# Project report for CSE 3025

# FACE MASK DETECTION


| | |
|---|---|
| **Sanjana Godiawala** | **19BCE1147** |
| **Neha Bheemisetty** | **19BCE1123** |
| **P Subiksha** | **19BCE1255** |

# Abstract

CoronaVirus disease 2019 has affected the world seriously. One major protection method for people is to wear masks in public areas. Furthermore, many public service providers require customers to use the service only if they wear masks correctly. It is not easy to check everyone whether they are wearing a mask or not, so we are proposing a face mask recognizer where it recognizes the particular person wearing a mask or not. In the present scenario due to Covid-19, there are no efficient face mask recognition applications that are now in high demand for transportation means, densely populated areas, residential districts, large-scale manufacturers, and some other enterprises to ensure safety. This system can therefore be used in real-time applications which require face-mask recognition for safety purposes due to the outbreak of Covid-19. This project can be integrated with the embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed. So, the main motive of this project is to identify a person whether that person is with a mask or not, using the artificial neural network technique and identify that person from the given database. For the implementation of the project, we are using Apache Nifi, OpenCV, and Tensorflow with programming in python.

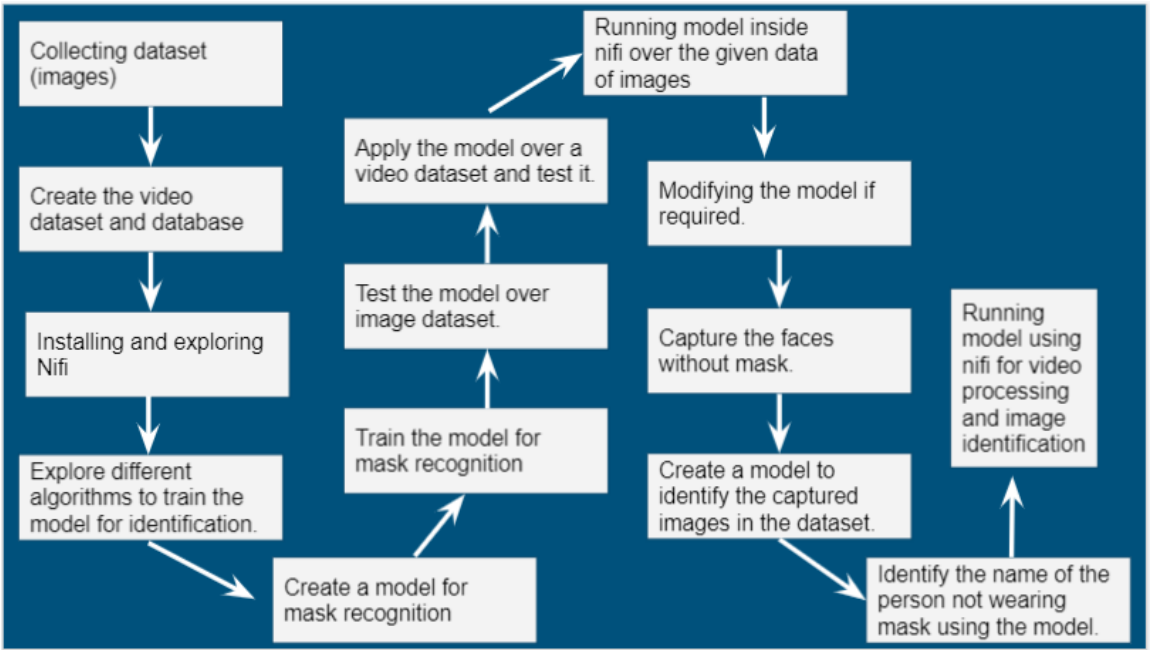
# Motivation

Due to the spread of Novel CoronaVirus, wearing masks on the face has become more crucial. Still, many people are unaware of the consequences of this life taking coronavirus and wearing the mask for namesake and some people take it for granted and most of the time refuse to wear the mask. It would affect not only the people who are not wearing them but also the healthy people. The spread of the coronavirus would be faster than the velocity of air when people refuse to wear masks in crowded places like airports, markets, railways. So, in order to alert them and to reduce the spread of this coronavirus, a real-time system is being designed which detects whether the people are wearing the mask or not and it also detects people who are wearing the mask improperly. Without a real-time system, the job to alert people to wear masks would be hectic and time-consuming. This system is designed using a deep learning model. Once the face of the people who are not wearing the mask is recognized using the face mask detection algorithm we would contact them and ask them to wear the mask to make their life and the life of other surrounding people on a safer side. Factories, Industries, Offices, and Government agencies can use this system at the entry gate to allow only the people that are wearing a face mask to reduce the spread of CoronaVirus.

# Introduction

As the number of cases are rising daily at an alarming rate,wearing masks is made compulsory for all people. A real time system has been designed using a deep learning model to detect whether people are wearing the mask or not.We have accomplished it in both ways, image processing as well as in video stream through which it detects the face of the people when the people doesn't wear mask or not wearing it properly and then it would alert them to wear the mask correctly.We have trained the face mask detector using OpenCV and python.And it would be very helpful in many places where large gatherings of people available for eg,offices,airports, hospitals,colleges,etc.From the dataset we would be able to infer, it displays name and contact number of the people who is not wearing mask or wearing it improperly.For comparing the images MSE(Mean Squared error) algorithm is been used and for resolving issues of MSE,SSIM(Structural Similarity Index) is used.SSIM is to model the perceived change in the structural information whereas MSE is actually for estimating the perceived errors.With the help of Face Mask Detection algorithm,we can detect firstly the presence and location of a face in an image and then secondly we can extract the embeddings that quantify each face in an image. And then facial landmarks would be computed for pre-processing and for aligning the face. Once face alignment and cropping is done,we can pass the face (input) through deep neural neutral.So,in this manner the network is able to learn to quantify faces and return highly robust and discriminating embeddings suitable for face recognition.Training the model is the first part of our project, the second part is testing using OpenCV and the final part is integrating it with person identification.

Flowchart of the project is as follows:



# Objectives

Coronavirus disease has affected the world seriously. One major protection method for people is to wear masks in public areas. Furthermore, many public service providers require customers to use the service only if they wear masks correctly. However, there are only a few research studies about face mask detection based on image analysis. Here, we propose a Face Mask detector, which is a high-accuracy and efficient face mask detector.

So, here we propose a Face Mask detector where it detects whether the particular person is wearing a mask or not wearing a mask along with the percentage. This is a high-accuracy and efficient face mask detector

The main objectives of this project are:-

1. Training the model for face detection followed by mask detection
2. Exploring Apache Nifi
3. Creating the a model for comparing images and identifying the person from database
4. Integrating the two models and testing it.

In this project we aim to create a model that could detect faces not wearing masks for an institution, where the details about the people working there are stored in a database, through which we can identify the person. As an extension a mechanism can be developed to alert/contact them and ask them to wear the mask correctly.

# Software and Hardware requirements

1)NiFi

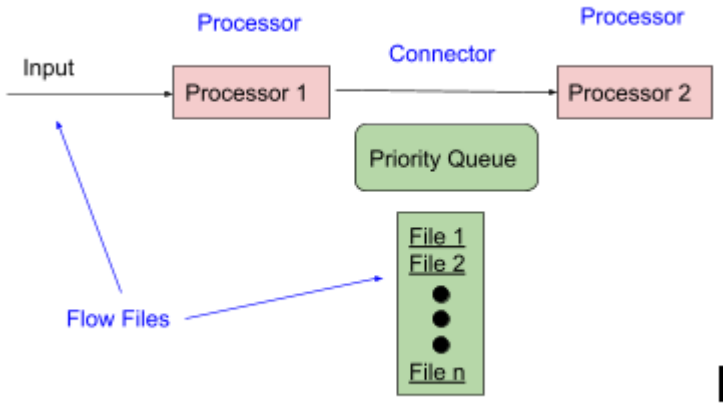
Apache Nifi is the software tool we are using in this project.It is an integrated data logistics platform for automating the movement of data between disparate systems.NiFi provides configurable plumbing platform for moving data, and enables tracing data in real time.NiFi can also adjust to fluctuating network connectivity that could impact delivery of data.

Apache Nifi supports directed graphs of data routing, transformation, and system median logic.It basically automates the flow of data between systems. It has a drag and drop interface which includes configuration of process, scalability across clusters, guaranteed delivery(no data loss), prioritization queuing with optimised latency and throughput.

Nifi is an efficient tool to use for reliable and secure data transfer between systems, delivery of data from sources to analytic platforms, enrichment and preparation of data like format conversion, extraction/ parsing, reduction, etc. Nifi is not much used for distributed computation, complex event process and joins, rolling windows and aggregate operations.

Three basic concepts of Nifi:-

1. **Flow files**:- It is basically data, it has 2 elements: the data itself and the attributes of that data. These attributes are key value pairs associated with the data. Gets persisted to disc after creation.
2. **Processor:-** Applies a set of transformations and rules to Flow files to generate new Flow files. Any processor can process any flow files. They pass flow files reference to each other to advance the data processing and they all run in parallel using different threads .
3. **Connector**:- It is basically a queue of all the flow files that are yet to be processed by the processor. Based on the rules we can prioritize this queue and also avoid overflow in the system.



2)Jupyter Notebook

Jupyter Notebook is a open-source,interactive web tool as a computational notebook,which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document, it comes in anaconda software which has other features too like, pyTorch, spider,etc along with most of the data science related packages pre installed. For our project we require opencv which is installed using the command “pip install opencv-python” in anaconda terminal. In this project we will be using the Keras Deep Learning Library and we will be running it on top of the TensorFlow backend. The libraries and packages used are:- pip, OpenCV, NumPy, Keras, Tensor Flow, Mtcnn, Matplotli, Pandas.

3) For coding using Sublime Text which is a code editor with a Python application programming interface. It natively supports many programming languages and markup languages, and functions can be added by users with plugins.

4)Central Processing Unit (CPU) — Intel Core i5 6th Generation processor or higher. An AMD equivalent processor will also be optimal.

RAM — 8 GB minimum, 16 GB or higher is recommended.

Graphics Processing Unit (GPU) — NVIDIA GeForce GTX 960 or higher. AMD GPUs are not able to perform deep learning regardless.

Operating System — Ubuntu or Microsoft Windows 10.

# Project Explanation

**Module Level Explanation:**

**Face detection:**

We will input the pic. With the face detector, the face will be detected for the pic here our Input is an Image consisting of faces and the Output is the Face is highlighted with a rectangle.

So, here in this, we give access to webcam, reading images or video and storing.

So, here our Input is an Image consisting of faces and the Output is Face is highlighted with a rectangle.

**Training:**

We will train the system with the databases. (faces with mask and without mask) with the use of the dataset, we train a facemask classifier using Tensor-Flow.

Here we load the dataset containing images of persons wearing masks and not wearing masks, with the use of the dataset we train a facemask classifier using TensorFlow.

So, here the input is a dataset containing images of persons wearing masks and not wearing masks and the output is the training of the facemask classifier.

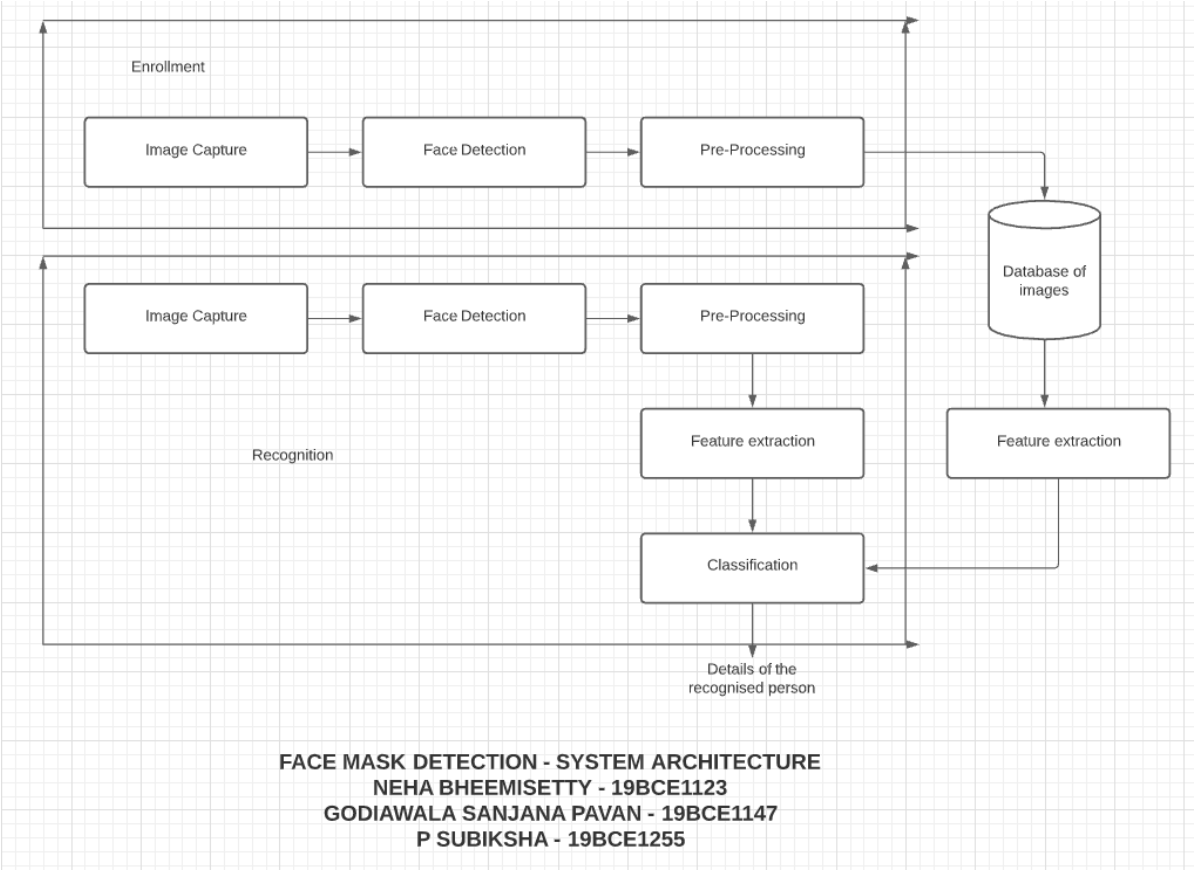
**Mask detection**

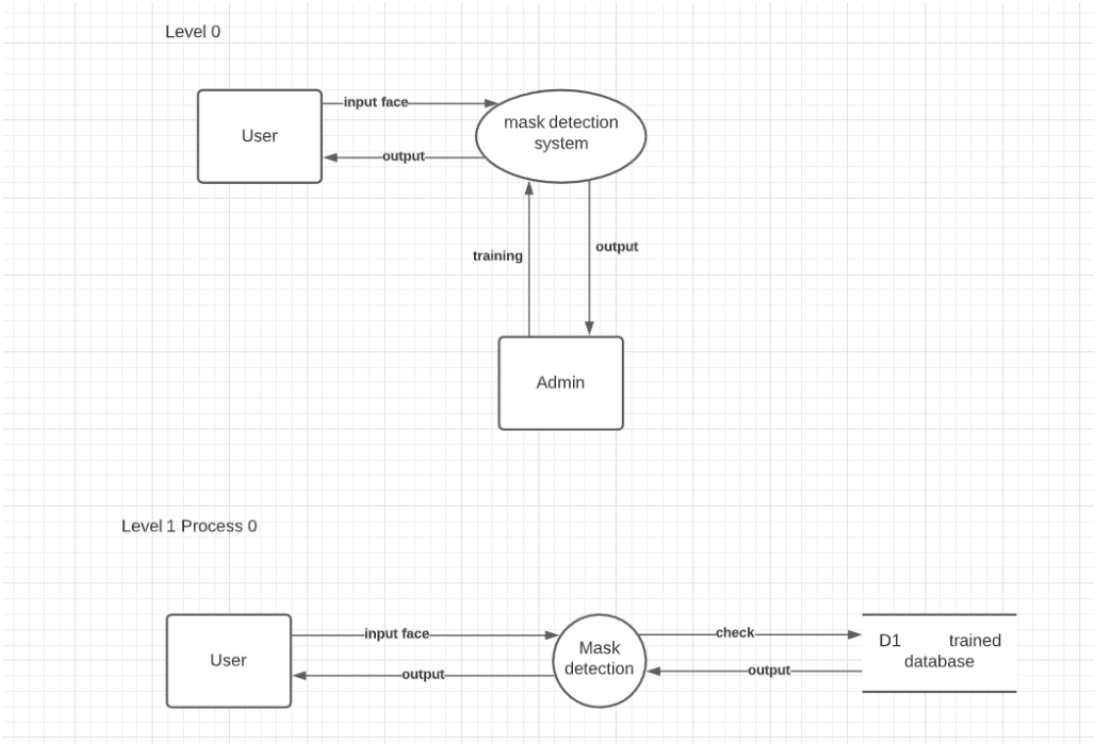
First, we will give the face input. With face detection, the face will be detected. And the system will apply a face mask classifier to each face RIO, and then detect the face with a mask or without a mask. If the person contains a mask the system will show a rectangular box with green color and it shows the detection percentage. If the person doesn't contain a mask it will show the rectangular box around the face with red color.

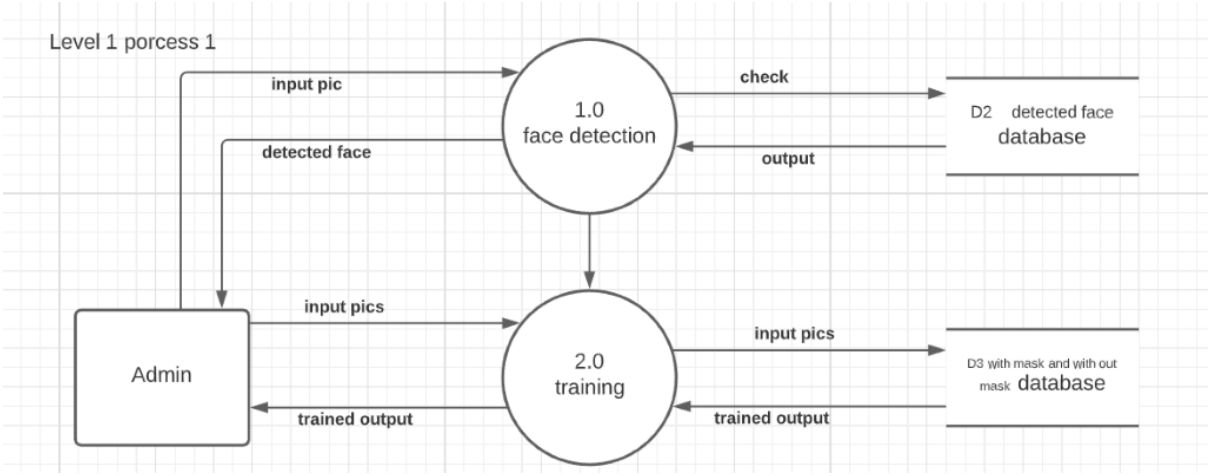
Moving on to the final step we do the following processes: load the facemask classifier, detect faces in the image or video stream, extract each face roi, apply facemask classifier to each face roi and detect whether the face is with a mask or without a mask.

So, here the input is an image or video stream and the output detects whether the particular person is wearing a mask or not wearing a mask along with the percentage.

## SYSTEM ARCHITECTURE:






**Face identification:**

Here we have created a folder containing the images of different people, and a csv file containing the details of each person. The details considered here are name, contact and the location of the image of that person. A python model based on Mean Square Error and Structural Similarity Index algorithm is used to find the person that matches the person from the livestream without a mask.

Here the image of the person is captured when the model detects without mask and is saved as the original image and then compared to all the other images present in the folder, by accessing the locations given by the csv file. After finding the best match it prints the details for that person onto the figure window as well as on the terminal.

# Large Scale Data Processing Approach:-

In our process the sense of large scale data processing can be considered as the video is basically a collection of multiple images, hence while processing video we are actually processing the multiple frames at the single time. Now for video we don't need so many frames/second as it won't be changing at that fast rate. So we can reduce the number of frames. Also considering the identification of the person module, in a real time scenario there would be a huge database with multiple people mentioned, so image identification would take a long time, instead if we think of using different nodes and dividing the matching process over distributed system and then output the most similar picture would reduce the delay in the image identification.

# How to use

1. Open Anaconda Prompt, change the directory to where the project is located.

   Using the command " cd <location>"

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Neha>cd C:\Users\Neha\Desktop\SEM4\LSDP\Face-Mask-Detection-master

(base) C:\Users\Neha\Desktop\SEM4\LSDP\Face-Mask-Detection-master>
```

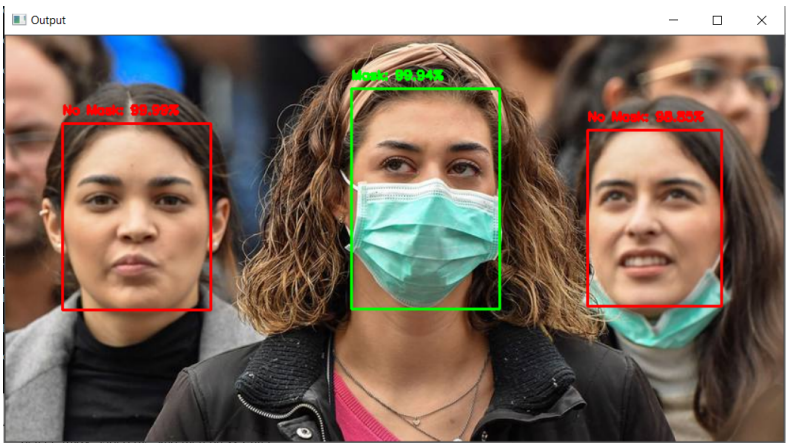
2. Once after changing the directory, now to run the image detection python file. Use the command "python detect_mask_image.py --image images/pic2.JPG" and then press Enter.

```
(base) C:\Users\Neha\Desktop\SEM4\LSDP\Face-Mask-Detection-master>python detect_mask_image.py --image images/pic2.JPG
2021-06-05 16:40:47.985725: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2021-06-05 16:40:47.986580: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do n
ot have a GPU set up on your machine.
```

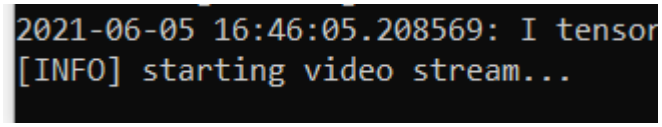
3. Then it prompts an output window with the labels on the picture mentioned. It shows a green-labeled box naming mask and the accuracy of wearing the mask around the face of the person. And a red labeled box naming no mask and the accuracy of not wearing a mask around the face of the person.

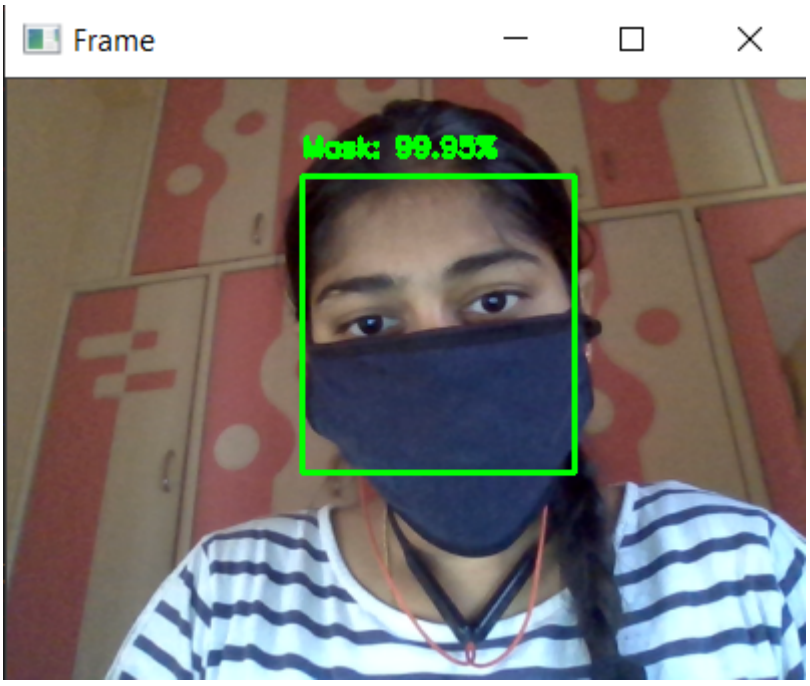
   Actual image in the dataset:

   Output after detecting the mask :

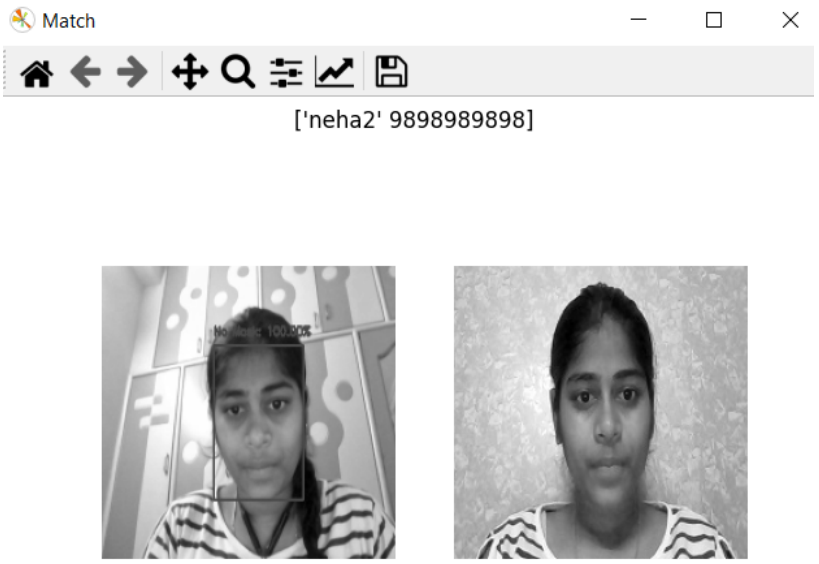



4. Now, to run the code for the live video processing. Type “python detect_mask_video.py” and then press Enter.
5. Then it prints “starting video stream...” and then prompts a live processing frame.




If the person is not wearing the mask it displays the details of the person on the output match tab and also in the command prompt.

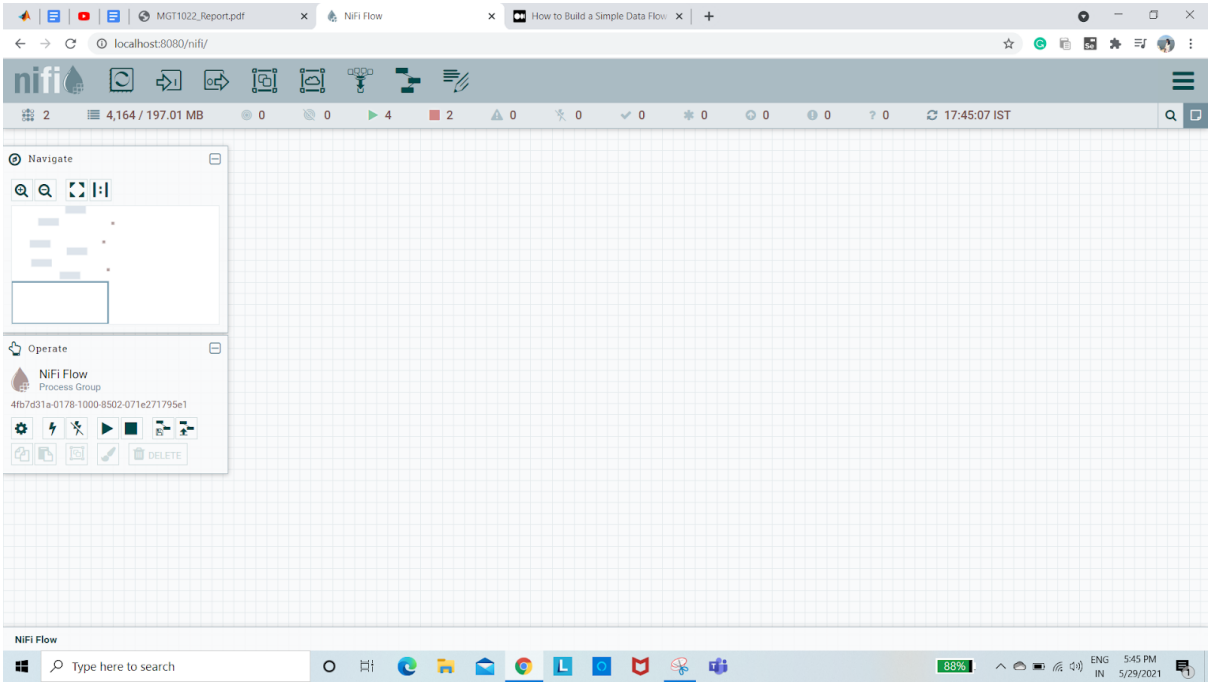


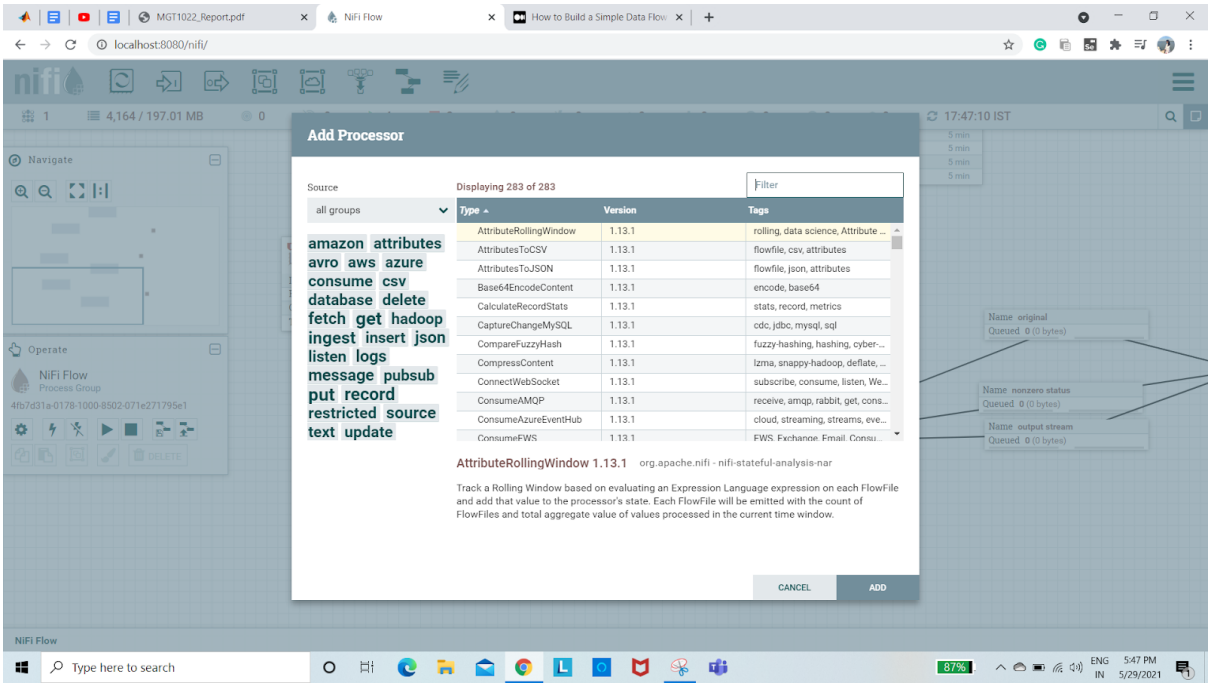

```
The details of the person who is not wearing the mask are:
['neha2' 9898989898]
```

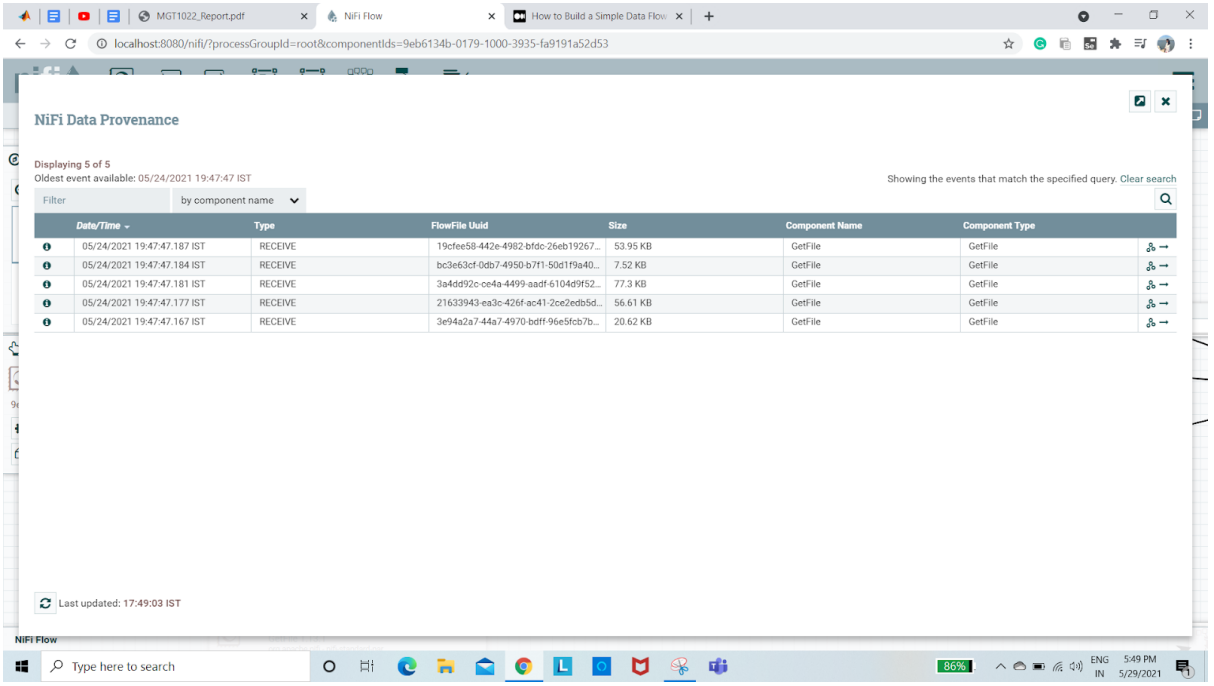
## Implementation in Nifi:-

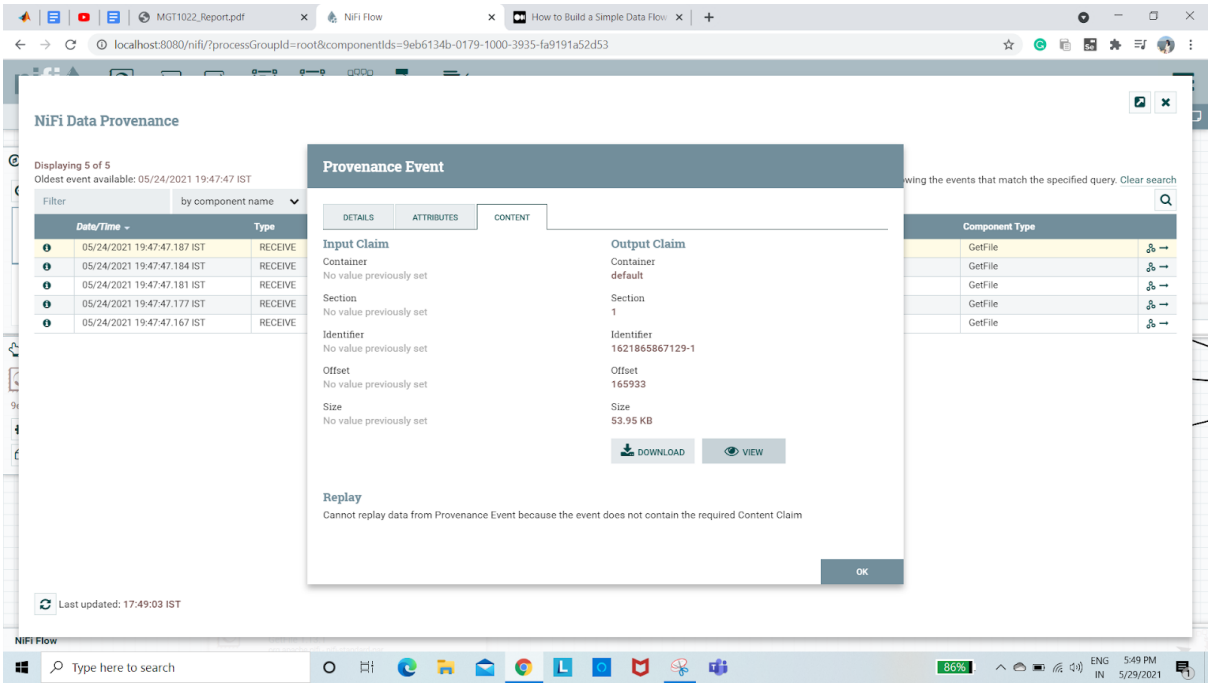
Steps for installing nifi:

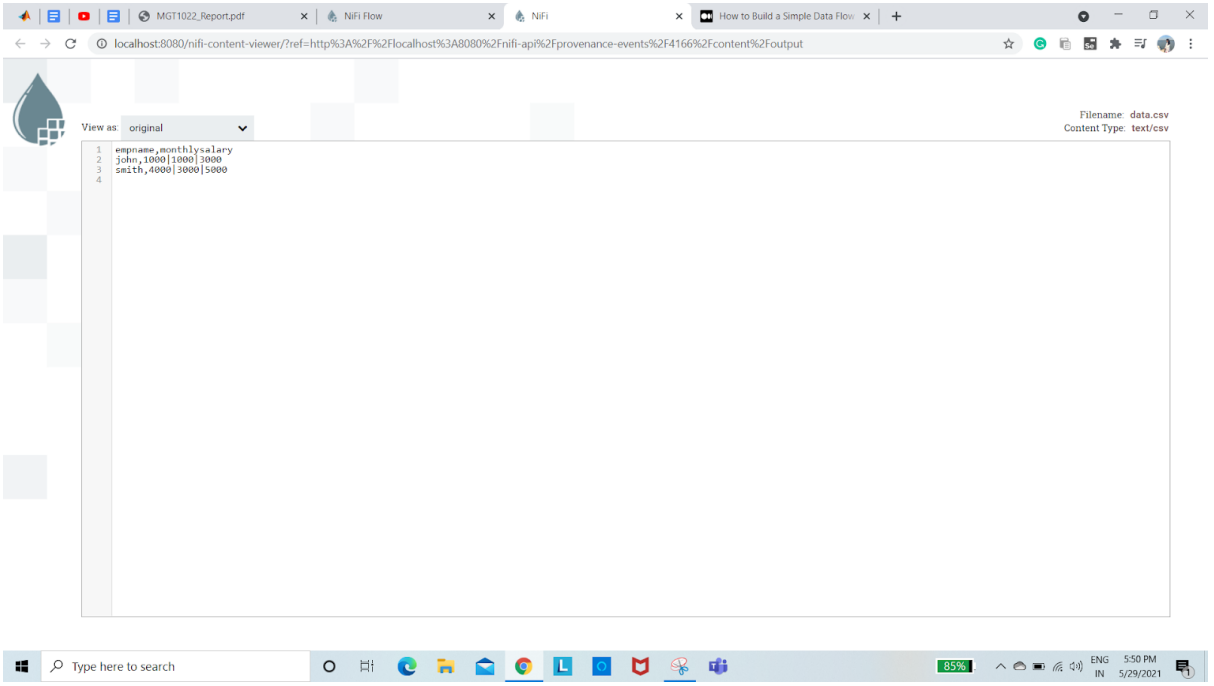
1. Download java jdk 8
2. Download the zip file from the nifi website.
3. If the 8080 port is not free, change the http port in the nifi-properties.
4. Run the run-nifi.bat file.
5. After running write the url localhost:port/nifi/
6. This will open nifi in the web browser.
7. To close it press ctrl+c in the command window where nifi was started.

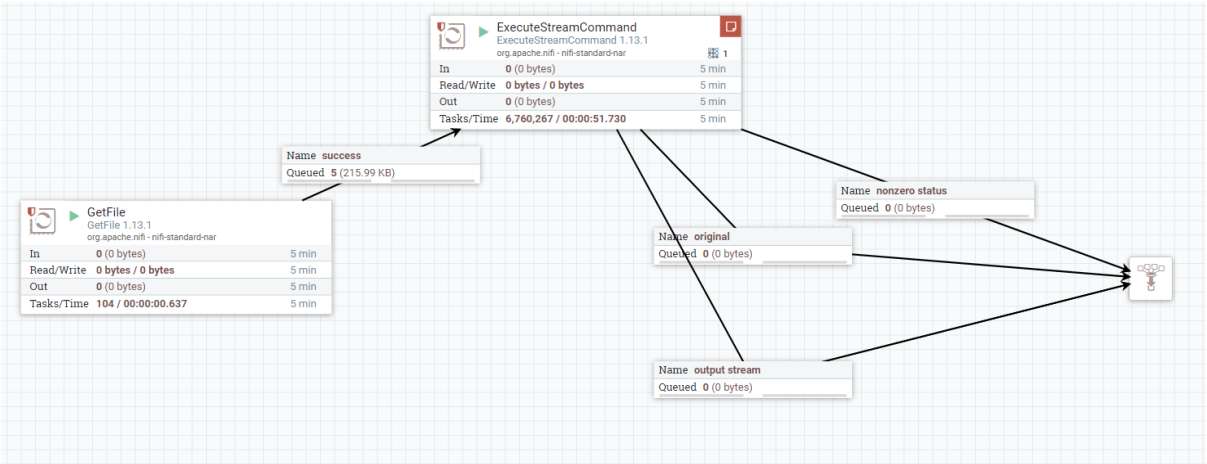

MGT1022_Report.pdf
NiFi Flow
How to Build a Simple Data Flow
localhost:8080/nifi/
4,164 / 197.01 MB
17:45:07 IST
Navigate
Operate
NiFi Flow
Process Group
4fb7d31a-0178-1000-8502-071e271795e1
DELETE
NiFi Flow
Type here to search
88%
ENG
IN
5:45 PM
5/29/2021

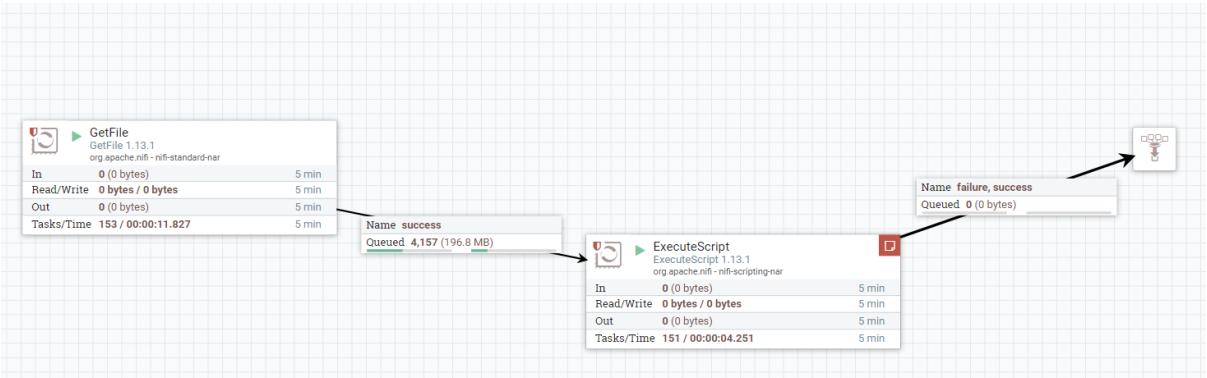

MGT1022_Report.pdf
NiFi Flow
How to Build a Simple Data Flow
localhost:8080/nifi/
4,164 / 197.01 MB
17:47:10 IST
Add Processor
Source
all groups
Displaying 283 of 283
Filter
Type
Version
Tags
AttributeRollingWindow 1.13.1
org.apache.nifi - nifi-stateful-analysis-nar
Track a Rolling Window based on evaluating an Expression Language expression on each FlowFile and add that value to the processor's state. Each FlowFile will be emitted with the count of FlowFiles and total aggregate value of values processed in the current time window.
CANCEL
ADD
NiFi Flow
Type here to search
87%
ENG
IN
5:47 PM
5/29/2021

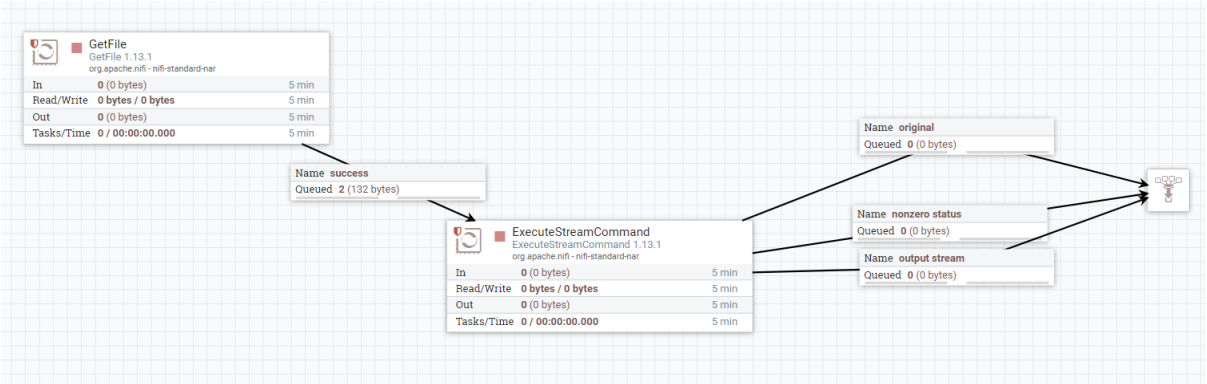

MGT1022_Report.pdf
NiFi Flow
How to Build a Simple Data Flow
localhost:8080/nifi/?processGroupId=root&componentIds=9eb6134b-0179-1000-3935-fa9191a52d53
NiFi Data Provenance
Displaying 5 of 5
Oldest event available: 05/24/2021 19:47:47 IST
Showing the events that match the specified query. Clear search
Filter
by component name
Date/Time
Type
FlowFile Uuid
Size
Component Name
Component Type
05/24/2021 19:47:47.187 IST RECEIVE 19cfee58-442e-4982-bfdc-26eb19267... 53.95 KB GetFile GetFile
05/24/2021 19:47:47.184 IST RECEIVE bc3e63cf-0db7-4950-b7f1-50d1f9a40... 7.52 KB GetFile GetFile
05/24/2021 19:47:47.181 IST RECEIVE 3a4dd92c-ce4a-4499-aadf-6104d9f52... 77.3 KB GetFile GetFile
05/24/2021 19:47:47.177 IST RECEIVE 21633943-ea3c-426f-ac41-2ce2edb5d... 56.61 KB GetFile GetFile
05/24/2021 19:47:47.167 IST RECEIVE 3e94a2a7-44a7-4970-bdff-96e5fcb7b... 20.62 KB GetFile GetFile
Last updated: 17:49:03 IST
NiFi Flow
Type here to search
86%
ENG IN
5:49 PM 5/29/2021

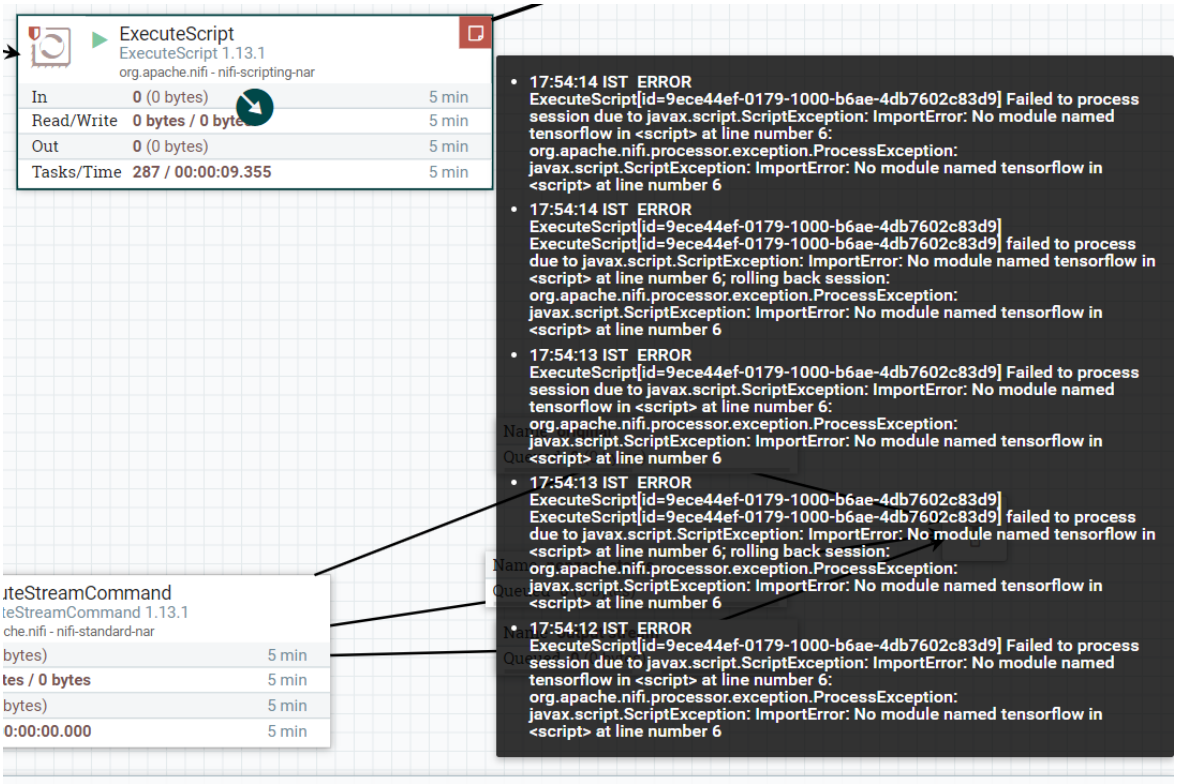

MGT1022_Report.pdf
NiFi Flow
How to Build a Simple Data Flow
localhost:8080/nifi/?processGroupId=root&componentIds=9eb6134b-0179-1000-3935-fa9191a52d53
NiFi Data Provenance
Displaying 5 of 5
Oldest event available: 05/24/2021 19:47:47 IST
Filter
by component name
Component Type
GetFile
Provenance Event
DETAILS
ATTRIBUTES
CONTENT
Input Claim
Container
No value previously set
Section
No value previously set
Identifier
No value previously set
Offset
No value previously set
Size
No value previously set
Output Claim
Container
default
Section
1
Identifier
1621865867129-1
Offset
165933
Size
53.95 KB
DOWNLOAD
VIEW
Replay
Cannot replay data from Provenance Event because the event does not contain the required Content Claim
OK
Last updated: 17:49:03 IST
NiFi Flow
Type here to search
86%
ENG IN
5:49 PM 5/29/2021


MGT1022_Report.pdf
NiFi Flow
NiFi
How to Build a Simple Data Flow
localhost:8080/nifi-content-viewer/?ref=http%3A%2F%2Flocalhost%3A8080%2Fnifi-api%2Fprovenance-events%2F4166%2Fcontent%2Foutput
View as: original
Filename: data.csv
Content Type: text/csv
empname,monthlysalary
john,1000|1000|3000
smith,4000|3000|5000
Type here to search
85%
ENG
IN
5:50 PM
5/29/2021


ExecuteStreamCommand
ExecuteStreamCommand 1.13.1
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 6,760,267 / 00:00:51.730 5 min
Name success
Queued 5 (215.99 KB)
GetFile
GetFile 1.13.1
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 104 / 00:00:00.637 5 min
Name nonzero status
Queued 0 (0 bytes)
Name original
Queued 0 (0 bytes)
Name output stream
Queued 0 (0 bytes)


GetFile
GetFile 1.13.1
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 153 / 00:00:11.827 5 min
Name success
Queued 4,157 (196.8 MB)
ExecuteScript
ExecuteScript 1.13.1
org.apache.nifi - nifi-scripting-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 151 / 00:00:04.251 5 min
Name failure, success
Queued 0 (0 bytes)




In our case we are getting errors as it is not recognising a python file, it considers it as a java or javascript file. We tried it for a small sample excel file but for that also it was showing the same error.

# Dataset Description

In this project, we would be using an image dataset as an input to the model, using which we will train the model to recognize whether the person is wearing a mask or not.

Using this model over a video we will identify people with or without masks in a video stream.

After this, we will try to capture the person without a mask and try to identify that person in the dataset that we will be creating as we could not find any relevant dataset regarding this. So we will create a few videos with and without masks and create our own dataset for person identification.

The dataset of the person identification currently includes 3 fields, which are name, contact number and the location of the image of that person. So for each person in the dataset the model will compare the captured image without having a mask and based on the matched values the details of the person would be displayed.

In this dataset adding more people would not hinder the working of the model as we can scale up the dataset by increasing the number of rows of the dataset and adding images to the specified image folder.

# Results

FOR IMAGE DETECTION:

Input data:

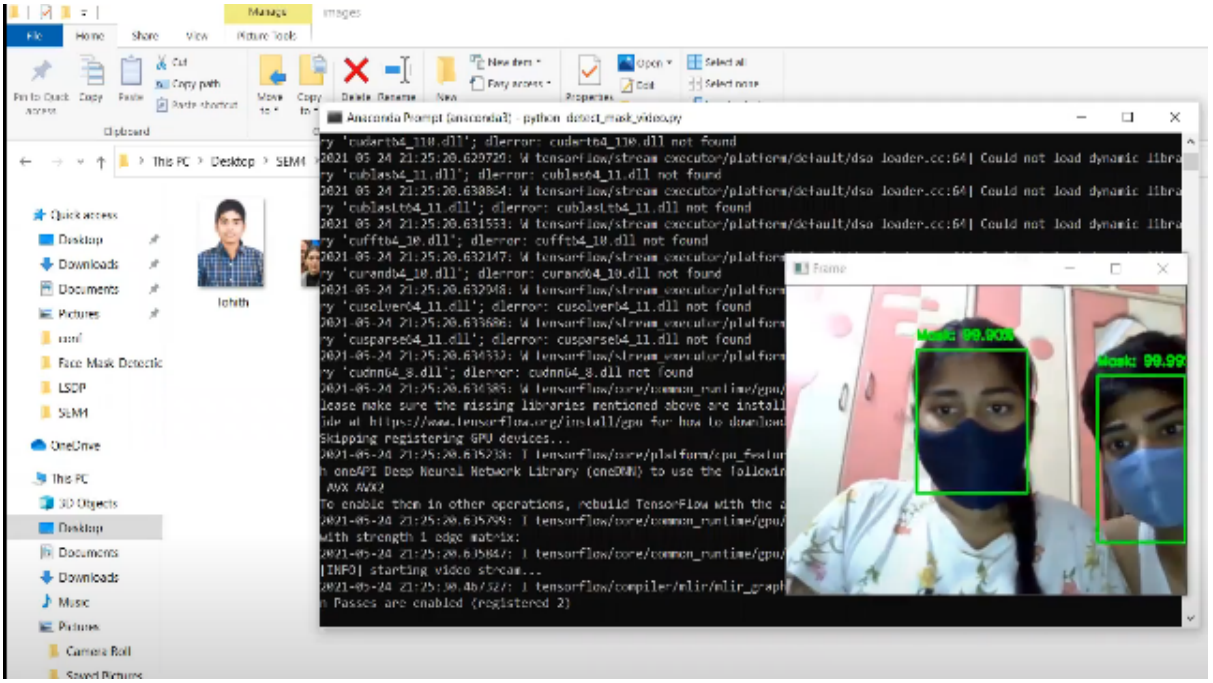
Output:

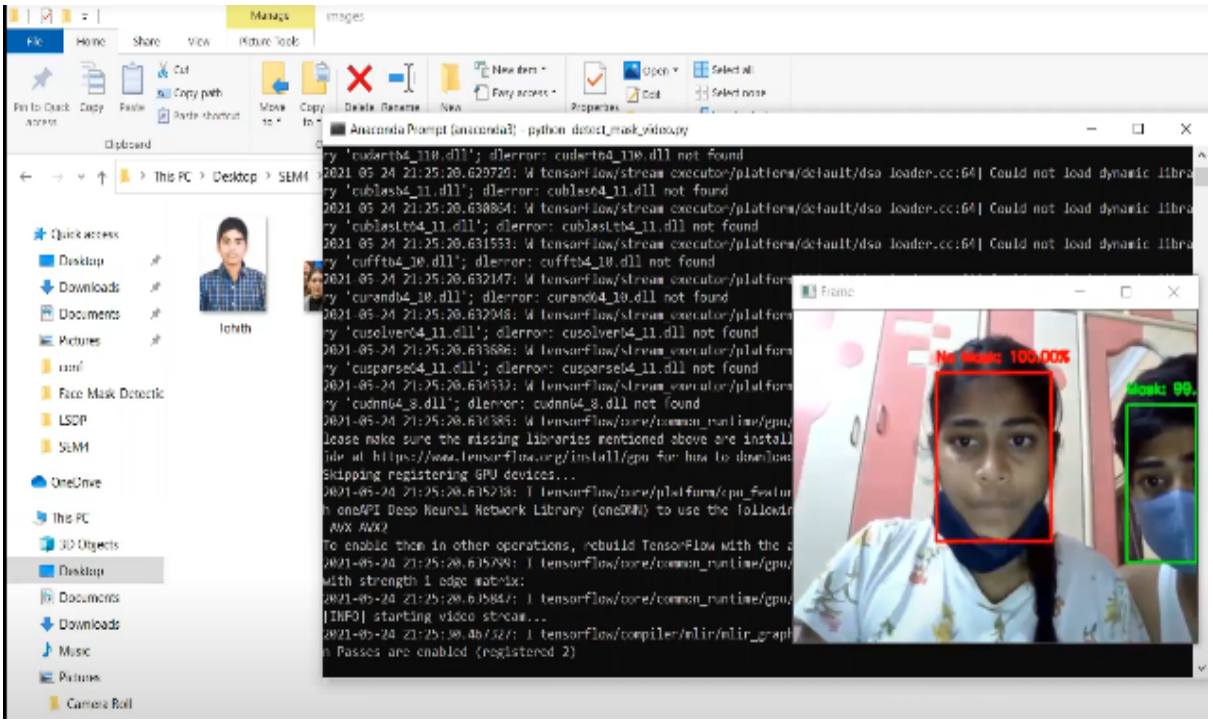
FOR VIDEO PROCESSING:

Multiple people:

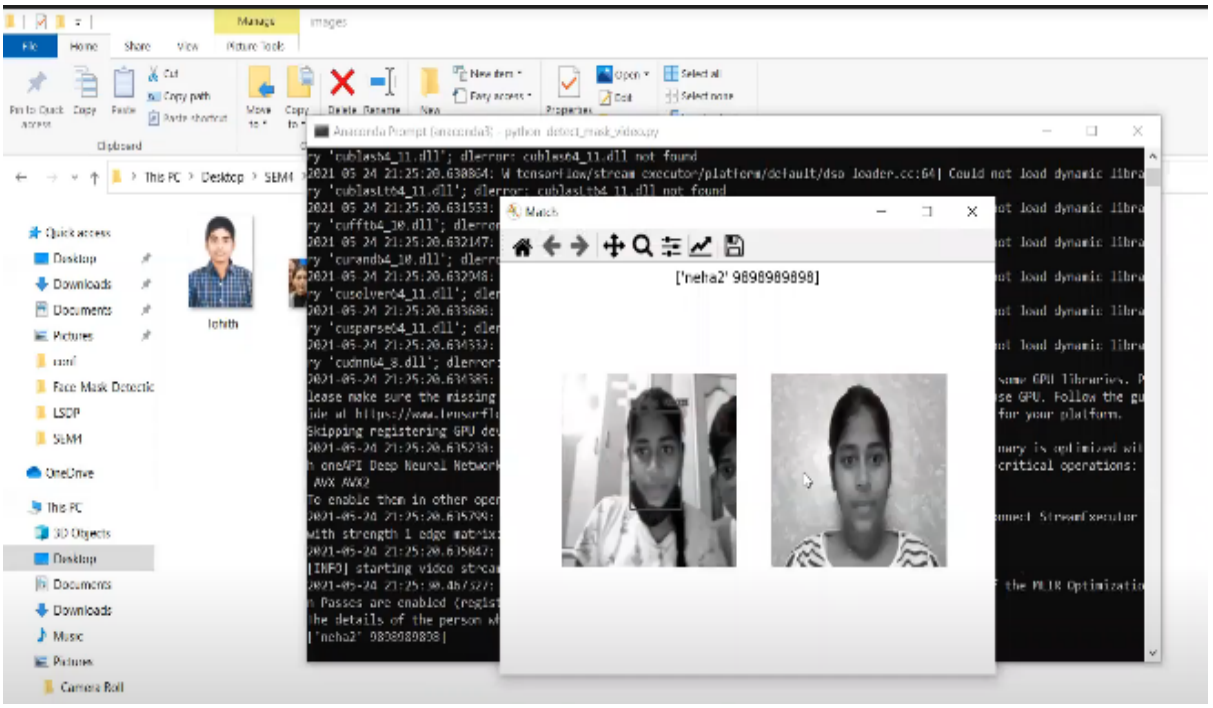
Input: Live video stream

Output:

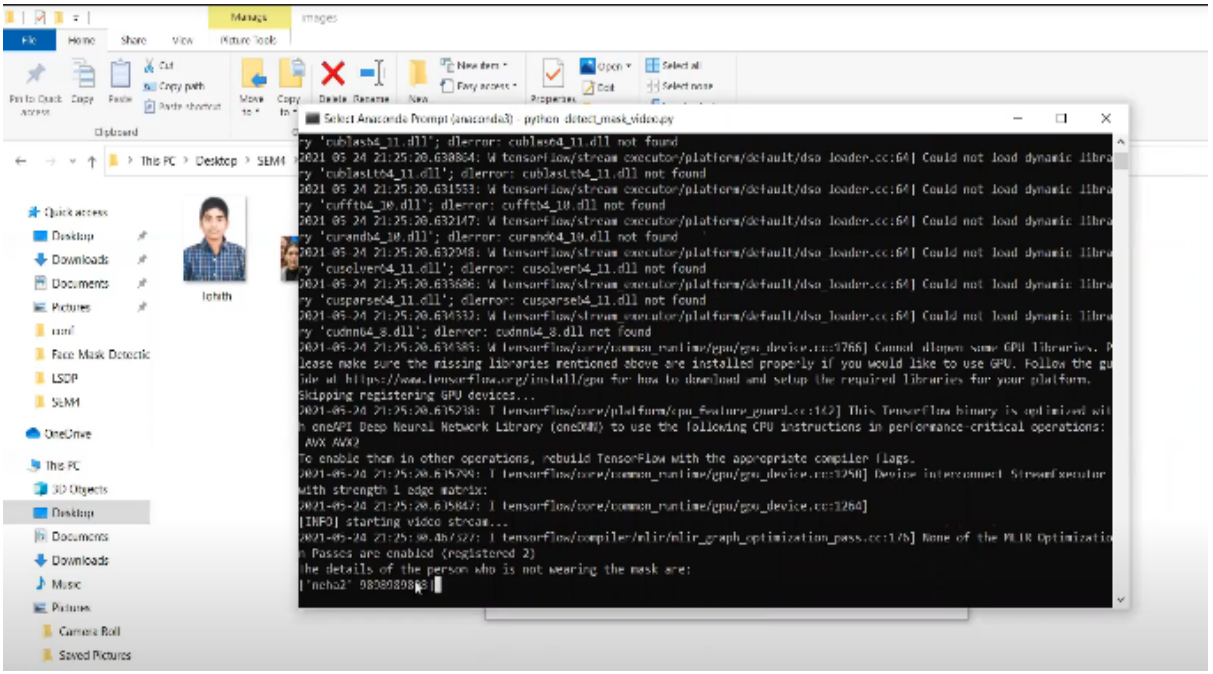




FOR VIDEO PROCESSING ALONG WITH IDENTIFICATION:




PRINTING THE DETAILS OF THE PERSON NOT WEARING A MASK:



# Conclusion

During COVID-19, One major protection method for people is to wear masks in public areas. Furthermore, many public service providers require customers to use the service only if they wear masks correctly. This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed. Using this we can not only detect people wearing and not wearing masks but also identify them and reach out to them for ensuring the safety of one and all.

# Future Enhancement

We can use them in the traffic cameras and cinema halls where It is able to detect the people with and without masks.

Nowadays, many public service providers require customers to use the service only if they wear masks correctly. It is not easy to check everyone whether they are wearing a mask or not, so we are proposing a face mask detector where it detects if the particular person is wearing a mask or not. This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

# References


1.https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/

2.https://data-flair.training/blogs/face-mask-detection-with-python/

3.https://www.mygreatlearning.com/blog/real-time-face-detection/

4.https://towardsdatascience.com/covid-19-face-mask-detection-using-tensorflow-and-opencv-702dd833515b

5.https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/#:~:text=To%20build%20our%20face%20recognition,and%20video%20streams%20with%20OpenCV

6.https://www.geeksforgeeks.org/python-opencv-capture-video-from-camera/

7.https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/

8.https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/

9.https://community.cloudera.com/t5/Community-Articles/Python-Script-in-NiFi/ta-p/246406#:~:text=The%20way%20we%20actually%20access,in%20NiFi%20called%20%22session%22.&text=is%20how%20we%20grab%20the,write()%20method.&text=This%20should%20help%20you%20get,python%20scripts%20inside%20of%20NiFi.

10.https://tienbm90.medium.com/run-pure-python-program-in-nifi-7cfd9df7c7e0

11.https://www.youtube.com/watch?v=rUJOS6LT5Jk