

High Level Design (HLD)

Face mask detection system

Realised by : Amri lheb

last date of revision 16/08/2021

Abstract :

COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society and to track the health protocols violators.

COVID-19 mask detector could potentially be used to help ensure your safety and the safety of others .

2 General Description

2.1 Product Perspective

The face mask detector solution is a deep learning (Computer Vision) based object detection model which will help us to detect the health protocols violators and take the necessary action.

2.2 Problem statement

To create an AI solution for surveillance using surveillance cameras and to implement the following use cases .

- To detect people entering without mask
- To detect people wearing mask incorrectly

2.3 Proposed solution

The solution proposed here is a camera surveillance system that can be implemented to detect and identify whether the person is wearing a mask or not. With a varying distance and color combination, it works for any person . When it detects a person without a mask or wearing it inappropriately it will take photos for evidence and send it immediately to the supervisor to take the necessary actions.

2.4 Further Improvements

The camera surveillance system can be added with more use cases like live temperature to detect and record the temperature of that person and raise an alert if it depasses the threshold.It **will gain the post of the worker at the door that takes temperature** .So it will :

- Gain money
- Gain time for workers/students by avoiding traffic at the entry.

2.5 Data Requirements

Data requirement completely depend on our problem statement,

- We need images data that is balanced and must have at least 1000 images.
- We require at least 30- 40 images for each class label with annotation.
- An image is nothing more than a two-dimensional array of numbers(pixels)
- Pixel value ranging between 0 to 255
- It is defined by the mathematical function $f(x, y)$, the value of $f(x, y)$ at any point
- is giving the pixel value at that point of an image

- Original image is in the format of (width, height, no of RGB channels).

There are numerous image file formats out there so it can be hard to know which type best suits your image needs (on your requirement).

- TIFF—Tagged image file format
- BMP—Bitmap image file form
- JPEG - Joint photographic experts' groups
- GIF - graphics interchange format
- PNG —portable network graphics
- EPS—encapsulated postscript
- RAW image files

Tiffs are great for printing. These are lossless image files meaning they don't need to compress or lose any image quality or information. These format images are high quality images.

- bmp format developed by Microsoft for windows. There is no compression or information loss; this format is generally recommended for high quality scans.
- JPEG is a lossy format meaning that the image is compressed to make a smaller file but this loss is not noticeable.
- JPEG is a very popular format for digital cameras.
- GIFs are widely used for web graphics because they are limited to only 256 colours, can allow for transparency and can be animated. These types of files are typically small in size and very portable.

- PNG are a lossless image format; these files are able to handle up to 16 million colours unlike the 256 colours supported by GIF.

If the data is in video format like (MP4) convert into images based on FPS (no, of frames displayed per second) in real time processing. There are number of tools to convert videos into images, Using cv we can convert video into images.

2.6 Tools used

Python programming language and frameworks such as Numpy,Pandas,Scikit-learn,TensorFlow,keras build the model .

- Sublime Text /Anaconda Prompt are used for the development .
- For visualization of the plots Matplotlib
- Firebase is used for the deployment
- Front end development is done using HTML/CSS and Javascript .
- Python is used for backend development
- GitHub is used as a version control system.

2.6 hardware Requirement

- Computer
- Camera For surveillance

2.7 Constraints

The camera surveillance system must be user friendly ,as automated as possible and users should not be required to know any of the workings .

2.8 Assumptions

The main objective of the project is to implement the use cases as previously mentioned for new dataset that comes through the surveillance camera that is capturing the live videos .Deep Learning based object detection model is used for capturing the above-mentioned use cases based on the input data .it is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting .

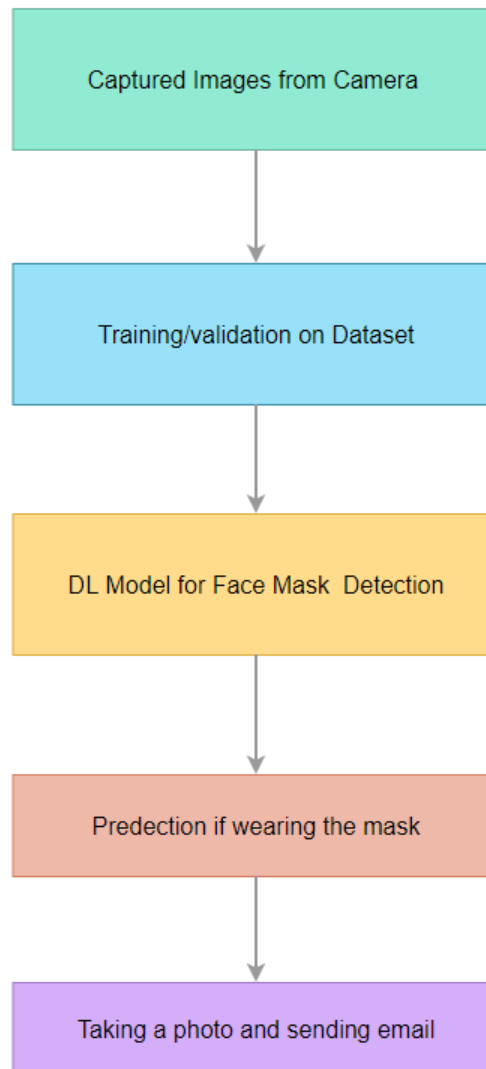
3 Design details

3.1 Process flow

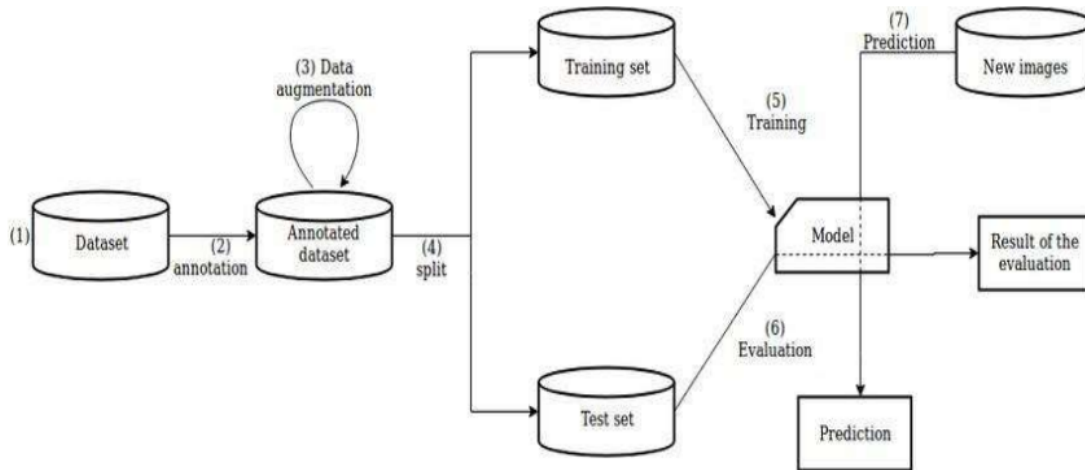
For identifying the persons who are wearing masks or not ,we will use a deep learning base model .

Below is the process flow diagram as shown below.

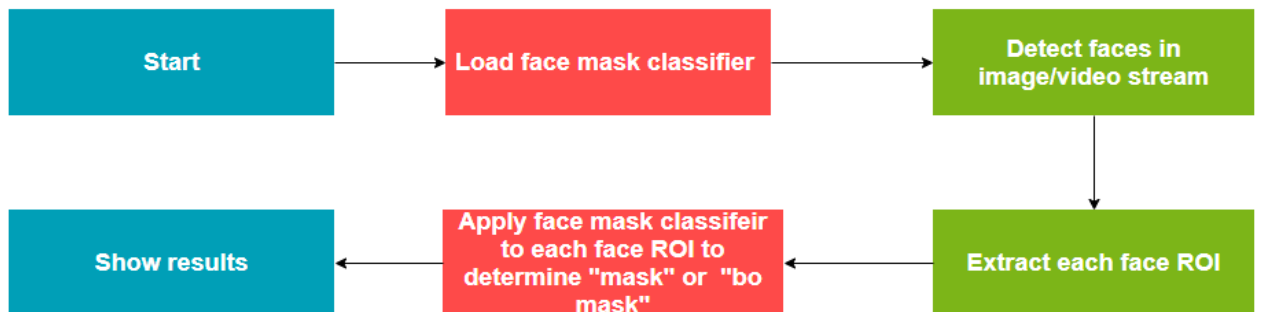
Proposed methodology :



3.1.1 Model training



3.1.2 Deployment Process



3.2 Event log

The system should log every event so that the user will know what process is running Internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developers can choose logging methods. You can choose database logging! File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 error handling

Should errors be encountered ,an explanation will be displayed as to what went wrong ? An error will be defined as anything that falls outside the normal and intended usage .

4 Performance

The camera surveillance mask detection solution is used for detection of policy violations in the group places .Whenever it detects any violation,it will inform concerned authorities and take necessary action, so it should be as accurate as possible.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problèmes.

4.2 Application Compatibility

The different components for this project will be using Python as an interface between

them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

4.4 Deployment



5 Conclusion

The camera mask detector will detect policy violations in the locality based on various mask/non mask data use to train our algorithm, so we can identify violators and help us to prevent the virus spreading.