PAPER NAME

JOURNAL CONFERENCE PAPER DWIPA (turnitin)

WORD COUNT

2237 Words

CHARACTER COUNT

11906 Characters

PAGE COUNT

4 Pages

FILE SIZE

650.7KB

SUBMISSION DATE

Nov 11, 2022 2:40 PM GMT+8

REPORT DATE

Nov 11, 2022 2:41 PM GMT+8

0% Overall Similarity

This submission did not match any of the content we compared it against.

- 0% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 25 words)
- Manually excluded sources

IoT and AI-enabled Physical Distance Monitoring Application to Prevent COVID19 Transmission

Mohammad Dwipa Furqan
Department of Electrical Engineering,
Faculty of Engineering,
Universitas Hasanuddin
Makassar, Indonesia
furqanmd17d@student.unhas.ac.id

Andani Achmad
Department of Electrical Engineering,
Faculty of Engineering,
Universitas Hasanuddin
Makassar, Indonesia
andani@unhas.ac.id

Wardi
Department of Electrical Engineering,
Faculty of Engineering,
Universitas Hasanuddin
Makassar, Indonesia
wardi@unhas.ac.id

Muhammad Niswar
Department of Informatics Engineering,
Faculty of Engineering,
Universitas Hasanuddin
Makassar, Indonesia
niswar@unhas.ac.id

***Abstract*— During COVID19 pandemic, people are encouraged to practice physical distancing at least 1 meter when interacting with other people to prevent the spread of the COVID19. This study aims to develop a system that can monitor the physical distancing and track physical contact in a room using internet of things (IoT) and artificial intelligent technology. The system consists of a small single-board computer (Raspberry Pi), webcam, and web application displaying physical contact information. The system uses YOLO algorithms to detect the human object and euclidean distance formula to determine the distance between human objects. We evaluated the performance of YOLOv3 and YOLOv3-tiny running on Raspberry Pi. The evaluation result shows that YOLOv3 consumes more CPU resources than YOLOv3-tiny but has better accuracy in detecting human objects. YOLOv3-tiny can process images and detect objects faster than YOLOv3.**

Keywords—COVID19, Physical Distancing, Internet of Things, Artificial Intelligence, You Only Look Once (YOLO).

I. Introduction

Covid19 has spread over the world causing many deaths everywhere. Although a vaccine has been developed for this infectious disease, physical distancing is still one of the best methods to prevent the spread of COVID-19. To reduce the transmission of COVID19, the government enforce the people to avoid activities in crowds and physical contact with other people by conducting self-isolation at home, working from home (WFH), and maintaining physical distance at least 1 meter from person to person in the workplace.

Recently, all countries in the world are implementing the WFH method which enforces citizens to work from home. However, some people are required to work in the office as essential worker including healthcare workers, engineers and law enforcements. so they tend to gather with other workers. In such circumstances required a physical distancing protocol in office to prevent the transmission of COVID19. To maintain the physical distancing protocol, we need a monitoring and warning system, so workers can obey the protocol.

To prevent the transmission of Covid19, we built a system that can monitor indoor distancing violations using IoT and AI technology. This system can record and store these violations in a database to be used for tracking when the person who violates it is infected with Covid19 at some point.

II. Related Work

There have been many studies related to physical distance monitoring systems. Reference [1] uses a PIR sensor attached to the arm like a wristband on each individual to monitor each other's distance. The components and sensors contained in the bracelet are two PIR Sensors for the front and back of the bracelet, Arduino Lilypad, Buzzer, push button and others. Reference [2] also proposed a system similar to reference [1] where the wearable prototype use Raspberry Pi camera, range-finder sensor, and accelerometer sensor to maintain the physical distance. Reference [3] starts to fully use cameras in monitoring physical distance, and in this study no longer uses wearables unit device to detect physical distance violations. Some of the components used in this research are raspberry pi and camera. Authors compared the performance of physical distance monitors using several devices including the Raspberry Pi 2B, Raspberry Pi 3, and high-spec laptop. Reference [4] describe implementation issues and describes several approaches to real-time physical distance monitoring solutions. The authors discuss different camera calibration approaches and their pros and cons during implementation.

The references that I mentioned earlier have their own advantages and disadvantages. By observing and utilizing some of these references, we have developed a system that utilizes cameras, and can track physical contact that has occurred using a web interface. We also use a Raspberry Pi 4B to detect human objects and calculate the physical distance between the human objects.

III. System Design

Our system composed of hardware and software design as describe as follow:

A. Hardware Design

The main hardware used in our system consists of the raspberry pi, webcam, wireless router, and web server. The webcam placed in the corner of the room. The webcam captures the image and the image will be processed in Raspberry pi using the physical distancing algorithm to determine the distance between people in a room. The Raspberry pi will send data to the web server and activate the buzzer when there are people who are close to each other (<1m), which is violate the physical distancing protocol.

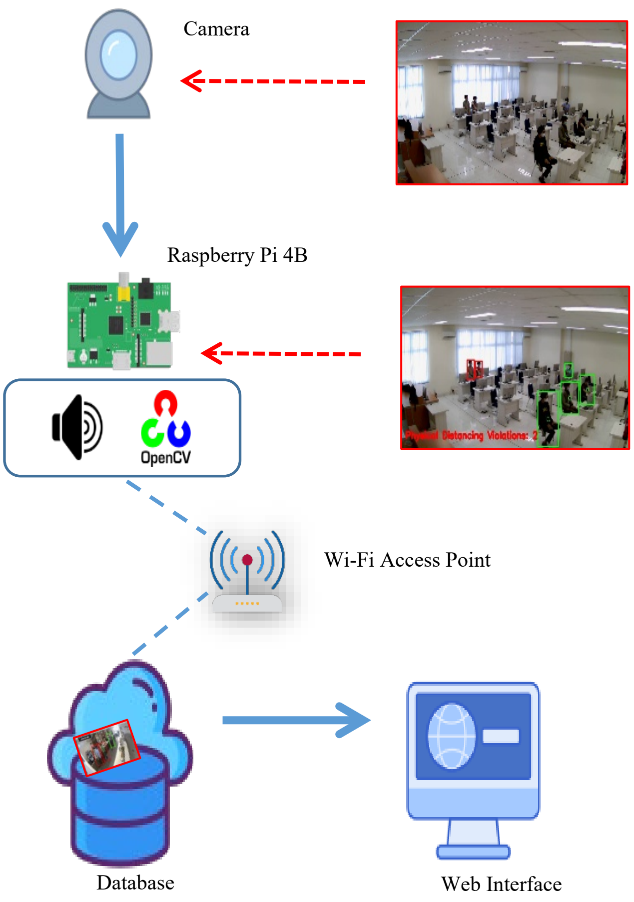

Fig. 1. System Design

B. Software Design

The software used in our system consists of OPENCV python, and PHP-based web application for monitoring and tracing contact. In this study, we used You Look Only Once (YOLO) [5] algorithm to detect objects (classify whether an object is a "human" class). The way YOLO works is by looking at the entire image once, then passing through the neural network to directly detect the object. While the method used to determine the distance between objects (humans) is using the 3D Euclidean Distance formula [6].

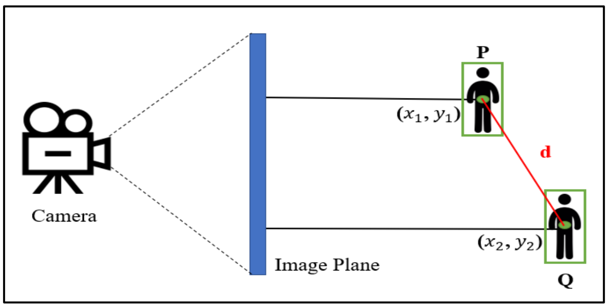

Fig. 2. Euclidean Distance Illustration

$$d(P,Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (1)$$

If a human object has been detected in the image, a bounding box will appear showing the coordinates of the object. The midpoint of the bounding box is called the centroid. We calculate the centroid distance of each bounding box in pixel between person P and person Q as shown in Fig 2. We apply the pythagorean theorem to obtain the distance d(P,Q) in pixels as shown in Eq.1.

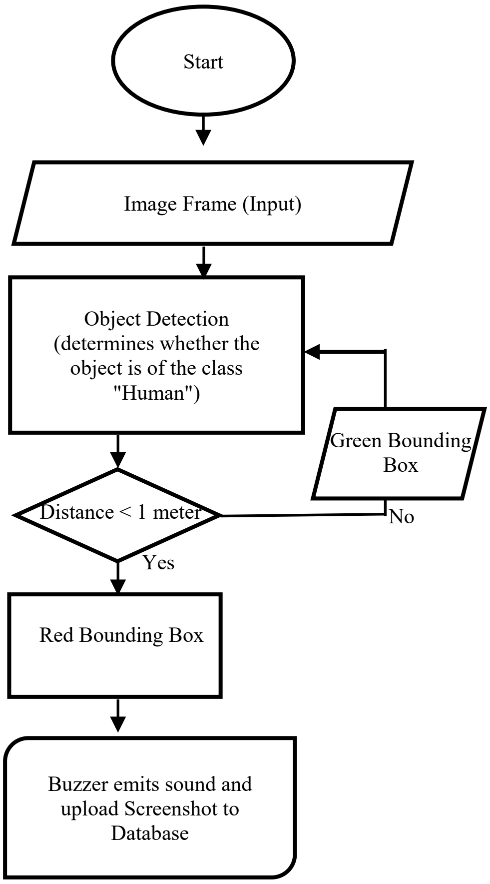

Fig. 3. System Workflow

Physical distancing monitoring pseudocode includes the detection of human objects, calculating distance between human objects, triggering a buzzer, and sending image data to database system. The following is the pseudocode:

Input : Image from Camera
Output : Send the screenshot of distance violation image to database

1. Load our YOLO object detector trained on COCO dataset
2. Connect to the Database System
3. Initialize minimum distance on pixel
4. Grab frame from camera
5. Ensure that at least two people detected in order to compute pairwise distance maps
6. If there are at least two object detected, then extract all centroid from objects (bounding boxes)
7. If there are at least two box detected, then compute the Euclidean distance between all pairs of the centroids
8. If distance between boxes < minimum distance, then capture the screenshot from distance violation frame

9. Send the screenshot of violation frame to the database

The system captured the image and stored in database when violation occurs.

IV. EXPERIMENTS AND ANALYSIS

We conducted experiment to evaluate our system. In this experiment, we recorded the activities of 6 people in a computer room. We have arranged the tables in the room so that there is a distance of 1 meter between one table and another.

Fig. 4. Hardware

A. Measurement of Actual distance and Pixel

We measured the distance (pixel) between two objects in the image by drawing a straight line from the center of the bounding box of one object to another. To determine a distance of 1 meter in an image, we take the image and measure the actual distance of two objects in meters using a tape measure and then compare it to the number of pixels in the image. From these measurements, we obtained that 1 meter is equal to 30 pixels. This value may change depending on the position and view angle of the camera.

Fig. 5. Measurement of actual distance vs number of pixels

As can be seen from Fig.5 that people with green bounding boxes identified that they are keeping a distance of one meter from each other. On the other hand, people with red bounding boxes identified that they are less than one meter apart and violate physical distancing protocol.

B. Web-based Monitoring

We developed a web-based monitoring displaying the image captured by camera when a violation occurs as shown in Fig 6. In addition to image, Raspberry pi send the number of violator and times when violation occurs to web server. The web server stores all data from Raspberry pi in a database and displays it to a web application so that users can monitor and conduct contact tracing when someone in the room is infected with covid19 in the future.

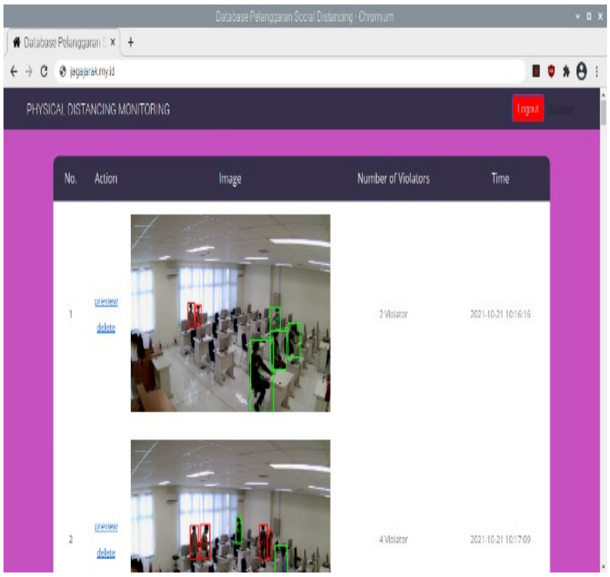

Fig. 6. Web-based Physical Distancing Monitoring

C. Comparison of YOLOv3 and YOLOv3-tiny

We have implemented both YOLOv3 and YOLOv3-tiny using OpenCV python. YOLO algorithm has been trained on the Common Objects in Context (COCO) dataset [8] for classification. We tested both algorithm on a Raspberry Pi 4B that has a Broadcom BCM2711 @1.5GHz CPU and a RAM size of 2GB LPDDR4-3200 SDRAM.

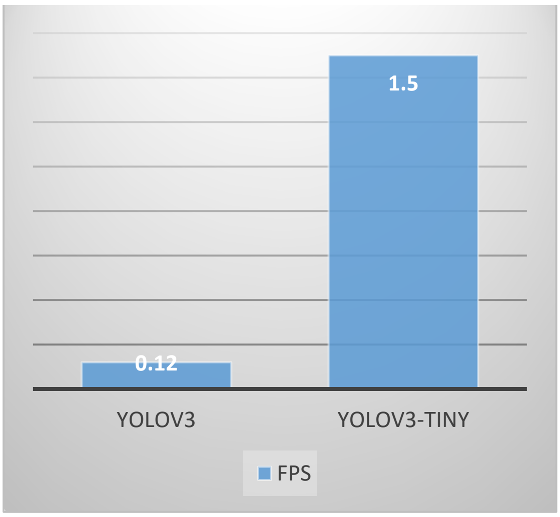

Fig. 7. FPS comparison of YoLov3 and YoLov3-tiny

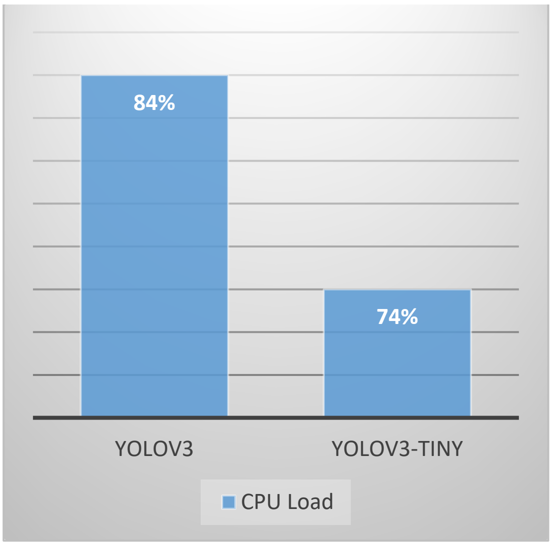

Fig. 8. CPU load comparison of YoLov3 and YoLov3-tiny

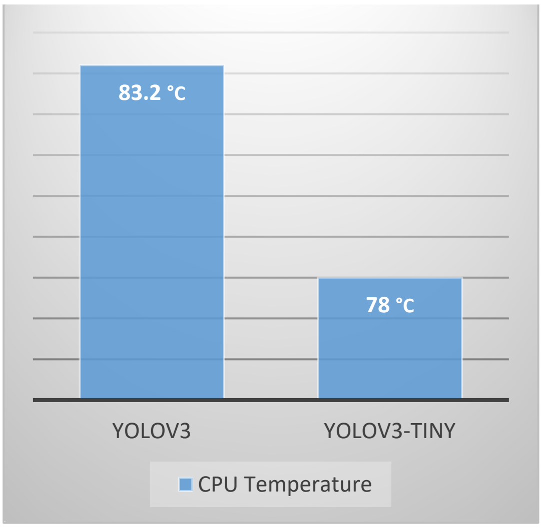

Fig. 9. CPU temperature comparison of YoLov3 and YoLov3-tiny

The frame rate of YOLOv3 is only around 0.12 Frame per Second (FPS) as shown in Fig 7, this indicate that YOLOv3 is slow in detecting objects in Raspberry Pi 4B. The frame rate of YOLOv3-tiny can reach around 1.5 FPS. This indicate that the frame rate of YOLOv3-tiny is better than YOLOv3 in terms of detecting objects in Raspberry Pi 4B.

Another difference between the two algorithms is in terms of CPU load and temperatures. The YOLOv3 has a CPU load value of 84% and a temperature of 83°C when running on a Raspberry Pi 4B. Meanwhile, the YOLOv3-tiny has a CPU load value of 74% and a temperature of 78°C. CPU load on YOLOv3-tiny is 10% less than YOLOv3 as shown in Fig 8. CPU temperature on YOLOv3-tiny is 5°C lower than YOLOv3 as shown in Fig 9. This indicate that YOLOv3 consumes more CPU resources than YOLOv3-tiny in detecting human objects.

We also conduct experiments to evaluate the accuracy of YOLOv3 and YOLOv3-tiny to detect human objects. When using YOLOv3, all human objects are detected properly. On the other hand, when using YOLOv3-tiny sometimes some human objects are not detected, hence, the bounding box does not appear. This may affected by view angle of the camera and the clarity of the object. When the object is clear or close to the camera, then YOLOv3-tiny can detect the object properly. When the object in the image is unclear or small or far away from camera, then the object sometimes cannot be detected by YOLOv3-tiny. It can be concluded that YOLOv3-tiny can process images and detect objects faster than YOLOv3 but less accuracy in detecting human objects.

V. CONCLUSIONS

In this study, we have developed a system that can monitor indoor physical distancing protocol violations using IoT and AI technology. This system aims to prevent Covid19 transmission in indoor environment. The system used YOLO algorithm to detect the human object and euclidean distance formula to determine the distance between objects (humans). The system will send data to the web server and activate the buzzer when there are people who are close to each other (<1m), which is violate the physical distancing protocol. We tested the distance accuracy by comparing the actual distance using a tape measure and the number of pixels in the image. We also conducted the experiment to evaluate the performance of YOLOv3 and YOLOv3-tiny in Raspberry Pi 4B. The experiment result shows that YOLOv3 consumes more CPU resources than YOLOv3-tiny but has better accuracy in detecting human objects. YOLOv3-tiny can process images and detect objects faster than YOLOv3 but less accuracy in Raspberry Pi 4B.

REFERENCES

[1] Kulkarni, M. D., Alfatmi, K., & Deshmukh, N. S. (2021). Social distancing using IOT approach. *Journal of Electrical Systems and Information Technology*, *8*(1). https://doi.org/10.1186/s43067-021-00040-z

[2] Alhmiedat, T., & Aborokbah, M. (2021). Social distance monitoring approach using wearable smart tags. *Electronics*, *10*(19), 2435. https://doi.org/10.3390/electronics10192435

[3] Salih, Husam & F. Jwaid, Mohanad & Al, & Mohammed, Alaa & Altemimi,. (2021). Designing the IoT based Social Distancing Monitoring System for Reducing the impact of Covid-19. 20. 4169-4177. 10.17051/ilkonline.2021.05.458.

[4] S. Das et al., "Computer Vision-based Social Distancing Surveillance with Automated Camera Calibration for Large-scale Deployment," 2021 IEEE 18th India Council International Conference (INDICON), 2021, pp. 1-6, doi: 10.1109/INDICON52576.2021.9691485.

[5] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection" University of Washington, Allen Institute for AI, Facebook AI Research. https://pjreddie.com/yolo

[6] Gupta, S., Kapil, R., Kanahasabai, G., Joshi, S. S., & Joshi, A. S. (2020). SD-Measure: A social distancing detector. *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*. https://doi.org/10.1109/cicn49253.2020.9242628

[7] Joseph Redmon, Ali Farhadi,."YOLOv3: An Incremental Improvement", Universiity of Washington. https://pjreddie.com/media/files/papers/YOLOv3.pdf

[8] Microsoft COCO: Common objects in context, in ECCV, 2014. [Online]. Available : https://cocodataset.org

0% Overall Similarity

NO MATCHES FOUND

This submission did not match any of the content we compared it against.

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Manually excluded sources
- Quoted material
- Small Matches (Less then 25 words)

EXCLUDED SOURCES

Mohammad Dwipa Furqan, Andani Achmad, Wardi, Muhammad Niswar. "IoT ... Crossref	84%