CHAPTER 1

INTRODUCTION

Face mask detection has emerged as a very interesting problem in image processing and computer vision. It has a range of applications from facial motion capture to face recognition which at the start needs the face to be detected with a very good accuracy. Face detection is more relevant today because it not only used on images but also in video applications like real time surveillance and face detection in videos. High accuracy image classification is possible now with the advancements of Convolutional networks. Pixel level information is often required after face detection which most face detection methods fail to provide. Obtaining pixel level details has been a challenging part in semantic segmentation. Semantic segmentation is the process of assigning a label to each pixel of the image. In our case the labels are either face or non-face. Semantic segmentation is thus used to separate out the face by classifying each pixel of the image as face or background. Also, most of the widely used face detection algorithms tend to focus on the detection of frontal faces.

This paper proposes a model for face detection using semantic segmentation in an image by classifying each pixel as face and non-face i.e., effectively creating a binary classifier and then detecting that segmented area. The model works very well not only for images having frontal faces but also for non-frontal faces. The paper also focuses on removing the erroneous predictions which are bound to occur. Semantic segmentation of human face is performed with the help of a fully convolutional network.

The ultrasonic sensor sends out 8 pulses of ultrasonic sound when you pull the trigger line high, these sound waves travel with the speed of sound. When the waves hit an obstacle, they bounce back, and the sensor receives the waves. The sensor then pulls the echo pin high for a few milliseconds. When connecting this sensor to a Raspberry pi, it is possible to measure the time between sending and receiving the pulses. Once we detect a person, with the use of Infrared sensor will detect the temperature of that person.

1.1 HISTORY OF MOBILENETV2

MobileNetV1 is a family of general-purpose computer vision neural networks designed with mobile devices in mind to support classification, detection and more. The ability to run deep networks on personal mobile devices improves user experience, offering anytime, anywhere access, with additional benefits for security, privacy, and energy consumption. As new applications emerge allowing users to interact with the real world in real time, so does the need for ever more efficient neural networks.

Today the availability of MobileNetV2 to power the next generation of mobile vision applications. MobileNetV2 is a significant improvement over MobileNetV1 and pushes the state of the art for mobile visual recognition including classification, object detection and semantic segmentation. MobileNetV2 builds upon the ideas from MobileNetV1, using depth wise separable convolution as efficient building blocks. However, V2 introduces two features to the architecture:

1. Linear bottlenecks between the layers

2. Shortcut connections between the bottlenecks

The basic structure is show below.

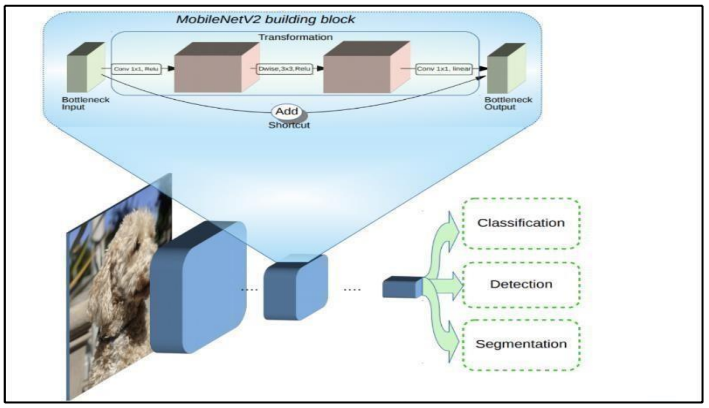

Figure 1.1: Overview of MobilenetV2 Architecture.

In the above figure1.1.1 the blue blocks represent composite convolutional building blocks. The intuition is that the bottlenecks encode the model's intermediate inputs and outputs while the inner layer encapsulates the model's ability to transform from lower-level

concepts such as pixels to higher level descriptors such as image categories. Finally, as with traditional residual connections, shortcuts enable faster training and better accuracy. Overall, the MobileNetV2 models are faster for the same accuracy across the entire latency spectrum. In particular, the new models use 2x fewer operations, need 30% fewer parameters and are about 30-40% faster on a google pixel phone than MobileNetV1 models, all while achieving higher accuracy.

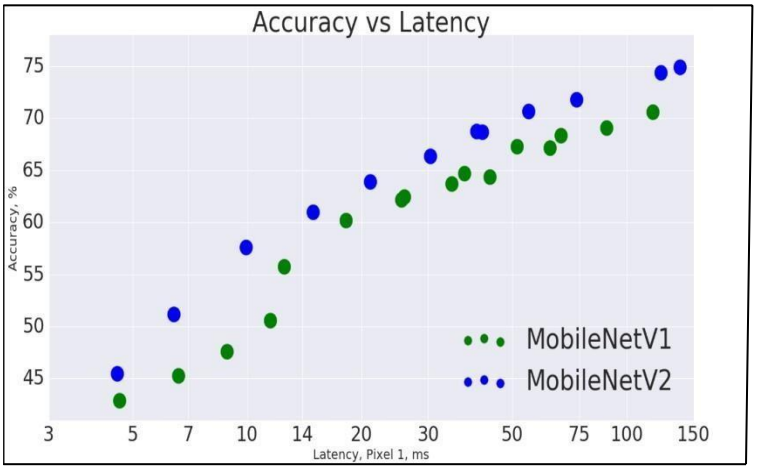

Figure 1.2: Accuracy graph of MobileNetV2

The above figure 1.1.2 illustrates that the MobileNetV2 improves speed (reduces latency) and increased ImageNet top 1 accuracy. MobileNetV2 is a very effective feature extractor for object detection and segmentation. For example, for detection when paired with newly introduced SSDLite the new model is about 35% faster with the same accuracy than MobileNetV1. To enable on-device semantic segmentation, we employ MobileNetV2 as a feature extractor in a reduced form of DeepLabv3, that was announced recently. On the semantic segmentation benchmark, PASCAL VOC 2012, our resulting model attains a similar performance as employing MobileNetV1 as feature extractor but require 5.3 times fewer parameters and 5.3 times fewer parameter operations in terms of Multiply-Adds. MobileNetV2 provides aa very efficient mobile-oriented model that can be used as a base for many visual recognition tasks.

1.2 PROBLEM STATEMENT

The end of 2019 witnessed the outbreak of coronavirus disease covid-19 which has continued to be the cause of flight for millions of lives and business even in 2020. As the world recovers from the pandemic and plans to return to a state of normally there is a wave of anxiety among all individuals especially those who intend to resume in person activity. Studies have proved that wearing mask significantly reduces the risk of viral transmission as well as provides a sense of protection. However, is not feasible to manually track the implementation of this policy. Technology hold the key here: we introduce a deep learning-based system that can detect instances where face mask or not used properly and to check the human temperature. This will help track safety violation promote the use of face mask and ensure a safe working environment.

1.3 EXISTING SYSTEM

Traditional object detection uses a multi-step process. A well-known detector is the Viola-Joins detector, which can achieve real-time detection. The algorithm extracts feature by Haar feature descriptor with an integral image method, selects useful features,and detects objects through a cascaded detector. Although it utilizes integral image to facilitate the algorithm, it is still very computationally expensive. In for human detection, an effective feature extractor called HOG is proposed, which computes the directions and magnitudes of oriented gradients over image cells. Later, deformable part-based model (DPM) detects objects parts and then connects them to judge classes that objects belong to. Rather than using handcrafted features, deep learning-based detector demonstrated excellent performance recently, due to its robustness and high feature extraction capability. There are two popular categories, one-stage object detectors and two-stage object detectors. Two-stage detector generates region proposals in the first stage and then fine-tune these proposals in the second stage.

The two-stage detector can provide high detection performance but with low speed. The seminal work R-CNN is proposed by R. Girshick et al. R-CNN uses selective search to propose some candidate regions which may contain objects. After that, the proposals are fed into a CNN model to extract features, and a support vector machine (SVM) is used to

recognize classes of objects. However, the second stage of R-CNN is computationally expensive since the network must detect proposals on a one-by-one manner and uses a separate SVM for final classification.

1.3.1 DRAWBACKS OF EXISTING SYSTEM

Due to the limited size of the face mask dataset, it is difficult for learning algorithms to learn better features. As deep learning-based methods often require larger dataset, transfer learning is proposed to transfer learned knowledge from a source task to a related target task. According to, transfer learning has helped with the learning in a significant way if it has a close relationship. In our work, we use parts of the network pretrained on a largescale face detection dataset - Wider Face, which consists of 32,203 images and 393,703 annotated faces. In Retina Face Mask, only the parameters of the backbone and neck are transferred from Wider Face, the heads are initialized by Kaiming's method. In addition, pre trained Imagenet weights are considered as a standard initialization of our backbones in basic cases. Post processing on the predicted mask obtained is performed so that the irregularities in the region can be filled and to remove the unwanted errors (which may have crept during the processing). This we perform by first passing the mask through Median filter and then performing the Closing Operation.

1.3.2 LIMITATIONS OF EXISTING SYSTEM

- Time consuming is more.
- Accuracy is high
- Not user friendly: The existing system is not user friendly because the retrieval of data is very slow, and data is not maintained efficiently.
- Time consuming: Every work is done manually so we cannot generate report in the middle of the session or as per the requirement.
- Managing a very large database is difficult.

1.4 REASONS BEHIND SELECTING THE PROJECT

We now know from recent studies that a significant portion of individuals with coronavirus lack symptoms (‘asymptomatic’) and that even those who eventually develop Symptoms (‘pre-symptomatic’) can transmit the virus to others before showing symptoms, according to the advisory published by the CDC “This means that the virus can spread between people interacting in close proximity -for example, speaking, coughing or sneezing -even if those people are not exhibiting symptoms. Therefore, wearing face mask is so important, hence we can reduce the spread of virus.

1.5 PROPOSED SYSTEM

To design an effective network for face mask detection, we adopt the object detector framework proposed in which suggests a detection network with a backbone, a neck, and heads. The backbone refers to a general feature extractor made up of convolutional neural networks to extract information in images to feature maps. we adopt ResNet as a standard backbone, but also include MobileNet as a backbone for comparison and for reducing computation and model size in deployment scenarios with limitedcomputing resources. In terms of the neck, it is an intermediate component between a backbone and heads, and it can enhance or refine original feature maps. FPN (Feature Pyramid Network) is applied as a neck, which can extract high-level semantic information and then fuse this information into previous layers’ feature maps by adding operation with a coefficient. Finally, heads stand for classifiers, predictors, estimators, etc., which can achieve the final objectives of the network. we adopt a similar multi-scale detection strategy as SSD to make a prediction with multiple FPN feature maps, because it can havedifferent receptive fields to detect various sizes of objects. Particularly, Face Mask utilizesthree feature maps, and each of them is fed into a detection head. Inside each detection head, we further add a context attention module to adjust the size of receptive fields and focus on specific areas, which is like Single Stage Headless (SSH) but with an attention mechanism. The output of the detection head is through a fully convolutional network

rather than a fully connected network to further reduce the number of parameters in the network. We name the detector as since it follows the architecture of which consists of a SSD and a FPN, and able to detect small face masks as well.

We propose a method of obtaining segmentation masks directly from the images containing one or more faces in different orientation. The input image of any arbitrary size is resized to 224 × 224 × 3 and fed to the FCN network for feature extraction and prediction. The output of the network is then subjected to post processing. Initially the pixel values of the face and background are subjected to global threshold. After that it is passed through median filter to remove the high frequency noise and then subjected to Closing operation to fill the gaps in the segmented area. After this bounding box is drawn around the segmented area.

1.6 OBJECTIVES

The objective of the project is

Extract's feature and predicts class input from digital camera video. Semantically segments the face present in that image & removes the unwanted noise and avoids the false class prediction. Detects multiple face mask in single frame. The proposed face mask is a one stage detector, which consists of a feature pyramid network to fuse high-level semantic information with multiple feature maps, and a novel context attention module to focus on detecting face masks. In addition, we also propose a novel cross-class object removal algorithm to reject predictions with low confidences and the high intersection of union. Detects the person, if any person isn't wearing mask then that person is identified in using face detection method and a warning email is sent.

CHAPTER 2

LITERATURE REVIEW

A literature review is a scholarly paper, which includes the current knowledge including the substantive findings, as well as theoretical and methodological contribution to a particular topic. Its basis for research in nearly every academic field and it may also be part of graduate and post-graduate student works, including the preparation of thesis, dissertation, or a journal article.

2.1 LITERATURE SURVEY

T. Ojala, M. Pietikainen, and T. Maenppa, published "Multi resolution gray scale and rotation invariant texture classification with local binary patterns," IEEE transactions a Pattern Analysis and Machine Intelligence, July 2002. This paper presents atheoretically very simple, yet efficient, multi resolution approach to gray-scale and rotation invariant texture classification based on local binary patterns and nonparametric discrimination of sample and prototype distributions. The method is based on recognizing that certain local binary patterns, termed ªuniform, º are fundamental properties of local image texture and their occurrence histogram is proven to be a very powerful texture feature. We derive a generalized gray-scale and rotation invariant operator presentation thatallows for detecting the ªuniformº patterns for any quantization of the angular space and for any spatial resolution and presents a method for combining multiple operators for multi resolution analysis. The proposed approach is very robust in terms of gray-scale variations since the operator is, by definition, invariant against any monotonic transformation of the gray scale. Another advantage is computational simplicity as the operator can be realized with a few operations in a small neighborhood and a lookup table. Excellent experimental results obtained in true problems of rotation invariance, where the classifier is trained at one particular rotation angle and tested with samples from other rotation angles, demonstrate that good discrimination can be achieved with the occurrence statistics of simple rotation invariant local binary patterns. These operators characterize the spatial configuration of local image texture and the performance can be further improved

by combining them with rotation invariant variance measures that characterize the contrast of local image texture. The joint distributions of these orthogonal measures are shown to be very powerful tools for rotation invariant texture analysis.

T. H. Kim, D. C. Park, D. M. Woo, T. Jeong, and S. y. Min, published "Multiclass classifier-based adaboost algorithm," in Proceedings of the Second Sino foreign interchange Conference on Intelligent Science and Intelligent Data Engineering, ser. IScIDE'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 122–127. A multi-class classifier based AdaBoost algorithm for the efficient classification of multi-class data is proposed in this paper. The traditional AdaBoost algorithm is basically a binary classifier and it has limitations when applied to multi-class data problems even though its multi-class versions are available. To overcome the problems of the AdaBoost algorithm for multi-class classification problems, we devise a AdaBoost architecture with its training algorithm that uses multi-class classifiers for its weak classifiers instead of series of binary classifiers. The proposed AdaBoost architecture can save its training time drastically and obtain more stable and more accurate classification results than a typical multi-class AdaBoost architecture based on binary weak classifiers. Experiments on an image classification problem with collected satellite image database are performed. The results show that the proposed AdaBoost architecture can reduce its training time 50%- 70% depending on the number of training rounds while maintaining its classification accuracy competitive when compared to Adaboost.M2.

P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Computer. Vision, vol. 57, no. 2, pp. 137–154, May 2004. This face detection system is most clearly distinguished from previous approaches in its ability to detect faces extremely rapidly. Operating on 384 by 288-pixel images, faces are detected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates. The first contribution of this work is a new image representation called an integral image that

allows for very fast feature evaluation. Motivated in part by the work of Papa-Georgiou et al. our detection system does not work directly with image intensities. Like these authors we use a set of features which are reminiscent of Haar Basis functions. To computethese features very rapidly at many scales we introduce the integral image representation for images. The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Harr-like features can be computed at any scale or location in constant time.

P. Viola and M. Jones, published "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers [6]. The third contribution is a method for combining increasingly more complex classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising objectlike regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest. In the domain of face detection, the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection.

Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, published "Towards real world human parsing: Multiple-human parsing in the wild," CoRR, vol. abs/1705.07206.The majority of existing human parsing methods formulate the task as semantic segmentation, which regard each semantic category equally and fail to exploit the intrinsic physiological structure of human body, resulting in inaccurate results. In this paper, we design a novel

semantic neural tree for human parsing, which uses a tree architecture to encode physiological structure of human body and designs a course to fine process in a cascade manner to generate accurate results. Specifically, the semantic neural tree is designed to segment human regions into multiple semantic subregions (e.g., face, arms, and legs) in a hierarchical way using a new designed attention routing module. Meanwhile, we introduce the semantic aggregation module to combine multiple hierarchical features to exploit more context information for better performance. Our semantic neural tree can be trained in an end-to-end fashion by standard stochastic gradient descent (SGD) with back-propagation. Several experiments conducted on four challenging datasets for both single and multiple human parsing, i.e., LIP, PASCAL-Person-Part, CIHP and MHP-v2, demonstrate the effectiveness of the proposed method.

Krizhevsky, I. Sutskever, and G. E. Hinton, published "Image net classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully connected layers with a final 1000-way SoftMax. To make training faster, we used no saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully connected layers we employed a recently developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

J.Simonyan and A. Zisserman, published "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of

increasing depth using an architecture with very small (3x3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16-19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localization and classification tracks respectively. We also show that our representations generalize well to other datasets, where they achieve state-of-the-art results. We have made our two-best performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, published “Going deeper with convolutions,” 2015**.**We propose a deep convolutional neural network architecture codenamed "Inception", which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One incarnation used in our submission for ILSVRC 2014 is called Google Net, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

K. He, X. Zhang, S. Ren, and J. Sun, published “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016. Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers - 8× deeper than VGG nets [40] but still having lower complexity. An ensemble of these residual nets achieves

3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions1, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

k. Li, G. Ding, and H. Wang, published "L-fcn: A lightweight fully convolutional network for biomedical semantic segmentation," in 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec 2018, pp. 2363–2367. For the past few years, deep learning-based methods have been widely used in the field of biomedical imaging. In biomedical image processing, the typical application of deep learning is semantic segmentation. However, the classical deep learning methods require higher hardware consumption and computational costs. To resolve this problem, we propose a new lightweight fully convolutional network (L-FCN). L-FCN consists of traditional watershed algorithm and fully convolutional network, which eliminates useless non-edge pixels to improves the efficiency of semantic segmentation. Experimental evaluations on ISBI 2012 dataset indicate that our L-FCN algorithm outperforms U-net intime efficiency and computational costs, meanwhile, maintain approximate image segmentation effect. Due to its portability, L-FCN can be applied to many biomedical areaseasily.

X. Fu and H. Qu.,1 published "Research on semantic segmentation of high-resolution remote sensing image based on full convolutional neural network," in 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Dec 2018. Remote sensing data is an important way to reflect the comprehensive information of surface. In this paper, based on the semantic segmentation of high-resolution remote sensing images, a segmentation method based on full convolutional neural network (FCN) is proposed. The method improves the traditional convolutional neural network (CNN) and replaces the final fully connected layer of the CNN network with a convolutional layer. And then optimize the convolution operation by using the matrix expansion technique. The

experimental results show that the FCN network with sufficient training and fine-tuning can effectively perform automatic semantic segmentation of high-resolution remote sensing images. The correct segmentation accuracy is higher than 85%, which improves the efficiency of convolution operations.

S. Kumar, A. Negi, J. N. Singh, and H. Verma, published "A deep learning for brain tumor MRI images semantic segmentation using fcn," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), Dec 2018, pp. 14.Attractive significance image base medicinal picture study and how to improve detection of brain tumours' MRI be achievement thought inside current period outstanding towards augmented require of competent with accurate reports using semantic segmentation. In this paper researchers are tended use All convolution network method in replacement of FULL CONVOLUTION NETWORK and to detect brain tumor more effectively and accurately we are going to use a combination of DEEP LEARNING and MACHINE LEARNING which will make detection in a better way. The ALL CONVOLUTION NETWORK will use CONV which are designed for a modern convolution network. In the case of assessing and identifying the tumor instead of using 2d detection and dice cut here we are going to use 3d segmentation in detection which makes it more accurate. Different algorithms are work best for different sub regions and fusing several best algorithms will yield a good result in full segmentation with the help of FCN. And FCN to CONV layer replacement means great reduction in the number of parameters. It's cool to save the memory, but it's loss of flexibility, nevertheless. In my experiments, CIFAR-10 classification accuracy dropped slightly after this single change, though it was certainly within one standard deviation. But I'm sure there are cases when this loss of flexibility makes a bigger impact.

CHAPTER 3

SYSTEM REQUIREMENTS

The system requirements specify features, components and behavior of system which is to be developed. The following sections describe about functional, non-functional, performance related, features and behavior of the solution. This includes the detailed description of the solution to be developed.

3.1 Functional Requirement

In software engineering, a functional requirement defines a system or its component. It describes the functions software must perform. A function is nothing but inputs, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification.

Functional Requirements are those requirements which show the working and functionality of a system and the expected behavior of a system based on certain situations and inputs. It defines specific functionality of a system.

Functional requirements of system are:

- Take digital camera video as input. Input video frames are fed into Fully Convolutional Network.
- Extract feature and class prediction using pre-defined training weights of MobileNetV2 architecture.
- The output of the network is fed into post processing.
- Initially pixel values of the face and background are subjected to global threshold.
- Then subjected to closing operation to fill the gaps in the segmented area.

3.2 Non-functional Requirement

- **Reliability** The model ought to be dependable and solid in giving the functionalities. When a people have encountered near the trained model the progressions must be made unmistakable by the model. The progressions made by the programmer ought to be unmistakable both to the project pioneer and in addition the test designer.
- **Maintainability** The trained model observing, and upkeep ought to be basic and target in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the face mask detections are running without lapses.
- **Performance** The model will be utilized by numerous people entering all the while. Since the model as trained using mobilenetv2 with a solitary database server out of sight, execution turns into a noteworthy concern. The trained model ought not to succumb when numerous people would encounter near the trained model all the while. It ought to permit quick facemask detection and temperature screening to people encountered. For instance, if multiple people encountered our model is well trained to detect multiple people in single frame.
- **Configuration and compatibility** Describe requirements such as those connected with individual customization or operations in specific competing environment.
- **Flexibility** is the capacity of a trained model to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable model is one that is anything but difficult to reconfigure or adjust because of diverse client and model prerequisites.
- **Usability** Describes items that will ensure the user friendliness of the software. Example: includes error messages that direct the user to a solution, input range checking soon as entries are made and order of choices and screen corresponding to user performances.

3.3 Hardware and Software Requirements

Hardware Requirements

Processor	: Pentium IV 2.4 GHz
RAM	: 1GB
Hard Disk	: 250 GB
Mouse	: Optical
Keyboard	: Multimedia

Software Requirements

Coding language	: Python
IDE Tool used	: Jupiter Notebook
Operating System	: Windows 7

3.4 RESOURCE REQUIREMENTS:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

CHAPTER 4

SYSTEM ANALYSIS

Analysis is the process of breaking a complex topic or substance into smaller parts to gain a better understanding of it. Analysts in the field of engineering look at requirements, structures, mechanisms, and systems dimensions. Analysis is an exploratory activity. The Analysis Phase is where the project lifecycle begins. The Analysis Phase is where you break down the deliverables in the high-level Project Charter into the more detailed business requirements. The Analysis Phase is also the part of the project where you identify the overall direction that the project will take through the creation of the project strategy documents.

4.1 Feasibility Study

The achievability of the scheme is scrutinized throughout this part and commerce tender is place onward with a terribly general organize for the project and a few prices approximate. All through system psychotherapy the viability study of the predictable system is to be prearranged this is habitually to make sure that the anticipated system isn't a lumber to the commercial. For viability analysis, some indulgent of the chief provisions for the structure is imperative.

Three key anxieties anxious within the achievability analysis are:

1. Economic Feasibility
2. Technical Feasibility
3. Social Feasibility
4. Operational Feasibility

1. Economic Feasibility

This study is mete out to foresee the financially viable crash that the scheme can show off the union the magnitude of endowment that the communal will decant into the psychoanalysis and enlargement of the scheme is prohibited. The expenditures ought to be even consequently the residential system in totaling inside the financial statement and this

was pull off accordingly of the largest part of the equipment's used are unreservedly on the souk solely the specially made commodities had to be procure.

2. Technical Feasibility

This cram is functional to see the procedural feasibleness, to be precise, the nominal chucks of the structure. Any organism urbanized should not have a lofty stipulate on the accessible procedural where withal. this can ground high strains on the accessible procedural possessions this can cause lofty strains being sited on the punter. The urbanized organism should have an unpretentious stipulate, as exclusively ostensible, or unsound amend are looked for executing this routine.

3. Social Feasibility

The visage of cram is to make certain the coverage of receiving of the scheme by the abuser. This embraces the routine of schooling the abuser to use the organism with competence. The abuser should not undergo defenseless by the organism, as an alternative should reconcile for it as a stipulation the coverage of approval by the abusers unaided depends on the traditions that are used to instruct the abuser concerning the organism and to figure him au fait with it. His echelon of assurance should be elevated so he's conjointly proficient to craft some productive denigration, that is salutation, as he's the definitive abuser of the system.

4. Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the user about the system and to make the user familiar with it. User level of confidence must be raised so that he/she will also be able to make some constructive criticism, which is welcomed, as he/she is the final user of the system.

- **Summary** The objective of this chapter is to know the proposed system is feasible or not. The chapter mainly describe about the various keys of feasibility analysis

i.e. Economic, Technical, Social and Operational.

CHAPTER 5

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture, and systems engineering.

If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non-software systems and organizations.

System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. The design will contain the specification of all these modules, their interaction with other modules and the desired output from each module. The output of the design process is a description of the software architecture.

The design phase is followed by two sub phases

- High Level Design
- Detailed Level Design

5.1 Fundamental Design Concepts

Fundamental design is developed in course of recent years. As year's passes, enthusiasm of creating new designs is evolved and each design has been tested. Software designer gets new ideas and foundation to build and test new design concepts. Fundamental framework is design to "getting it right". A major plan idea, for example, deliberation, and refinement, modularity, and programming engineering and data encryption is applied to meet the requirement of proposed work.

5.1.1 High Level Design

In the high-level design, the proposed functional and non-functional requirements of the software are depicted. Overall solution to the architecture is developed which can handle those needs. This chapter involves the following consideration.

• System Architecture

5.1.2 Low Level Design

During the detailed phase, the view of the application developed during the high-level design is broken down into modules and programs. Logic design is done for every program and then documented as program specifications. For every program, a unit test plan is created.

- Data flow Diagram
- The information stream outline demonstrates the graphical portrayal, similar to game plans it is utilized to speak to the information through the sources of info, different sorts of information examination will be completed and the coveted yield will be produced.
- These parts will be utilized to demonstrate the framework and it will be displayed by to contemplate quickly regarding the information. In facemask detection outline the DFD will demonstrate the stream of whole parts. The stream of data will in arrangement of change utilizing this framework.
- During the detailed phase, the view of the application developed during the high-level design is broken down into modules and programs. Logic design is done for every program and then documented as program specifications. Dataflow diagrams are used to graphically represent the flow of data in a business information system.

DFD describes the processes that are involved in a system to transfer data from the input to the file storage and report generation.

- It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities.
- It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

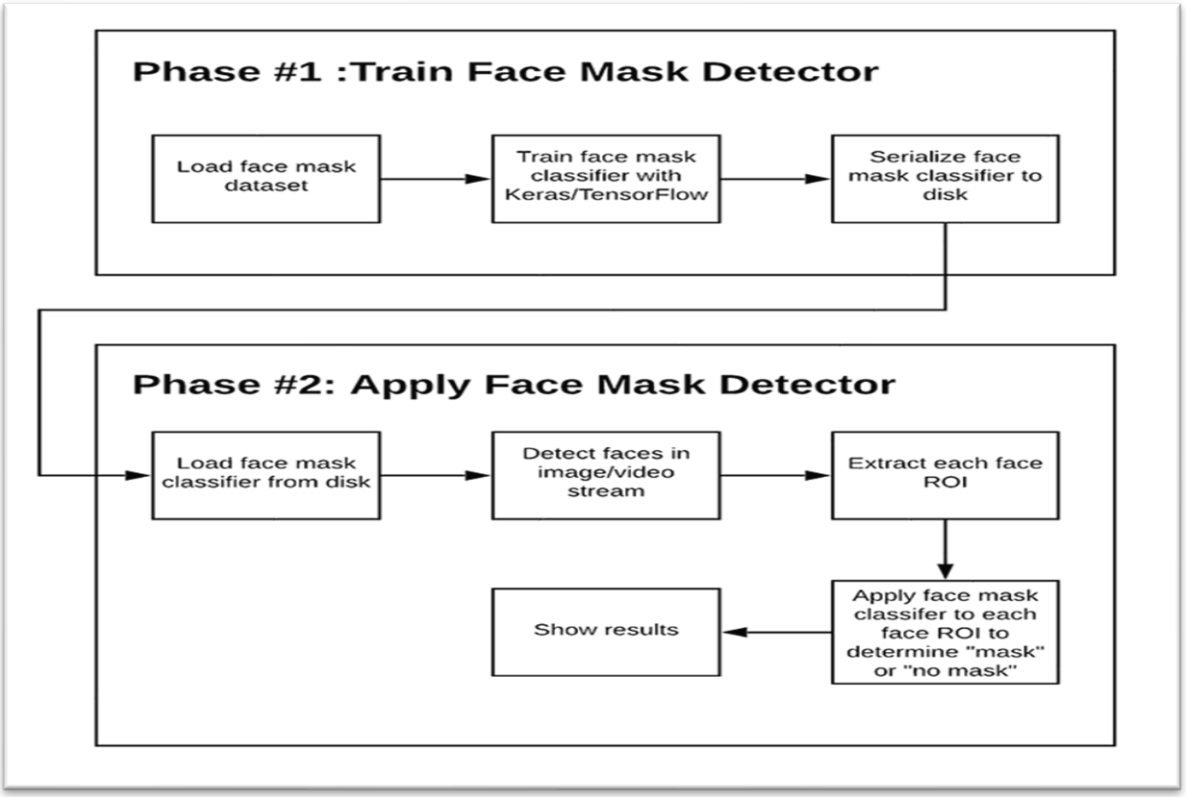

Figure 5.1: Data Flow diagram

The above diagram 5.1 illustrates the phases and individual steps for building a COVID19 face mask detector with computer vision and deep learning using Python, OpenCV, and Tensor Flow/Keras.

To train a custom face mask detector, we need to break our project into two distinct phases, each with its own respective sub-steps:

1. **Training:** here we'll focus on loading our face mask detection dataset from disk, training a model (using Keras /Tensor Flow) on this dataset, and then serializing the face mask detector to disk.
2. **Deployment:** once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with mask or without mask.

5.2 System Architecture

The diagram below shown illustrates that the Source images are also called as input tensors; these are given to MobilenetV2 as inputs. MobilenetV2 was chosen as an algorithm to build a model device. A customized fully connected layers on top of the MobilenetV2 model was developed.

The layers are:

1. Average pooling layer with 7×7 weights
2. Linear layer with ReLu activation function
3. Dropout layer
4. Linear layer with SoftMax activation function with the result of 2 values.

The final layer SoftMax activation function gives the result of two probabilities each one represents the classification of mask or no-mask.

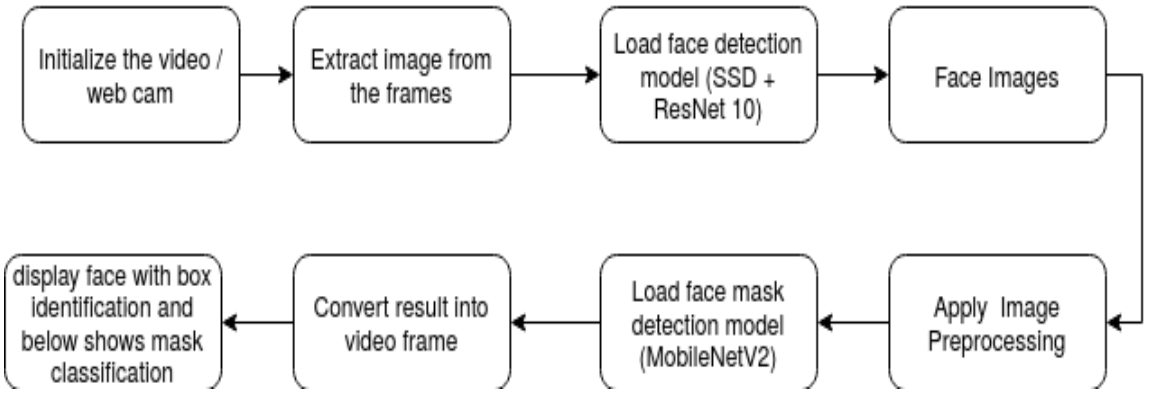

Figure 5.2: System architecture

At the initial stage we take the input from digital camera then from that input we extract the video frames and load that frames into face detection model to identify that person from the database then the image is processed and that person is recognized from the database where we have the information of that person auto-generated email is sent as a warning. Before all these things happen the person is identified whether the person is wearing face mask or not.

5.3 Use Case Diagram:

A utilization case in programming designing and frameworks building is a portrayal of a framework's conduct as it reacts to a demand that starts from outside of that framework. As it were, a utilization case depicts "who" can do "what" with the framework being referred to. The utilization case method is utilized to catch a framework's behavioral necessities by specifying situation driven strings through the useful prerequisites.

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

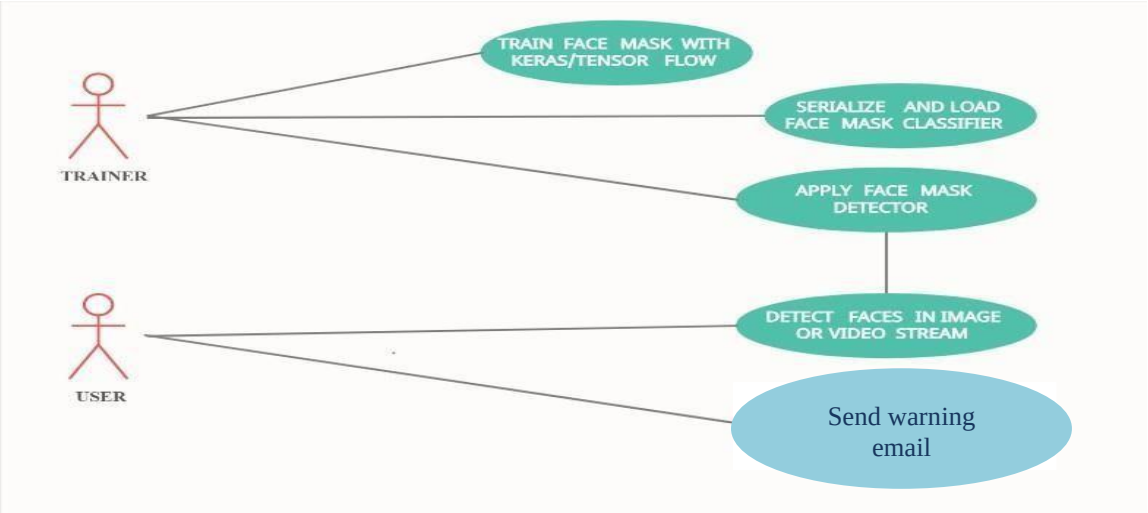

Figure 5.3: Use case Diagram of facemask detection

Given the trained COVID-19 face mask detector, we'll proceed to implement two more additional scripts used to:

- Detect covid-19 face masks in images.
- Detect face masks in real-time video streams.

5.4 Implementation of Frontend:

The MEAN stack is JavaScript-based framework for developing web applications. MEAN is named after MongoDB, Express, Angular, and Node, the four key technologies that make up the layers of the stack.

- MongoDB - document database
- Express(.js) - Node.js web framework
- Angular(.js) - a client-side JavaScript framework
- Node(.js) - the premier JavaScript web server

There are variations to the MEAN stack such as MERN (replacing Angular.js with React.js) and MEVN (using Vue.js). The MEAN stack is one of the most popular technology concepts for building web applications.

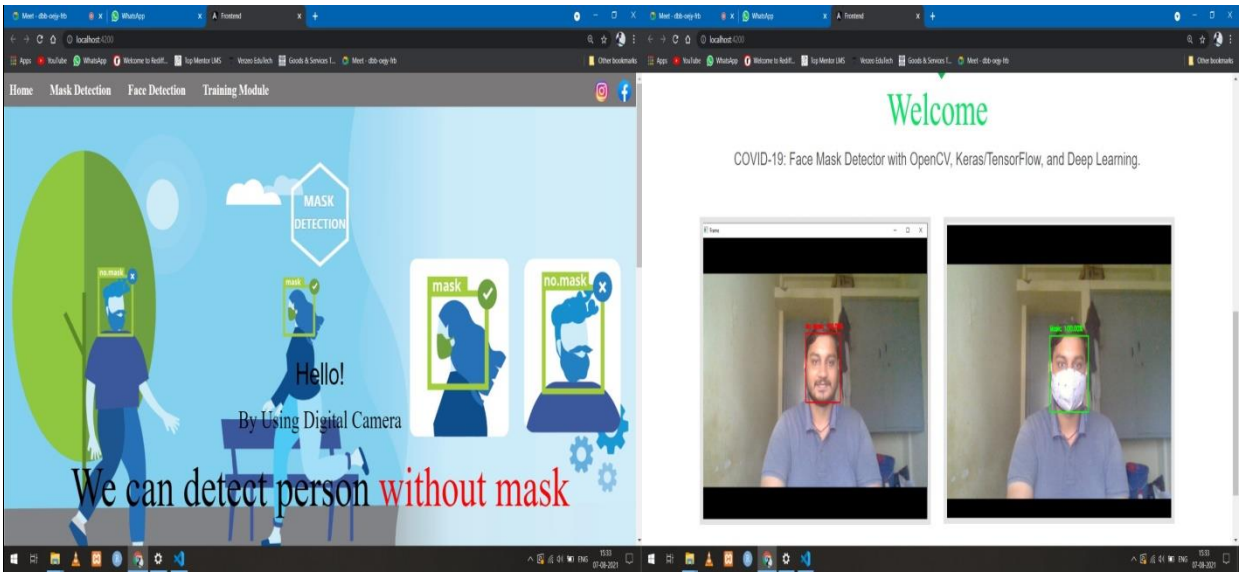

Fig: 5.4: Home Page

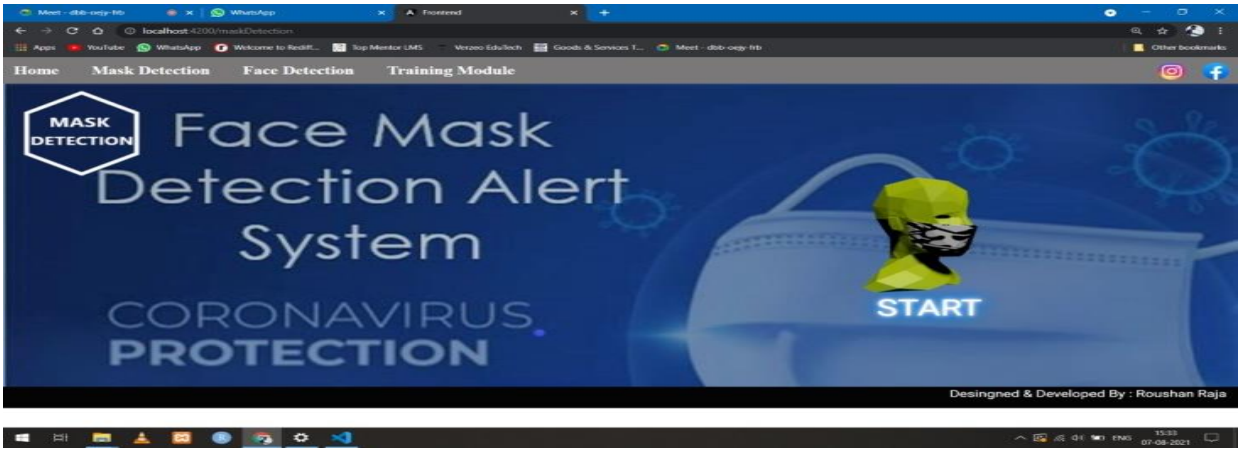

Fig:5.5: Face Mask Detection

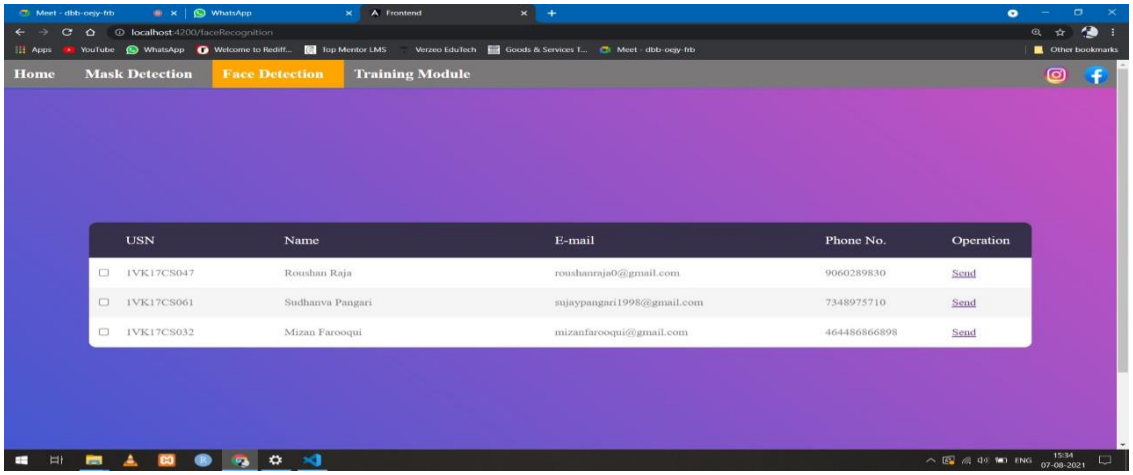

Fig:5.6: List of People Detected without Mask

CHAPTER 6

METHODOLOGY

We propose this project with twin objective of creating a Binary face classifier which can detect faces in any orientation irrespective of alignment and train it in an appropriate neural network to get accurate results. The model requires inputting an digital camera video of any arbitrary size to the model. The model's basic function is feature extraction and class prediction. The output of the model is a feature vector which is optimized using Gradient descent and the loss function used is Binomial Cross Entropy. Figure 6 represents the end to end pipeline of our method along with sample demonstration of obtained output at eachstep.

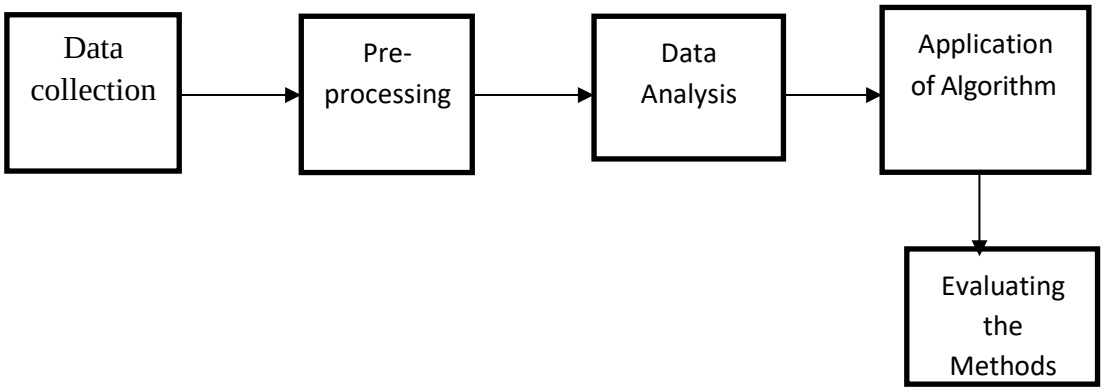

Figure 6.1: Flow Chart of face mask detection

6.1 Data Collection:

The figure 6.2 shown below consists of "with mask**"** & "without mask" images. We will use the dataset to build a covid-19 facemask detector with computervision and deep leaning using python OpenCV and tensor flow/Keras. This method is a loteasier than it sounds once you apply facial landmarks to the problem.

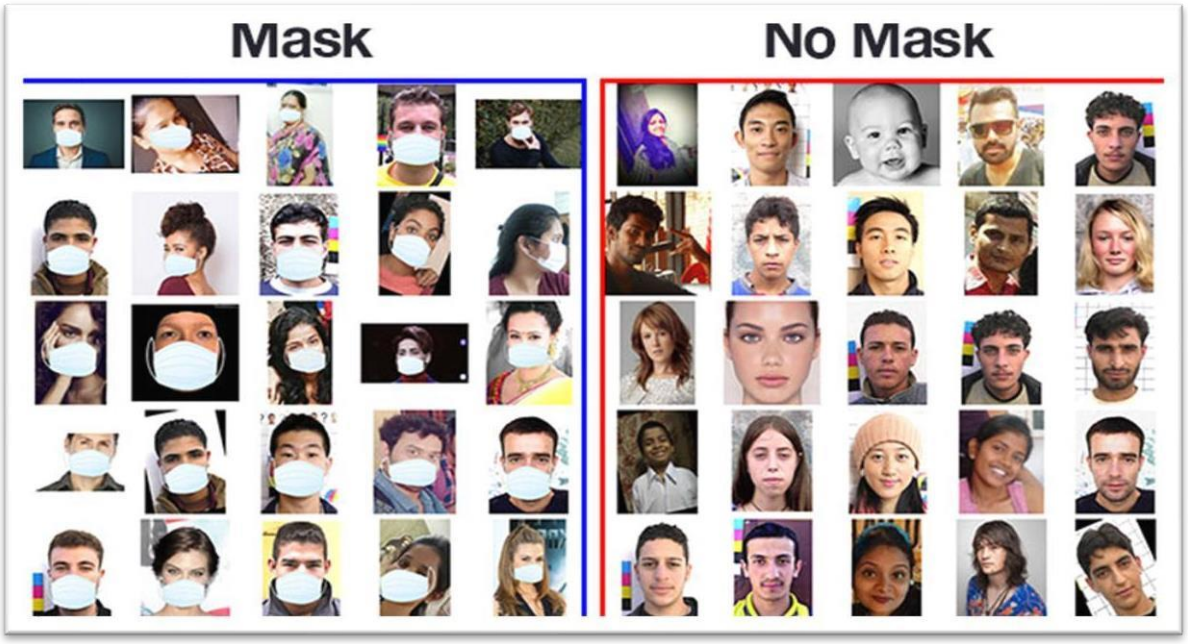

Figure 6.2: Images containing with and without Face Masks.

6.2 Model Training:

Facial landmarks allow us to automatically infer the locationof facial structures,

Including:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

To use facial landmarks to build a dataset of faces wearing face masks, we need to first start with an image of a person not wearing a face mask:

Figure 6.5.1 illustrates to build a COVID-19/Corona virus pandemic face mask dataset; we'll first start with image of a person not wearing a face mask.

Figure 6.3: Image of Face Mask Detection.

From there, we apply face detection to compute the bounding box location of the face in the image:

The next step is to apply face detection, here we have used a deep learning method to perform face detection with OpenCV.

Once we know where in the image the face is, we can extract the face Region of Interest (ROI):

Figure 6.4: Image of multiple faces not wearing mask in single frame

And from there, we apply facial landmarks, allowing us to localize the eyes, nose, mouth, etc.

Facial landmark detection is done using Dlib so that we know where to place a mask on the face. Next, we need an image of a mask (with a transparent Background).

Figure 6.5: An example of a COVID-19 Face mask.

This Mask will be overlaid on the original face ROI automatically, since we know the face landmark locations. This mask will be automatically applied to the Face by using the facial landmarks (namely the points along the chin and nose). To compute where the mask will be placed. The mask is then resized and rotated, placing it on the face:

In this above Figure the Face mask is placed on the persons face in the original frame. It is difficult to tell briefly that the COVID-19 Mask has been applied with computer vision byway of OpenCV and Dlib Face landmarks.

We can then repeat this process for all of our input images, thereby creating our artificial Face mask dataset.

Figure 6.6: An Artificial set of COVID-19 face mask images.

The dataset shown above is a part of our "with mask"/ "without mask" Dataset for COVID-19 facemask detection with computer Vision and deep learning using Python, OpenCV, and Tensor Flow/Keras.

If we use a set of images to create an artificial dataset of people wearing masks, we cannot "re-use" the images without masks in our training set-we still need to gather nonface mask images that where not used in the artificial generation process.

If we include the original images used to generate the face mask samples as nonface mask samples, our model will become heavily biased and fail to generalize well. Avoid that at all costs by taking the time to gather new examples of faces without masks.

CONCLUSION

In this survey, we present a comprehensive review of the COVID-19 pandemic. We have covered almost all the different aspects related to the pandemic and outbreak. We have covered the different stages that the COVID-19 goes through in course of its spread in an area. To give a brief comparison, we have also compared the COVID-19 pandemic with other previous pandemics in last century in terms of different statistics. The clinical features, myths, diagnosis procedure, and vaccination attempts are discussed in good detail. The global impact of the outbreak on different major industries around the globe is also discussed. Various technological institutions, research organizations, and industries are trying to use different modern technologies to manage and prevent the community spread of the outbreak. We have discussed all the latest technologies that are being used in different parts of the world for this purpose. Majorly we discuss the use of IOT, Machine Learning, Deep Learning to manage the outbreak.

REFERENCES

[1] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, July 2002.

[2] T.-H. Kim, D.-C. Park, D.-M. Woo, T. Jeong, and S.-Y. Min, "Multi-class classifier based adaboost algorithm," 2012.

[3] P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Dec 2001.

[5] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, "Towards real world human parsing: Multiple-human parsing in the wild," CoRR, vol. abs/1705.07206.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016.

[10] K. Li, G. Ding, and H. Wang, "L-fcn: A lightweight fully convolutional network for biomedical semantic segmentation," in 2018.

[11]X. Fu and H. Qu, "Research on semantic segmentation of high-resolution remote sensing image based on full convolutional neural network," Dec 2018.

[12]S. Kumar, A. Negi, J. N. Singh, and H. Verma, "A deep learning for brain tumor mri images semantic segmentation using fcn," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), Dec 2018.

[13] Samrat Kumar Dey, Arpita Howlader, "MobileNet Mask: A Multiphase Face Mask Detection Model to Prevent Person- To- Person Transmission of SARS- Conv2," in 2020 17th December International Conference Paper.

[14] Ming Yui Cheng, Lung S Chan, I J Lauder, Cyrus Rustam Kumana, "Detection of Body Temperature with Infrared Thermography: accuracy in detection of fever," in 2012 Hong Kong Medical Journal.

[15] Arun Francis G, Arulselvan M, ElangKumaran P, Keerthivarman S, Vijaya Kumar J, "Object Detection Using Ultrasonic Sensor," in 2019 International Journal of Innovative and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue- 6S, April 2019.