ALERT SYSTEM USING CNN TO DETECT FACE MASKS

Sonal Patil[1], Swati Patil[2], Shital Jadhav[3], Nilesh V.Ingale[4]

Article History: **Received: 15.02.23** **Revised: 10.04.23** **Accepted: 05.06.23**

Abstract

The main motive of writing this paper is to create a facial mask detector that can find whether a person has worn a facemask properly or not. This paper is used to solve real-life problems that nowadays we must avoid. So it helps any person. After implementing this detector, we can easily classify people who are wearing a face mask and the people who are not wearing the mask and people who are incorrectly wearing facemasks. The proposed method of this paper uses the YOLO algorithm. This detector can be used for precautionary purposes during the spread of COVID-19.

Keywords: Automated Face mask detection, Conventional Neural Network, YOLO, Machine Learning.

[1]Assistant Professor, CSE Department, GHRIEBM, Jalgaon
[2]Assistant Professor, CSE Department, GHRIEBM, Jalgaon
[3]Assistant Professor, CSE Department, GHRIEBM, Jalgaon
[4]Assistant Professor, CSE Department, GHRIEBM, Jalgaon

DOI: 10.48047/ecb/2023.12.8.45

Introduction

The spread of coronavirus is increasing everywhere in the world. Talking about India, it also increased rapidly in the initial time but after some time now it is in control, so the government gets time to recover unbalanced things. This virus spreads through the droplets, airborne, and contact with COVID people. So later WHO proposed safety measures to reduce the spread of COVID-19 such as every person must wear a mask as well as maintain proper social distance in crowded places and try to increase immune power. Therefore, to protect each other, every person must have and wear the masks properly and increase the use of sanitizers. Even knowing these safety measures, some people won't follow the safety precautions for so many reasons. But must take action against such people.

To tackle this problem, a working face mask detecting system must be built. The object detection techniques are an effective technique to tackle the issue of face mask detection. Also using integrated API (Application Programming Interface) of Mail to send alerts to a system admin about the overall scenario so that they can take necessary actions and decide the precautions and safety measures. We are using machine learning, which is a subfield of artificial intelligence. Machine learning is used to detect patterns in the data and can be used to classify previously unseen data. In our case, we are subcategorized whether they are masked, unmasked, or worn the mask incorrectly or not. So, on successful implementation, our research paper will be able to detect masked and unmasked people by using a camera as a source for input. We are also planning to monitor people's lives. So if a person does not wear a mask, we can perform specific actions on him/her instantly.

Face mask detection is a project that depends on a variety of fields like machine learning, artificial intelligence and others. Face mask detection is an artificial intelligence technology that recognizes face masks by a single person or crowd of people. It's similar to an object detection system, which detects a certain class of items. We hope that by constructing this detector, we would be able to help in ensuring people's safety in public areas like malls, train stations, bus stands and hospitals. This technique can be utilized in a number of settings which include public and private sectors such as shopping malls, schools, colleges, airports and bus and train stops, among others. Face-Mask detection is a way for matching a person's mask from a video or image by using a relevant dataset of face masks. It is typically used to confirm customers via ID verification services, and it really works with the aid of finding and measuring facial capabilities from a given video or image. Facial identification systems are used on smartphones, laptops and computers and in different sorts of technology which include robotics in current years. Organization of Paper: In this paper we have discussed about automated face mask recognition system, which involves literature survey, comparative study about various CNN based algorithms and final outcome.

Literature Survey

We have reviewed a few associated techniques that aim to solve the same problem as we are going to propose. As face mask detection has been in the demand of being practiced in recent covid situations, there is a lot of published work in this field. In March 2020, WHO confirmed the novel Coronavirus illness (COVID-19) case. In December 2019, Wuhan, the capital of China Province in the Republic of China,COVID-19 spread incredibly fast, a virus that may be passed from humans to other human by droplet or airborne dispersion. According to multiple surveys conducted, use of face mask in public places heavily decreases the risk of a wide spread of Covid-19.

In [1], the authors BVarshiniHRYogeshSyed DanishPashaMaazSuhailVMadhumithaArchana Sasi made a proposition that is capable of identifying users by using Internet of Things technology. Smart door-openers monitor body heat and detect face masks using machine learning. He revealed a system in which face mask detection is carried by TensorFlow.The author uses a without contact temperature sensor to monitor temperature .This model contains some limitations like low measurement accuracy and long computation time. In [2], the authors Susanto Susanto; Febri Alwan Putra; Riska Analia; Ika Karlina Laila Nur Suciningtyas made a detector by using YOLO deep learning and constructs its mask detection algorithm. The advantage of this detector is that it can detect a face mask correctly even if there is some noise and disturbance in that area.

In [3], the authors Sohaib Asif, Yi Wenhui, Yi Tao, Si Jinhai proposed a framework to determine the mask region. First, he used OpenCV and ML to detect and track the faces of people and then used facial frames with their suggested deep transfer model named MobileNetV2. This framework has the disadvantage that the detector algorithm cannot detect people with light-colored clothes wearing surgical masks. In [4], the publishers Alok Negi, Prachi Chauhan, Krishan Kumar, R.S.Rajput aim to train a deep learning model which helps in recognizing face masked people. This model uses TensorFlow, python scripts, and CNN. They use the concept of model prun with a case study named as Keras-Surgeon. The advantage of model pruning is it is easily implemented, and it assists in decreasing the dimension of the proposed model. The training technique sometimes does not work efficiently when it validates for classification of input into multiple classes on the output. In [5], the authors Mohammad Marufur Rahman,Md. Motaleb Hossen Manik,Md. Milon Islam,Saifuddin Mahmud makes an automated system by using deep learning and CNN to detect people who not wearing face protection properly in smart cities. They are monitored through CCTV cameras and take proper action on those people who violate the rule of face mask detection.In [6], the authors P.Raguraman, A. Meghana, Y. Navya, Sk. Karishma proposed a model by using computer vision. He used three colors Red, Green and Blue for tracking fundamentals of computer vision and made use of deep learning based facemask detection techniques.

In [7], the authors R. Suganthalakshmi, A. Hafeeza, P. Abinaya, A.Ganga Devi made a model that can be used in conjunction with CCTV cameras to detect unmasked people. The model is built by using OpenCv, Tensor-Flow. Along with that they also used Keras and deep learning and standard machine learning knowledge. They got good accuracy by taking less time during training and detection. In [8] and [9], the authors Mr.Alok Neg,Krishan Kumar,Prachi Chauhan,R. S. Rajput,G. Deore, R. Bodhula, V. Udpikar and V. More used CNN and VGG16 technology to detect the masked and unmasked person. In [8], they used many concepts like data augmentation and normalization. The difference is that in [9], the author uses a custom dataset named SMFD. In [10], the authors Y. Heydarzadeh, A. T. Haghighat and N. Fazeli make a new algorithm that runs faster than similar available algorithms to detect mask worn and unmasked persons.

[11] The purpose of this article from the authors A. Negi, P. Chauhan, K. Kumar and R. S. Rajput is to train face mask detection classifiers and model trimming using Kerasurgen. This individual model is used to observe whether the subject has worn a mask or not. Model pruning can be easily implemented because it can be effective to reduce the size of the model, so you can easily implement it on the built-in system. From the paper [12] the authors W. Bu, J. Xiao, C. Zhou,

M. Yang and C. Peng provides a new cnn used cascaded framework that includes 3 developed neural networks to identify mask characters. We pre-train with a wider dataset of face masks to avoid overfitting, pretending a wider dataset.In [13], the authorsI.B. Venkateswarlu, J. Kakarla and S. Prakash, introduced MobileNet as a pool block for face mask identification. A global integration layer is applied to the proposed model to consider the vector element. The global integration block of the suggested model prevents the model from overheating.

Comparative Study

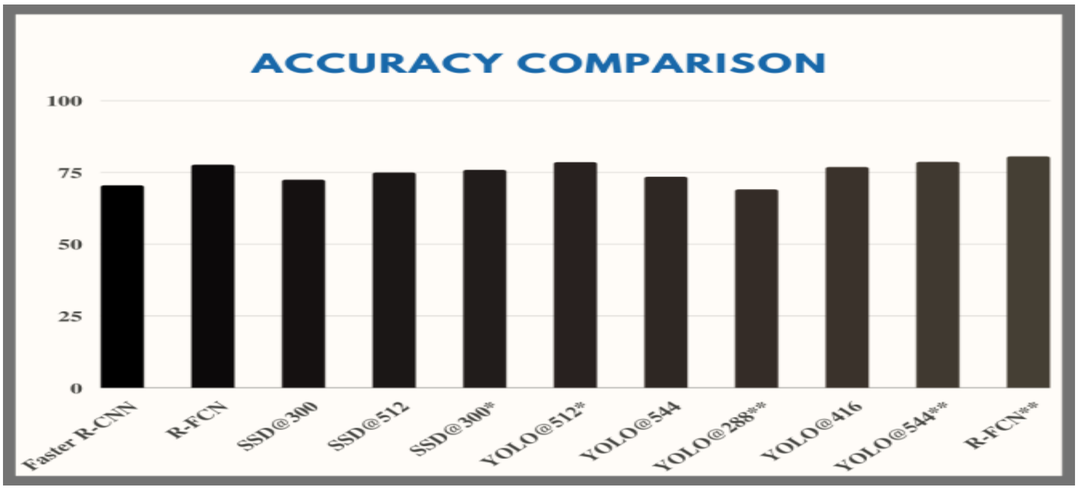

Figure. 2.1 Accuracy Comparison of various technologies Source: Data taken from Ref. No.[15]

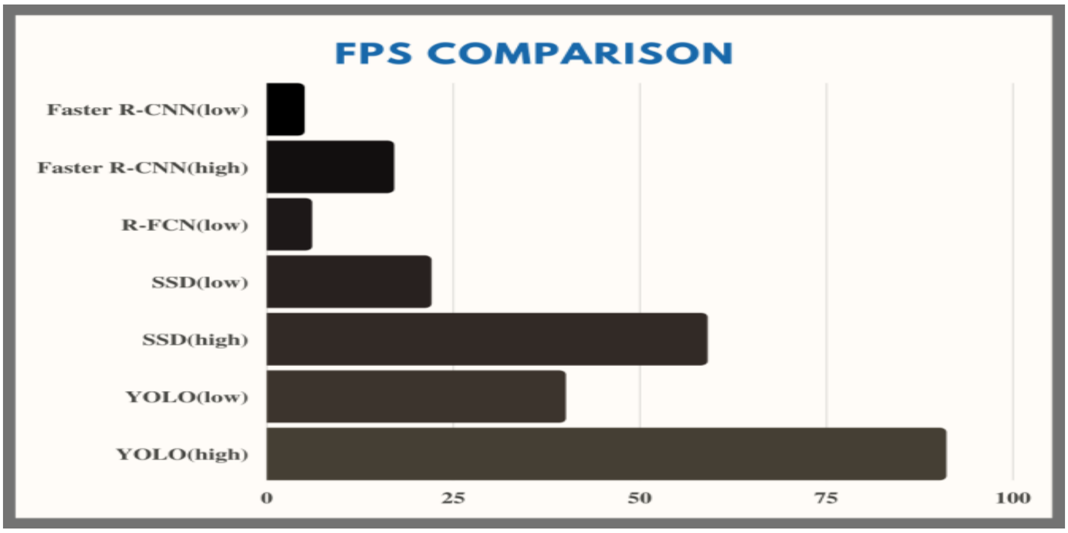

Figure. 2.2. Frames per second comparisonSource: Data taken from Ref. No. [15]

Object Detection Algorithms	Frames per second (FPS)
R-FCN	6
Faster R-CNN(low)	5
Faster R-CNN(high)	17
SSD300	22
YOLO V3	91

Object Detection Algorithms	Accuracy
R-FCN	31.5
Faster R-CNN	21.9
SSD300	23.2
YOLO V3	33
RetinaNet	40.8
FPN	33.9

The RFCN and SSD models are faster on average, but if speed is not considered, they cannot perform better than Faster RCNN in terms of accuracy. Faster RCNNs require a time of at least 100ms per frame. Using only low-resolution feature maps for detection significantly reduces accuracy. The resolution of the input image has a great influence on its accuracy. Reducing a given custom image size by half the width and height not only reduces the accuracy by 15.88% on average, but also reduces the inference time by 27.4% on average. The detection accuracy of R-CNN and R-FCN is much faster. But it is less dependent on the SSD. A comparison of frames per second is made in Figure 3. If mAP is calculated with only one IOU, use mA P@ IOU=0.75.

Accuracy wise

Proposed Design Approach

Faster R-CNN with Inception ResNet with 300 suggestions is the most accurate single model. Each image takes one second to render. An ensemble model with multi-crop inference is the most accurate model. In the previous few years, it achieved the highest level of development by detection of objects in terms of accuracy. It selects the five most diverse models using the average precision vector. Within the quickest detectors, SD with MobileNet provides the best accuracy-to-speed ratio. Within the quickest detectors, SSD with MobileNet delivers the best accuracy-to-speed ratio. Although SSDs are quick, they perform poorly when it comes to little objects when compared to other storage options. With lighter and quicker extractors, SSD can exceed R-FCN and Faster R-CNN in accuracy for largely visible objects. The comparative graph (figure 2) in overall terms of Accuracy of other various algorithms is shown above.

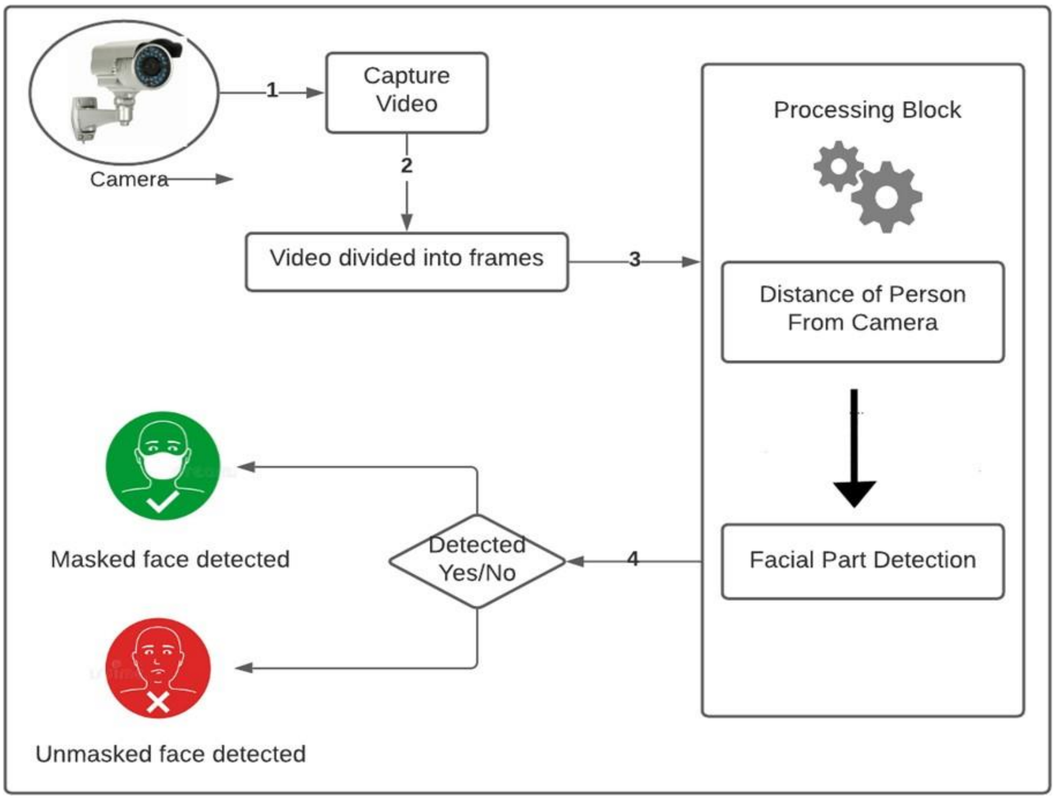

Figure. 3.1. Basic Architecture of the Model

Above image shows that overall working in short. We take input in the form of a video file/image file/ integrated webcam. We divided our implementation of the research paper problem statement into five phases. The working and importance of each phase is written as follows:

Custom Dataset Gathering

The model will be trained and tested using a custom dataset of face masks, and that custom dataset will be prepared by taking real-life images of people with and without face masks. After creating a new dataset, perform training and testing operations on it. This dataset uses a pre-trained set of weights that are derived from image classification and trained on specific data. Then try to add the same number of images to both classes (masked and unmasked). After creating the dataset, it is split into three ways: validation, training, and testing. If someone faces an overfitting issue, then that can be resolved by separating the data. Our main goal is to achieve generalization. So that our detector system performs well on training and test data. To train models, we used a training set. Also, we adjust parameters to obtain better accuracy. Hyperparameters tuned by the validation dataset.

Data augmentation for Best Results

Data augmentation is a phase used to increase the quantity of data by adding slightly edited copies of already present data or newly generated synthetic data from previous data. Since the training dataset is always countable in size, data augmentation is used to extend the data in the dataset for training by manipulating images of people in face masks in different scenarios for the dataset artificially. By blurring, horizontally flipping, shearing, contrasting, zooming, and rotating functions are used to increase custom dataset model training images. Using rescaling for the input image and processing it to a single channel, thus the size of the current model is reduced.

Training Model

Now we are going to train models. It can detect a mask on each person's face very fast. Also, it detects from every possible video camera angle. It takes the RGB input image from any orientation to obtain the required output. This is used to extract features from photos and classify them. By using feature extraction system technique, the image gets sketched. After that it creates a new image. The new generated image is more efficient than the previously used image. Various phases are shown in the figure given. The processing block analyses the distance between the person and the camera. Then it detects the face mask and eyeline in order to differentiate between a person with or without a mask. In the next step, we will combine the facemask eyes in order to differentiate between a person with or without a mask. The last step involves combining the face mask eyeline with the whole facial part of the person in order to be categorized as a mask detected or no mask worn person.

Processing & Analyzing Data for Classification

In this phase, we are creating a new window of a minimum size that is larger than the actual image size. Then it crops and passes to the YOLO algorithm to make predictions. Clipped images are stored in one place. Again, we crop the new image to the window size. The same steps must be followed for larger window size images. Now we pass the processed images to the YOLO algorithm for predictions. Earlier, stored clipped images were combined with the bounding box of the object. That's a process called "object detection" by using sliding windows. It represesses the bounding box of the high intersection over the union with a high confidence score bounding box. Repeat this process to obtain the final specific bounding box.

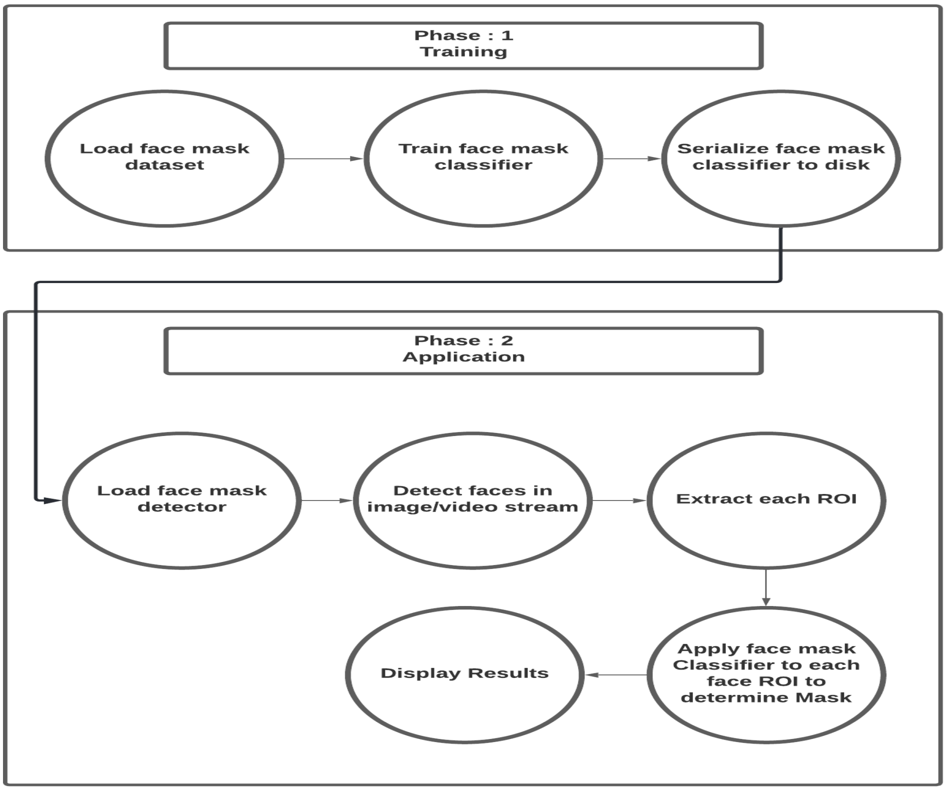

Figure. 3.2 Train and Apply Face Mask DetectorSource: Image taken from Ref. No. [14]

YOLO Algorithm

The YOLO algorithm uses a singly neural network based on CNN to process the full image or video as per the requirements. The image is then divided into regions, with bounding boxes and predicted confidence scores for each. Then we have to train custom models by keeping differences in mask-wearing styles in images and videos, and different camera angles. The system reads the input frame by frame. If it reaches EOF (End of Frame), then the system will stop. Otherwise, it will continue using the algorithm until it gets the bounding boxes for the face and the person in the frame. After processing, we get results as a person who wears a face mask properly, who's bounding box should be displayed in one color, and on the other hand, if a person is not wearing a face mask properly, then that person's bounding box should be displayed in a different color. The

YOLO method sections the target image into N grids, each of which is further responsible for detecting and localizing the object contained within it. Then, in these grids, we calculate predicted bounding box coordinates relative to actual bounding box coordinates. Also, we detect the object name as its label and the probability of the object being present in these grids. The YOLO algorithm uses the Non-Maximal Suppression technique.

All bounding boxes with lower confidence scores in the Non-Maximum Suppression technique are repressed by YOLO. The YOLO performs this by considering the highest score bounding box and then comparing it with all other probability scores that are linked with each decision. The YOLO represses bounding boxes that have a high intersection over union value with respect to the current highest probability bounding box. This process is repeated until we get a complete bounding box.

The YOLO algorithm is a combination of three techniques. Let's analyse each technique one by one:

Residual Blocks

First image is converted into different grids. The dimension of each cells are 2 dimensional like a*b matrix. As per the shown in Fig.5

Figure. 3.3 Residual Block Source: Image taken from Ref. No. [14]

There may be different grid cells of similar size in the image. The grid cell will detect an object if and only if it appears within those grid cells. If an object center seems to be within a particular grid cell, then that cell will be responsible for detecting the definite object.

Bounding box regression

Bounding box is an outline which shows the specific objects in the image. For example, in the below image the bounding box is around a car, which shows objects from an image.

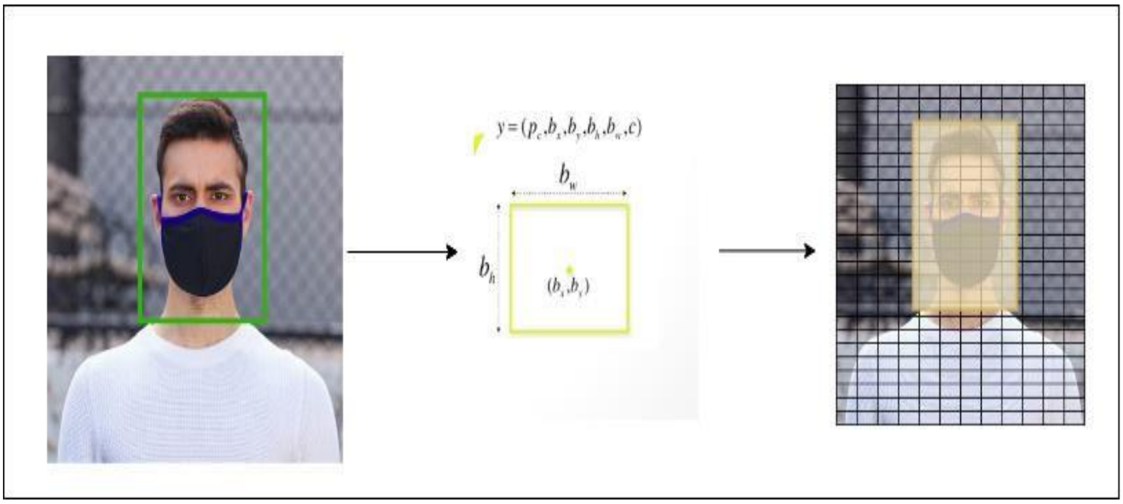

Figure. 3.4 Boundary Box RegressionImage taken from Ref. No. [14]

The YOLO algorithm uses a single bounding box regression technique to predict the height, center, class, and a width of a specific object. Confidence score of an object appearing in an image is shown in the above graphic image.

Intersection over Union (IOU)

Intersection over union displays the accuracy of the algorithm. The results are 100% accurate when the actual and predicted bounding boxes are the same. Otherwise, there are some percentage errors that we have to resolve later by doing image preprocessing or any other method. Each grid cell helps in calculating their confidence scores. Also, it helps in the prediction of the bounding box. The following image Fig. 7 shows how intersection over union works. Now in the below image of cat there are two boxes green one and blue one, blue one represents the predicted image of cat whereas green on is actual image nwo the algorithm tries to minimize the differences between these 2 boxes.

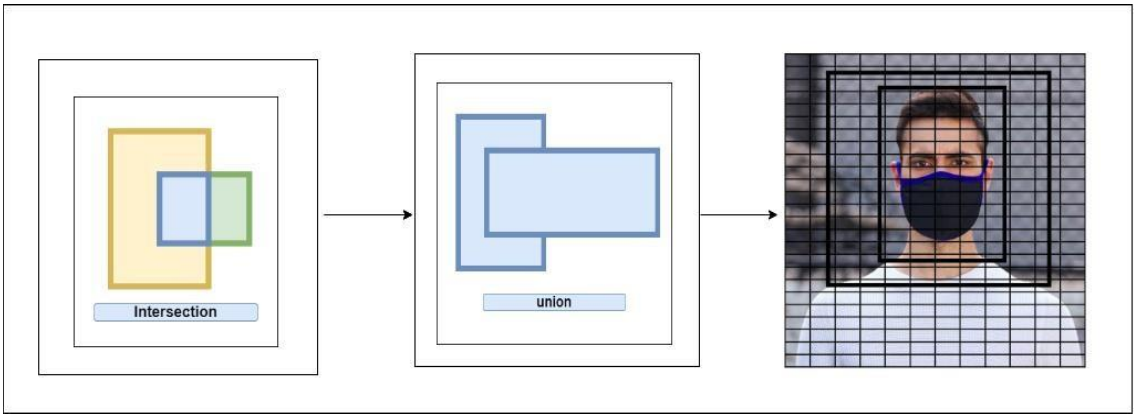

Figure. 3.5 Intersection over Union Source: Image taken from Ref. No. [14]

Combination of the three techniques

All the three techniques are combined to get more accurate results.

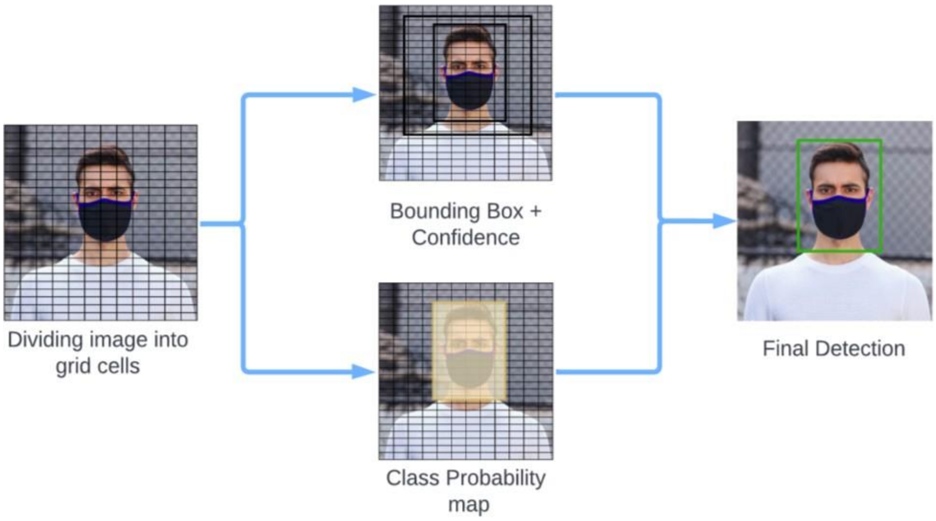

Figure. 3.6 Combination Of All Three TechniquesImage taken from Ref. No. [14]

First, the given image is converted into different grid cells, and then each object's confidence score is displayed. In the above image, there are three objects: a bicycle, a dog, and a car. The YOLO algorithm predicts all these objects at the same time. Then IOU is used to show the actual bounding boxes are equal to the predicted bounding boxes of each object. This technique is useful in removing unwanted bounding boxes that don't relate to the height, width, and other parameters of the object. Finally, we only get the modified distinct bounding boxes that are required for us. In the above image, the bicycle is in the yellow bounding box, the dog is displayed in the blue bounding box and the car is displayed in the pink bounding box. Similar procedure is applicable for face mask detection.

Implementation

For Implementation We have chosen a custom dataset. After Labeling the images Masked and Unmasked, we have cleaned the dataset that is removing zero area annotation, removing annotations outside the frames which may confuse the model. There are various websites for doing this job, for example roboflow. Further to increase the size of the dataset one can use augmentation options like to mirror image etc. One of the most important steps is to resize the image and there are various options in this step like zoom into the image or scale it or stretch the image etc.

Now we trained the model by using the Yolo-v4 object detection algorithm. For that first we have to set some parameters like learning rate, steps, max batches, subdivisions, approx. epochs and some more. We import required libraries and initialize hardware drivers. Then initialize face mask detection setup like loading class labels, yolo weights and model configuration and dataset. Then we declared the bounding box and made necessary changes to look great and precise with the human face. Then it processes the model and displays the output on the console. Also we get information in the output console like mask count and non-mask count. By considering this information it displays the message like a person is in danger or is safe. If a person is in danger it alerts the system admin by using email. It sends the email with all information like Camera ID, Status of person, No-mask count, mask-count and date and time of alert detection.

Results

The detector labeled the person's face depending on whether they are wearing a mask or not as you can see in the below images. To get insightful and intuitive results, we combined previous work and evaluated the proposed YOLO v4 algorithm based on accuracy. The comparison results show that the proposed YOLO v4 achieved higher Mean Average Accuracy (mAP) value, which resulted in higher detection accuracy by the proposed algorithm. The proposed algorithm achieved 93.95% accuracy for facial masks detection with the YOLO v4 Fig 5.1, which means higher accuracy in detecting images with faces worn masks for each class in the data set. Furthermore, the proposed algorithm achieved an intersection over union (IoU) of 71.02% in the dataset used, which means that the proposed YOLO v4 is able to detect overlapping objects with high accuracy.

Data Analysis

We gather many images and annotations from the internet so we can get the best accuracy of the model. We collect two folders to store: images: which comprises 853 .png files and annotations: which comprises 853 corresponding .xml annotations.

Then we split the images dataset in order to train, test and validate. In order to train our model and validate it during the training phase, we have to split our data into two sets, the training, and the validation set. The proportion was 90–10% respectively. So I created two new folders and I put 86 images with their corresponding annotations into the test folder and the rest 767 images into the train folder.

Sets Name	Number Of Images	Objects With Mask	Objects Without Mask
Total Images	923	3829	1074
Training Set	701	3048	869
Validation Set	101	279	50
Testing Set	121	504	157

Result Analysis: Detection in real time:

Figure. 5.1. Face Mask Detection Image

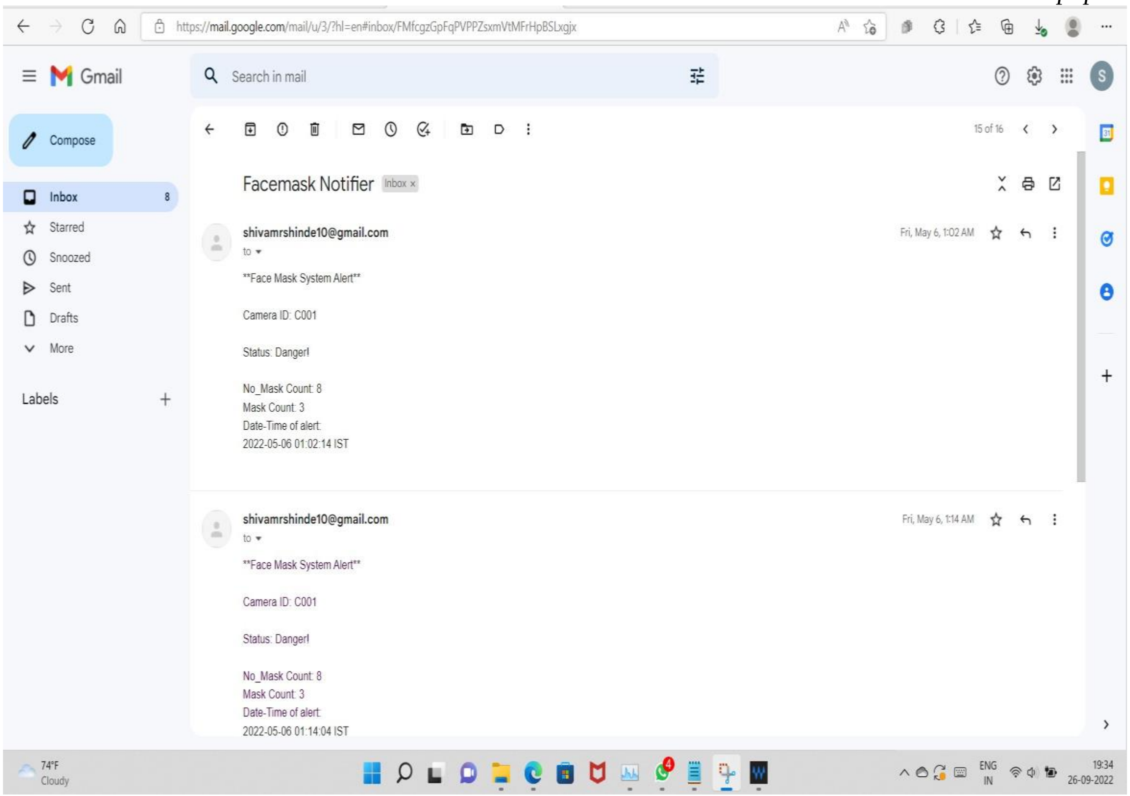

Figure. 5.2 Email warning

Performance Analysis

Model Name	Accuracy
Our Model	93.95%
MobileNet v2 Model	90.7%
Keras/Tensorflow Model	91.48%
OpenCV Based Model	86%

Conclusion

We developed a CNN-based framework to match moving video from CCTV or web cameras and then check if a person is wearing a face mask properly or not at all. We used the YOLO algorithm to develop face mask detection. This technique is based on deep learning technology that can correctly identify objects. From the scan results, the algorithm is in a position to notice and distinguish between a non-wearing and a wearing mask in any situation in the surrounding environment. Also if a person violates the rule then it sends an email to the admin. In email it displays date and time, count of masked and unmasked people. In the future, we will add heat sensing to this device to make the guard's job easier. This gadget is also expected to be put in additional crowd areas that require a face mask detector.

References

1. B.Varshini,Yogesh,Syed Danish Pasha,Maaz Suhail,V Madhumitha ,Archana Sasi.Issued online 12 August 2021 ,“IoT-Enabled Smart Doors for Monitoring Body Temperature and Face Mask Detection”
2. Susanto Susanto; Febri Alwan Putra; Riska Analia; Ika Karlina Laila Nur, “The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam”
3. Sohaib Asif,Yi Wenhui,Yi Tao,Si Jinhai ,“Real Time Face Mask Detection System using Transfer Learning with Machine Learning Method in the Era of Covid-19 Pandemic”
4. Alok Negi, Prachi Chauhan, Krishan Kumar, R.S.Rajput. ,“Face Mask Detection Classifier and Model Pruning with Keras-Surgeon”,DOI: 10.1109/ICRAIE51050.2020.935833

5. Mohammad Marufur Rahman,Md. Motaleb Hossen Manik,Md. Milon Islam,Saifuddin Mahmud,"An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network",INSPEC No: 20052690
6. P.Raguraman,A. Meghana,Y. Navya,Sk. Karishma,"Color Detection of RGB Images Using Python and OpenCv", ISSN:2456-3307
7. R. Suganthalakshmi, A. Hafeeza, P. Abinaya,A.Ganga Devi,"Covid-19 Facemask Detection with Deep Learning and Computer Vision",Paper ID : IJERTCONV9IS05017 Volume & Issue : ICRADL – 2021 (Volume 09 – Issue 05) Published (First Online): 27-03-2021 ISSN (Online) : 2278-0181
8. Mr.Alok Neg,Krishan Kumar,Prachi Chauhan,R. S. Rajput ,"Deep Neural Architecture for Face mask Detection on Simulated Masked Face Dataset against Covid-19 Pandemic" Published in: 2021 International Conference on Computing Date of Conference: 19-20 Feb. 2021 Date Added to IEEE Xplore: 12 April 2021 INSPEC Accession Number: 20633697 DOI: 10.1109/ICCCIS51004.2021.9397196 Publisher: IEEE
9. G. Deore, R. Bodhula, V. Udpikar and V. More, "Study of masked face detection approach in video analytics", 2016 Conference on Advances in Signal Processing (CASP), 2016, pp. 196-200, doi:10.1109/CASP.2016.7746164.
10. Y. Heydarzadeh, A. T. Haghighat and N. Fazeli, ‘‘Utilizing Skin Mask and Face Organs Detection for Improving the Viola Face 24 Detection Method" 2010 Fourth UK Sim European Symposium on Computer Modeling and Simulation, 2010, pp. 174-178, doi: 10.1109/EMS.2010.38.
11. A. Negi, P. Chauhan, K. Kumar and R. S. Rajput "Face Mask Detection Classifier and Model Pruning with Keras-Surgeon", "Face Mask Detection Classifier and Model Pruning with Keras-Surgeon," 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering(ICRAIE),2020,pp.1-6,doi:10.1109/ICRAIE51050.2020.9358337.
12. W. Bu, J. Xiao, C. Zhou, M. Yang and C. Peng , "A cascade framework for masked face detection," 2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2017, pp. 458-462, doi: 10.1109/ICCIS.2017.8274819.
13. I.B. Venkateswarlu, J. Kakarla and S. Prakash ,"Face mask detection using MobileNet and Global Pooling Block", "Face mask detection using MobileNet and Global Pooling Block," 2020 IEEE 4th Conference on Information & Communication Technology(CICT),2020,pp.1-5,doi:10.1109/CICT51604.2020.9312083.0.
14. https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/#:~:text=YOLO%20algorithm%20can%20be%20applied%20in%20the%20following,systems%20to%20enforce%20security%20in%20an%20area.%20
15. https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359