Machine Vision Surveillance System - Artificial Intelligence For Covid-19 Norms

Ritika Nigam, Arpan Kundu, Xinrui Yu and Jafar Saniie
Embedded Computing and Signal Processing (ECASP) Research Laboratory (*http://ecasp.ece.iit.edu*)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago IL, U.S.A

***Abstract*—This paper presents the design and implementation of the Machine Vision Surveillance System - Artificial Intelligence (MaViSS-AI) for Covid-19 Norms using jetson nano. This system is designed to be cost-effective, accurate, efficient, and secure. The proposed system tracks and counts humans for monitoring social distancing and detects face masks using object detection methods. We used YOLO as an object detection method and neural network to detect a person and count them. And for social distancing monitoring the concept of the centroid is based on calculating the distance between pairs of centroids, and thus checking whether there are any violations of the threshold or not. To detect the face mask, a YOLO V4 deep learning model is used as the mask detection algorithm. The system also raises alerts when any suspicious event occurs. Given this alert, security personnel can take relevant actions. This research aims to provide a holistic approach to overcoming the real-time challenges encountered during the monitoring of Covid-19 norms.**

Keywords—Automated Surveillance System, Artificial Intelligence, Real-time object detection YOLO, Real-time Alerts Module

I. INTRODUCTION

In March 2020, World Health Organization (WHO) declared the pandemic due to COVID-19 since then the coronavirus outbreak has caused a global disaster with its deadly spreading [1]. Though vaccines have been developed by various nations, as stated by the World Health Organization (WHO), vaccines rarely protect 100% of the recipients and vaccinated individuals still run the risk of contracting the disease. And also with the increase in the mutation of the virus, new variants of coronavirus are being emerged which is eventually decreasing the effectiveness of vaccines against the coronavirus [2]. To curb this pandemic, it is important to monitor whether people are abiding by all the necessary precautions i.e. maintaining social distancing norms, wearing face masks, and crowd management.

Manual monitoring of these norms is difficult and tends to be quite inefficient and inaccurate. This necessitates the design and realization of an automated machine vision system for monitoring the covid norms in real-time [3-4]. This encouraged us to design an Artificial Intelligence-based Machine Vision Surveillance System (MaViSS-AI) for real-time monitoring of COVID-19 norms. This system is cost-effective, accurate, practical, and secure and would overcome the real-time challenges faced during manual monitoring of norms. Since the outbreak of the pandemic, many systems have been developed to monitor different aspects of COVID-19 norms like face mask usage and social distancing, but most of them have addressed only specific aspects of the norms and do not have a holistic approach [5-6]. Moreover, these systems mainly focus on visualizing the system output but do not have any alerts component to keep track of violations in norms and notify the monitoring user about the same [7]. Our proposed system has a more holistic approach to address these shortcomings and offers a complete platform for monitoring the COVID-19 norms, as well as sending alerts in real-time directly to the monitoring user's smartphone using an instant messaging service like Telegram. This system would replace many physical eyes with computer visions and thereby providing an accurate and efficient monitoring system. The system will be used for monitoring three different tasks.

- Detecting and tracking humans for monitoring social distancing norms and counting the total humans for crowd management.
- Detecting face masks and keeping the track of face mask usage by the detected people.
- Raising real-time alerts using a telegram bot whenever any of the following norms are breached.

Our research aims to limit the impact of the coronavirus epidemic through an automated surveillance system consisting of three different modules which monitor three different tasks including real-time detection, tracking, and alarming alleviating the COVID-19 surge.

II. SYSTEM MODULES

The system MaViSS-AI is the integration of three different modules. Each module performs a different monitoring task with the help of a real-time object detection method (YOLO) and OpenCV library of Python.

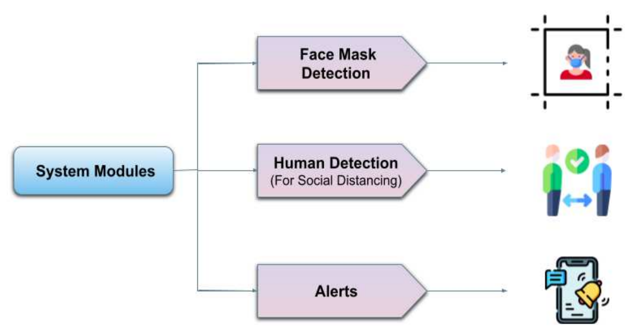

Fig. 1. Flowchart Representing Different System Modules

A. Face Mask Detection

Face mask detection module detects the face mask usage and classifies it into three classes using different shades of bounding box: In the first class, the green bounding box is annotated with a "good" remark which represents that the person is well masked with nose and mouth fully covered. The second class is represented by the orange bounding box is annotated with a "bad" remark which represents that the person is not well masked i.e. his/her nose or mouth is not fully covered. Similarly, the third class is represented by the red bounding box is annotated with a “none” remark which represents that the person is not wearing a mask and is violating the norms.

Fig. 2. Representing different classes of Face Mask Detection Module

This module uses the mask-YOLOv4-tiny model which is a neural network model based on the darknet framework and YOLOv4 architecture [8-9]. This neural network is trained by cansik [8] and the dataset for this pre-trained network (consisting of 678 images of people with and without masks) is the Medical Masks Dataset (MMD), published by Mikolaj Witkowski [10]. This model was trained on an NVIDIA GTX 1080Ti for about 2 hours over 6000 iterations with a batch size of 64 and 16 subdivisions. The weights were trained on an image size of 416x416.

To address the issue of people not facing the camera like people on the left of Fig. 3, we propose that the mask detection cameras will be installed at the entrances and exits of facilities.

Fig. 3. Sample test run of Face Mask Detection module

We tested this module on Jetson Nano and achieved the following performance figures:

TABLE I. FACE MASK DETECTION MODULE PERFORMANCE

Model	**Frames per second (FPS Range)**	
	CPU	**GPU**
mask-YOLOv4-tiny	1.37-1.77	3-6

B. Human Detection

The human detection module detects humans and keeps a human count (for crowd management), calculates the distance between each pair of detected people (for social distancing). This information is then calibrated with the safe distance set by the user and each person is classified into three classes using different colors of bounding box: The person enclosed within a green bounding box denotes that person is at safe distance (i.e. 2m) from others (No Violation). The person enclosed within a yellow bounding box denotes that person is at a minimum safe distance (i.e. 1m) but not at safe distance (i.e. 2m) from others (Abnormal Violation). The person enclosed within a red bounding box denotes that person is not at a minimum safe distance (i.e. 1m) from others (Serious Violation).

Fig. 4. Representing Different Classes of Human Detection Module

This module uses the YOLOv3-608 model which is a neural network model based on the darknet framework and YOLOv3 architecture [11]. This pre-trained neural network was trained on the COCO (Common Objects in Context) dataset and can detect 80 different classes of objects. In our case, it is used to detect humans.

The distance between any two humans has been calculated by utilizing the concept of centroid i.e. calculating distances between the pair of centroids.

Firstly, the Euclidean distance L is calculated between the two persons (in pixels). Lx and Ly are distances in pixels on the x and y axes.

$$L = \sqrt{(Lx)^2 + (Ly)^2} \quad (1)$$

Next, this distance is calibrated into meters by multiplying it with the calibration factor k.

$$k = ((\frac{1}{H1} + \frac{1}{H2})/2) * H \quad (2)$$

Where H1 and H2 are the heights of the two persons (in pixels) respectively and H is the average height of a person (in centimeters), which we have assumed to be 170 cm.

Finally, the calibrated distance D (in cm) is obtained as:

$$D = k * L \quad (3)$$

Using the average height of a person in our equation deals with range estimation issues, e.g., when two persons seem close to each other but actually not (see the right image in Fig. 4.).

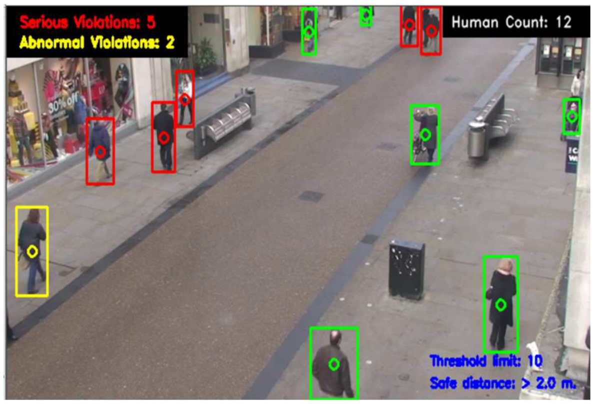

Fig. 5. Sample Test Run of Human Detection Module

We tested this module on the Jetson Nano and achieved the following performance figures as shown in Table II.

TABLE II. HUMAN DETECTION MODULE PERFORMANCE FIGURES

Model	Frames per second (FPS Range) (Jetson Nano)	
	CPU	GPU
YOLOv3-608	0.21	**0.71-0.79**

C. Alerts

The alerts module sends alert messages in real-time to the monitoring user through a Telegram bot, whenever a COVID-19 norms violation (in social distancing or face mask usage) is detected by the system. This module consists of a trigger function containing a personalized alert message for different norms violations cases along with count statistics. This function sends a GET request (containing the alert message) to the Telegram bot's server, which in turn conveys the alert message to the monitoring user.

Fig. 6. Telegram Bot Sending Alert Messages

III. SYSTEM DESIGN

Our system MaViSS-AI is an integration of hardware and software components. The system design comprises a computational unit, software implementation, and methodology of the system.

A. Computational Unit

The major hardware component utilized in our system is Nvidia Jetson Nano for computation. Nvidia Jetson Nano Developer Kit is a small, powerful computer that runs multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing [13]. It is an easy-to-use platform that runs in as little as 5 watts. It is booted by inserting a microSD card with a system image. It is used for building practical AI applications.

The second hardware component used in our project is an IMX 219-77 camera or a webcam for capturing videos. An IMX 219-77 camera is a high-quality camera with an 8-megapixel Sony IMX219 image sensor. It is capable of viewing images at a high resolution of 3280x2464. It has a high fov (field of view) to capture more area. It is suitable to use with the Nvidia Jetson Nano and Nvidia Jetson Xavier NX Development Kits.

The third hardware component used in our project is an external monitor used for visualizing the output of our system and also for monitoring the norms. This external monitor is connected to Jetson Nano using the HDMI cable. Along with the external monitor, a USB mouse and a keyboard is also connected to the Jetson Nano through USB cables.

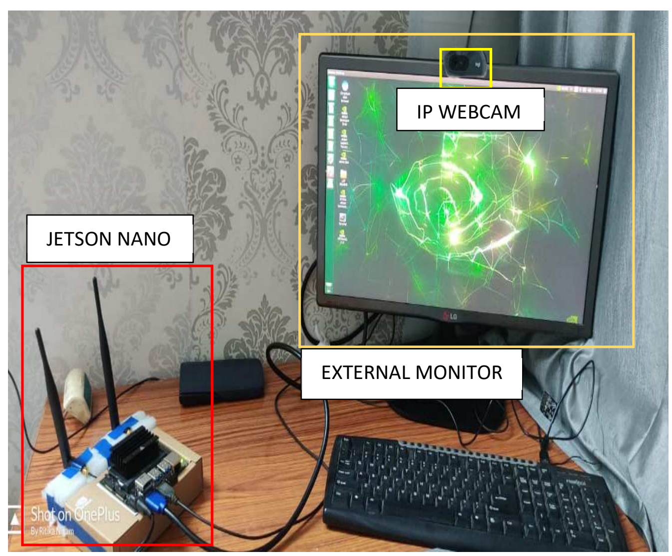

Fig. 7. Hardware components used in MaViSS-AI

B. OpenCV with CUDA Implementation

The system is coded in the Python 3.7 programming language. And the Opencv computer vision library is the main library used in it. Our system uses Opencv 4.5.2 and imutils for the video and image processing tasks. Also, the Opencv dnn (Deep Neural Networks) module is used to facilitate the deep learning inference on the videos/live streams we are processing [12]. This module is compatible with the YOLO (You Only Look Once) architecture that forms our main detection model. The OpenCV 4.5.2 version we're using was compiled with CUDA backend support, to utilize the GPU capabilities of the Nvidia Jetson Nano.

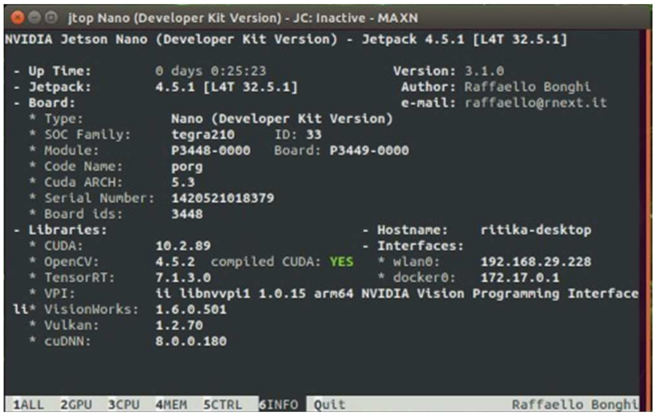

Fig. 8. Compilation of OpenCV with CUDA

SciPy and NumPy libraries were used to facilitate various scientific calculations and calibrations (e.g. distance calculation and calibration) used in our project. The urllib and requests libraries were used in the alerts module of our project for sending GET requests to the Telegram bot's server containing the alert messages.

C. Methodology

Thus, these hardware and software tools integrate to form the system MaViSS-AI. The system workflow has six important phases:

- Video: Video is captured from a source like an IP Camera, CCTV or Webcam. Frames are extracted from this video source.
- Preprocessing: Preprocessing is done on these received frames and they are resized for model inference.
- Model Inference: Model Inference is done by using the YOLO architecture neural network for state-of-the-art, real-time human and face mask detection.
- Calibration: Calibration involves computing parameters like social distancing & face mask metrics, validating them with the norms, and identifying violations.
- Output: Output is generated in real-time to the monitoring user, displaying the social distancing metrics, color-coded bounding boxes for person detection & tracking and face mask usage, and information regarding any violations.
- Alerts: Alerts are sent directly to the monitoring user's smartphone in real-time through a Telegram bot.

IV. RESULTS

By combining all the system modules, MaViSS-AI forms an integrated system that enables the user to monitor social distancing norms and face mask usage in the scene captured by the surveillance camera and any norm breach is reported directly to the user as an alert message.

Fig. 9 shows the monitoring window of our system MaViSS-AI with all the metrics displayed on the screen. The metric at the right-hand corner represents the human counter which counts the total number of humans present in the scene at any particular moment. It also represents the counter for masked, improperly masked, and unmasked parameter which shows the number of humans wearing a mask, not wearing a mask, or is improperly masked. On the other hand, in the left corner, there is a metric for social distancing parameter. It represents the counter for serious and abnormal violations present in the scene. Whenever these counters exceed the threshold value an Alert message is displayed on the monitoring screen and as well as an alert message through the telegram bot is directly sent to the user on their smartphones.

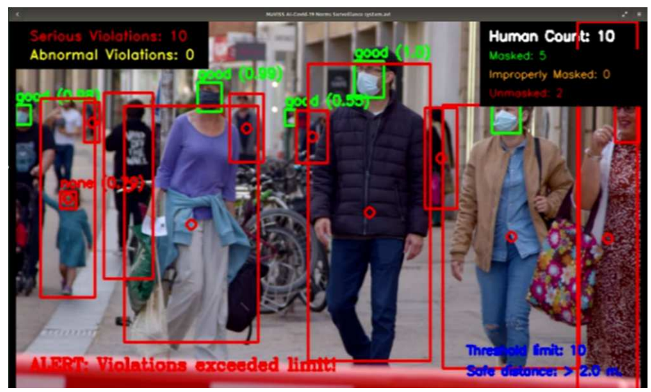

Fig. 9. Monitoring window of system MaViSS-AI

On combining all the modules, the final performance of the system on Jetson Nano is as follows:

TABLE III. MAVISS-AI PERFORMANCE FIGURES

System	**Frames per second (FPS)**	
	CPU	**GPU**
MaViSS AI (YOLOv3-608 + mask-YOLOv4-tiny)	0.15-0.2	**0.65-0.83**

Thus the final inference is that MaViSS-AI (mask-YOLOv4-tiny + YOLOv3-608) utilizes the powerful GPU of Jetson Nano with CUDA backend to improve its performance by approximately 4 times better than that achieved on CPU and runs at a frame rate of 0.65-0.83 FPS.

The analytical chart above shows the comparative analysis of our system MaViSS-AI vs its performance in CPU and GPU of the Jetson Nano. The bar graph indicates that the fps obtained on GPU is much higher than that achieved on the CPU. Thus, with the help of the NVIDIA Maxwell 128 cores GPU of Jetson Nano, we improved the overall performance of our system.

V. CONCLUSION

Therefore, taking into account the importance of social distancing and face mask usage in managing and reducing the probability of COVID-19 disease from continuously spreading which can cause the healthcare system to collapse due to the high number of patients, this paper presents—MaViSS-AI an integrated automated surveillance system for monitoring all the necessary norms that are needed to be followed. It monitors the face mask usage, social distancing parameters and also counts the total number of humans present in the scene. The system raises real-time alerts through a telegram bot whenever any of the following norms are breached. Thus MaViSS-AI surpasses several limitations of the manual monitoring systems and provides an efficient and accurate way of monitoring and reporting breaches in COVID19 norms.

Thus MaViSS-AI is suitable for both indoor and outdoor surveillance scenarios. It can be used by humanitarian and law enforcement professionals to gauge whether people are abiding by public health guidance. It can also be used significantly in various busy places like railway stations, airports, megastores, malls, streets, etc. where manual monitoring is very difficult. Moreover, whenever there are violations in norms, monitoring authorities can be immediately alerted through their smartphones.

The system can also be used for broader applications as a generic human detection and tracking system, not just limited to COVID-19 norms monitoring but also for various real-world applications such as human action and anomaly detection in security systems, pedestrians detection and tracking in autonomous vehicles, crowd management in shops, lifts, public transports, etc.

REFERENCES

[1] Emergencies, Coronavirus pandemic, World Health Organization,[Online],Available:https://www.who.int/emergencies/diseases/novel-coronavirus-2019/global-research-on-novel-coronavirus-2019-ncov, [Accessed: Feb 21, 2022]

[2] Disease, Covid-19 Vaccines, World Health Organizations, [Online], Available:https://www.who.int/emergencies/diseases/novelcoronavirus-2019/covid-19-vaccines/advice, [Accessed: Feb 22, 2022]

[3] M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud and J. -H. Kim, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216386.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[5] Mohd Aquib Ansari , Dushyant Kumar Singh, "Monitoring social distancing through human detection for preventing/reducing COVID spread",Int.j.inf.Tecnol,[Online],Available:https://pubmed.ncbi.nlm.nih.gov/33870073/

[6] P. Shukla, R. Kundu, A. Arivarasi, G. Alagiri and J. shiney, "A Social Distance Monitoring System to ensure Social Distancing in Public Areas," *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2021, pp. 96-101, doi: 10.1109/ICCIKE51210.2021.9410745.

[7] Article, "Smart Social Distancing", Neuralet, Available: https://neuralet.com/article/smart-social-distancing/, [Accessed: Feb 27, 2022]

[8] Mask detection method, Mask YOLO v4 tiny model (for face mask detection), Available: https://github.com/cansik/yolo-mask-detection, [Accessed: Mar 01, 2022]

[9] Object detection method, YOLOv4 architecture, Available: https://arxiv.org/abs/2004.10934, [Accessed: Feb 28, 2022]

[10] Humans in the Loop. Medical Mask Dataset. Available: https://humansintheloop.org/medical-mask-dataset [Accessed Feb 28, 2022]

[11] Object detection method, YOLO v3 model (for human detection), Available: https://pjreddie.com/darknet/yolo/, [Accessed: Feb 17, 2022]

[12] OpenCV with CUDA and dnn modules, Available: https://www.pyimagesearch.com/2020/02/03/how-to-use-opencvs-dnn-module-with-nvidia-gpus-cuda-and-cudnn/

[13] Configuration of Nvidia Jetson Nano, Computational Unit, Available: https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit, [Accessed: Feb 12, 2022]