A Project Report

on

IMAGE CLASSIFICATION BASED ON FACEMASK

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. D Rajesh Kumar
Assistant Professor
Department of Computer Science and Engineering

Submitted By

18SCSE1010321 - RAVEENA RAJPAL

18SCSE1010445 - SRISHTI SINGH

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled " **IMAGE CLASSIFICATION BASED ON FACEMASK**" in partial fulfillment of the requirements for the award of the BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of JULY-2021 to DECEMBER-2021, under the supervision of Dr. D Rajesh Kumar, Assistant Professor, Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other place.

18SCSE1010321 - RAVEENA RAJPAL

18SCSE1010445 - SRISHTI SINGH

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Dr. D Rajesh Kumar, Assistant Professor)

Acknowledgement

The Final Thesis/Project/ Dissertation Viva-Voce examination of 18SCSE1010321 - Raveena Rajpal, 18SCSE1010445 - Srishti Singh has been held on _________________ and his/her work is recommended for the award of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.

Signature of Examiner(s) Signature of Supervisor(s)

Signature of Project Coordinator Signature of Dean

Date: 21/12/21

Place: Greater Noida

ABSTRACT

In this model we are going to classify an image on the basis of a face mask using MobileNet. This project is about building a system which can detect whether a person is wearing a mask or not on an image or live webcam video using Deep Learning and Computer Vision. I've used MobileNet for this purpose and then use transfer learning on it. This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed. This project helps us to spread awareness among people using face masks properly. It detects the face mask on your face whether the person is hiding his/her face by mask or not. It also checks if the face mask is properly covering your nose and mouth. MobileNet has fewer parameters and higher classification accuracy. In order to further reduce the number of network parameters and improve the classification accuracy, dense blocks that are proposed in DenseNets are introduced into MobileNet. In Dense-MobileNet models, convolution layers with the same size of input feature maps in MobileNet models are taken as dense blocks, and dense connections are carried out within the dense blocks.

Table of Contents

Title	Page No.
Candidates Declaration	1-2
Acknowledgement	3
Abstract	4
Table of Contents	
Acronyms	5
Chapter 1 Introduction	7
1.1 Architecture Diagram for Proposed method	8
1.2 MERITS OF PROPOSED SYSTEM	11
Chapter 2 Literature Survey	12
Chapter 3 Implementation	
3.1-Tools Required For Implementation	13
3.2-Implementation and Description of Project Modules	14
3.3 Screenshots	21
Chapter 4	
4.1-Network Architecture:	23
4.2-Inductive Transfer	24
4.3 Error Function	25
4.4 Speculation	26
4.5 Results	27
4.6 Discussion	28
Chapter 5 Conclusion and Future Scope	
5.1 Conclusion	29
5.2 Future Scope	30
References	31

Acronyms

- FMD- FACE MASK DETECTION
- C-19 - CORONAVIRUS 2019
- ML- Machine Learning
- DL - Deep Learning
- CNN - Convolutional Neural Networks

CHAPTER-1

Introduction

As a lightweight deep neural network, MobileNet has fewer parameters and higher classification accuracy. In order to further reduce the number of network parameters and improve the classification accuracy, dense blocks that are proposed in DenseNets are introduced into MobileNet. In Dense-MobileNet models, convolution layers with the same size of input feature maps in MobileNet models are taken as dense blocks, and dense connections are carried out within the dense blocks. For building this model, we will be using the face mask dataset provided by Prajna Bhandary. It consists of about 1,376 images with 690 images containing people with face masks and 686 images containing people without face masks. We are going to use these images to build an Image Classification model using TensorFlow and MobileNet to detect if you are wearing a face mask by giving a particular image.

Architecture Diagram for Proposed method

Outcome and Probe

Presentation of Face Mask contrasted the fundamental social impact distributed by the information base maker. Because of the restricted alternatives proposed for this information base, we have utilized ResNet and MobileNet as discrete spines for correlation. Broad1 demonstrated that the Retina FaceMask with ResNet-spine accomplished a sophisticated exactness than MobileNet-spine by about 9% meticulousness and review arrangement. What's more, RetinaFaceMask with ResNet-spine picks up a cutting edge result contrasted with the essential model. Specifically, RetinaFaceMask is 2.3% and 1.5% higher than the standard impact on vis-à-vis identification individually, and 11.0% and 5.9% higher than the review premise. Other test outcomes appear in the picture, here red and green block allude forecasts, individually.

Network Removal

We carry out elimination deals with exchange studying, attention devices and spines. The test effects are summed up in Board2 and subtleties are provided below:

Transfer Studying

The examinations receive pretrained masses from Imagenet record for spines. This association is taken into consideration as a pattern result in our examination. The removal research of move studying implies that the organization utilizes pretrained masses from a comparative undertaking - face recognition, and organized on a particular record Broader Aspects. Hence, consequences utilizing MobileNet backbone display that flow studying can drastically extend the discovery execution by means of 5-6% in facial areas as well as veil area tests. One potential explanation is that progressed component removal capability is upgraded by using the usage of pretrained hundreds as a firmly associated challenge.

Attention Identification

We carry out investigations to assess the exhibition of the explained placing attention identification region. They have a tendency to verify that, through utilizing attention factor. These consequences show off that the attention modules can zero in on the perfect facial area and cover highlights to enhance the last discovery execution. Albeit distinctive organization segments can expand the recognition execution, the greatest precision improvement is accomplished by the ResNet spi.

Merits of Proposed system

The proposed system helps to ensure the safety of the people at public places by automatically monitoring whether they maintain a safe social distance, and also by detecting whether or not an individual wears a face mask. This section briefly describes the solution architecture and how the proposed system will automatically function in an automatic manner to prevent the coronavirus spread. The proposed system is a deep learning solution that uses OpenCV and TensorFlow, to train the model. We combine the deep learning MobileNetV2 modal with the TensorFlow framework for a fast and efficient deep learning solution for real-time human detection in video streams and use a triangular similarity technique to measure distance between persons detected by camera in real time in public places and comprises customized data collection to resolve a face mask detection model with variance in the types of face masks worn by the public in real time by means of a transfer of learning to a pre-trained SSD face detect

CHAPTER-2

Literature Reviews/Comparative study

The World Health Organization (WHO) ninety-six of the standing report undraped the corona virus has affected many folks worldwide as well as instigated a hundred and eighty thousand deaths. In addition, there are many major similarities. We tend to measure major metabolism diseases, like associate degree acute metabolism syndrome (SARS) and also the geographic region tract syndrome (MERS), that occurred a few years ago. It had been rumored that the birth variety of the Coronavirus high paralleled the SARS. Consequently, a lot of as well as a lot of folk are upset regarding their health, the public health is taken into account as a priority for governments. luckily, it showed the surgery. Facial masks will cut back the unfold of the coronavirus. Meanwhile, the WHO claims that the folks garb front covers if they need metabolism symptoms, or look after folks with symptoms. In associate degree addition, the bulk of the communities' service suppliers need customers to use the service only if they wore masks. Since, the discovery of a face mask has been an important computer-based activity to help the international community, but research on the discovery of face masks has been limited. MobileNet has fewer parameters and higher classification accuracy. In order to further reduce the number of network parameters and improve the classification accuracy, dense blocks that are proposed in DenseNets are introduced into MobileNet. In Dense-MobileNet models, convolution layers with the same size of input feature maps in MobileNet models are taken as dense blocks, and dense connections are carried out within the dense blocks. As in this paper, we had developed a face detector, i.e. Face Mask, which can detect facial expressions and contribute to public health care. FaceMask is the first keen face mask machine. For network design, FaceMask uses Feature Network Pyramid (FPN) for mixing high-level semantic data. For better discovery, we suggest the context attention-grabbing header and high-resolution object algorithm to improve the ability to find. In addition, for small-screen mirror databases where features can be difficult to remove, we use reading transfer transferring learned characters from networks trained in the same face recognition function to a wider database. For building this model, we will be using the face mask dataset provided by Prajna Bhandary. It consists of about 1,376 images with 690 images containing people with face masks and 686 images containing people without face masks.We are going to use these images to build a Image Classification model using TensorFlow and MobileNet to detect if you are wearing a face mask by giving a particular image.

CHAPTER 3

3.1-TOOLS REQUIRED FOR IMPLEMENTATION

Google Colab:

Google Colab is a free cloud service and now it supports free GPU! You can; improve your Python programming language coding skills. develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV.

TensorFlow:

TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine Learning is easy. If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

Keras:

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation

MoblieNet:

MobileNet is a CNN architecture model for Image Classification and Mobile Vision. There are other models as well but what makes MobileNet special is that it has very less computation power to run or apply transfer learning to.

Python:

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.

3.2-Implementation and Description of Project Modules

Step 1: Data Visualization

In the first step, let us visualize the total number of images in our dataset in both categories. We can see that there are *690* images in the '*masked*' class and *686* images in the '*unmasked*' class.

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

from keras.models import load_model
from tensorflow.keras.applications.mobilenet import preprocess_input
```

Step 2: Data Augmentation

In the next step, we augment our dataset to include more number of images for our training. In this step of *data augmentation*, we *rotate* and *flip* each of the images in our dataset. We see that, after data augmentation, we have a total of 2751 images with 1380 images in the 'masked' class and '1371' images in the 'unmasked' class.

```
[ ]  mobilenet = MobileNet(include_top = False)
     x = mobilenet.output

     x=GlobalAveragePooling2D()(x)


     x=Dense(1024,activation='relu')(x)

     x=Dense(512,activation='relu')(x)

     preds = Dense(2,activation='softmax')(x)

     model = Model(inputs = mobilenet.input, outputs=preds)
```

Step 3: Splitting the data

In this step, we *split* our data into the *training set* which will contain the images on which the CNN model will be trained and the *test set* with the images on which our model will be tested. In this, we take *split_size =0.8*, which means that *80%* of the total images will go to the *training set* and the remaining *20%* of the images will go to the *test set.* After splitting, we see that the desired percentage of images have been distributed to both the training set and the test set as mentioned above**.**

```
[ ] (trainX, testX, trainY, testY) = train_test_split(data, labels)
```

```
[ ]
    model.compile(loss="binary_crossentropy", optimizer = 'adam', metrics=["accuracy"])

    H = model.fit( trainX, trainY, epochs = 15)
```

```
Epoch 1/15
33/33 [==============================] - 49s 1s/step - loss: 0.0374 - accuracy: 0.9971
Epoch 2/15
33/33 [==============================] - 49s 1s/step - loss: 0.0244 - accuracy: 0.9961
Epoch 3/15
```

Step 4: Building the Model

In this step download the dataset. Preprocess it and then Train our neural network. This network has been trained using transfer learning of MobileNet. We've achieved a testing accuracy of 99.2% which is really good. We are saving the model with the name 'MaskNet.hdf5' after training.

This code file first loads the pretrained model. We need to add the path of the image which we want to process. Then the model predicts whether the mask is detected or not. (0 = 'Not Masked', 1 = 'Masked). For testing you can use images which are present in the repository named as masked and unmasked.

```
X = []
y = []
for i in os.listdir('/content/observations/experiements/data/with_mask'):
    img = load_img('/content/observations/experiements/data/with_mask/'+str(i), target_size = (224,224))
    img = img_to_array(img)
    img = preprocess_input(img)
    X.append(img)
    y.append([1,0])

for i in os.listdir('/content/observations/experiements/data/without_mask'):
    img = load_img('/content/observations/experiements/data/without_mask/'+str(i), target_size = (224,224))
    img = img_to_array(img)
    img = preprocess_input(img)
    X.append(img)
    y.append([0,1])

data = np.array(X)
labels = np.array(y)
```

Step 5: Pre-Training the CNN model

After building our model, let us create the '*train_generator*' and '*validation_generator*' to fit them to our model in the next step. We see that there are a total of *2200* images in the *training set* and *551* images in the *test set*.

```
for layer in model.layers[:-9]:
    layer.trainable = False
```

Step 6: Training the CNN model

This step is the main step where we fit our images in the training set and the test set to our Sequential model we built using *keras* library. I have trained the model for *30 epochs* (iterations). However, we can train for more number of epochs to attain higher accuracy lest there occurs *over- fitting*. We see that after the 30th epoch, our model has an accuracy of *98.86%* with the training set and an accuracy of *96.19%* with the test set.

```
[ ] (trainX, testX, trainY, testY) = train_test_split(data, labels)
```

```
[ ] model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

    H = model.fit( trainX, trainY, epochs = 15)
```

```
Epoch 1/15
33/33 [==============================] - 10s 45ms/step - loss: 0.2944 - accuracy: 0.8642
Epoch 2/15
33/33 [==============================] - 1s 40ms/step - loss: 0.0103 - accuracy: 0.9975
Epoch 3/15
33/33 [==============================] - 1s 40ms/step - loss: 0.0012 - accuracy: 1.0000
Epoch 4/15
33/33 [==============================] - 1s 41ms/step - loss: 7.0545e-04 - accuracy: 1.0000
Epoch 5/15
33/33 [==============================] - 1s 41ms/step - loss: 0.0039 - accuracy: 0.9991
Epoch 6/15
33/33 [==============================] - 1s 40ms/step - loss: 6.5402e-04 - accuracy: 0.9997
Epoch 7/15
33/33 [==============================] - 1s 40ms/step - loss: 2.9771e-04 - accuracy: 1.0000
Epoch 8/15
33/33 [==============================] - 1s 41ms/step - loss: 2.8864e-06 - accuracy: 1.0000
Epoch 9/15
33/33 [==============================] - 1s 40ms/step - loss: 7.0726e-06 - accuracy: 1.0000
Epoch 10/15
33/33 [==============================] - 1s 41ms/step - loss: 8.9190e-07 - accuracy: 1.0000
```

```
Epoch 9/15
33/33 [==============================] - 1s 40ms/step - loss: 7.0726e-06 - accuracy: 1.0000
Epoch 10/15
33/33 [==============================] - 1s 41ms/step - loss: 8.9190e-07 - accuracy: 1.0000
Epoch 11/15
33/33 [==============================] - 1s 40ms/step - loss: 4.4193e-06 - accuracy: 1.0000
Epoch 12/15
33/33 [==============================] - 1s 40ms/step - loss: 1.3883e-05 - accuracy: 1.0000
Epoch 13/15
33/33 [==============================] - 1s 40ms/step - loss: 2.5538e-05 - accuracy: 1.0000
Epoch 14/15
33/33 [==============================] - 1s 40ms/step - loss: 0.0032 - accuracy: 0.9983
Epoch 15/15
33/33 [==============================] - 1s 40ms/step - loss: 6.2610e-07 - accuracy: 1.0000
```

```
[ ] model.evaluate(testX , testY)
```

```
11/11 [==============================] - 1s 57ms/step - loss: 0.1271 - accuracy: 0.9855
[0.12709519267082214, 0.9854651093482971]
```

Step 7: Labeling the Information

After building the model, we label two probabilities for our results. *['0' as 'without_mask' and '1' as 'with_mask']*. I am also setting the boundary rectangle color using the RGB values.*['RED' for 'without_mask' and 'GREEN' for 'with_mask]*

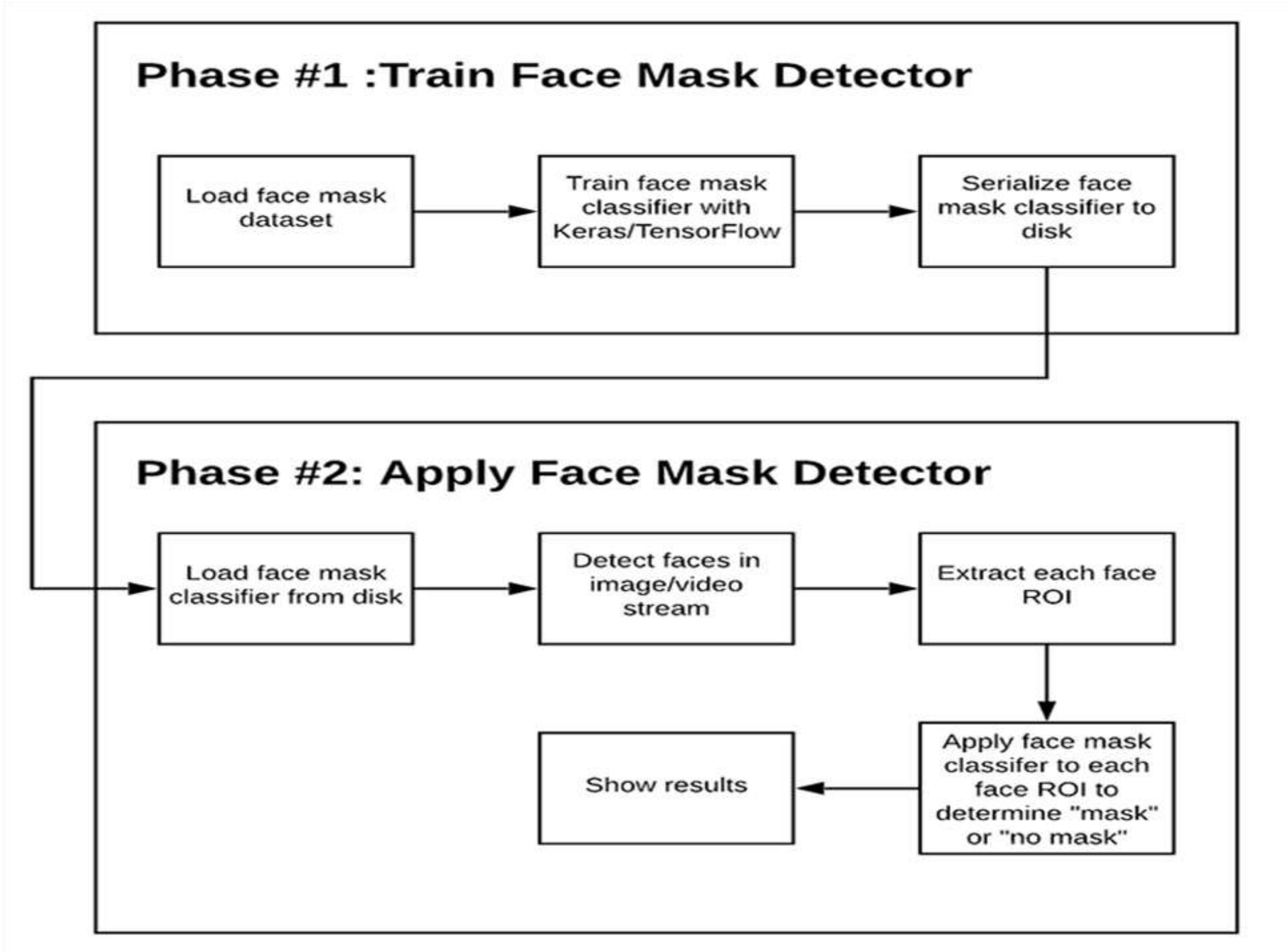
```
img = load_img('/content/observations/experiements/data/without_mask/376.jpg', target_size = (224,224))
img = img_to_array(img)
img = preprocess_input(img)
plt.imshow(img)

img = np.expand_dims(img, axis = 0)
model.predict(img)
```

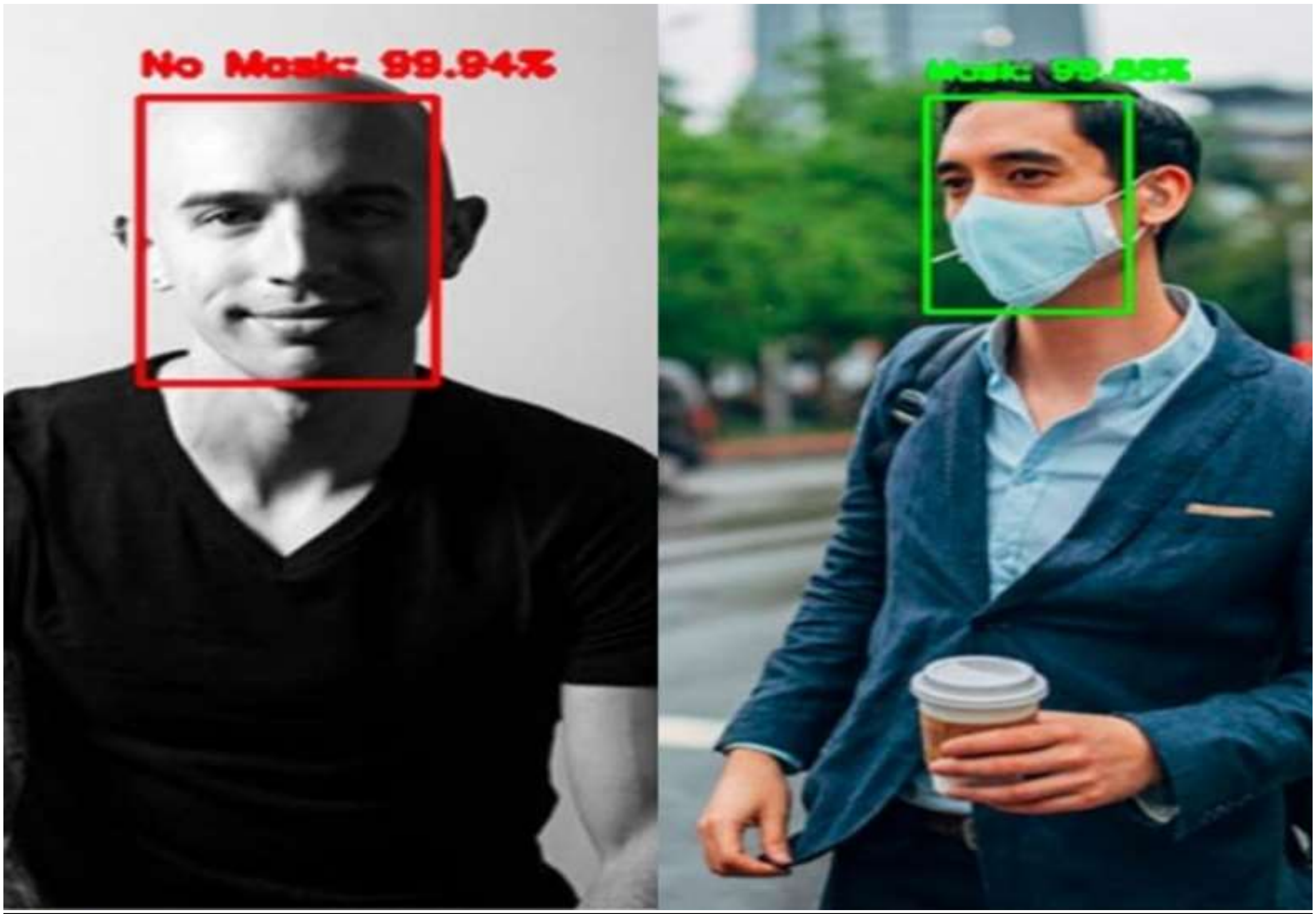
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
array([[0., 1.]], dtype=float32)
```

Step 8: Detecting the Faces with and without Masks

In the last step, we will give a path of image to our model which we have to classify.The model will predict the possibility of each of the two classes ([without_mask, with_mask]). Based on which probability is higher, the label will be chosen and displayed around our face.

3.3-SCREENSHOTS


Phase #1 :Train Face Mask Detector
Load face mask dataset
Train face mask classifier with Keras/TensorFlow
Serialize face mask classifier to disk
Phase #2: Apply Face Mask Detector
Load face mask classifier from disk
Detect faces in image/video stream
Extract each face ROI
Apply face mask classifer to each face ROI to determine "mask" or "no mask"
Show results


No Mask: 99.94%

CHAPTER 4

4.1-Network Architecture: -

The projected FaceMask design is displayed. Build an active facial network with the availability of a mask, we accept the outline of the proposed object in it, proposing a retrieval network for the spine, neck and head. The spine refers to a common catheter feature formed by convolutional neural organizations to separate data from pictures to embed maps. The Neural Network architecture is made of individual units called neurons that mimic the biological behavior of the brain. Its main function is to give importance to those features that contribute more towards the learning. It does so by introducing scalar multiplication between the input value and the weight matrix. For example, a negative word would impact the decision of the sentiment analysis model more than a pair of neutral words. The job of the transfer function is to combine multiple inputs into one output value so that the activation function can be applied. It is done by a simple summation of all the inputs to the transfer function. It introduces non-linearity in the working of perceptrons to consider varying linearity with the inputs. Without this, the output would just be a linear combination of input values and would not be able to introduce non-linearity in the network.

4.2-**Inductive Transfer: -**

Because of the restricted extent of appearance cover dataset, it is hard on behalf of knowledge calculations to know about enhanced highlights. The same number of the profound learning techniques ordinarily need bigger dataset, move learning is intended to move took in data from a premise errand to a connected objective undertaking. As indicated by means of, flow studying has assisted with the getting to know in a huge way as lengthy as it has a comfy relation. In our needs, we use quantities of the organization pre organized for full-size scope face recognition. Two inductive knowledge transfer approaches – multitask learning (MTL) and Feature Net (FN) – have been used to build predictive neural networks (ASNN) and PLS models for 11 types of tissue-air partition coefficients (TAPC). Unlike conventional single-task learning (STL) modeling focused only on a single target property without any relations to other properties, in the framework of inductive transfer approach, the individual models are viewed as nodes in the network of interrelated models built in parallel (MTL) or sequentially (FN). It has been demonstrated that MTL and FN techniques are extremely useful in structure−property modeling on small and structurally diverse data sets, when conventional STL modeling is unable to produce any predictive model. The predictive STL individual models were obtained for 4 out of 11 TAPC, whereas application of inductive knowledge transfer techniques resulted in models for 9 TAPC. Differences in prediction performances of the models as a function of the machine-learning method, and of the number of properties simultaneously.

Meta-Searching for Problem Solvers

A different research direction in inductive transfer Explores complex scenarios where the software architecture itself evolves with experience. The main idea is to divide a program into different components that Can be re-used during different stages of the learning process. As an illustration, one can work Within the space of (self-delimiting binary) programs to propose an optimal ordered problem Solver. The goal is to solve a sequence of problems, deriving one solution after the other, as Optimally as possible. Ideally the system should be capable of exploiting previous solutions and Of incorporating them into the solution to the current problem. This can be done by allocating Computing time to the search for previous solutions that, if useful, become transformed into Building blocks. We assume the current problem can be solved by copying or invoking previous Pieces of code (i.e., building blocks or knowledge). In that case the mechanism will accept those Solutions with substantial savings in computational time.

4.3-Error Function: -

FaceMask provides 2 yields information picture, confinement balance expectation, Ybloc ∈ St×4 and characterization forecast, Ybc ∈ St ×d, here t and d indicate quantity produced secures as well as quantity of programs. An error function, E(X, ₩theta)E(X,θ), which defines the error between the desired output ₩vec{y_i}yi and the calculated output ₩hat{₩vec{y_i}}yi^ of the neural network on input ₩vec{x_i}xi for a set of input-output pairs ₩big(₩vec{x_i}, ₩vec{y_i}₩big) ₩in X(xi,yi)∈X and a particular value of the parameters ₩thetaθ.

4.4-Speculation: -

In surmising, the version created the item limitation $A \in St\times4$ and article actuality $B \in St\times3$, in which the second one section of B, $Bz \in St\times1$, was the certainty of face; the 1/3 segment of B, $Ba \in St\times1$, changed into the understanding of cover. Detecting negative information is essential in most text mining tasks such as sentiment analysis, since Negation is one of the most common linguistic means to change polarity. The literature on sentiment Analysis and opinion mining in particular has emphasized the need for robust approaches to negation Detection, and for rules and heuristics for assessing the impact of negation on evaluative words and Phrases, i.e., those which convey the author's opinion towards an object, a person or another opinion. Popescu, & Asher (2012)) have shown that this common linguistic construction is highly relevant to Sentiment analysis, and that different types of negation have subtle effects on sentiment. In addition, They argue that the automatic study of opinions requires fine-grained linguistic analysis techniques as Well as substantial effort in order to extract features for either machine learning or rule-based systems, So that negation can be appropriately incorporated. Distinguishing between objective and subjective facts is also crucial for sentiment analysis since Speculation tends to correlate with subjectivity. Authors such as Pang & Lee (2004) show that Subjectivity detection in the review domain helps to improve polarity classification. Applying an SVM classifier, the proposed method surpasses the baseline by as much as about 20% in The negation cue detection and about 13% in the scope recognition, both in terms of F1. To the best of Our knowledge, this is the first system that addresses speculation in the review domain. The results Achieved in the speculation cue detection are close to those obtained by a human rater performing the Same task. In the scope detection phase, the results are also promising and they represent a Substantial improvement on the baseline (up by roughly 10%). In addition, the extrinsic evaluation Demonstrates that the proposed system could improve results in sentiment analysis. The proposed machine learning negation and speculation detection system is presented. The Evaluation framework is then detailed and the results are provided and discussed. An error analysis is Provided, and the potential of the developed method for addressing sentiment analysis

4.5-RESULTS

We compare the images of the traditional logistic loss function with our proposed maximum interval loss function. It can be clearly seen that the value of the loss function increases with the increase of the severity of the misclassification, which indicates that the loss function can effectively express the error degree of the classification.

4.6-DISCUSSION

We compare the images of the traditional logistic loss function with our proposed maximum interval loss function. It can be clearly seen that the value of the loss function increases with the increase of the severity of the misclassification, which indicates that the loss function can effectively express the error degree of the classification. Researchers working in image analysis and computer vision fields understand that leveraging AI, particularly CNNs, is a revolutionary step forward in image classification. Since CNNs are self-training models, their effectiveness only increases as they are fed more data in the form of annotated images (labeled data). That being said, it is high time for you to implement your image classification using CNN if your company has a dependency on image classification and analysis.

CHAPTER 5

5.1-CONCLUSION

As in this paper, we discussed another face indicator, the FaceMask, that could add to general medical services. The FaceMask design comprises MobileNet for spine as well as logical consideration components with equal headers. ResNet spine and MobileNet light spine can be utilized for high and low estimation circumstances, separately. To deliver strong highlights, we use moves to figure out how to utilize devices in the obtaining of a similar surface, preparing a huge information base. Likewise, we had recommended bean element for zero in for that embodiment of the novel to focus on face-to-face masks, just as a cross-sectional calculation, ORCC, low-certainty expulsion and high IoU. This proposed approach accomplishes proficient outcomes in open public face information bases.

5.2-FUTURE SCOPE

As the existing model doesn't gave us much accuracy for the given model . So, we have pre trained the model again and again to get the higher accuracy for the model. For that, we have used more epochs than before so, that might gave us better accuracy than before.Hence, for that step downloads the dataset. Preprocess it and then Train our neural network. This network has been trained using transfer learning of MobileNet. We've achieved a testing accuracy of 99.2% which is really good. We are saving the model with the name 'MaskNet.hdf5' after training.

This code file first loads the pretrained model. We need to add the path of the image which we want to process. Then the model predicts whether the mask is detected or not. (0 = 'Not Masked', 1 = 'Masked). For testing you can use images which are present in the repository named as masked and unmasked.

References

[1] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable component version," in 2008 IEEE convention on laptop

imaginative and prescient and sample reputation. IEEE, 2008, pp. 1–eight.

[2]]P. A. Rota, M. S. Oberste, S. S. Monroe, W. A. Nix, R. Campagnoli, J. P. Icenogle, S. Penaranda, B. Bankamp,

[3] W. H. organization et al., “Coronavirus sickness 2019 (covid-19): situation record, ninetysix,” 2020.

[4] Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklöv, “The reproductive wide variety of covid-19 is higher as compared to sars coronavirus,” magazine of journey medication, 2020.

[5] Y. Fang, Y. Nie, and M. Penny, “Transmission dynamics of the covid-19 outbreak and effectiveness of government interventions: A records-driven evaluation,”

[6] Magazine of medical virology, vol. ninetytwo, no. 6, pp. 645–659, 2020.

[7] N. H. Leung, D. k. Chu, E. Y. Shiu, k.-H. Chan, J. J. McDevitt, B. J. Hau, H.-L. Yen, Y. Li, D. KM, J. Ip et al., “respiration virus dropping in exhaled breath and efficacy of face mask.”

[8] S. Feng, C. Shen, N. Xia, W. music, M. Fan, and B. J. Cowling, “Rational use of face masks inside the covid-19 pandemic,” The Lancet respiratory remedy, 2020.

[9] D. Chiang., “discover faces and decide whether or not people are sporting mask,” https://github.com/AIZOOTech/ FaceMaskDetection, 2020.

[10] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, ok. Jiang, N. Wang, Y. Pei et al., “