Journal of Advances in Mathematical & Computational Sciences
An International Pan-African Multidisciplinary Journal of the SMART Research Group
International Centre for IT & Development (ICITD) USA

CrossREF Member Listing - https://www.crossref.org/06members/50go-live.html

Yolo Real-Time Face Mask Detection In An Automatic Door System

[1]Abiola O. A., [2]Okechukwu R. I. & [3]Akinola S.O.
[12,3]Department of Computer Science
University of Ibadan
Ibadan, Nigeria.
E-mails: oladimejiarowolo@yahoo.co.uk, solom202@yahoo.co.uk, icicle621@yahoo.com
Phone Nos: +2347055535537, 08169748281, 08135098680

ABSTRACT

COVID-19 posed a huge challenge to the world when in early 2020 it spread rapidly to the rest of the world from China and forced the world into lockdown. The mask played a big role in helping to curtail the spread of the virus and in flattening the curve. Before COVID-19, some industries do rely on facemask usage for their workers to avoid the risk of exposing them to harmful gaseous chemicals that would be hazardous to their health. To prevent the risk of exposing people to the COVID-19 virus and other harmful substances when they go to public places like schools, hospitals, parks, etc., would require efficient safety checks by monitoring and regulating people wearing facemasks and those who are defaulting. While this can be achieved through sheer human effort, Artificial Intelligence through Computer Vision can provide a more efficient, accurate, hassle-free and sustainable solution. The focus of this study is providing a solution for facemask detection that could be incorporated into an automatic door system using YOLOv5 arduino uno (the latest in the YOLO architectures) for operating entrance doors in public places. With this system, people who wear facemasks are detected and received positive response of having the door opened for them while masks were not detected in the defaulters' cases, and they received the negative feedback having the door remain shut on them and warning beep signal lighting. The system if fully implemented will help to help stop the spread of COVID-19 virus and other air-borne diseases.

Keywords: Arduino uno, COVID-19, Facemask detection, Kaggle dataset, YOLO V5 architecture

Abiola O. A., Okechukwu R. I. & Akinola S.O. (2024): Yolo Real-Time Face Mask Detection In An Automatic Door System. Journal of Advances in Mathematical & Computational Science. Vol. 12, No. 1. Pp 87-99.
Available online at www.isteams.net/mathematics-computationaljournal. dx.doi.org/10.22624/AIMS/MATHS/V12N1P7

1. INTRODUCTION

Coronavirus Virus Disease (COVID-19) became a household name due to its devastating global effect early 2020, even though Corona virus has been around for quite a while and there has been some different outbreaks in the past twenty years (Mandal, 2021).

One of the modes of transmission of COVID-19 is airborne transmission where people breathe in droplets released by an infected person through coughing, sneezing etc. Since the COVID-19 pandemic swept the globe, governments around the world have taken severe but necessary measures to stem its spread. Normal day-to-day operations came to a total halt as a result of this. When the curve flattened in multiple countries after months of lockdown, the community became agitated. Relevant organizations, like World Health Organization (WHO), have long issued guidelines to limit people's exposure to the virus. Some of the measures include frequent hand-washing or use of hand sanitizer, social distancing and use of face masks. As people are coming back to everyday life from the government imposed lockdown, the government has further mandated the compulsory use of face masks in public places.

Even before the pandemic, numerous businesses, such as chemical, paint, and cotton industries, relied heavily on masks to protect individual lives from dangerous diseases. We are currently concentrating on preventing the spread of COVID-19 in public locations such as ATM galleries, where people did not wear masks and thereby disseminated or became infected with the virus. It is one thing to mandate people to use face masks in a public place, it is another thing to effectively enforce it. Trying to manually monitor people's compliance with using facemasks in public places such as schools, hospitals, malls, etc. would be very tasking in terms of human, financial and time resources. Moreover, having a human monitor would require a lot of mental effort and concentration which humans may find hard to sustain. This is where automation is needed. Having an automatic system that checks for face mask usage and acts on it will go a long way in ensuring compliance.

This was achieved by building an automatic door system that allows doors to open automatically by sensing and responding to the approach of a person. In this case, though, the automatic door will be sensing for whether the approaching person is using a face mask or not. Our proposed solution in this study makes use of the YOLO architecture, and integrates Arduino uno, micro-controllers for automatic opening and closing of doors in public places. YOLO (You Only Look Once) family of models offer faster object detection and especially, they do that in real-time as opposed to R-CNN (Region-based Convolutional Neural Networks). YOLO architecture is one of the latest and a very important algorithm in object detection. Object detection is a task in computer vision that involves identifying the presence, location and type of one or more objects in a given photograph. It builds upon object recognition, object localization and object classification.

Computer vision is one of the most successful areas of research in machine learning. This field of AI helps computers to derive meaningful information from images, videos and other visual inputs and take action based on it. In simple terms, a person wearing facemask would be allowed to enter through the automatic door system, whereas a defaulter would be denied entry. Deep learning is easier and faster to build and deploy than prior types of machine learning. Because of the availability and advancements in hardware and cloud computing resources, deep learning and deep neural networks have gone from the theoretical realm to practical applications. One of such hardware applications which is the focus of this study is integrating it in an automatic door system. An automatic door, often known as an auto door, is one that opens automatically when it detects a person approaching. By definition, an Automated Door System is one in which the door opens automatically rather than manually. Automated Doors can be used in a variety of settings, including hospitals, retail malls, airports, and businesses - anywhere where an autonomous door experience is desired.

YOLO is extremely fast in frame detection as a regression problem and don't need a complex pipeline. It runs neural network on a new image at test time to predict detections (Redmon, et. At., 2016). YOLO came on the computer vision scene with a paper released in 2015 by Joseph Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection," and immediately got a lot of attention from fellow computer vision researchers. Convolutional Neural Networks (CNN) such as Region Convolutional Network (R-CNN) used Regions Proposal Networks (RPNs) before YOLO was invented, it produces proposal bounding boxes on the input image first, then runs a classifier on the bounding boxes and then apply post-processing to remove duplicate detections and refine the bounding boxes. It was not suitable for training individual stages of the R-CNN network separately. The R-CNN network was both difficult and sluggish to optimize. Inspired by the GoogleNet architecture, YOLO's architecture has a total of 24 convolutional layers with 2 fully connected layers at the end.

So far, combining with many of the most innovative ideas coming out of the computer vision research community, YOLO has been upgraded to five versions and assessed as one of the outstanding object detection algorithms. The 5th generation of YOLO, YOLOv5, is the latest version not developed by the original author of YOLO. However, the performance of the YOLOv5 is higher than the YOLOv4 in terms of both accuracy and speed,(Do Thuan 2021). YOLOv5 is an open-source project that consists of a family of object detection models and detection methods based on the YOLO model pre-trained on the COCO dataset. It is maintained by Ultralytics and represents the organization's open-source research into the future of Computer Vision works. This paper addresses the development of an automated door system that uses YOLO (You Only Look Once) algorithm in object detection and recognition of facemask. YOLO is a state-of-the-art, real-time object detection system.

2. Related Works

Artificial Intelligence (AI) algorithms can tackle learning, perception, problem-solving, language-understanding and/or logical reasoning, Mohammed (2019). Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think. Sharma (2020), developed model uses the YOLOv5 and TensorFlow technologies for processing images and real-time videos. This tool allows the model to load the images for the testing process. From the results it can be said that the developed model is able to detect whether an individual is wearing a facemask or not by accurately classifying the persons who wear a mask as well as persons who are not wearing masks. The model quickly learns the parameters. It collects the video from the camera, process the video, identifies the objects, and finds if a person wears mask or not. OpenCV is used to train the model for finding the person with and without a facemask.

For doing this task, the Deep Neural Network (DNN) module was used from OpenCV, which contains a 'Single Shot Multibox Detector' (SSD) as a face detector PanelPreeti, *et, al,.* (2021) object detection model with ResNet-10 as its backbone architecture. Deep learning-based approach for detecting masks over faces in public places to curtail the community spread of Coronavirus is presented. The concept of transfer learning is applied to utilize already learned attributes of a powerful pre-trained convolutional neural network in extracting new features for the model Sethi, *et al*. (2021). Suresh *et al.* (2021), proposed an Optimistic Convolution Network that helps to automatically monitor and ensure whether in public the people are wearing masks or not.

Their system uses the TensorFlow and Keras algorithm to detect whether or not, an individual is wearing a face mask along with the Convolutional Neural network model. The proposed system loaded image dataset from Keras and then the images are converted into an array, later MobileNet is used to preprocess input image and to append image to the data list. In the proposed system the main contribution includes people face identification and face mask detection.

Sanjaya and Rakhmawan (2020) conducted their experiments on two original datasets which are the datasets from the Kaggle dataset and the Real-World Masked Face dataset (RMFD). The datasets were used for the training, validation, and testing phases. The model was implemented based on the dataset gathered from 25 cities in Indonesia. Some cities were chosen based on data availability. The dataset was taken from some sources, for instance, public place CCTV, shop, and traffic lamp camera. After the training, validation, and testing phase, the model provides the percentage of people using face mask in some cities with high accuracy.

To predict whether or not a person has worn a mask correctly, Nagrath, *et al.* (2021) proposed a system named SSDMNV2. The initial stage would be to train the model using a proper dataset. After training the classifier, an accurate face detection model is required to detect faces, so that the SSDMNV2 model can classify whether the person is wearing a mask or not. The task in this paper is to raise the accuracy of mask detection without being too resource-heavy. For doing this task, the DNN module was used from OpenCV, which contains a 'Single Shot Multibox Detector' (SSD)

A pre-trained MobileNet was proposed with a global pooling block for face mask detection. The pre-prepared MobileNet takes a shading picture and creates a multi-dimensional component map. The worldwide pooling block that has been used in the proposed model changes the element map into an element vector of 64 highlights. At long last, the softmax layer performs paired order utilizing the 64 highlights, Riya and Rutva (2020). Areas where computer vision techniques have been applied and also applications of image processing and machine vision can be further obtained from Rasch (2021).

From the literature research, it is found out that existing works only detect facemasks after training. In this present study, Arduino automatic door opening was incorporated in such that after detecting facemasks, the door opens automatically and closes between two to three seconds so as to prevent any other unauthorized entry.

3. RESEARCH METHODOLOGY

Building an automatic door system involves the design and implementation of the hardware and software components that make up the system. The major steps that were undertaken are as follows:

(a) Data Collection: The data collected for the model is a dataset publicly available on Kaggle, supplemented by images from the internet. The dataset is made up of 848 images classified as "with facemask", "without facemask", and "incorrect facemask".
(b) Model Training and Testing: The dataset was used to train our model (weight) using Google Colab (Virtual Machine) and inferences were run on some sample images as tests.
(c) Design and Implementation of the Automatic Door System: Here, the software model that we have trained was integrated in an automatic door system and was physically implemented (simulated) by employing the use of an Arduino device.

(d) Implementation: The proposed system was implemented using Python programming language for the face mask detection and C++ code for the automatic door system

3.1 Physical Modeling

This represents the design of the physical entities of the system. It deals with the partitioning of entire system into modules. It includes the following diagrams: A context diagram is a data flow diagram that shows the system boundaries, external entities that interact with the system, and the major information flows. The context diagram has only one process which represents the entire system itself. The source/sink represents its environmental boundaries. Figure 1 represents the context diagram of the system.

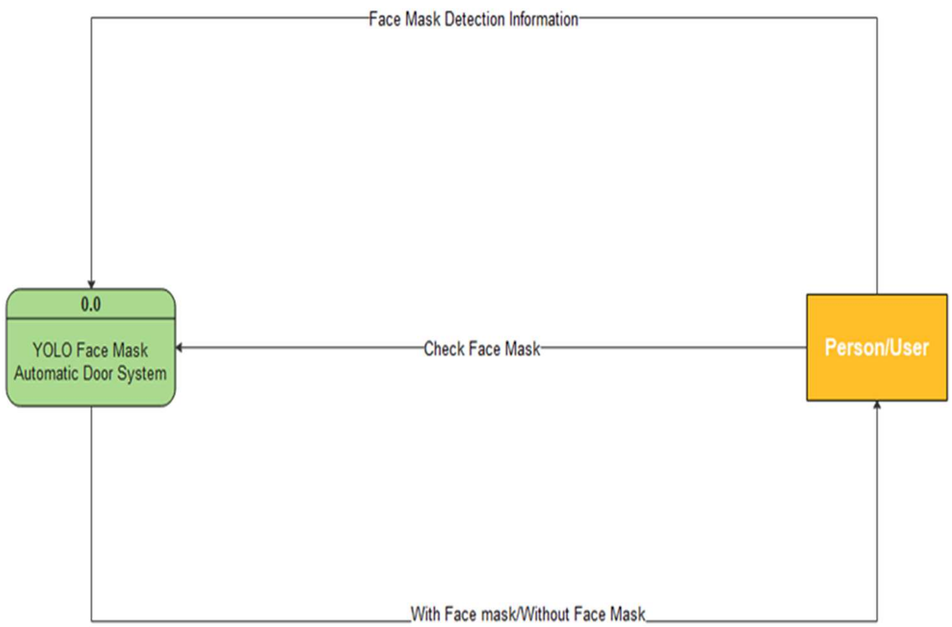

Figure 1: Context Dataflow Diagram of the Face mask detection system

3.1.1 Description of the Components of the Context Dataflow Diagram

1. **Process**: The context dataflow diagram of the system contains only one process named; Face mask detection System. This process represents the entire process of the system.
2. **Data flows**: The arrows in the system represent the data flows. The data flows are labeled and depicts the kind of data that go into and out of the system.
3. **Data source / sink**: The data sources/sinks in the facemask detection system is the user of the system. These data sources/sinks can send and/or receive data from the system.

Figure 2 is a flow chat that illustrates the working principle of the facemask detection system

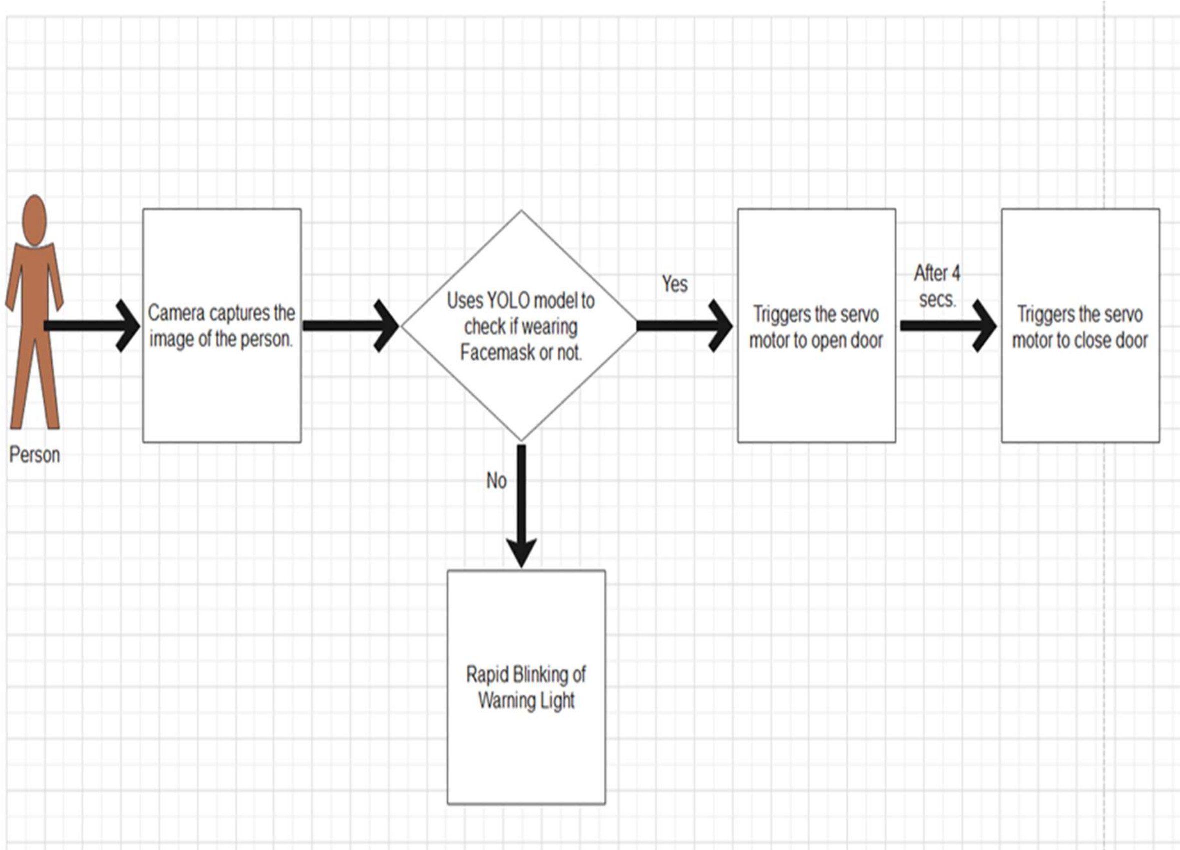

Figure 2: System's Flow chat

3.1.2 Use Case Modeling

Certainly! Let's rephrase that:

A use case provides a valuable method for representing and prioritizing user-centric requirements.

Use-case-based prioritization involves ranking use cases to achieve business value based on specific criteria. It helps ensure that the most critical functionalities are addressed early in the development process (Odeh and Al-Saiyd, 2023). It is used to describe the proposed functionality of a new system. It represents a discrete interaction between a user and the machine (Somerville, 2010). Figure 3 shows the Use Case Model for the Face Mask Detection System.

The Use Case shows that users need to stand in front of the system for authentication before they will be granted access to enter.

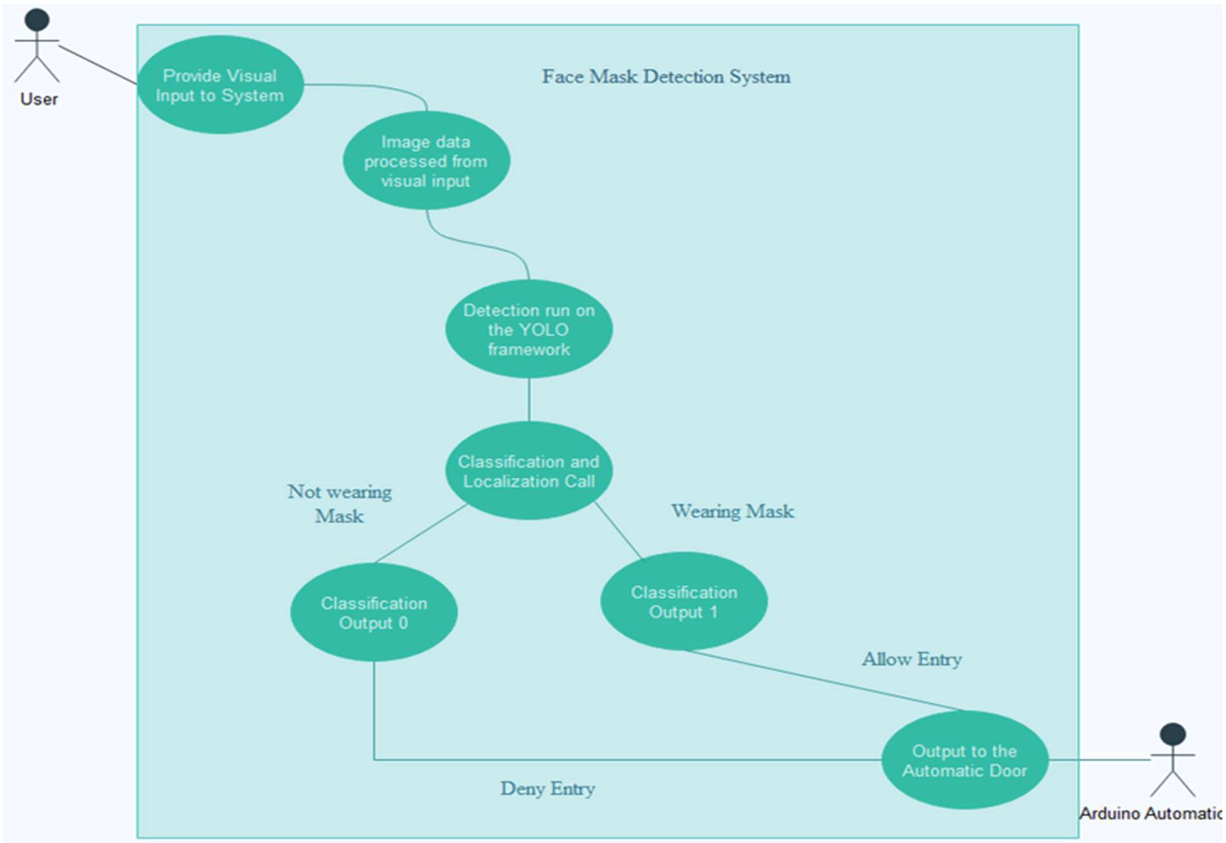

Figure 3: Use case diagram of the Facemask detection system

3.1.3 System Architecture

System architecture is the conceptual model that defines the structure, behaviour and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system (Valacich *et al.*, 2012). The output of an architectural design process is an architecture model that describes how the system is organized as a set of communicating components. Figure 4 depicts the interaction between the system components, both external and internal.

The process for getting the operational Facemask detection model starts with getting the required dataset for training the model. The dataset is made up of publicly available image data available on Kaggle. This dataset was supplemented with images gathered from the internet. The dataset is made up of 848 images classified as “with facemask”, “without facemask” and “incorrect facemask”. The image data that were collected from the internet were annotated on Roboflow (a Computer Vision developer framework for better data collection to preprocessing, and model training techniques) using its Labelling interface by drawing bounding boxes around the area of detection in the images (the nose and mouth area and then filling in the text field with the appropriate class (with or without facemask).

After this, the output dataset increased and was split into three subsets of 1600 training data, 170 validation data and 85 testing data. The dataset that was generated were exported into a Yolov5 PyTorch format for use in training the Model. The exported dataset contains separate folders for each split (train, validation and test) with each of them containing the .jpg augmented images, the corresponding .txt annotation files and a _pytorch.labels file with the labels in the order:

- incorrect_mask # label 0
- with_mask # label 1
- without_mask # label 2

The training was done using one of the official YOLOv5's pre-trained weights, Yolov5s.pt which is the lightest in order to derive our own custom trained weight on Google Colab.

Inferences on some images were run using the custom-trained Facemask detection model by running the command *detect.py* and feeding it the location of the images to run the inferences on. Further modifications were made to change the training *data.yaml* from 3 classes to two:

- with_mask #1
- without_mask #0

This had to be done because the objective of our facemask detection is only concerned with checking if people are wearing facemasks or not and is not concerned with how people are wearing the masks. After ensuring that the model was running inferences correctly on the test images, the custom-trained weight was exported from the Google Colab environment and downloaded it into the local system which serves as the operational environment.

The hardware components needed to set up the automatic door system are:

(a) An Arduino uno board with built-in LED display
(b) Servo motor
(c) Jumper wires
(d) System port cable

The only software needed is the Arduino IDE for writing and sending programming signals to the hardware. The code was written in C++ programming language. The Arduino board was correctly connected to its other components using the jumper wires and the board pins. The Arduino was connected to the servo motor and to the port which would serve as both data and power input from the laptop. The code for operating the system was written and saved. The project files were gathered in one project folder with the major ones being the YOLOv5 dependencies, the weights file (Facemask.pt) and the Arduino code (servo.ino). A new Python detection function for our custom trained model was written to integrate the Arduino. A serial connection was established between the Computer System using the System-to-Arduino port and cable. The necessary driver for the Arduino was also downloaded and installed.

Inferences were run as usual using the new Python detection function. More modifications were introduced to the detection function to prevent it from false-detecting when no one is on the camera. Due to the speed of detection of the YOLOv5 algorithm, delays and halts were introduced to accommodate the time it would take for someone to enter the door. Beep signals of 0 and 1 would be sent to the Arduino depending on if no mask was detected or mask was detected.

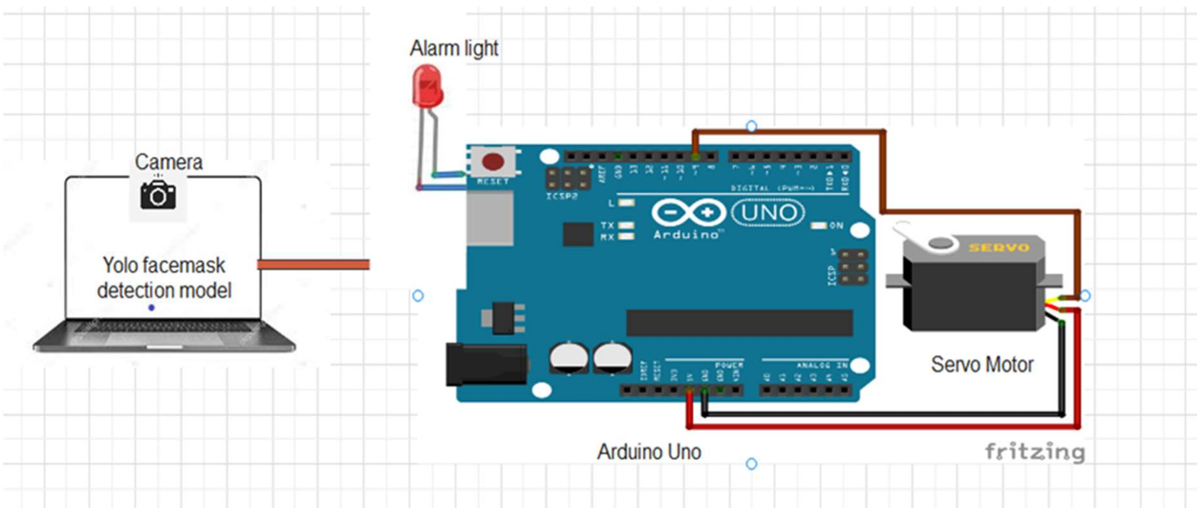

Figure 4: System Architecture

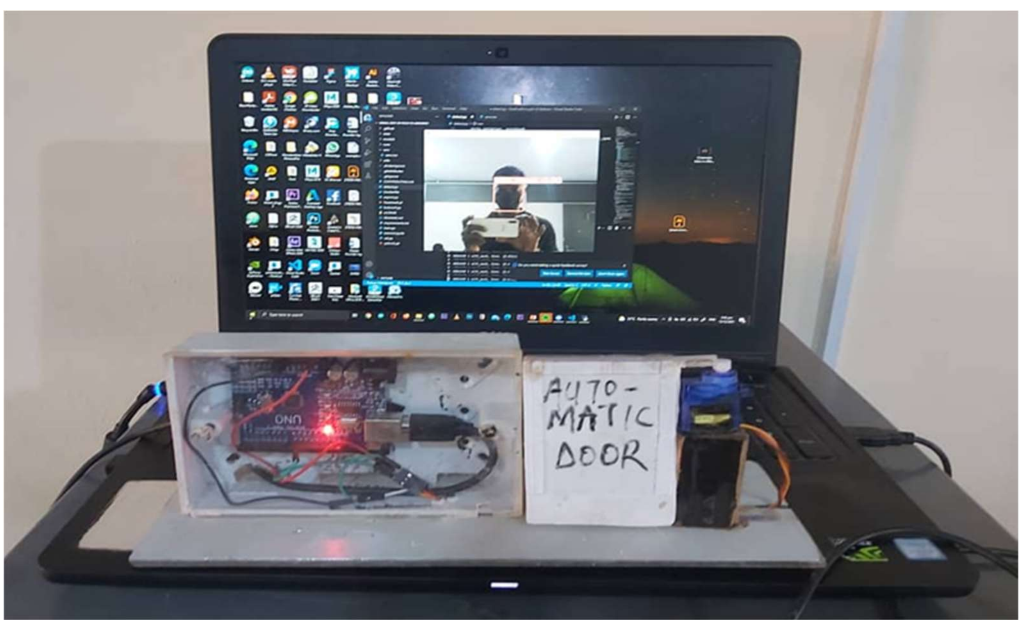

Figure 4: Live Demonstration of System

4. RESULTS AND CONCLUSION

Inferences run with the web camera gave the following results depicted in Figures 5 and 6. In Figure 5, the Facemask detection model was able to detect a bounding box containing the object (a person) and the corresponding label (without mask). In Figure 6, the Facemask detection model was able to detect a bounding box containing the object (a person) and the corresponding label (with mask).

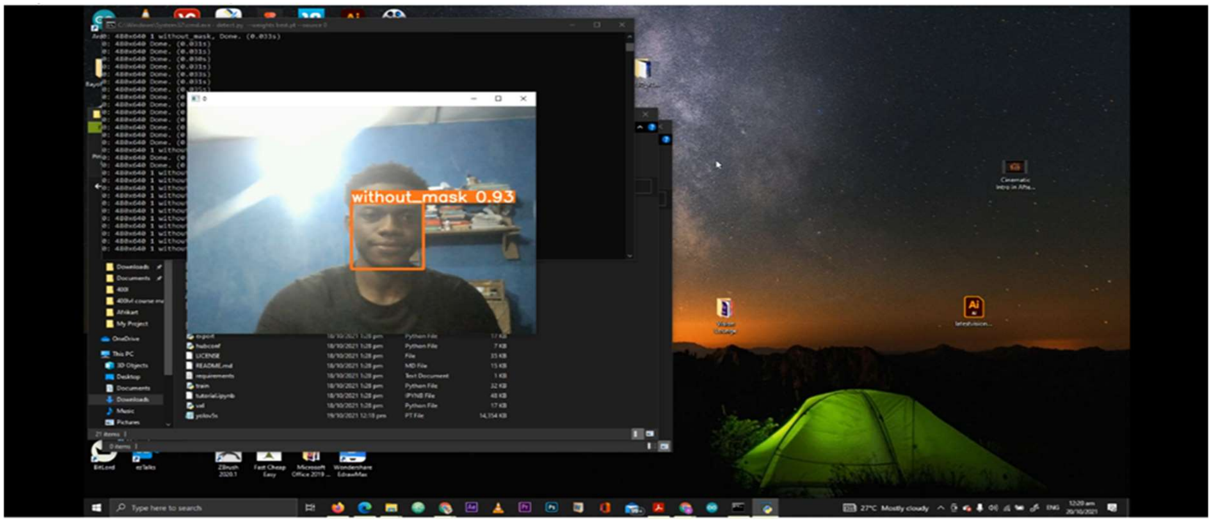

Figure 5: When no mask is detected on a face

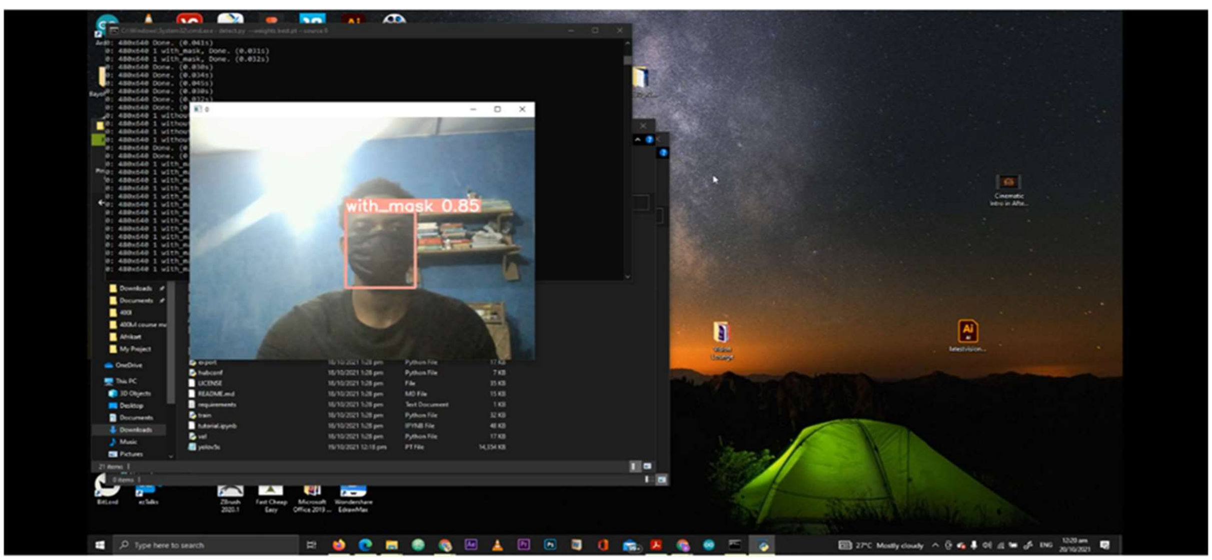

Figure 6: When facemask is detected.

From Figure 7, the door is simulated in such a way that the servo motor would remain in the default 90 degrees position (closed) if no mask is detected (check first photo by the left). When a facemask is detected on the face, the servo motor moves 0 degrees (opens) to allow entry (check middle photo), and then closes after 4 secs (returns to 90 degrees) (check last photo by the right). A delay of 50 beeps was introduced in the detection to ensure a detection-door cycle runs completely before another one starts.

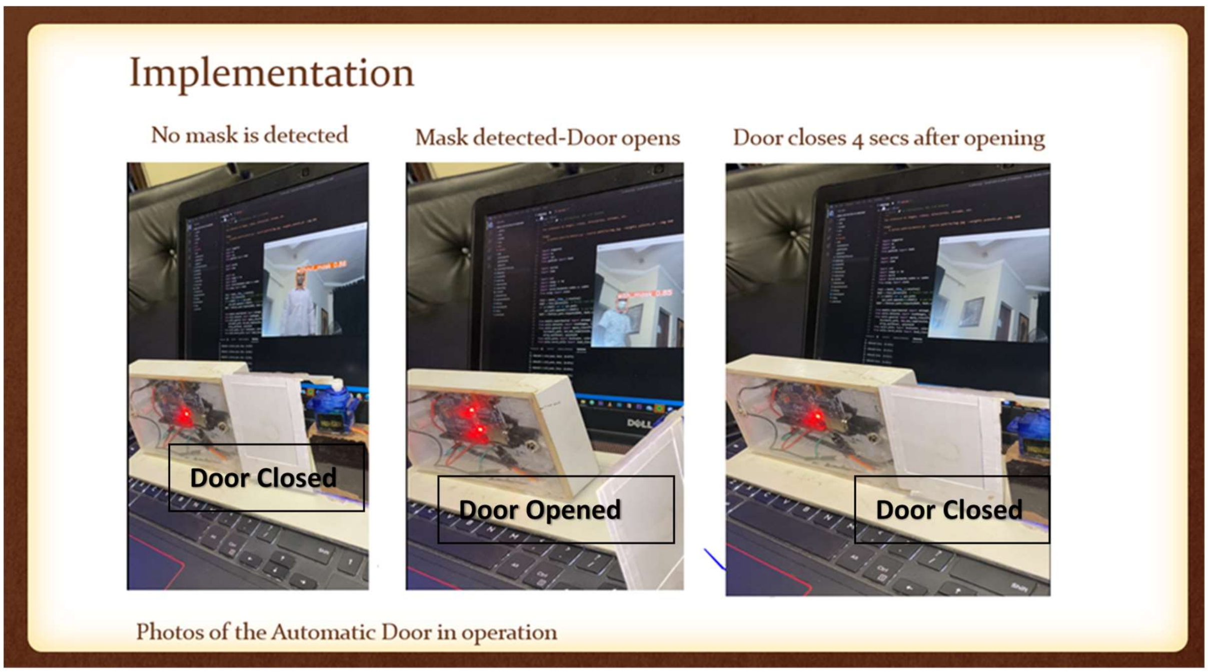

Figure 7: Automatic Door Operations

5. CONCLUSION

This research integrated an automatic door system that works with real time face mask detection where object detection takes place using YOLO v5. The two classes that were simultaneously detected are the masked and unmasked faces. After rigorous testing, it was noticed that the model was detecting too fast and as a result was false-detecting positively even when there was no one in view. Thus, the algorithm was optimized by introducing delays in the detection which reduces the false positive. Without any addition of time consuming computations or image warping, this light-weight model has been calibrated to work well and can be used in real time due to high FPS (Frame Per Second) detection and good accuracy. It was also refreshing to see that the model accurately ran detections on all types of human faces irrespective of the racial differences between the training data and the data that was eventually used for the inference.

The implemented YOLOv5 algorithm facemask detection in an automatic door system can be applied in public places or buildings like Pedestrian walks, Hospitals, Schools that require strict observance of the COVID-19 safety protocol of wearing facemask. Also, manufacturing industries like Paint factories, Cotton factories, etc. where facemask usage is absolutely necessary and failure to abide by the safety measure will expose the defaulter to harmful gaseous chemicals. Observations from this study shows that while the system is trained to detect Facemask usage where the face mask is covering the nose and mouth, the system made false positive detections when the nose and mouth areas were covered with hands or in fact anything that is not a nose mask. Thus, the system would be easily by-passed if the users know they can exploit this flaw. Further research is encouraged to find out if this flaw can be fixed. A possible way to mitigate this problem is to add more diversities in the training images and running additional data augmentation techniques.

Other problem observed with this system is that the system would not be able to make accurate detections in poor lighting environments. It needs to be exposed to proper lighting in order to be able to run proper inference on the visual data being fed to it real time. This area could be further researched to see where improvements can be made in the system, be it the camera or the trained model itself.

REFERENCES

1. Anisimov, Dmitriy and Khanova, Tatiana. (2017). Towards lightweight convolutional neural networks for object detection, *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, 1-8. 10.1109/AVSS.2017.8078500.
2. Anita Mandal and Prabir K. Mandal (2021). COVID-19 Pandemic, Disparity and Vaccine: An Update, *J. Biotechnology and Bioprocessing* 2(1).
3. Do Thuan (2021). Evolution of YOLO Algorithm and YOLOv5: The State-Of-The-Art Object Detection Algorithm. *Artificial Intelligence, Intelligence system. Tutorialspoint, simply easily learning*. https://tutorialspoint.com
4. Face Mask Detection using YOLOv5 for COVID-19 https://scholarworks.calstate.edu› downloadsPDF
5. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C. (2016). Ssd: Single shot multibox detector. *In European conference on computer vision* (pp. 21-37). Springer, Cham.
6. Nagrath *et al.* (2021). Corrigendum to "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 [Sustain. Cities Society. 66 (2021) 102692].
7. Odeh, Y.; Al-Saiyd, N. (2023). Prioritizing Use Cases: A Systematic, literature review. Computer 2023, 12. 136. http://doi.org/10.3390/computers 2070136.
8. *PanelPreeti et al. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 [Sustainable Cities and Society Volume 6, March (2021), 102692].*
9. Redmon, J. Divvala, S. Girshick, v and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

10. Rasch C. (2021). Computer Vision: An Overview For Enthusiasts. https://www.researchgate.net/publication/336460083. Accessed December, 2021.
11. Riya C. S. and Rutva J. S. (2022). Detection of Face Mask using Convolutional Neural Network, https://arxiv.org/ftp/arxiv/papers/2106/2106.05728.pdf, accessed March 2022.
12. Sanjaya S. A. and Rakhmawan S. A. (2020) Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic, *International Conference on Data Analytics for Business and Industry: Way towards a Sustainable Economy* (ICDABI)
13. Sharma V· (2020). Face Mask Detection using YOLOv5 for COVID-19. An MSc Dissertation, California State University San Marcos November 24, 2020.
14. Shilpa Sethi, Mamta Kathuria, Trilok Kaushik (2021). Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread, *Journal of Biomedical Informatics*.
15. Suresh K1, Palangappa MB2, Bhuvan S3 (2021). Face Mask Detection by using Optimistic Convolutional Neural Network. *Proceedings of the Sixth International Conference on Inventive Computation Technologies* [ICICT 2021] IEEE Xplore Part Number: CFP21F70-ART; ISBN: 978-1-7281-8501-9.
16. Ziyad Mohammed (2019). Artificial Intelligence Definition, Ethics and Standards, *Electronics and Communications: Law, Standards and Practice* | 18ELEC07I.