

## Covid symptom detection + Mask detection

Snehal Batule<sup>1</sup>, Shrushti Dongare<sup>2</sup>, Vedant Kulkarni<sup>3</sup>, Akshay Dawkare<sup>4</sup>, Harshada Joshi<sup>5</sup>

<sup>1</sup>(Student, N.B.N.School of Engineering, Ambegaon (Bk.) Pune), [batulesnehal2704@gmail.com](mailto:batulesnehal2704@gmail.com),

<sup>2</sup>(Student, N.B.N.School of Engineering, Ambegaon (Bk.) Pune) [shrushtidongre06@gmail.com](mailto:shrushtidongre06@gmail.com),

<sup>3</sup>(Student, N.B.N.School of Engineering, Ambegaon (Bk.) Pune) [ved.k96@gmail.com](mailto:ved.k96@gmail.com),

<sup>4</sup>(Student, N.B.N.School of Engineering, Ambegaon (Bk.) Pune), [akshaydawkare@gmail.com](mailto:akshaydawkare@gmail.com),

<sup>5</sup>(Assistant Professor, N.B.N.School of Engineering, Ambegaon (Bk.) Pune) [harshadajoshi93@gmail.com](mailto:harshadajoshi93@gmail.com)

### To Cite this Article

Snehal Batule, Shrushti Dongare, Vedant Kulkarni, Akshay Dawkare, Harshada Joshi, "COVID SYMPTOM DETECTION + MASK DETECTION", *Journal of Science and Technology*, Vol. 06, Special Issue 01, August 2021, pp06-10..

### Article Info

Received: 15.07.2021

Revised: 24.07.2021

Accepted: 10.08.2021

Published: 16.08.2021

**Abstract** – Head pose classification is widely used for the preprocessing before face recognition and multi-angle problems, because algorithms such as face recognition often require the input image to be a front face. But affected by the COVID-19 pandemic, people wear face masks to protect themselves safe, which makes cover most areas of the face. This makes some common algorithms cannot be applied to head pose classification in the new situation. Therefore, this project established a method HGL to deal with the head pose classification by adopting color texture analysis of images and line portrait. The proposed HGL method combines the H channel of the HSV color space with the face portrait and grayscale image, and train the CNN to extract features for classification and its hardware pulse sensor check the oxygen level of person and notify and also check Temperature by using temperature Sensor. Notify the level of Temperature low or not. The evaluation on MAF dataset shows that compared with the algorithms based on facial landmark detection and convolutional neural network, the proposed method has achieved a better performance.

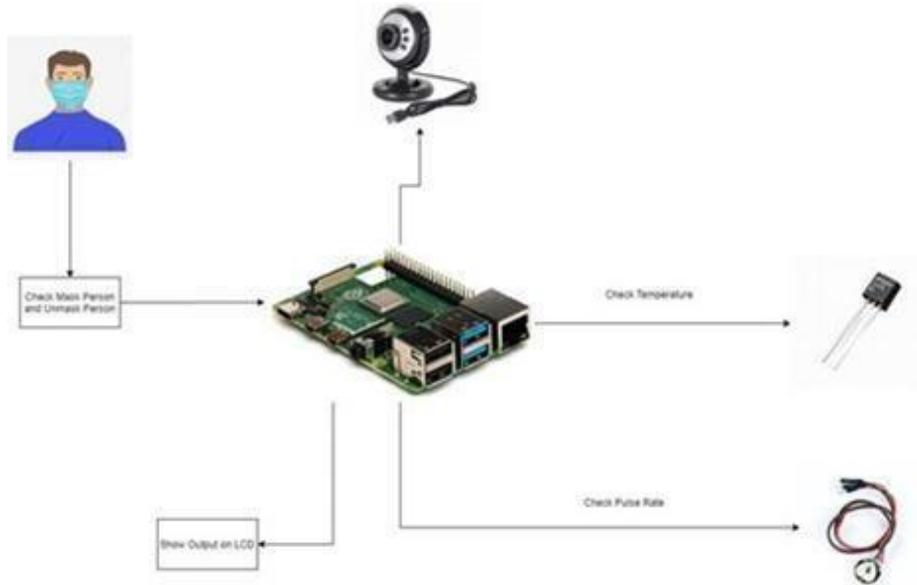
**Keywords**—Generation, Maglev turbine, Solar tracking, Wind power generation, Ard

## I. Introduction

Presented that coronavirus disease 2019 (COVID-19) has globally infected over 2.7 million people and caused over 180,000 deaths. In addition, there are several similar large scale serious respiratory diseases, such as severe acute respiratory syndrome (SARS) and the Middle East respiratory syndrome (MERS), which occurred in the past few years. Liu et al. reported that the reproductive number of COVID-19 is higher compared to the SARS. Therefore, more and more people are concerned about their health, and public health is considered as the top priority for governments. Fortunately, Leung et al. showed that the surgical face masks could cut the spread of coronavirus. At the moment, WHO recommends that people should wear face masks if they have respiratory symptoms, or they are taking care of the people with symptoms. Furthermore, many public service providers require customers to use the service only if they wear masks. Therefore, face mask detection has become a crucial computer vision task to help the global society, but research related to face mask detection is limited.

## II. Construction And Methodology

### System architecture



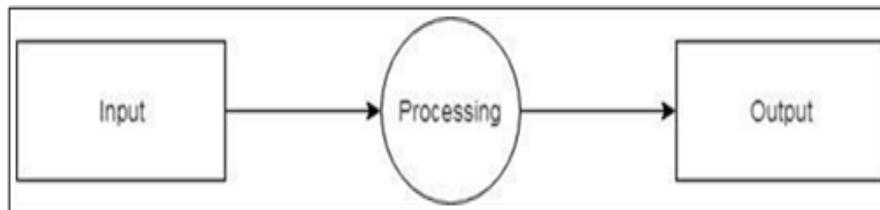
Figureno 1: System Architecture

### Module

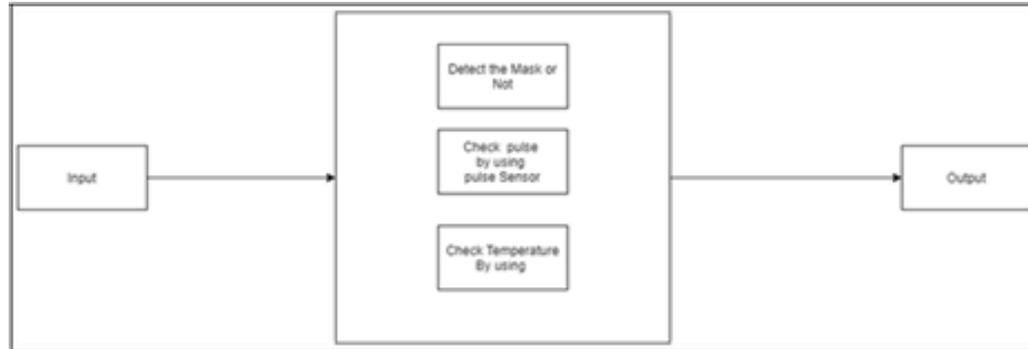
Pre-processing: User login/Registration for security purpose in this model Camera take a image of mask and unmask person. Feature Extraction: In this model given of images they remove unwanted data, remove noisy part of image. classification: In this model they apply algorithm for classifying the image to detect unmask person then show final output.

### Data Flow Diagram

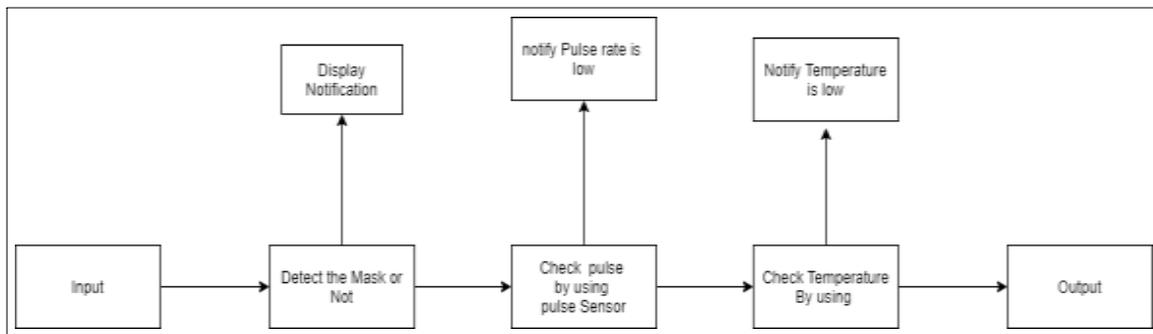
In Data Flow Diagram, we show that flow of data in our system in DFD0 we show that base DFD in which rectangle present input as well as output and circle show our system, In DFD1 we show actual input and actual output of system input of our system is text or image and output is Unmask person detected likewise in DFD 2 we present operation of user as well as admin.



Figureno 2: Data flow diagram1



**Figureno 3:**Data flow diagram 2

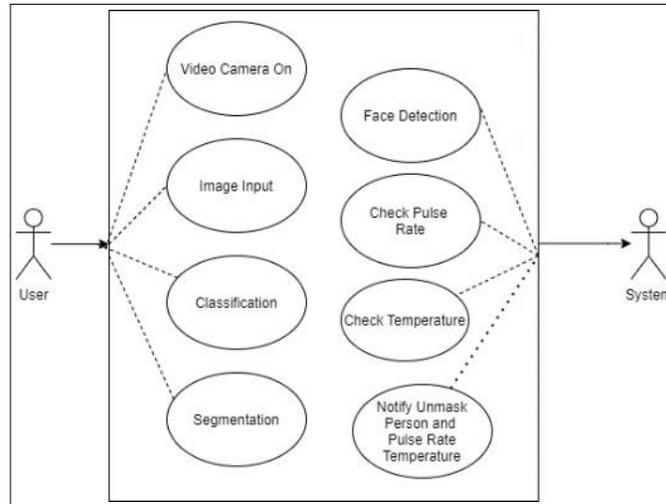


**Figureno 4:**Data flow diagram 3

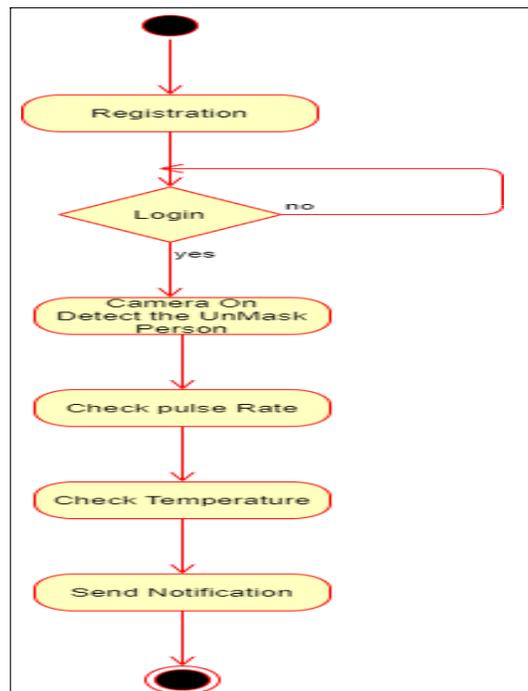
### Umidigram

Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system. UML is process independent, although optimally it should be used in process that is usecase driven, architecture-centric, iterative and incremental. The Number of UML Diagram is available.

Use case Diagram. Component Diagram. Activity Diagram. Sequence Diagram



**Figureno 5:** UseCaseDiagram



**Figureno 6:** SequenceDiagram



### **Software information**

Python is an interpreter, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release." [30] No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development. Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project.

Alternative to the command line interface (CLI). The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and there is no conda exists. When pip install a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages [citation needed]. It will install a package and any of its dependencies regardless of the state of the existing installation [citation needed]. Because of this, a user with a working installation of, for example, GoogleTensorFlow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by TensorFlow. In some cases, the package may appear to work but produce different results in detail. In contrast, conda analyses the current environment including everything currently installed, and, to gather with any version limitations specified (e.g. the user may wish to have TensorFlow version 2.0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done. Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda 2 includes Python 2.7 and Anaconda 3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

### **Anaconda**

Anaconda: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and mac OS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free. Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Mini conda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the anaconda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical bit.

### **III. Conclusion**

We will be building a model which will detect whether a person is wearing a mask or not and will also determine if the person has low temperature or oxygen ratio to notify, so we reduce Covid Patient.

### **IV. Futurescope**

As experts forecast a future with more pandemics, rising levels of air pollution, persisting authoritarian regimes and a projected increase in bushfires producing dangerous smoke – it's likely mask-wearing will become the norm for at least a proportion of us.

### **REFERENCES**

- [1]. V. Chamola et al., "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Block chain, and 5G in Managing its Impact," *IEEE Access*, vol. 8, 2020, pp. 90225–65
- [2]. Y. Zhang et al., "Edge Intelligence in the Cognitive Internet of Things: Improving Sensitivity and Interactivity," *IEEE Network* vol. 33, no. 3, May/June 2019, pp. 58–64.
- [3]. M. Arshad et al., "5G and Explainable Deep Learning Assisted Healthcare Vertical at the Edge-COVID-19 Perspective," *IEEE Network*, vol. 34, no. 4, July/August 2020. DOI: 10.1109/MNET.011.2000353, pp. 1–8.
- [4]. G. Muhammad, M.F. Alhamid, and X. Long, "Computing and Processing on the Edge: Smart Pathology Detection for Connected Healthcare," *IEEE Network*, vol. 33, no. 6, Nov.-Dec. 2019, pp. 44–49.
- [5]. S. M. Lundberg et al., "From Local Explanations to Global Understanding with Explainable AI for Trees," *Nat. Mach. Intell.* 2, 2020, pp. 56–67.
- [6]. A. Srinivasan, L. Vig, and M. Bain, "Logical Explanations for Deep Relational Machines Using Relevance Information," *J. Machine Learning Research*, vol. 20, 2019, pp. 1–47.
- [7]. S. Andreev et al., "Future of Ultra-Dense Networks Beyond 5G: Harnessing Heterogeneous Moving Cells," *IEEE Commun. Mag.*, vol. 57, no. 6, June 2019, pp. 86–92.
- [8]. A. Ahlert et al., "Review of Block chain-Based Distributed Energy: Implications for Institutional Development," *Renewable and Sustainable Energy Reviews*, vol. 107, 2019, pp. 200–11.