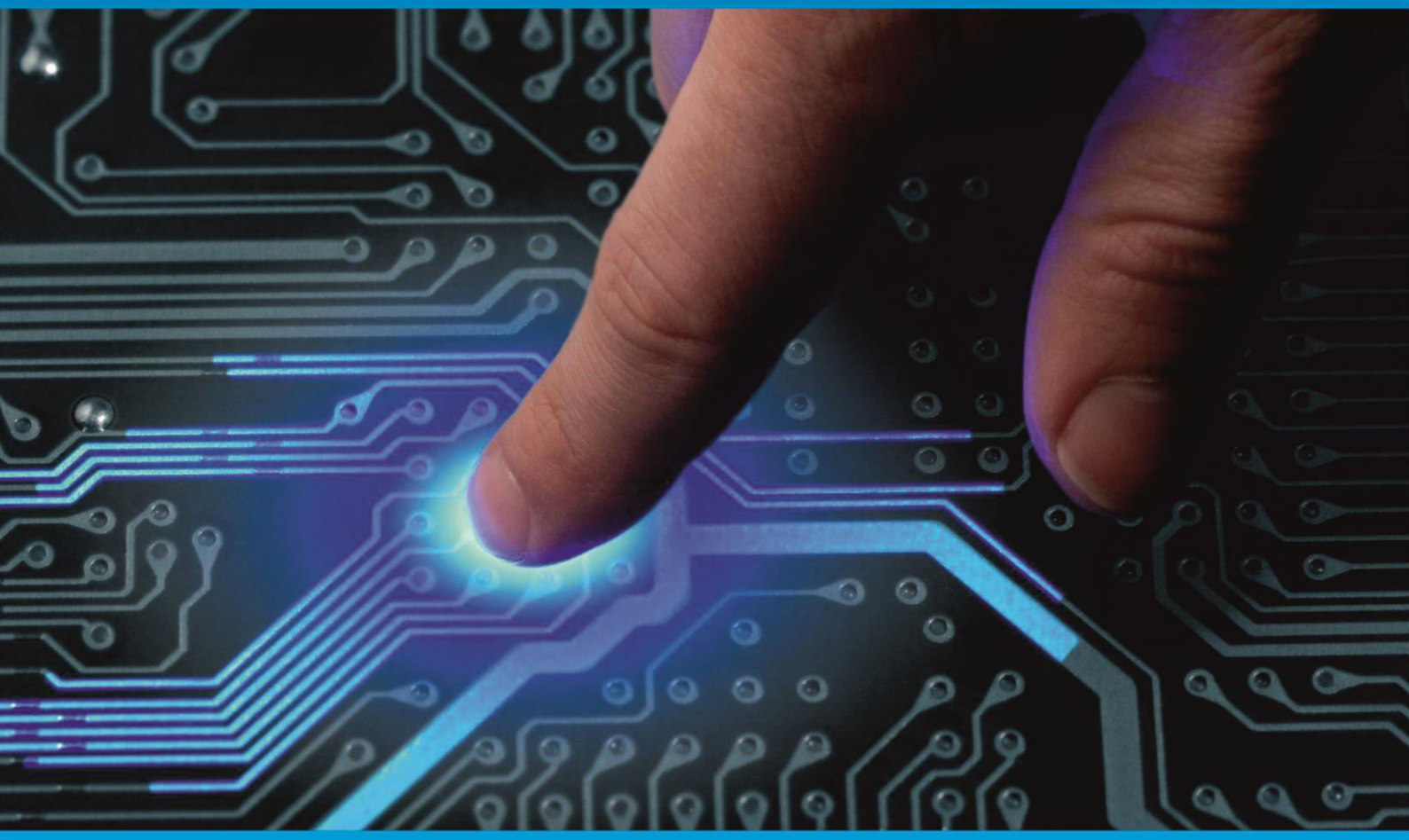
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 3, March 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

Design of Face Mask Detection System Using Deep Learning Algorithm

M.Sentamilselvi[1], S.Karthika[2], T.Ragavi[3], P.Selvi[4], T.Subathra[5]

Assistant Professor, Department of Electronics and Communication Engineering, Cheran College of Engineering, Tamilnadu, India[1]

UG Student, Department of Electronics and Communication Engineering, Cheran College of Engineering, Tamilnadu, India[2,3,4,5]

ABSTRACT: In this project ,the face mask detection system designed by using Deep learning algorithm for avoid the COVID virus spreading after the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. The basic aim of the project was detection of the presence of a face mask on human faces on live streaming video as well as on images. The system used the deep learning algorithm develop our face mask detection method. The object detection based on Single Shot Detector (SSD) because of its good performance accuracy and high speed. The basic concept of Neural Network System used in the data transfer. Finally the output was presence or absence of a face mask in an image, image processing based algorithm implemented for recognize the face image and it provided alert message to the person.

KEYWORDS:- CNN, COVID-19, Kernel, SSD and Deep Learning

I. INTRODUCTION

The outbreak of the novel coronavirus disease (COVID-19) in China spread worldwide, becoming an emergency of major international concern. SARS-CoV-2 infection causes clusters of severe respiratory illness similar to severe acute respiratory syndrome coronavirus. Human-to-human transmission via droplets, contaminated hands or surfaces has been described, with incubation times of 2-14 days. Early diagnosis, quarantine, and supportive treatments are essential to cure patients. This paper reviews the literature on all available information about the epidemiology, diagnosis, isolation and treatments of COVID-19. Treatments, including antiviral agents, chloroquine and hydroxyl chloroquine, corticosteroids, antibodies, convalescent plasma transfusion and vaccines, are discussed in this article. In addition, registered trials investigating treatment options for COVID-19 infection are listed. In a household study, loss of taste and/or smell was the fourth most reported symptom (26/42; 62%) among COVID-19 case-patients and had the highest positive predictive value (83%; 95% CI: 55–95%) among household contacts. Olfactory and taste dysfunctions should be considered for COVID-19 case identification and testing prioritization. The global coronavirus disease 2019 (COVID-19) pandemic caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has prompted robust public health investigations to characterize the disease course. Early published reports identified fever, cough, and shortness of breath as predominant symptoms of COVID-19 (1, 2), and these classic symptoms have been used for case identification and testing prioritization [1]-[5].

During the rapid rise in COVID-19 illnesses and deaths globally, and notwithstanding recommended precautions, questions are voiced about routes of transmission for this pandemic disease. Inhaling small airborne droplets is probable as a third route of infection, in addition to more widely recognized transmission via larger respiratory droplets and direct contact with infected people or contaminated surfaces. While uncertainties remain regarding the relative contributions of the different transmission pathways, we argue that existing evidence is sufficiently strong to warrant engineering controls targeting airborne transmission as part of an overall strategy to limit infection risk indoors. Appropriate building engineering controls include sufficient and effective ventilation, possibly enhanced by particle filtration and air disinfection, avoiding air recirculation and avoiding overcrowding. Often, such measures can be easily implemented and without much cost, but if only they are recognized as significant in contributing to infection control goals. The world is in the middle of a historic public health crisis. As of March 30, 2020, over a third of the population in the United States were under “stay at home” orders given by state governors to protect the vulnerable and the unexposed. Unprecedented steps have been taken by governments globally to contain the novel coronavirus disease

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

2019 (COVID-19), a rapidly spreading pandemic. This has resulted in more than 690,000 cases and over 33,000 deaths worldwide (Supplementary Table 1). The index case of the disease, caused by the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2) was identified more than 3months ago. Since then, public health authorities worldwide have taken aggressive measures to blunt the exponential spread of this coronavirus. Furthermore, several nations, including Italy, Spain, and France, have imposed nationwide lockdown measures to enforce social distancing to further prevent the spread of COVID-19 in their respective countries**.** 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 linage" which allows the features used by our detector to be computed very quickly [6]-[10]. 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 classers. The third contribution is a method for combining increasingly more complex classers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

II. RELATED WORK

Neural Networks (NN) are important data mining tool used for classification and clustering. It is an attempt to build machine that will mimic brain activities and be able to learn. NN usually learns by examples. If NN is supplied with enough examples, it should be able to perform classification and even discover new trends or patterns in data. Basic NN is composed of three layers, input, output and hidden layer. Each layer can have number of nodes and nodes from input layer are connected to the nodes from hidden layer. Nodes from hidden layer are connected to the nodes from output layer. Those connections represent weights between nodes Artificial Neural Networks (ANN) consider classification as one of the most dynamic research and application areas. The major disadvantage in using ANN is to find the most appropriate grouping of training, learning and transfer function for classifying the data sets with growing number of features and classified sets. The different combinations of functions and its effect while using ANN as a classifier is studied and the correctness of these functions are analyzed for various kinds of datasets.

The dataset classification uses the most effective tool called back propagation neural network. The usage of Back Propagation Neural Network (BPNN) for classifying the image. The neural network is given the target outputs on to which it should map its inputs, i.e. it is given in paired data of input and output. The error arising from the discrepancy between the network output and the target is used to optimize the network parameters. Once the network has been trained, it is used to produce an output for unseen data FFNNs are a kind of multilayer neural network which allows signals to travel one way only, from input to output. First, the network is trained on a set of paired data to determine input-output mapping. The weights of the connections between neurons are then fixed and the network is used to determine the classifications of a new set of data. During classification the signal at the input units propagates all the way through the net to determine the activation values at all the output units. Each input unit has an activation value that represents some feature external to the net. Then, every input unit sends its activation value to each of the hidden units to which it is connected. Each of these hidden units calculates its own activation value and this signal are then passed on to output units. The activation value for each receiving unit is calculated according to a simple activation function. The function sums together the contributions of all sending units, where the contribution of a unit is defined as the weight of the connection between the sending and receiving units multiplied by the sending unit's activation value. This sum is usually then further modified, for example, by adjusting the activation sum to a value between 0 and 1 and/or by setting the activation value to zero unless a threshold level for that sum is reached.

III. PROPOSED WORK

The convolutional neural network (CNN) is a class of deep learning Neural Networks. CNNs represent a huge breakthrough in image recognition. They're most commonly used to analyze visual imagery and are frequently working behind the scenes in image classification. They can be found at the core of everything from Facebook's photo tagging to self-driving cars. In this project working hard behind the scenes in everything from healthcare to security. It is fast and efficient. Image classification is the process of taking an input and outputting a class or a probability that the input is a particular class. This means that this type of network is ideal for processing 2D images. Compared to other image classification algorithms, CNNs actually use very little pre-processing. This means that they can learn the filters that have to be hand-made in other algorithms. CNNs can be used in tons of applications from image and video recognition, image classification, and recommender systems to natural language processing and medical image analysis. CNNs are inspired by biological processes. They're based on some cool research done by Hubel and Wiesel in the 60s regarding vision in cats and monkeys. The pattern of connectivity in a CNN comes from their research regarding the organization

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

of the visual cortex. In a mammal's eye, individual neurons respond to visual stimuli only in the receptive field, which is a restricted region. The receptive fields of different regions partially overlap so that the entire field of vision is covered. This is the way that a CNN works CNNs have an input layer, and output layer, and hidden layers. The hidden layers usually consist of convolutional layers, RELU layers, pooling layers, and fully connected layers. Convolutional layers apply a convolution operation to the input. This passes the information on to the next layer. Pooling combines the outputs of clusters of neurons into a single neuron in the next layer. Fully connected layers connect every neuron in one layer to every neuron in the next layer. In a convolutional layer, neurons only receive input from a subarea of the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer. A CNN works by extracting features from images. This eliminates the need for manual feature extraction. The features are not trained! They're learned while the network trains on a set of images. This makes deep learning models extremely accurate for computer vision tasks. CNNs learn feature detection through tens or hundreds of hidden layers. Each layer increases the complexity of the learned features. At this point, everything is trained through forward- and backward propagation through many, many epochs. We wind up with a very well defined neural network where all the weights and features are trained. We started with an input image and applied multiple different features to create a feature map. We applied the RELU to increase non-linearity and applied a pooling layer to each feature map. (We did that to make sure we have spatial variance in our images, to reduce the size of the images, and to avoid over fitting of the model to the data while still preserving the features we're after.) We flattened all the pooled images into one long vector. We input the vector into our fully connected artificial neural network. This is where all the features were processed through the network. This gave us the final fully connected layer which provided the "voting" of the classes that we're after. All of this was trained through forward propagation and back propagation until we wound up with a well defined Neural Network where the weights and feature detectors were trained. Using MATLAB with Deep Learning Toolbox enables you to design, train, and deploy CNNs. MATLAB provides a large set of pre trained models from the deep learning community that can be used to learn and identify features from a new data set. This method, called transfer learning, is a convenient way to apply deep learning without starting from scratch. Models like Google Net, Alex Net and Inception provide a starting point to explore deep learning, taking advantage of proven architectures built by experts.

Designing and Training Networks Using Deep Network Designer, you can import pre trained models or build new models from scratch. A CNN Starts with an input image Applies many different filters to it to create a feature map Applies a RELU function to increase non-linearity Applies a pooling layer to each feature map Flattens the pooled images into one long vector. Inputs the vector into a fully connected artificial neural network. Processes the features through the network. The final fully connected layer provides the "voting" of the classes that we're after. Trains through forward propagation and back propagation for many, many epochs. This repeats until we have a well-defined neural network with trained weights and feature detectors. At the very beginning of this process, an input image is broken down into pixels. For a black and white image, those pixels are interpreted as a 2D array (for example, 2x2 pixels). Every pixel has a value between 0 and 255. (Zero is completely black and 255 is completely white. The grey scale exists between those numbers.) Based on that information, the computer can begin to work on the data. For a color image, this is a 3D array with a blue layer, a green layer, and a red layer. Each one of those colors has its own value between 0 and 255. The color can be found by combining the values in each of the three layers.

For better generalize ability of the model, a very common regularization technique is used i.e. to add a regularization term to the objective function. This term ensures that the model doesn't capture the 'noise' in the dataset or does not over fit the training data. Objective function = Loss Function (Error term) + Regularization term Hence the objective function can be written as:

Objective function = $L(F(xi),\theta) + \lambda f(\theta)$
Where $L(F(xi),\theta)$ is the loss function expressed in terms of the model output $F(xi)$ and the model parameters θ. The second term $\lambda f(\theta)$ has two components — the regularization parameter λ and the parameter norm $f(\theta)$.
There are broadly two types of regularization techniques (very similar to one in linear regression) followed in CNN:
L1 norm: $\lambda f(\theta) = ||\theta||1$ is the sum of all the model parameters
L2 norm: $\lambda f(\theta) = ||\theta||2$ is the sum of squares of all the model parameters
A dropout operation is performed by multiplying the weight matrix W1 with an α mask vector as shown below.
Then, the shape of a vector α will be (3,1). Now if the value of q is .66, the α vector will have two 1s and one 0. Hence, the α vector can be any of the following three: [1 1 0] or [1 0 1] or [0 1 1].
One of these vectors is then chosen randomly in each mini-batch. Let's say that, in some mini-batch, the mask α=[1 1 0] is chosen. Hence, the new(generalized) weight matrix will be.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

Dropouts All elements in the last column become zero. Thus few neurons(shown in the image below) which were of less importance are discarded, making the network to learn more robust features and thus reducing the training time for each epoch.

This technique allows each layer of a neural network to learn by itself a little bit more independently of other previous layers. For example- In a feed-forward neural network
h4=σ(W4.h3+b4)=σ(W4.(σ(W3.(σ(W2.(σ(W1.x+b1))+b2))+b3))+b4)
h4 is a composite function of all previous networks(h1,h2,h3). Hence when we update the weights (say) W4, it affects the output h4, which in turn affects the gradient ∂L/∂W5. Thus, the updates made to W5 should not get affected by the updates made to W4. Thus Batch normalization is performed on the output of the layers of each batch, H(l). O/p layer is normalized by the mean vector μ and the standard deviation vector ^σ computed across a batch. Understanding the above techniques, we will now train our CNN on CIFAR-10 Datasets. The last thing you want is for your network to look for one specific feature in an exact shade in an exact location. That's useless for a good CNN. You want images that are flipped, rotated, squashed, and so on. You want lots of pictures of the same thing so that your network can recognize an object (say, a leopard) in all the images. No matter what the size or location. No matter what the lighting or the number of spots, or whether that leopard is fast asleep or crushing prey. You want spatial variance, You want flexibility. That's what pooling is all about.
Pooling progressively reduces the size of the input representation. It makes it possible to detect objects in an image no matter where they're located. Pooling helps to reduce the number of required parameters and the amount of computation required. It also helps control over fitting.
Over fitting can be kind of like when you memorize super specific details before a test without understanding the information. When you memorize details, you can do a great job with your flashcards at home. You'll fail a real test, though, if you're presented with new information. (Another example: if all of the dogs in your training data have spots and dark eyes, your network will believe that for an image to be classified as a dog, it must have spots and dark eyes. If you test your data with that same training data, it will do an amazing job of classifying dogs correctly! But if your outputs are only "dog" and "cat," and your network is presented with new images containing, say, a Rottweiler and a Husky, it will probably wind up classifying both the Rottweiler and the Husky as cats. You can see the problem) Without variance, your network will be useless with images that don't exactly match the training data. Always, always, always keep your training and testing data separate. If you test with the data you trained on, your network has the information memorized! It will do a terrible job when it's introduced to any new data. So for this step, you take the feature map, apply a pooling layer, and the result is the pooled feature map. The most common example of pooling is max pooling. In max pooling, the input image is partitioned into a set of areas that don't overlap. The outputs of each area are the maximum value in each area. This makes a smaller size with fewer parameters. At this step, we add an artificial neural network to our convolutional neural network.

The main purpose of the artificial neural network is to combine our features into more attributes. These will predict the classes with greater accuracy. This combines features and attributes that can predict classes better. At this step, the error is calculated and then back propagated. The weights and feature detectors are adjusted to help optimize the performance of the model. Then the process happens again and again and again. If, for example, you have two output classes, one for a cat and one for a dog, a neuron that reads "0" is absolutely uncertain that the feature belongs to a cat. A neuron that reads "1 is absolutely certain that the feature belongs to a cat. In the final fully connected layer, the neurons will read values between 0 and 1. This signifies different levels of certainty. A value of 0.9 would signify a certainty of 90%.
At this point, everything is trained through forward- and backward propagation through many, many epochs. We wind up with a very well defined neural network where all the weights and features are trained. Now we have something that can recognize and classify images. A convolutional neural network is trained on hundreds, thousands, or even millions of images. When working with large amounts of data and complex network architectures, GPUs can significantly speed the processing time to train a model. We started with an input image and applied multiple different features to create a feature map. We applied the RELU to increase non-linearity and applied a pooling layer to each feature map. (We did that to make sure we have spatial variance in our images, to reduce the size of the images, and to avoid over fitting of the model to the data while still preserving the features we're after.) We flattened all the pooled images into one long vector. We input the vector into our fully connected artificial neural network. This is where all the features were processed through the network. This gave us the final fully connected layer which provided the "voting" of the classes that we're after. All of this was trained through forward propagation and back propagation until we wound up with as well defined neural network where the weights and feature detectors were trained.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

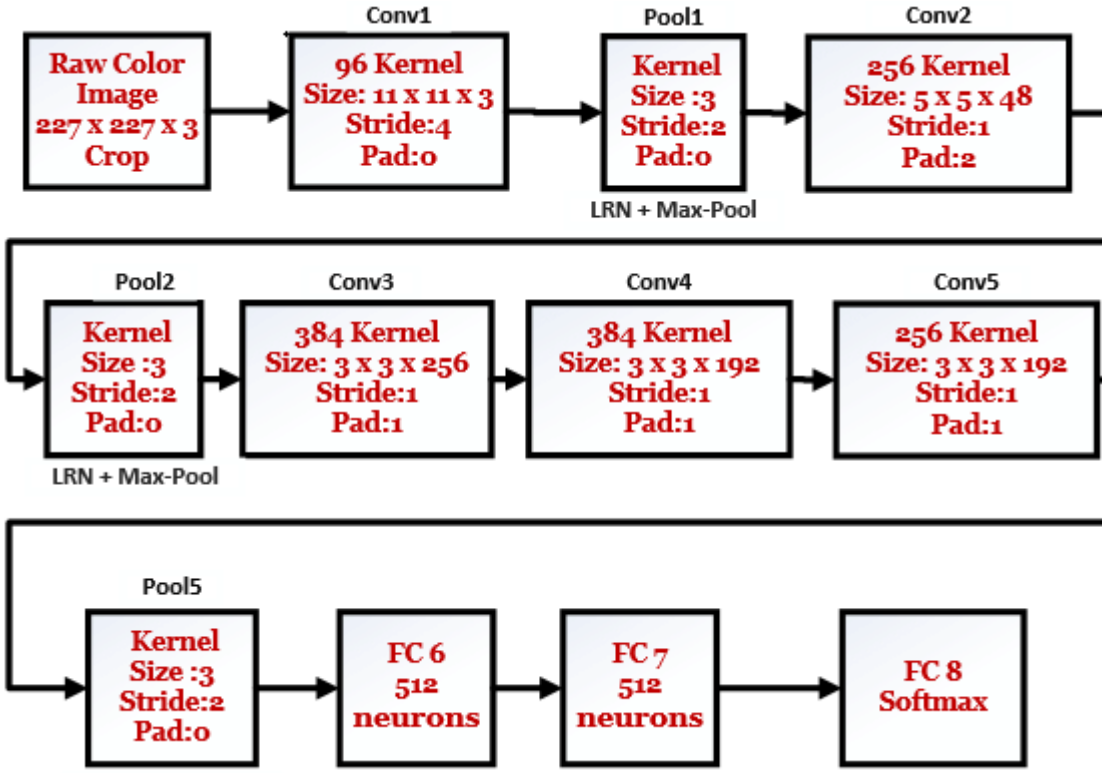


Figure.1. CNN layer Flow Chart

IV. RESULT AND DISCUSSION

The below Figure had more number of human beings. This project recognized the face masking and the face mask didn't wear human being. The figure 1 is the example picture.

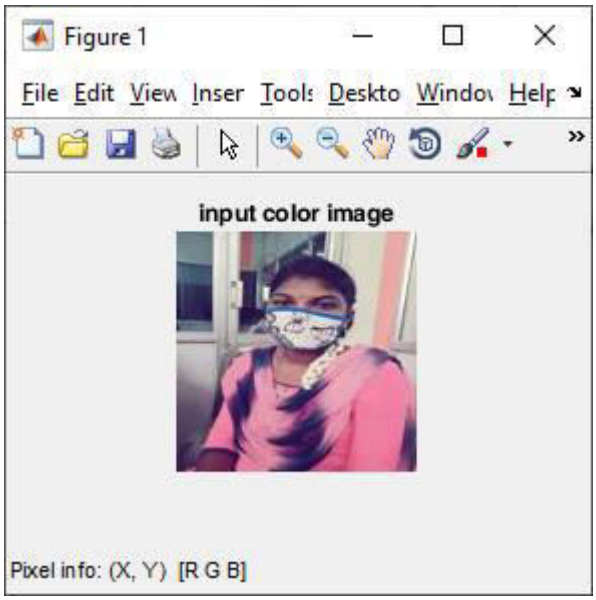


Figure.2. Input face image

Figure 2 had face detection example picture. The face mask identified by coding which based on CNN and SSD.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

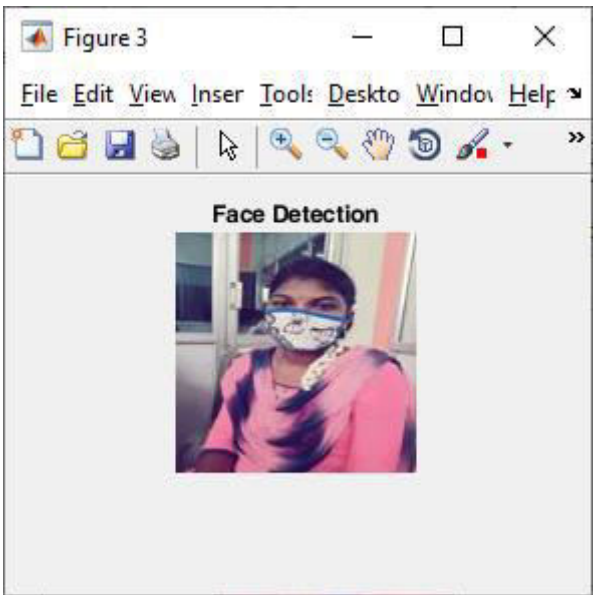


Figure.3. NORMALIZATION

```
layers =

  10x1 Layer array with layers:

     1   ''   Image Input             120x120x3 images with 'zerocenter' normalization
     2   ''   Convolution             20 5x5 convolutions with stride [1  1] and padding [0  0  0  0]
     3   ''   ReLU                    ReLU
     4   ''   Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
     5   ''   Convolution             20 5x5 convolutions with stride [1  1] and padding [0  0  0  0]
     6   ''   ReLU                    ReLU
     7   ''   Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
     8   ''   Fully Connected         2 fully connected layer
     9   ''   Softmax                 softmax
    10   ''   Classification Output   crossentropyex
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:00 |       38.46% |       0.7933 |      1.0000e-04 |
|      20 |          20 |       00:00:04 |      100.00% |       0.0146 |      1.0000e-04 |
|=========================================================================================|
```

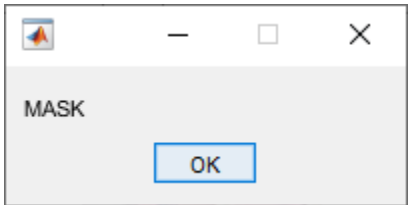


Figure.4.CNN LAYERS TRAINING

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903183 |

V. CONCLUSION

A new facemask-wearing identification system was designed, by using Deep learning Algorithm. In the face mask wearing identification system, the input images were processed with image pre-processing, facial detection, cropping and identification. Finally, SRC Net achieved a 98.70%accuracy and outperformed traditional end-to-end image classification methods by over 1.5% in kappa. Our findings indicate that the proposed SRC Net can achieve high accuracy in facemask-wearing condition identification, which is meaningful for the prevention of epidemic diseases including COVID-19 in public. There are a few limitations to our study. Firstly, the Medical Masks Dataset

REFERENCES

[1].Yu, P., Zhu, J., Zhang, Z., & Han, Y. (2020). A Familial Cluster of Infection Associated With the 2019 Novel Coronavirus Indicating Possible Person-to-Person Transmission During the Incubation Period. The Journal of infectious diseases, 221(11), 1757–1761. https://doi.org/10.1093/infdis/jiaa077

[2] Chavez, S., Long, B., Koyfman, A. & Liang, S. Y. Coronavirus Disease (COVID-19): A primer for emergency physicians. Am J Emerg Med, https://doi:10.1016/j.ajem.2020.03.036 (2020).

[3] World Health Organization. Coronavirus disease 2019 (COVID-19) Situation Report– 142, 2020, [cited 10 June 2020], https://www.who.int/docs/default-source/coronaviruse/situationreports/20200610-covid-19-sitrep-142.pdf?sfvrsn=180898cd_6 65

[4] Bai, Y., Yao, L., Wei, T., Tian, F., Jin, D. Y., Chen, L., & Wang, M. (2020). Presumed Asymptomatic Carrier Transmission of COVID-19. JAMA, 323(14), 1406–1407. Advance online publication. https://doi.org/10.1001/jama.2020.2565

[5] Centres for Disease Control and Prevention. Interim Infection Prevention and Control Recommendations for Patients with Suspected or Confirmed Coronavirus Disease 2019 (COVID-19) in Healthcare Settings. 2020 [cited 5 June 2020]. https://www.cdc.gov/coronavirus/2019-ncov/hcp/infection-controlrecommendations.html

[6] Korea Centres for Disease Control and Prevention. Infection Prevention and Control Recommendations for Patients with Suspected or Confirmed Coronavirus Disease 2019 (COVID-19) in Healthcare Settings [in Korean]. 2020 [cited 5 June 2020]. http://ncov.mohw.go.kr/duBoardList.do?brdId=2&brdGubun=28

[7] Sim, S. W., Moey, K. S. & Tan, N. C. The use of facemasks to prevent respiratory infection: a literature review in the context of the Health Belief Model. Singapore Med J 55, 160-167, https://doi:10.11622/smedj.2014037 (2014).

[8] Cowling, B. J. et al. Facemasks and hand hygiene to prevent influenza transmission in households: a cluster randomized trial. Ann Intern Med 151, 437-446, https://doi:10.7326/0003-4819-151-7-200910060- 00142 (2009).

[9] Tracht, S. M., Del Valle, S. Y. & Hyman, J. M. Mathematical modelling of the effectiveness of facemasks in reducing the spread of novel influenza A (H1N1). PLoS One 5, e9018, https://doi:10.1371/journal.pone.0009018 (2010).

[10] Jefferson, T. et al. Physical interventions to interrupt or reduce the spread of respiratory viruses. Cochrane Database System Rev, CD006207, https://doi:10.1002/14651858.CD006207.pub4 (2011)

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.488

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details