a bit of cleanup

cv2.destroyAllWindows()

vs.stop()

##### Training the model

# import the necessary 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 the initial learning rate, number of epochs to train for,

# and batch size

INIT\_LR = 1e-4

EPOCHS = 20

BS = 32

DIRECTORY = r"D:\face detect\Face-Mask-Detection-master\dataset"

CATEGORIES = ["with\_mask", "without\_mask"]

# grab the list of images in our dataset directory, then initialize

# the list of data (i.e., images) and class images

print("[INFO] loading images...")

data = []

labels = []

for category in CATEGORIES:

path = os.path.join(DIRECTORY, category)

for img in os.listdir(path):

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

image = load\_img(img\_path, target\_size=(224, 224))

image = img\_to\_array(image)

image = preprocess\_input(image)

data.append(image)

labels.append(category)

# perform one-hot encoding on the labels

lb = LabelBinarizer()

labels = lb.fit\_transform(labels)

labels = to\_categorical(labels)

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

labels = np.array(labels)

(trainX, testX, trainY, testY) = train\_test\_split(data, labels,

test\_size=0.20, stratify=labels, random\_state=42)

# construct the training image generator for data augmentation

aug = ImageDataGenerator(

rotation\_range=20,

zoom\_range=0.15,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.15,

horizontal\_flip=True,

fill\_mode="nearest")

# load the MobileNetV2 network, ensuring the head FC layer sets are

# left off

baseModel = MobileNetV2(weights="imagenet", include\_top=False,

input\_tensor=Input(shape=(224, 224, 3)))

# construct the head of the model that will be placed on top of the

# the base model

headModel = baseModel.output

headModel = AveragePooling2D(pool\_size=(7, 7))(headModel)

headModel = Flatten(name="flatten")(headModel)

headModel = Dense(128, activation="relu")(headModel)

headModel = Dropout(0.5)(headModel)

headModel = Dense(2, activation="softmax")(headModel)

# place the head FC model on top of the base model (this will become

# the actual model we will train)

model = Model(inputs=baseModel.input, outputs=headModel)

# loop over all layers in the base model and freeze them so they will

# \*not\* be updated during the first training process

for layer in baseModel.layers:

layer.trainable = False

# compile our model

print("[INFO] compiling model...")

opt = Adam(lr=INIT\_LR, decay=INIT\_LR / EPOCHS)

model.compile(loss="binary\_crossentropy", optimizer=opt,

metrics=["accuracy"])

# train the head of the network

print("[INFO] training head...")

H = model.fit(

aug.flow(trainX, trainY, batch\_size=BS),

steps\_per\_epoch=len(trainX) // BS,

validation\_data=(testX, testY),

validation\_steps=len(testX) // BS,

epochs=EPOCHS)

# make predictions on the testing set

print("[INFO] evaluating network...")

predIdxs = model.predict(testX, batch\_size=BS)

# for each image in the testing set we need to find the index of the

# label with corresponding largest predicted probability

predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report

print(classification\_report(testY.argmax(axis=1), predIdxs,

target\_names=lb.classes\_))

# serialize the model to disk

print("[INFO] saving mask detector model...")

model.save("mask\_detector.model", save\_format="h5")

# plot the training loss and accuracy

N = EPOCHS

plt.style.use("ggplot")

plt.figure()

plt.plot(np.arange(0, N), H.history["loss"], label="train\_loss")

plt.plot(np.arange(0, N), H.history["val\_loss"], label="val\_loss")

plt.plot(np.arange(0, N), H.history["accuracy"], label="train\_acc")

plt.plot(np.arange(0, N), H.history["val\_accuracy"], label="val\_acc")

plt.title("Training Loss and Accuracy")

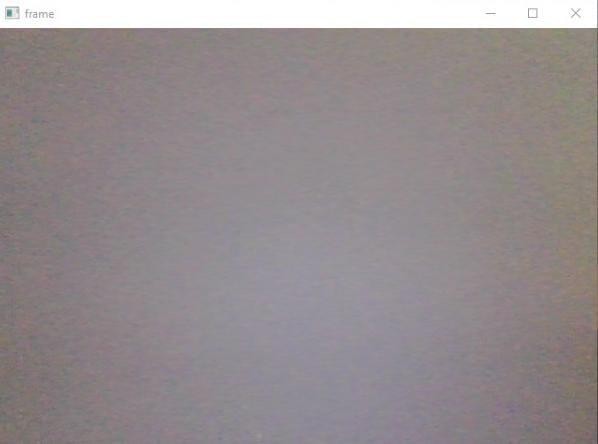
plt.xlabel("Epoch #")

plt.ylabel("Loss/Accuracy")

plt.legend(loc="lower left")

plt.savefig("plot.png")

##### LIVE WEB IMAGE:



Face Mask Detection in webcam stream:

The flow to identify the person in the webcam wearing the face mask or not. The process is two-fold.

* + - 1. To identify the faces in the webcam 2.Classify the faces based on the mask.

Identify the Face in the Webcam:

To identify the faces a pre-trained model provided by the OpenCV framework was used.

The model was trained using web images. OpenCV provides 2 models for this face detector.

#### FUNCTIONAL REQUIREMENTS

The primary purpose of computer results is to deliver processing results to users. They are also employed to maintain a permanent record of the results for future use.

In general, the following are many types of results:

External results are those that are exported outside the company.

* Internal results, which are the main user and computer display and have a place within the organization.
* Operating results used only by the computer department.
* User-interface results that allow the user to communicate directly with the system.
* Understanding the user's preferences, the level of technology and the needs of his or her business through a friendly questionnaire.

#### NON-FUNCTIONAL REQUIREMENTS

#### SYSTEM CONFIGURATION

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

**CHAPTER 6:**

**SOFTWARE REQUIREMENT**

**HARDWARE**

Software:

* Python idle :3.9
* Anaconda navigator 3.3
* Adas lane vehicle

Hardware:

RASPBERRY PI

IR SENSOR

DHT11

SERVO MOTOR

CAMERA

**Raspberry Pi**

Raspberry Pi is a small single-board Computer developed in UK by Raspberry Pi foundation to promote the teaching of computer science in schools and in developing countries.

Original model become far more popular than anticipated sealing outside of its target market, for uses such as robots.

**History**

Raspberry Pi has mainly three generations Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3 and also a reduce simple inexpensive Raspberry Pi zero.

* The first model of Raspberry Pi was launched in February 2012 i.e. Raspberry Pi 1 Model B followed by a simple inexpensive Model A.
* In April 2014 “Compute Model” for embedded application Raspberry Pi 1 model B+ improved versions of A and B was launched.
* In November 2015 with reduced I/O and GPIO Raspberry Pi zero came into market.
* In February 20-15 advance model with 40 GPIO pins, Ethernet, 4 USB slots Raspberry Pi 2 was launched.
* In February 2016 an upgraded model with inbuilt Bluetooth and Wi-Fi Raspberry Pi 3 Model B was launched.
* Recently in February 2017, “Raspberry Pi – Zero W” with in built Wi-Fi and Bluetooth come into the market.

**Features**

The heart of the Raspberry Pi is a Broadcom System on Chip (SOC) which includes ARM compatible CPU and on-chip graphic processing unit and Vediocore IV.

The key feature from First generation to the Third generation includes:

* CPU speed ranges from 700 MHz to 1.2 GHz.
* On board Memory (RAM) ranges from 256 MB to 1 GB.
* USB slot differs from 1 slot to USB slots.
* HDMI, composite video output and 3.5mm phone jack.
* Low level output is provided by GPIO pins which support common. protocols like I2C (inter-integrated circuit).
* Ethernet 8 Position 8 Contact (8P8C).

**Processor**

The processor at the heart of the Raspberry Pi is a Broadcom BCM28XX.

This is the Broadcom System on Chip (SOC) chip use in the Raspberry Pi. The processor from first to third generations include:

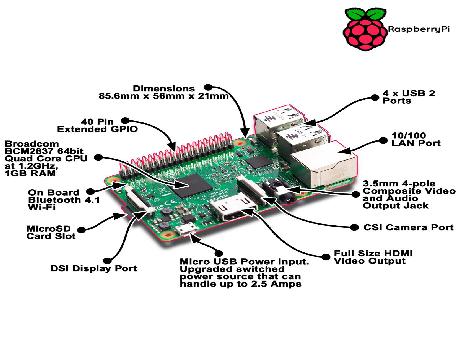
* Raspberry Pi 1: Broadcom BCM2835 SOC with 700MHz CPU speed, L2 cache of 128kb with ARM compatibility AR1176JZF-S (ARMv6) 32-bit RISC ARM.
* Raspberry Pi 2: Broadcom BCM 2836 SOC with 900MHz CPU speed, L2 cache of 256kb with 32-bit quad-core ARM cortex-A7 (ARMv7).
* Raspberry Pi 3: Broadcom BCM2837 SOC with 1.2GHz 64-bit quad-core –A53 with 512 kb shared L2 cache (64-bit instruction set ARMv8).

**Raspberry Pi 3**

In this project we are using Latest version of Raspberry i.e. Raspberry Pi 3The processor at the heart of the Raspberry Pi 3 is a Broadcom BCM2837, and the later models of the Raspberry Pi 2. The underlying architecture of the BCM2837 is identical to the BCM2836. The Only significant difference is the replacement of the ARMv7 quad core cluster with quad-core ARM Cortex A53 (ARMv8) cluster.

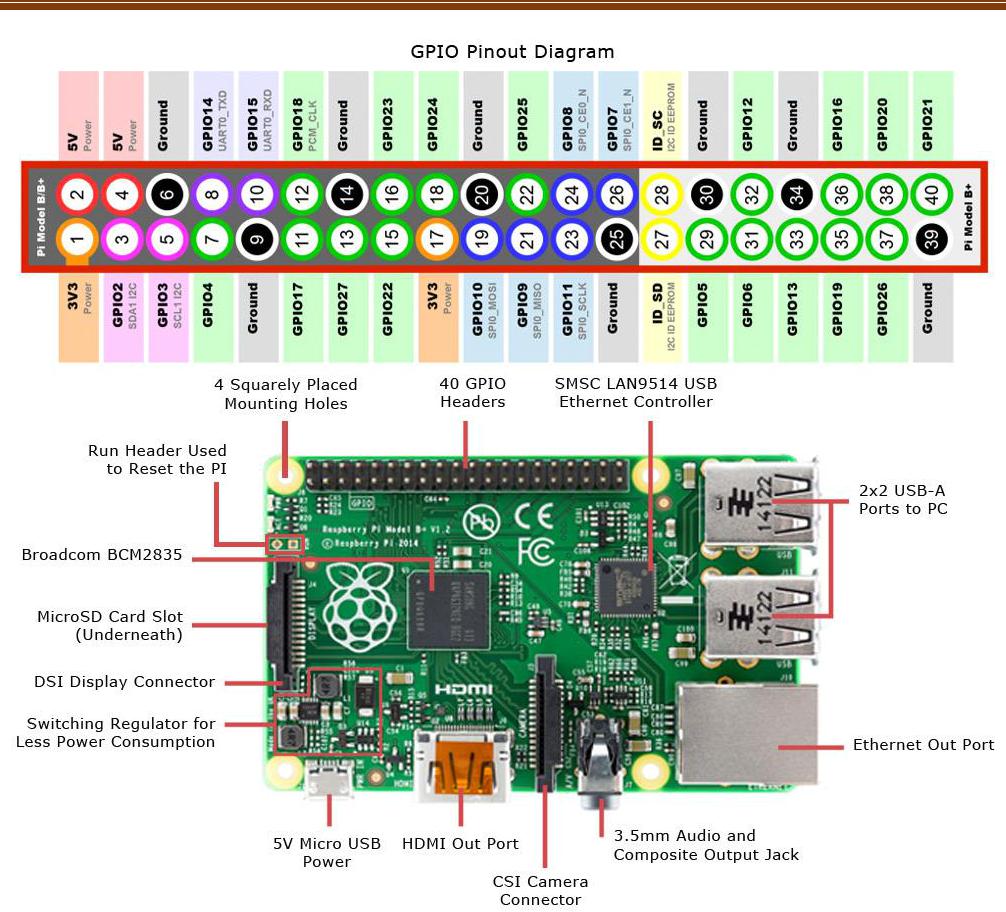
The ARM cores run at 1.2GHz, making the device about 50% faster than the Raspberry Pi 2 with a on board memory of 1GB RAM. The videocoreIV runs at 400MHz.

**GPIO: Raspberry Pi 3**



**Fig. 3.1 Raspberry Pi 3**

One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the top edge of the board. These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on and off (output).Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins).



**Note:** When programming the GPIO pins there are two different ways to refer tothem. GPIO numbering and physical numbering.

**GPIO NUMBERING**

This are the GPIO pins as the computer sees them. The numbers don’t make any sense to humans, they jump about all over the place, so there is no easy way to remember them. We suggest you to take a printed reference.

**PHYSICAL NUMBERING**

The other way to refer to the pins is by simply counting across and down from pin 1 at the top left (nearest to the SD card). This is ‘physical numbering’ and it looks like the Fig. 3.2.

**WHICH SYSTEM SHOULD I USE?**

Beginners and young children may find the physical numbering system simpler, you just have to count the pins. Generally we recommend using the GPIO numbering. Its good practice and most resources use them.

## cameras

[Smart cameras](https://www.safewise.com/resources/home-security-cameras/) take the traditional security camera concept and turn the dial up to eleven. These cameras incorporate Wi-Fi connections that allow them to be viewed and controlled remotely using a smartphone app. Many can also upload the footage to the Cloud for storage, eliminating the need for a potentially complicated on-site storage setup.

There are a lot of options in this space. Choosing one starts with knowing whether you want indoor or outdoor cameras. The main difference is construction: Outdoor cameras need to survive the elements, which usually means they are bulkier and made from tougher materials like metal. Indoor cameras can be smaller and more discreet, because they aren’t as likely to get beaten up or tampered with.

In either case, features you’ll want to look for include high resolution, a wide field of view, motion detection, and audio recording. Normally we would include Wi-Fi connectivity as an essential feature, but because we’re talking about smart cameras, Wi-Fi is a given.

Some recommendations for great indoor cameras are the Nest Cam and [Vimtag 361 HD](https://www.amazon.com/Fujikam-Wireless-Monitoring-Surveillance-security/dp/B00JAJ9U8K/ref=as_li_ss_tl/?ie=UTF8&qid=1456365109&sr=8-1&keywords=Vimtag%2BVT-361%2BHD&linkCode=ll1&tag=safewicom-20&linkId=8acf5fd59efbbd8e08bf665defa3c415&kbid=62750" \t "_blank). For outdoor cameras, check out the [Amcrest QCAM](https://www.amazon.com/dp/B01BN1FD7E/ref=as_li_ss_tl/?ie=UTF8&linkCode=ll1&tag=safewicom-20&linkId=ad9d8b4cdd6fb3bb920b533fc107de65&kbid=62750" \t "_blank) and the [ZOSI 960H](https://www.amazon.com/ZOSI-1000TVL-120Feet-outdoor-Security/dp/B00J21DFGE/ref=as_li_ss_tl/?s=hi&ie=UTF8&qid=1463080324&sr=1-1&keywords=ZOSI%2B960H&linkCode=ll1&tag=safewicom-20&linkId=764f0c15eba08c4dda08f7c03e85403a&kbid=62750). Any of these options will serve you well.

**SD-CARD**:

SD cards are commonly used in **digital cameras, baby monitors, or handheld computers**. Because it is flash memory, it can be used to store files, similar to a USB flash drive. However, you need to use a card reader to read or write to the SD card on a regular desktop.



**Buzzer**

A buzzer or beeper is an [audio](https://en.wikipedia.org/wiki/Sound) signalling device, which may be [mechanical](https://en.wikipedia.org/wiki/Machine), [electromechanical](https://en.wikipedia.org/wiki/Electromechanics), or [piezoelectric](https://en.wikipedia.org/wiki/Piezoelectricity) (piezo for short). Typical uses of buzzers and beepers include [alarm devices](https://en.wikipedia.org/wiki/Alarm_devices), [timers](https://en.wikipedia.org/wiki/Timer), and confirmation of user input such as a mouse click or keystroke



**How to use a Buzzer**

A buzzeris a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customised with help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

## Types

**Electromechanical**

Early devices were based on an electromechanical system identical to an [electric bell](https://en.wikipedia.org/wiki/Electric_bell) without the metal gong. Similarly, a [relay](https://en.wikipedia.org/wiki/Relay) may be connected to interrupt its own actuating [current](https://en.wikipedia.org/wiki/Electric_current), causing the [contacts](https://en.wikipedia.org/wiki/Switch) to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

**Mechanical**

A [joy buzzer](https://en.wikipedia.org/wiki/Joy_buzzer) is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

**Piezoelectric**



**Piezoelectric disk beeper**

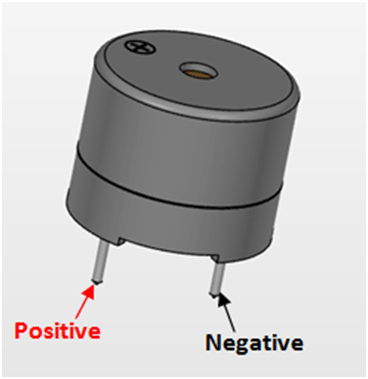
A [piezoelectric](https://en.wikipedia.org/wiki/Piezoelectric) element may be driven by an [oscillating](https://en.wikipedia.org/wiki/Oscillation) electronic circuit or other [audio signal](https://en.wikipedia.org/wiki/Audio_signal) source, driven with a [piezoelectric audio amplifier](https://en.wikipedia.org/wiki/Piezoelectric_audio_amplifier). Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.



Interior of a readymade loudspeaker, showing a piezoelectric-disk-beeper (With 3 electrodes ... including 1 feedback-electrode (the central, small electrode joined with red wire in this photo), and an oscillator to self-drive the buzzer.

A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or [Helmholtz resonance](https://en.wikipedia.org/wiki/Helmholtz_resonance) to produce an audible beep

### Buzzer Pin Configuration



|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Positive | Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC |
| 2 | Negative | Identified by short terminal lead. Typically connected to the ground of the circuit |

### Specifications

Rated Voltage: 6V DC

Operating Voltage: 4-8V DC

Rated current: <30mA

Sound Type: Continuous Beep

Resonant Frequency: ~2300 Hz

Small and neat sealed package

Breadboard and Perf board friendly

**Application**

Alarming Circuits, where the user has to be alarmed about something

Communication equipments

Automobile electronics

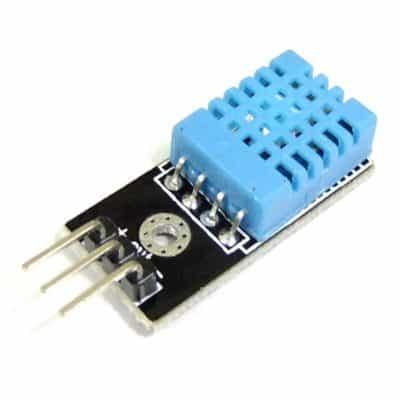
Portable equipments, due to its compact size

**DHT-11**

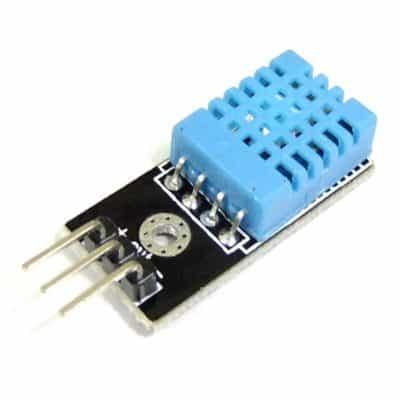
DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor.  To measure the surrounding air this sensor uses a [thermistor](https://www.elprocus.com/introduction-to-thermistor-types-with-its-workings-and-applications/) and a capacitive humidity sensor.

1. Dht11:

This sensor is used here to monitor the humidity variation of the environment where the crops are cultivated. This is a digital sensor and measures the humidity value in percentage format.



**DHT11 Sensor**



DHT11 is a low-cost digital sensor for sensing temperature and humidity.  This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously.

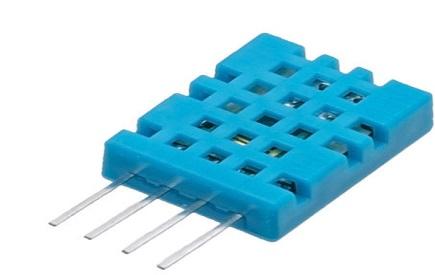
**Working**

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature.  The humidity sensing [capacitor](https://www.elprocus.com/construction-of-capacitor-with-working/) has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

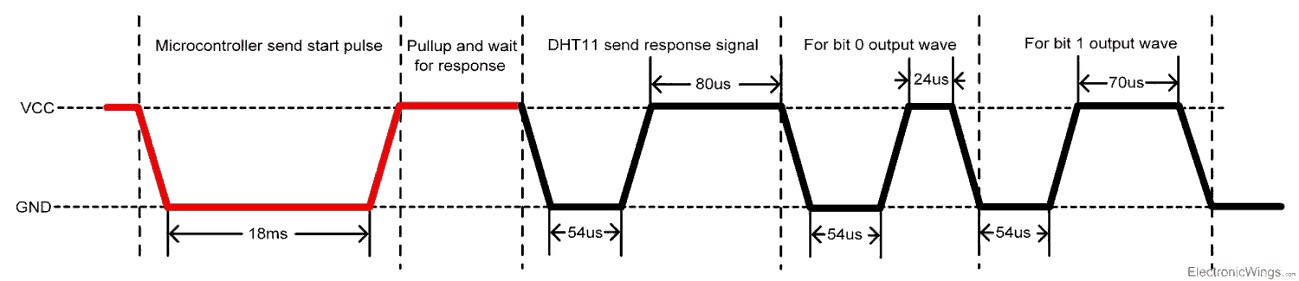
The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second.  DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.



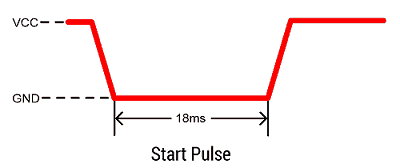
**HUMIDITY (DHT11):**

* DHT11 uses only one wire for communication. The voltage levels with certain time value defines the logic one or logic zero on this pin.
* The communication process is divided in three steps, first is to send request to DHT11 sensor then sensor will send response pulse and then it starts sending data of total 40 bits to the microcontroller.



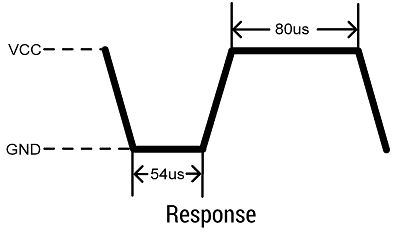
**Communication process**

Start pulse (Request)



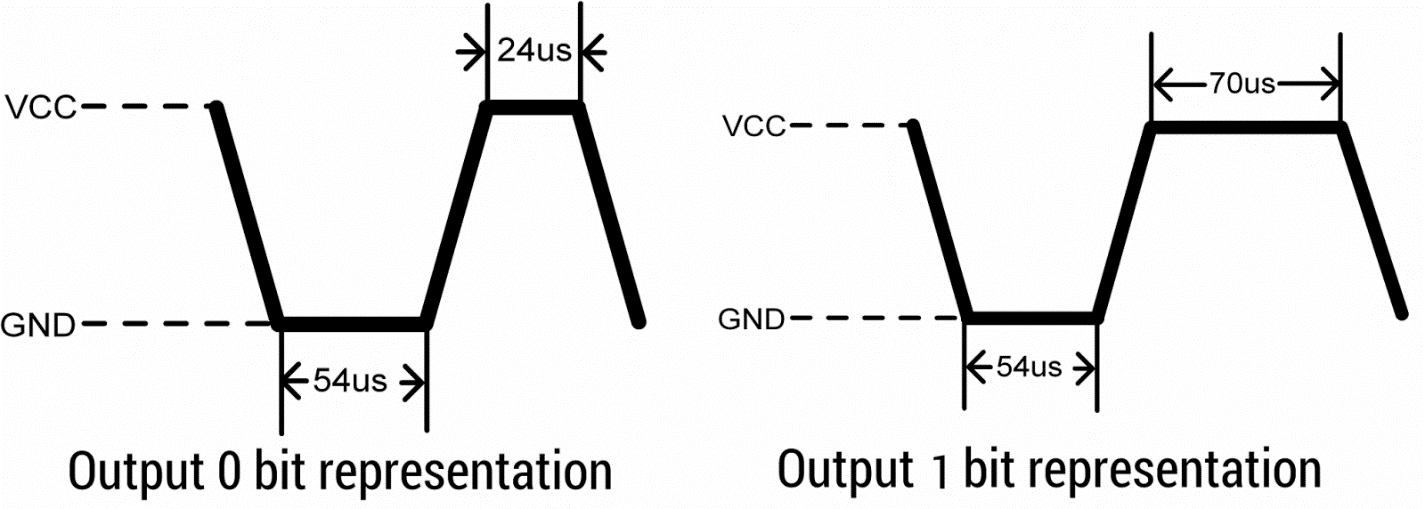
* To start communication with DHT11, first we should send the start pulse to the DHT11 sensor.
* To provide start pulse, pull down (low) the data pin minimum 18ms and then pull up, as shown in diag.

**Response**



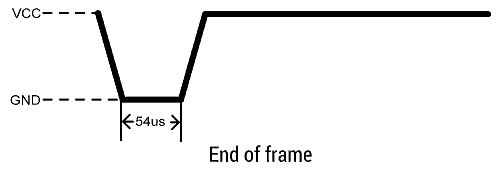
* After getting start pulse from, DHT11 sensor sends the response pulse which indicates that DHT11 received start pulse.
* The response pulse is low for 54us and then goes high for 80us.

**Data**



* After sending the response pulse, DHT11 sensor sends the data, which contains humidity and temperature value along with checksum.
* The data frame is of total 40 bits long, it contains 5 segments (byte) and each segment is 8-bit long.
* In these 5 segments, first two segments contain humidity value in decimal integer form. This value gives us Relative Percentage Humidity. 1st 8-bits are integer part and next 8 bits are fractional part.
* Next two segments contain temperature value in decimal integer form. This value gives us temperature in Celsius form.
* Last segment is the checksum which holds checksum of first four segments.
* Here checksum byte is direct addition of humidity and temperature value. And we can verify it, whether it is same as checksum value or not. If it is not equal, then there is some error in the received data.
* Once data received, DHT11 pin goes in low power consumption mode till next start pulse.

**End of frame**



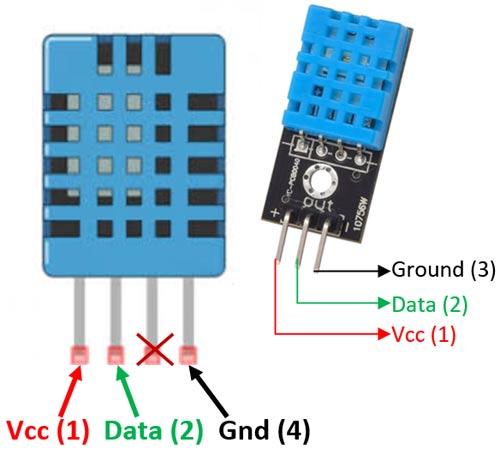
* After sending 40-bit data, DHT11 sensor sends 54us low level and then goes high. After this DHT11 goes in sleep mode.

**DHT11 vs DHT22**

Two versions of the DHT sensor, they look a bit similar and have the same pinout, but have different characteristics and specifications:

[**DHT11**](http://www.adafruit.com/products/386)

* Ultra-low cost
* 3 to 5V power and I/O
* 2.5mA max current use during conversion (while requesting data)
* Good for 20-80% humidity readings with 5% accuracy
* Good for 0-50°C temperature readings ±2°C accuracy
* No more than 1 Hz sampling rate (once every second)
* Body size 15.5mm x 12mm x 5.5mm
* 4 pins with 0.1" spacin

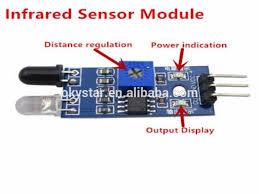


## Applications

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions.  The humidity[sensor](https://en.wikipedia.org/wiki/Humidity) is used as a preventive measure in homes where people are affected by humidity.  Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure.

**IR (Infrared Sensor)**

IR sensor is a simple electronic device which emits and detects IR radiation in order to find out certain objects/obstacles in its range. Some of its features are heat and motion sensing.



IR sensors use infrared radiation of wavelength between 0.75 to 1000µm which falls between visible and microwave regions of electromagnetic spectrum. IR region is not visible to human eyes. Infrared spectrum is categorized into three regions based on its wavelength i.e. Near Infrared, Mid Infrared, Far Infrared.

**Wavelength Regions of Infrared Spectrum**

* Near IR – 0.75µm to 3 µm
* Mid IR – 3 µm to 6 µm
* Far IR –   > 6 µm

## Working Principle of Infrared Sensor

InfraredSensors works on three fundamental Physics laws:

### IR Transmitter

IR Transmitter acts as source for IR radiation. According to Plank’s Radiation Law, every object is a source of IR radiation at temp T above 0 Kelvin. In most cases black body radiators,  tungsten lamps, silicon carbide, infrared lasers, LEDs of infrared wavelength are used as sources.

### Transmission Medium

As the name suggests, Transmission Medium provides passage for the radiation to reach from IR Transmitter to IR Receiver. Vacuum, atmosphere and optical fibers are used as medium.

### IR receiver

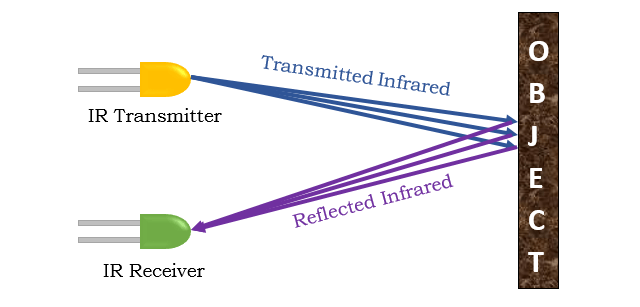
Generally IR receivers are photo diode and photo transistors. They are capable of detecting infrared radiation. Hence IR receiver is also called as IR detector. Variety of receivers are available based on wavelength, voltage and package.

IR Transmitter and Receivers are selected with matching parameters. Some of deciding specifications of receivers are photosensitivity or responsivity, noise equivalent power and detectivity.

## How Infrared Sensor Works

An Infrared Sensor works in the following sequence:

* IR source (transmitter) is used to emit radiation of required wavelength.
* This radiation reaches the object and is reflected back.
* The reflected radiation is detected by the IR receiver.
* The IR Receiver detected radiation is then further processed based on its intensity. Generally, IR Receiver output is small and amplifiers are used to amplify the detected signal.



Incidence in an IR Detection System may be direct or indirect. In case of Direct Incidence, there is no hindrance in between transmitter and receiver. Whereas, in Indirect Incidence IR Transmitter and Receiver are kept side by side and the object is in front of them.

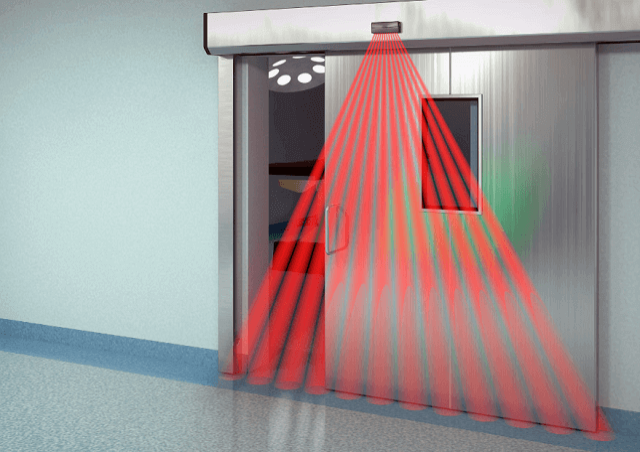
**Types of Infrared Sensor**

IR sensors can be classified in two types based on presence of IR source:

* Active Infrared Sensor
* Passive Infrared Sensor

**Active Infrared Sensor**

Active Infrared Sensor contains both transmitter and receiver. Most of the cases LED or laser diode is used as source. LED for non-imaging IR sensor and laser diode for imaging IR sensor are used.



Active IR Sensor works by radiating energy, received and detected by detector and further processed by signal processor in order to fetch information required.

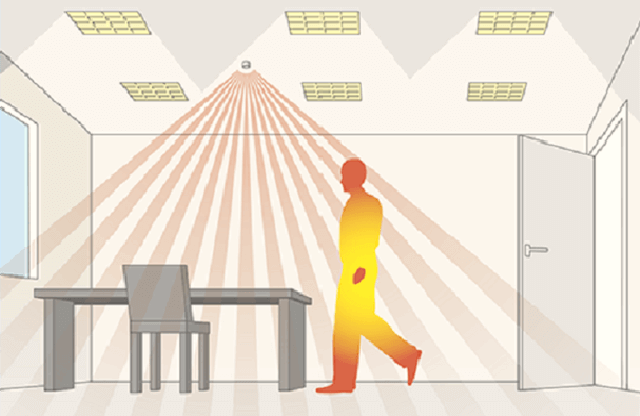
Examples of Active IR Sensor: Break Beam Sensor, Reflectance Sensor.

### Passive Infrared Sensor

Passive Infrared Sensor contains detectors alone. There won’t be a transmitter component.

These type of sensors use object as IR source/ transmitter. Object radiates energy and it is detected by IR receivers. A Signal processor is then used to interpret the signal to fetch information required.

Example of Passive IR Sensor: Thermocouple-Thermopile, Bolometer, Pyro-Electric Detector, etc.



There are two types of Passive Infrared Sensor:

* Thermal Infrared Sensor
* Quantum Infrared Sensor

#### Thermal Infrared Sensor

Thermal Infrared sensors are independent of wavelength. They use heat as energy source.

Thermal detectors are slow with their detection time and response time.



#### Quantum Infrared Sensor

Quantum Infrared Sensor are dependent on wavelengths. They have high detection time and response time. These type of IR sensors require frequent cooling for precise measurement.

## Applications of Infrared Sensor

IR sensors have found their applications in most of today’s equipment. Following are the list of sensors which are named after its usage.

### Proximity Sensor

These are used in smart phones to find distance of object. They use principle called Reflective Indirect Incidence. Radiation transmitted by transmitter is received by receiver after being reflected from object. Distance is calculated based on the intensity of radiation received.

### Item Counter

This use direct incidence method to count the items. Constant radiation is maintained in between transmitter and receiver. As soon as object cuts the radiation, item is detected and count is increased. The same count is shown on display system.

### Burglar Alarm

This is one of widely and commonly used sensor application. It is another example for direct incidence method.

It works similar to item counter, where transmitter and receiver are kept on both the sides of door frame. Constant radiation is maintained between transmitter and receiver, whenever object crosses path alarm starts off.

**Radiation Thermometers**

It is one of key application of Infrared sensors. Working of radiation thermometer depends on temperature and type of object.

These have faster response and easy pattern measurements. They can do measurement without direct contact of object.

**Human Body Detection**

This method is used in intrusion detection, auto light switches, etc. Intrusion alarm system sense temperature of human body.

If the temperature is more than threshold value, it sets on the alarms. It uses electromagnetic system which is suitable for human body in order to protect it from unwanted harmful radiations.

**Gas Analyzers**

Gas Analyzersare used to measure gas density by using absorption properties of gas in IR region. Dispersive and Non Dispersive types of gas analyzers are available.

**Other Applications**

IR sensors are also used in IR imaging devices, optical power meters, sorting devices, missile guidance, remote sensing, flame monitors, moisture analyzers, night vision devices, infrared astronomy, rail safety, etc.

### IR Sensor Module Features

* 5VDC Operating voltage
* I/O pins are 5V and 3.3V compliant
* Range: Up to 20cm
* Adjustable Sensing range
* Built-in Ambient Light Sensor
* 20mA supply current
* Mounting hole

**Advantages of Infrared Sensor**

The advantages of Infrared Sensor are:

* Their low power requirements make them suitable for most electronic devices such as laptops, telephones, PDAs.
* They are capable of detecting motion in presence/ absence of light almost with same reliability.
* They do not require contact with object to for detection.
* There is no leakage of data due to beam directionality IR radiation.
* They are not affected by corrosion or oxidation.
* They have very strong noise immunity.

**Disadvantages of Infrared Sensor**

The disadvantages of Infrared Sensor are:

* Required Line of sight.
* Get blocked by common objects.
* Limited range.
* Can be affected by Environmental conditions such as rain, fog, dust, pollution.
* Transmission Data rate is slow.

**Applications**

* Obstacle Detection
* Industrial safety devices
* Wheel encode

**Servo motor**

**Abstract:**

The servo motor is most commonly used for high technology devices in the industrial applications like automation technology. It is a self-contained electrical device, that rotates parts of machine with high efficiency and great precision. Moreover the output shaft of this motor can be moved to a particular angle. Servo motors are mainly used in home electronics, toys, cars, airplanes and many more devices.

Thus this blog discusses the definition, types, mechanism, principle, working, controlling, and lastly the applications of a servo machine.

**Definition:**

A servo motor is a rotary actuator or a motor that allows for a precise control in terms of the angular position, acceleration, and velocity. Basically it has certain capabilities that a regular motor does not have. Consequently it makes use of a regular motor and pairs it with a sensor for position feedback.



**Principle of working:**

Servo motor works on the PWM (Pulse Width Modulation) principle, which means its angle of rotation, is controlled by the duration of pulse applied to its control PIN. Basically servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears.

**Mechanism of servomotor:**

Basically a servo motor is a closed-loop servomechanism that uses position feedback to control its motion and final position. Moreover the input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The motor is incorporates some type of encoder to provide position and speed feedback. In the simplest case, we measure only the position. Then the measured position of the output is compared with the command position, the external input to controller. Now if the output position differs from that of the expected output, an error signal generates. This then causes the motor to rotate in either direction, as per need to bring the output shaft to the appropriate position. As the position approaches, the error signal reduces to zero. Finally the motor stops.

The very simple servomotors can position only sensing via a potentiometer and bang-bang control of their motor. Further the motor always rotates at full speed. Though this type of servomotor doesn’t have many uses in industrial motion control, however it forms the basis of simple and cheap servo used for radio control models.

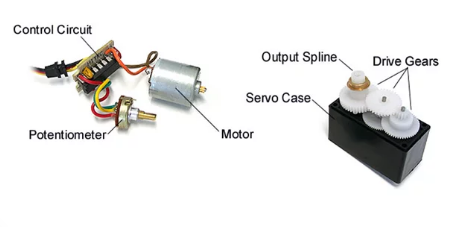
Servomotors also find uses in optical rotary encoders to measure the speed of output shaft and a variable-speed drive to control the motor speed. Now this, when combined with a PID control algorithm further allows the servomotor to be in its command position more quickly and more precisely with less overshooting

**Working of servomotors**

Servo motors control position and speed very precisely. Now a potentiometer can sense the mechanical position of the shaft. Hence it couples with the motor shaft through gears. The current position of the shaft is converted into electrical signal by potentiometer, and is compared with the command input signal. In modern servo motors, electronic encoders or sensors sense the position of the shaft

We give command input according to the position of shaft. If the feedback signal differs from the given input, an error signal alerts the user. We amplify this error signal and apply as the input to the motor, hence the motor rotates. And when the shaft reaches to the require position error signal become zero , and hence the motor stays standstill holding the position.

The command input is in form of electrical pulses as the actual input to the motor is the difference between feedback signal (current position) and required signal, hence speed of the motor is proportional to the difference between the current position and required position. The amount of power require by the motor is proportional to the distance it needs to travel.

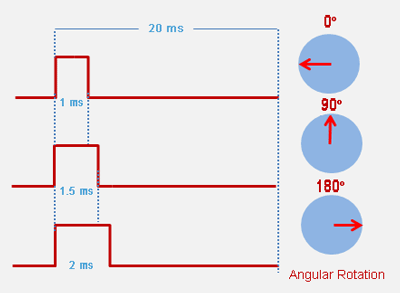


### Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on required angle.



Servo motor can be rotated from 0 to 180 degree, but it can go up to 210 degree, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. Pulse of 1 ms (1 millisecond) width can rotate servo to 0 degree, 1.5ms can rotate to 90 degree (neutral position) and 2 ms pulse can rotate it to 180 degree.

All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume, if you are planning to use more than two servo motors a proper servo shield should be designed.

**Applications:**

1. Robotics: At every joint of the robot, we connect a servomotor. Thus giving the robot arm its precise angle.

2. Conveyor belts: servo motors move, stop, and start conveyor belts carrying product along to various stages, for example, in product packaging/ bottling, and labeling.

3. Camera auto focus: A highly precise servo motor build into the camera corrects a camera lens to sharpen out of focus images.

4. Solar tracking system: Servo motors adjust the angle of solar panels throughout the day and hence each panel continues to face the sun which results in harnessing maximum energy from sunup to sundown.

**Advantages:**

* If a heavy load is placed on the motor, the driver will increase the current to the motor coil as it attempts to rotate the motor. Basically, there is no out-of-step condition.
* High-speed operation is possible.

**Disadvantages:**

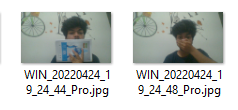
* Since the servomotor tries to rotate according to the command pulses, but lags behind, it is not suitable for precision control of rotation.
* Higher cost.
* When stopped, the motor’s rotor continues to move back and forth one pulse, so that it is not suitable if you need to prevent vibration

#### INPUT AND OUTPUT

INPUT DATASET

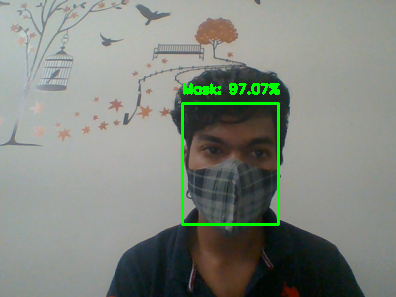


* **With mask data set**

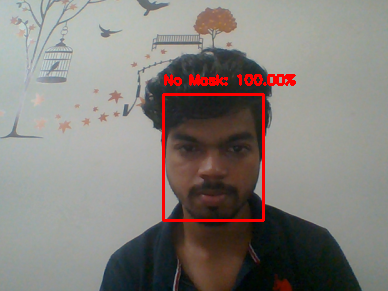
**** ****

* **Without mask data set**

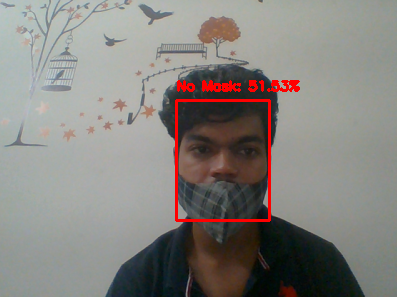
**REAL TIME OUTPUT**



Output with mask message and its prediction rate of percentage of wearing a mask

****

Output with no mask message and its prediction rate of percentage of no mask



Output with improper mask message and its prediction rate of percentage of improper mask

**CHAPTER 6** **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. The architecture consists of Mobile Net as the backbone it can be used for high and low computation scenarios. 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. We used OpenCV, tensor flow, and NN 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

**FUTURE ENHANCEMENT**

Jingdong's recognition accuracy is stronger than 99 percent. We created the MFDD, RMFRD, and SMFRD datasets, as well as a cutting-edge algorithm based on them. The algorithm will provide contactless face authentication in settings such as community access, campus governance, and enterprise resumption. Our research has given the world more scientific and technological strength.

**References**

1. **Xinqi Fan, Mingjie Jiang and Hong Yan, *A Deep Learning Based Light Weight Face Mask Detector with Residual Context Attention And Gaussian Heat Map To Fight Against Covid-19,* IEEE Access, Volume 9, pp. 96964 – 96974, 14, July 2021.**
2. J. D. Arias-Londono, J. A. Gomez-Garcia, L. Moro-Velazquez, and J. I. Godino-Llorente, ‘‘Artificial intelligence applied to chest X-ray images for the automatic detection of COVID-19. A thoughtful evaluation approach,’’ IEEE Access, vol. 8, pp. 226811–226827, 2020.
3. S. Zhao, Q. Lin, J. Ran, S. S. Musa, G. Yang, W. Wang, Y. Lou, D. Gao, L. Yang, D. He, and M. H. Wang, ‘‘Preliminary estimation of the basic reproduction number of novel coronavirus (2019-nCoV) in China, from 2019 to 2020: A data-driven analysis in the early phase of the outbreak,’’ Int. J. Infectious Diseases, vol. 92, pp. 214–217, Mar. 2020
4. Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklöv, ‘‘The reproductive number of COVID-19 is higher compared to SARS coronavirus,’’ J. Travel Med., vol. 27, no. 2, pp. 1–4, Mar. 2020.
5. S. Hu, Y. Gao, Z. Niu, Y. Jiang, L. Li, X. Xiao, M. Wang, E. F. Fang, W. Menpes-Smith, J. Xia, H. Ye, and G. Yang, ‘‘Weakly supervised deep learning for COVID-19 infection detection and classification from CT images,’’ IEEE Access, vol. 8, pp. 118869–118883, 2020
6. M. Loey, F. Smarandache, and N. E. M. Khalifa, ‘‘Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning,’’ Symmetry, vol. 12, no. 4, p. 651, Apr. 2020
7. N. H. L. Leung, D. K. W. Chu, E. Y. C. Shiu, K.-H. Chan, J. J. McDevitt, B. J. P. Hau, H.-L. Yen, Y. Li, D. K. M. Ip, J. S. M. Peiris, W.-H. Seto, G. M. Leung, D. K. Milton, and B. J. Cowling, ‘‘Respiratory virus shedding in exhaled breath and efficacy of face masks,’’ Nature Med., vol. 26, no. 5, pp. 676–680, May 2020.
8. M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, ‘‘Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection,’’ Sustain. Cities Soc., vol. 65, Feb. 2021, Art. no. 102600
9. 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, Jan. 2021, Art. no. 108288.
10. S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, “Rational use of face masks in the COVID19 pandemic,” The Lancet Respiratory Medicine, vol. 8, no. 5, pp. 434–436, May 2020
11. SHEETAL RAJPAL , MANOJ AGARWAL , VIRENDRA KUMAR , ANAMIKA GUPTA , AND NAVEEN KUMAR ” Triphasic DeepBRCA-A Deep Learning-Based Framework for Identification of Biomarkers for Breast Cancer Stratification”   
    Year: 2021 | [Volume: 9](https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=9312710) | Journal Article | Publisher: IEEE
12. Y. Chen, M. Hu, C. Hua, G. Zhai, J. Zhang, Q. Li, and S. X. Yang, ‘‘Face mask assistant: Detection of face mask service stage based on mobile phone,’’ IEEE Sensors J., vol. 21, no. 9, pp. 11084–11093, May 2021.
13. E. P. Fischer, M. C. Fischer, D. Grass, I. Henrion, W. S. Warren, and E. Westman, ‘‘Low-cost measurement of face mask efficacy for filtering expelled droplets during speech,’’ Sci. Adv., vol. 6, no. 36, Sep. 2020, Art. no. eabd3083
14. A. S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil, and S. Gupta, ‘‘Deep learning framework to detect face masks from video footage,’’ in Proc. 12th Int. Conf. Comput. Intell. Commun. Netw. (CICN), Sep. 2020, pp. 435–440.
15. 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, Jan. 2021, Art. no. 108288.
16. M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, ‘‘Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection,’’ Sustain. Cities Soc., vol. 65, Feb. 2021, Art. no. 102600.
17. M. Jiang, X. Fan, and H. Yan, ‘‘RetinaMask: A face mask detector,’’ 2020, arXiv:2005.03950. [Online]. Available: http://arxiv.org/abs/2005.03950
18. D. Chiang. (2020). Detecting Faces and Determine Whether People are Wearing Mask. [Online]. Available: https://github.com/AIZOOTech/ FaceMaskDetection
19. B. Roy, S. Nandy, D. Ghosh, D. Dutta, P. Biswas, and T. Das, ‘‘MOXA: A deep learning based unmanned approach for real-time monitoring of people wearing medical masks,’’ Trans. Indian Nat. Acad. Eng., vol. 5, no. 3, pp. 509–518, Sep. 2020.
20. P. Viola and M. Jones, ‘‘Rapid object detection using a boosted cascade of simple features,’’ in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Dec. 2001, pp. 1–9.
21. N. Dalal and B. Triggs, ‘‘Histograms of oriented gradients for human detection,’’ in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2005, pp. 886–893. [26] P. Felzenszwalb, D. McAllester, and D. Ramanan, ‘‘A discriminatively trained, multiscale, deformable part model,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.
22. Z. Zou, Z. Shi, Y. Guo, and J. Ye, ‘‘Object detection in 20 years: A survey,’’ 2019, arXiv:1905.05055. [Online]. Available: http://arxiv.org/ abs/1905.05055
23. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘‘You only look once: Unified, real-time object detection,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.
24. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘SSD: Single shot multibox detector,’’ in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 21–37.
25. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ‘‘Focal loss for dense object detection,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.
26. R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘‘Rich feature hierarchies for accurate object detection and semantic segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587
27. [Automatic Smart and Safety Monitoring System for Kitchen Using Internet of Things](https://ieeexplore.ieee.org/document/9065663/) [Harika Pudugosula](https://ieeexplore.ieee.org/author/37088371616) [2019 International Conference on Intelligent Computing and Control Systems (ICCS)](https://ieeexplore.ieee.org/xpl/conhome/9046169/proceeding) Year: 2019 , Publisher: IEEE
28. [G. Piñeres-Espitia](https://ieeexplore.ieee.org/author/37088860375) ;[Shariq Aziz Butt](https://ieeexplore.ieee.org/author/37086929773);[M . Cañate-Masson](https://ieeexplore.ieee.org/author/37088861776) ;[A. Alvarez-Navarro](https://ieeexplore.ieee.org/author/37088859793);[Syed Areeb Hassan](https://ieeexplore.ieee.org/author/37088859804);[Saikat Gochhait](https://ieeexplore.ieee.org/author/37088479695) ” [Gas Sensing System using An Unmanned Aerial Vehicle](https://ieeexplore.ieee.org/document/9418000/)”