Face Mask Detection System

1st Tahmidur Rahman
dept. CSE Stamford University
CSE 067 07903
tahmid0059@icloud.com

2nd Kaosar Ahmed
dept. CSE Stamford University
CSE 068 07943
kaosar1971@gmail.com

***Abstract*—To fight against COVID-19 taking safety measures is must, such as wearing mask and maintaining safe distance. This paper reflects application of Machine Learning Algorithm, To keep public gatherings under surveillance and prevent further transmission of a virus that transmits via expiratory particles, By Identifying civilians without mask and raising awareness among them.**

***Index Terms*—covid-19, machine learning, surveillance, cloud-platfom**

I. Introduction

Covid-19 is infectious virus. Transmission of the virus mainly occurs from infected to susceptible individual via cough or sneeze. During a sneeze or a cough, droplet sprays of virus-laden respiratory tract fluid impact directly on a susceptible individual Because The virus mostly transmits via expiratory particles, thus wearing mask can create significant impact, preventing further spread of the virus by 88.5% [1]. Application of machine learning algorithm can bring significant impact when it comes to raising awareness to wear mask.

Our Application Model is kept simple and user-friendly while maintaining core functionality to ensure accuracy, efficiency and efficacy. We used a open source library enabling access to machine learning algorithms and models in the browser which relies on tensorflow.js [2].

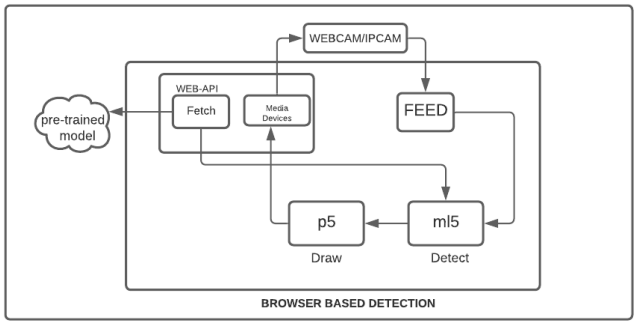


Fig. 1. Browser-based Detection Model

Detection Model from fig-1 Starts with Fetching Pre-Trained Model over the internet The library p5 communicates with Webcam via Browser's WEB-API using webRTC protocol. and gets the feed. ml5 detects realtime frames from the webcam feed and does feature extraction enabling abilities to detect face mask on object.

II. Detection Approach

A. Detection Using Pre-Trained Model

Using Pre-Trained Model the detection process is much easier. Necessary libraries and pre-trained models are embedded into electron-app bundling a single executable file ensuring accessibility. Detection will occur automatically upon webcam access.

Fig. 2. Detection Using Pre-Trained Model

B. Detection Using Customized Image Classifier

Users are allowed to Train Model and Customized Image Classification According to Need ml5 generates custom Image Classifier when active data set is given to.

Fig. 3. Generating Custom Image Classifier

From the webcam image frame training on the fly is possible through simple user interaction. using mask-on and mask-off buttons user simply classify images which are with mask and which are not. After providing enough data set. user clicks train button to generate custom model and classifier mask on and mask off. upon clicking on train button ml5 generates custom model using feature extraction which is known as learning transfer. user can reuse the trained model and use it as described in II-A.

III. Methodology

We applied Agile Methodology to ensure agility in our Software Development Life-cycle and we used scrum as a framework to implement agile Methodology. Agile methodology is the iterative way for developing the software project for frequent changes, fast delivery and reduce risk [3]

A. *Plan*

In this phase we came up with a plan to develop face mask detection system. and analyzed required tools and technological knowledge.

B. *Design*

After analyzing requirements we we designed possible solution to detect face mask on a object maintaining enough accuracy

C. *Develop*

We Implemented our design using JavaScript programming language and machine learning algorithm using pre-trained model

D. *Test*

During the development Life-cycle We did various Test

- Alpha-Test: We did this test to ensure functionality while in development environment
- Blackbox-Test: We did black-box test to check all functionality are usable or not
- Bet-Test: We Released Beta-Version and got feedback from our fellow classmates

E. *Deploy*

We deployed final release in github as a open-source software Ensuring all the test are passed we did continuous Deployment using github Action Flow.

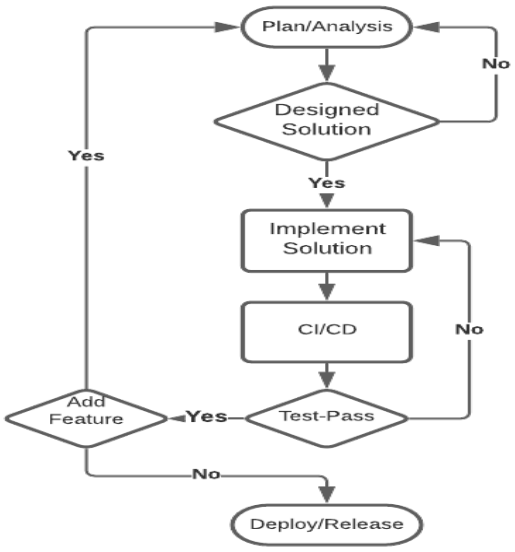


Fig. 4. Agile SDLC

IV. UML Diagram

A.

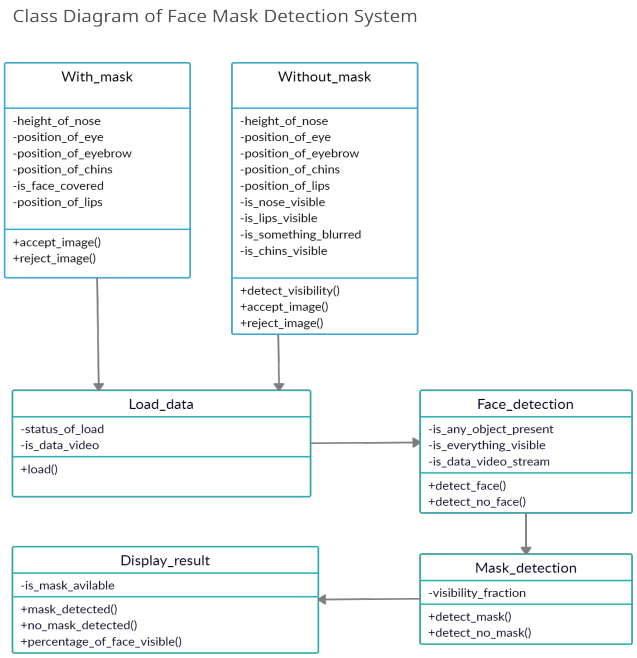


Fig. 5. Class Diagram

B.

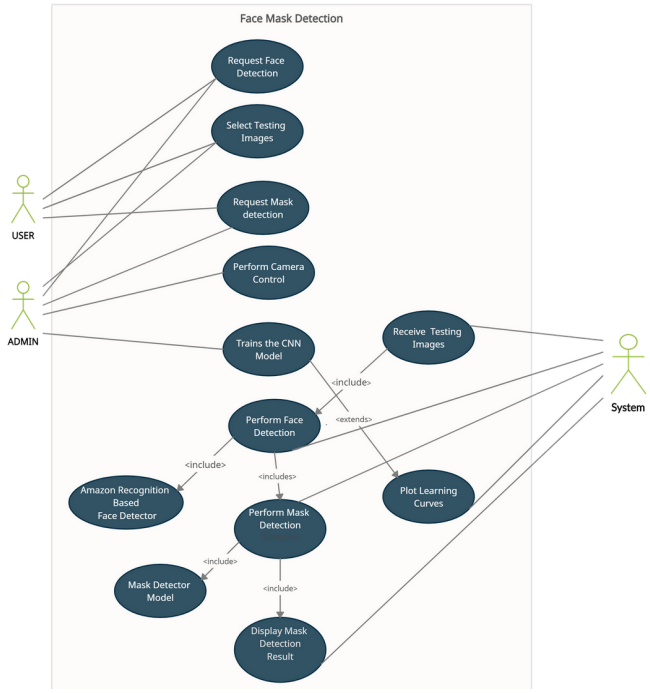


Fig. 6. Usecase Diagram

V. Portability

For Ease of Access We Ported Our Application for different Operating System Architecture, So that application is accessible from different devices.

- Android arm64
- Windows x86_64
- Linux x86_x64

VI. Implementation

Android devices are used by most people in the world. So, building an application for android platform can gain more reach to the people. A robust face mask detection system needs algorithm that can detect object. This objective can be attained by using TensorFlow machine learning model and OpenCV library.

"TensorFlow is free and open-source software library for machine learning and artificial intelligence. It can be used across the range of tasks but has a particular focus on training and inference of deep neural network. [4]" "TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. [5]" TensorFlow is available on 64-bit Linux, Windows, macOS and mobile computing platform including Android and iOS. "TensorFlow can run on multiple CPUs and GPUs (with optional CUDA, and SYLC extensions for general purpose graphics processing unit). [6] "Computations in TensorFlow are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google. [7]"

TensorFlow can be used using Python and C++. For this project to create the data model training set we are going to use python.

"OpenCV (Open-Source Computer Vision Library) is an open-source 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. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These 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. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies. [8]"

VII. Hardware Element

This part will explain about the hardware architecture of the system. It can be illustrated by following figure.

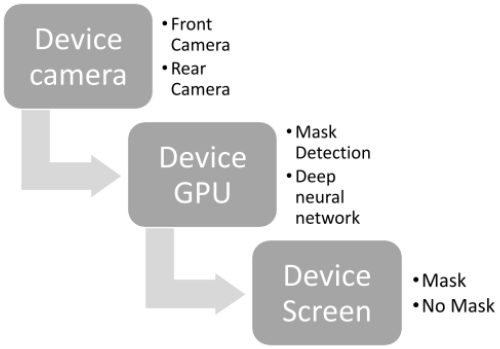


Fig. 7. Block Diagram of face mask detection hardware

The whole hardware system is consisted of a camera (Front or Rear) with minimum 8 Mega Pixel resolution, A GPU and a screen to show the output. To detect if there is mask on face or not, a deep neural network is used as a detection method. All the computation is done by the GPU which is integrated to CPU. The system is a real time application in which camera detected the user and if the use wears mask or not. A rectangle box will appear around the face of the user and if the user wears a mask, then a green border will be showed with the message "mask" along percentage of the coverage of region of interest. If the user doesn't wear a mask, then read border will appear on the screen with the message "no mask". If mask is not worn properly a blue border will show the percentage of coverage in the ROI.

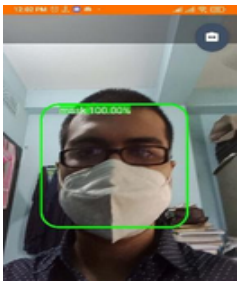


Fig. 8. Right Angle Between Device and Face

VIII. Detection Model

This work is implemented by TensorFlow model and OpenCV library. Both TensorFlow and OpenCV are easy to implement and support both Python and C++ programming

language [9]. CNN algorithm is used to implement the deep neural networking. CNN is a class of artificial neural network which is used to analyze imagery [7]. “A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically, this includes a layer that performs a dot product of the convolution kernel with the layer’s input matrix. This product is usually the Fresenius inner product, and its activation function is commonly REL. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers. [10]” In CNN an input is tensor with a shape: (I x H x W x C) – where I is number of inputs, H is input height, W is input width and C is input channels. After Passing through a convolutional layer, the image becomes abstracted to a feature map. [10] The input image of this work needs to be handled in the resolution around 1920x1080 pixel in real time application to ensure the detection. In this part the convolutional 3xm matrix is built and huge number of parameters will be selected as backbone input layer. In this backbone, the input network resolution will be shrunk into 512x512 pixel with respective field size of 729x729. The second stage predictor sparse prediction which applies the CNN method. In this stage the 3x3 layer is inputted and the output is ‘mask’ or ‘no mask’.

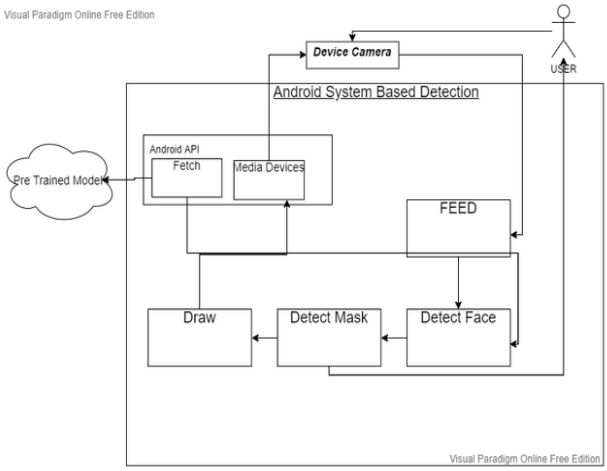


Fig. 9. Detection method in Face mask detection system in android platform

IX. Training Machine Learning Model

To train the model real world masked dataset was collected. The aim is to collect 2500+ images for each “with_mask” and “without_mask” classes. A python script is needed to run to find image with multiple queries. The first experiment which is depicted in Fig-A has been completed the first trial before it is implemented for video stream. Fig-B is showing detection of mask while the mask is not worn properly. Fig-C and D are showing no mask since no mask is worn by the user. Fig-D is showing “mask” since mask is worn properly.

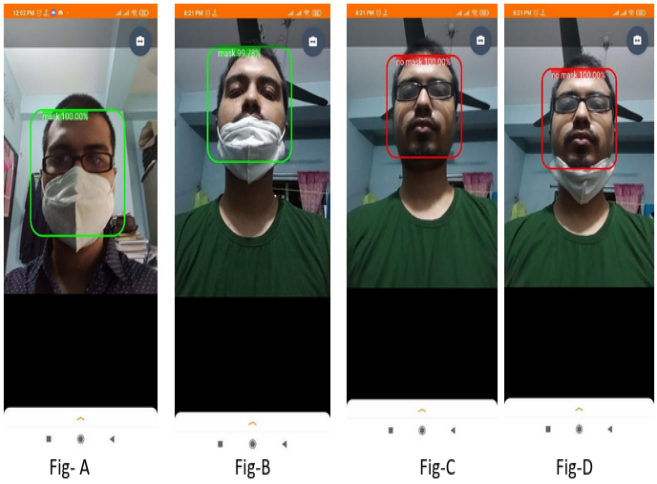
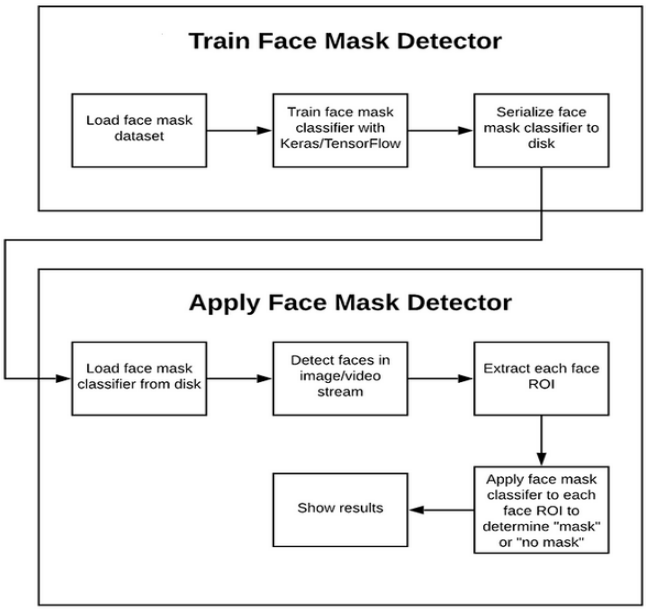


Fig- A Fig-B Fig-C Fig-D

In Training Part We Prepare Our Training Model with Various Image dataset



X. Risk Analysis and Management

Risk analysis and management are a series of steps that help to understand uncertainty of a software project. Many problems can plague a software project. “A risk is a potential problem – it might happen [11] .” “Project risks threaten the project plan that is, if project risks become real [11] .” Risk Identification is a systematic attempt to identify and specify the threats to the project plan. “One method for identifying risks is to create risk item check list. The check list can be

used for risk identification and focus on some subset of known and predictable risks in the following generic subcategories

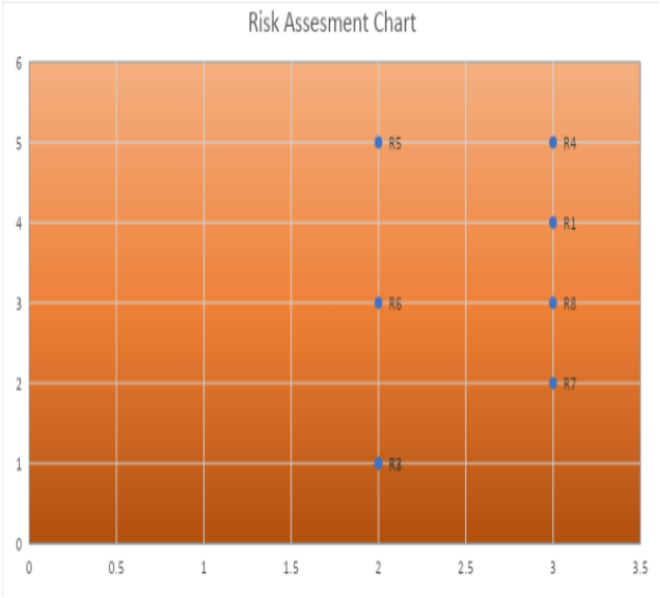
- Product size- Risk associated with the overall size of the software to be built or modified
- Process Definition- Risk associated with the degree to which the software process has been defined and is followed by the development organization
- Development environment- The Risk associated with the availability and quality of the tools to be used to build the product
- Technology to be built- Risk associated with the complexity of the system to be built and the 'newness' of the technology that is packaged by the system.
- Staff size and experience- Risk associated with the overall technical and project experience of the software engineers who will do the work
- Business impact- Risk associated with constraints imposed by management or the marketplace
- Stakeholder characteristics- Risk associated with the sophistication of the stakeholders and developer's ability to communicate with stakeholders. [11]"

The risk components are defined by following manner

- Performance risk- The degree of uncertainty that the product will meet its requirements and be proper to its intended use.
- Cost risk- The degree of uncertainty that the project budget will be maintained
- Support risk- The degree of uncertainty that the resultant software will be easy to correct, adapt and enhance.
- Schedule risk- The degree of uncertainty that the project schedule will be maintained and the product will be delivered in time.

XI. Risk Assessment

Factor	Risks	Category	Probability	Impact
R1	Size estimation may be significantly low	Product	High	Critical
R2	Large Number of user than planned	Product	Very low	Marginal
R3	Less reuse than planned	Product	Very low	Marginal
R4	Project may cross the deadline	Business	Very High	Critical
R5	CASE tools cannot be integrated	Project/Tools	Very High	Marginal
R6	Code generated by CASE tool may be inefficient	Project/Tools	Moderate	Marginal
R7	The Machine Learning algorithm may be defective	Product	Low	Critical
R8	Programmers / Software Engineers may be sick during project time	Project	Moderate	Critical



a) Size Estimation: It may cause severe risk to the product. It occurs often when analyst team underestimate the volume of the product. For face mask detection system, the estimated volume of android app is 50 MB and for computer software is 100 MB.

b) Number of User: For face mask detection system, number of users can vary, for organizational use like school, office and other workplace number of users is fixed. But for public places like shopping mall and rail station the number of users is not fixed. The face mask detection app can work properly for any number of users.

c) Amount of reuse: The number of reuse of resources in the facemask detection app may vary. It is optimized by the analyst team that the number of reuse of resources are limitless for the system.

d) Deadline: The project may cross the deadline to finish. There are several reasons for that, changing of staffs, requirement changing and overall miss prediction of time span. This system may take longer time to build and train, so Extra one week is added to the overall schedule.

e) Case Tools: CASE tools may not work properly during coding and building up the project. For this Project Android Studio was the main CASE tool and there are several defects that can occur while working in Android Studio, such are: Emulator breakdown, unbinding, package failure, backdated built in methods and so on. So before starting the project latest version of Android studio was installed and checked thoroughly.

f) Code Generation By Case Tools: Code generated by CASE tool may not be compatible with the other parts of code. So, compatibility was checked before any generation and integration of code by CASE tool.

g) Defective Algorithm: Defective algorithm causes critical situation for software. This may rise from two way; one is the limitations of an algorithm and other is code written

from the algorithm is not appropriate. Defective algorithm may generate wrong answer. So, for face mask detection app Convolutional Neural Networking (CNN) and K Nearest Neighbor (KNN) algorithm are used. And also, the code is written from this algorithm are thoroughly checked if any error exists.

h) Staff illness: The crucial staffs for the project, for this project programmer and analyst may become sick and unable to carry one work. Since the outbreak of Covid-19 pandemic the probability of being sick is very high. To avoid this situation vaccinated / healthy staffs can be recruited. Also, backup worker can be kept.

XII. Software Testing

Software testing is very important phase of a software building project. "Software is tested to uncover errors that were made inadvertently as it was design and constructed. [12]" Many strategies can be used to test software. "A testing strategy that is chosen by many software teams falls between the two extremes. It takes an incremental view of testing, beginning with the testing of individual program units, moving to tests designed to facilitate the integration of the units (sometimes on a daily basis), and culminating with tests that exercise the constructed system. [13]" When object-oriented software is considered, the concept of the unit changes.

"Encapsulation drives the definition of classes and objects. An Encapsulated class is main focus of unit testing. [13]"However, methods within the class are testable smallest unit. Object-oriented software does not have an obvious hierarchical control structure, traditional top-down or bottom-up approaches have little meaning. Integration operation one at a time is often impossible because "the direct and indirect interaction of the components that make up the class [14]" "There are two different strategies for integration testing of OO systems. [15]" One is thread based testing and other is cluster testing.

A. *Test Strategies used in Face Mask Detection System*

- The content Model for the application reviewed to uncover errors.
- The interface model is reviewed to ensure that all use cases can be accommodated.
- The design model for the application is reviewed to uncover navigation error
- Each functional component is unit tested
- Performance tests are conducted.
- Security tests are conducted in an attempt to exploit vulnerabilities in the application or environment.
- User experience testing is held.
- Device compatibility testing are done on several platform and several architectures.
- It is ensured that the application meets every standards established by the app stores that will distribute it.

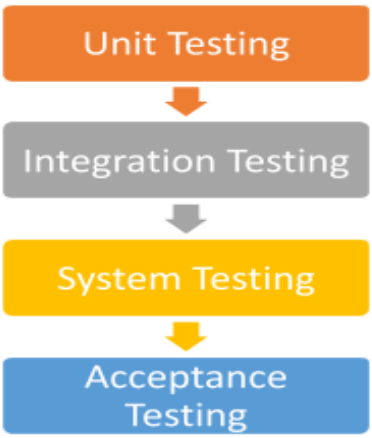


Fig. 10. Testing flow from low level to high level

B. *Unit Testing*

Unit tests are being held by developers' team as well as Tester's team. The Face mask detection system is divided into three separate parts: design Layout, Machine Learning model, and I/O part.

- Design Layout: It's the interface of the application. A rectangle shape layout is created by the system, a rectangle box appears around the face with a message. The result of this test is 100% accurate interface.
- ML Model: Here TensorFlow model is used as machine learning model. It uses CNN algorithm and K-NN algorithm. The model is trained and then tested if it can generate 100% correct output. The application passed this test too with 100% accuracy.
- I/O: The application will take video feed from camera and generates output message either 'mask' or 'no mask' with percentage of face covered. The application can take feed and generate output correctly with 100% accuracy

a) Stress Testing: Stress testing executes a system or program in a manner that demands resources in abnormal quantity or volume. In essence, "How long we can crank it up until it fails?" For the facemask detection system, the computer version worked for 16 hours at a straight and the android version worked 12 hours straight without fails.

b) Compatibility Testing: Compatibility testing assures that the software will work properly in different kind of platform and architectures. The android version of the system was tested in Android '11', '10', '9', '8' version and worked properly without any failure. The computer version was tested in both Windows and Ubuntu and worked properly.

c) Install-Uninstall Test: The system works properly after install and uninstall (reinstalling).

d) Performance Test: "The performance testing is designed to test the run time performance of software within the context of an integrated system. [3]"Performance tests were done along stress tests and was being monitored the performance of both hardware and software components.

e) Alpha-Beta Test: When a software is built, several acceptance tests are conducted to enable the customers to validate all requirements. Alpha tests are conducted by the developer's team. The result was highly satisfying, the software performed well and met user's all requirements. For beta testing 10 people were selected and they use system, their satisfaction is assured

XIII. Future Scope

We will continue our work on adding more features in our system. We can add database to our system. We can add face recognition model to our system and a unique ID will be generated for each face. The data obtained from day-to-day use will be stored in the database. Those data can be used for many statistical purpose in future. We can Implement Missing Person Identification Via Our Application Which will put great impact in society we can add missing person database from local authority and by saving each person in our database and matching against missing person database we can find missing person and notify to relevant authority

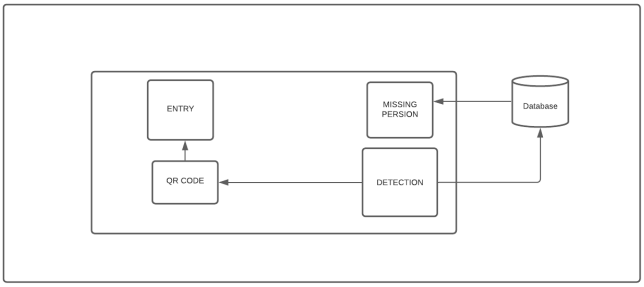


Fig. 11. Feature-Missing person Detection

XIV. Conclusion

For modern machine learning methods, it is possible for us to test presence of face mask without human observation. With the help of these methods, we are able to improve public health care condition. With the help of TensorFlow and OpenCV as well as MobileNetV2 architecture we have succeeded to obtain a more efficient and accurate Face Mask Detection System, which can be used in various high and low computational scenarios. We have added some robust features and trained our model using vast and dynamic dataset so that the system can hundred percent accurately detect a face and mask put on over it from a real time video stream. We have had several difficulties on our endeavor to peruse a robust system; eventually we had better results to compare so many other existing technologies. The system was tested thoroughly and the results are all satisfactory. We worked hard to obtain more efficiency and optimization of the system. This system can play a very good role of prevention of spreading Covid-19 disease by assuring mask wearing of people.

References

[1] D. K. Chu et al. "Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis". The Lancet, Volume 395, Issue 10242, 2020, pp1973 – 1987
[2] "Teachable Machine," Google, 2021. [Online]. Available: https://teachablemachine.withgoogle.com/. [Accessed 2 September 2021].
[3] F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab and M. Abbas, "The Influence of Agile Methodology (Scrum) on Software Project Management," 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2019, pp. 145-149, doi: 10.1109/SNPD.2019.8935813.
[4] "TensorFlow," [Online]. Available: https://en.wikipedia.org/wiki/TensorFlow.
[5] TensorFlow, "tensorflow Release 1.0.0," [Online]. Available: https://github.com/tensorflow/tensorflow/blob/07bb8ea2379bd459832b23951fb20ec47f3fdbd4/RELEASE.md. [Accessed September 2 2021].
[6] C. Metz, "TensorFlow, Google's Open Source AI, Points to a Fast-Changing Hardware World," 15 November 2015.
[7] Machine Learning: Google I/O, 2016
[8] About," [Online]. Available: https://opencv.org/about/.
[9] The Face Mask Detection For Preventing the Spread of COVID-19," in 2020 3rd International Conference on Applied Engineering (ICAE), 2020
[10] "Convolutional neural network," [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network.
[11] Roger S. Pressman, Bruce R. Maxim, Software Engineering : A Practitioner's Approach, Eighth Edition,McGRAW-Hill, 2015, pp. 777, 778, 780.
[12] B. R. M. Roger S. Pressman, Software Engineering: A Practitioner's Approach, 8th Edition, McGraw-Hill, 2015, p. 466.
[13] B. R. M. Roger S. Pressman, Software Engineering: A Practitioner's Approach, 8th Edition, McGraw-Hill, pp. 473, 487.
[14] B. E., Essays On Object-Oriented-Software-Engineering, vol. 1, Addison-Weasly, 1993.
[15] R. Binder, Design for Reuse Is for Real, vol. 6, American Programmer, 1993, pp. 33-37.