

Real-time multiple face mask and fever detection using YOLOv3 and TensorFlow lite platforms

Ali A. Abed¹, Alaa Al-Ibadi¹, Issa Ahmed Abed²

¹Department of Computer Engineering, University of Basrah, Basrah, Iraq

²Department of Engineering Technical College Basrah, Southern Technical University, Basrah, Iraq

Article Info

Article history:

Received Jun 10, 2022

Revised Aug 13, 2022

Accepted Oct 6, 2022

Keywords:

COVID-19

Deep learning

Google Colab

Infra-red sensor

Raspberry Pi

TensorFlow Lite

You only look once version 3

ABSTRACT

COVID-19 has caused disruptions to many aspects of everyday life. To reduce the impact of this pandemic, its spreading must be controlled via face mask wearing. Manually mask-checking for everybody is embarrassing and uncontrollable. Hence, the proposed technique is used to help for automatic mask-checking based on deep learning platforms with real-time surveillance live infra-red (IR) camera. In this paper, two recent object detection platforms, named, you only look once version 3 (YOLOv3) and TensorFlow lite are adopted to accomplish this task. The two models are trained with a dataset consisting of images of persons with/without masks. This work is simulated with Google Colab then tested in real-time on an embedded device mated with fast GPU called Raspberry Pi 4 model B, 8 GB RAM. A comparison is made between the two models to verify their performance in relation to their precision rate and processing time. The work of this paper is also succeeded to realize multiple face masks real-time detection up to 10 facemasks in a single scene with high inference speed. Temperature is also measured using IR touchless sensor for each person with sound alarming to alert fever. The presented detector is cheap, light, small, and fast, with 99% accuracy rate during training and testing.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ali A. Abed

Department of Computer Engineering, University of Basrah

Basrah, Iraq

Email: ali.abed@uobasrah.edu.iq

1. INTRODUCTION

The dangerous COVID-19 is a global pandemic strongly deployed all over the world, and about 533.4 million cases along with about 6.3 million deaths are recorded globally till June 8, 2022 [1]. Individuals infected with COVID-19 suffer from flu, fever, and some other symptoms [2]. The few physicians and specialists and the lack of immunity against COVID-19 leads to the susceptibility of the community. According to World Health Organization (WHO), mask wearing is the primary possible way to protect people from infection with this pandemic. Therefore, the whole community is forced to look to this protection measure, beside to the social distance to stop the virus spreading. Even though vaccines are right now available, but unfortunately it does not protect the vaccinated person from infection 100% [3]. Hence, until this virus is totally disappeared, wearing face masks permanently should be considered to assist preventing the spread of infection and keep humans safer. Face masks may be considered as an effective way for infection avoidance. Since COVID-19 is a new disease, face mask detection is accordingly a recent subject that has not covered considerably by researchers worldwide.

This paper contributes the following: i) an approach for mask detection depending on TensorFlow Lite based on TensorFlow embedded on an edge device leading to a real-time single-object, tiny, cheap, low-

power consuming, and high-efficiency artificial intelligent (AI) model; ii) another method for face mask detection using you only look once version 3 (YOLOv3) model in docker container operating environment embedded on the same tiny, cheap, low-power consuming, and high-efficiency AI edge device but with a real-time multiple-objects facility; iii) a temperature measuring with alert module embedded inside the two proposed deep learning models; and iv) a custom dataset is made for training of the two machine learning platforms.

The main difficulties appeared in face mask detection will be surveyed. Transfer learning, which means a model built for some tasks is reused after adaption as the initial point for a model on a related purpose [4], is used in this work. It is a famous strategy in deep learning where pre-trained models are adopted as the initial point in vision tasks. This leads to the large time and computational resources needed to build neural network models from the basis of these problems and the large transferring in professionalism that they provide on related issues.

YOLOv3 and TensorFlow are the platforms adopted for making new developed models and then their results are compared to verify the best. You only look once (YOLO) is an convolutional neural network (CNN) developed originally for real-time object detection [5]. The algorithm uses one neural network to the input image, and then splits the image into partitions and produces probabilities and bounding boxes for each partition [6]. Training of these models is so hard, considering variation in camera angles and mask kinds, leading to a big challenge with this issue. Another challenge is the lack of a large dataset for training this type of detection systems, hence a custom dataset is made, and a transfer learning technique is applied to achieve this task. The proposed work of this paper provides contributions to the field of AI by building platforms that use small size, cheap, low-power, and high-efficiency AI embedded devices like Raspberry Pi 4 model B, with 8 GB memory.

Several past works are presented for face mask detection and the related subjects based on CNNs and deep learning platforms. Sethi *et al.* [7] integrated three commonly used machine learning models, ResNet50, Alexnet, and MobileNet to obtain a model with accuracy of 98.2% and minimized inference time. Jagadeeswari and Theja [8] proposed a model in which individuals who do not wear masks are pointed by using learning approaches. An alarm is triggered if the model distinguishes a non mask person. Das *et al.* [9] presented a simple method that uses some libraries like: Keras, TensorFlow, Scikit-Learn, and OpenCV. The suggested method observed the face in the picture and indicated the presence of a mask. Using two different datasets, the method satisfied an accuracy of 95.77% and 94.58%, respectively. Rao *et al.* [10] showed a facial recognition system capable of identifying an individual with a mask by indicating individuals without masks. They used two convolutional layers having 100 filters each, neglecting 1/5%, and activated the internal layers using ReLu and Softmax activation functions. The model is optimized with "Adam" and a cascade classifier was adopted with 1,500 images leading to an accuracy of 91.21%. Suresh *et al.* [11] presented a thorough facemask detection and notification system trained with Kaggle datasets. The person was notified via a text message if not wearing a mask. Sen and Sawant [12] presented a mask detection system able to detect masks of different shapes in a recorded video. Mohandas *et al.* [13] presented a face mask detection system that modified to enter and exit control system. The designed real-time system satisfied an accuracy of 89% for face mask detection and inference time of less than 3 ms. Nguyen *et al.* [14] suggested a face mask wearing alert system depended on a simple CNN to adapt with a low-computation devices. The system worked in two phases: face detection and facemask recognition. The proposed networks are trained and evaluated on benchmark datasets. The system worked in real-time 26.18 frames per second (FPS) on a NVIDIA Maxwell GPU.

The sections of the paper are presented as follows. Section 2 contains the research method. In this section, YOLOv3 and TensorFlow are detailed for the sake of achievement of the required detection models. Section 3 monitors the experimental results and a comparison between the two models is given. In section 4, the main conclusions have been discussed and some future works are suggested.

2. METHOD

In this paper, the real-time detector for mask and fever measurement based on the trained CNN models is presented. Live infra-red (IR) camera and touchless IR temperature sensor are included to that proposed detector. Two main platforms are proposed for the desired task and will be explained in the subsequent sections.

2.1. TensorFlow and TensorFlow Lite

TensorFlow is an open-source set of tools used to build, evaluate, and train machine learning models [15]. It is a well-known framework adopted in machine learning that can be interacted via its Python library. In this paper, TensorFlow Lite, a package of tools for spreading TensorFlow models to mobile and embedded hardware kits, is used to invoke the proposed model on-device. This light-weight version of TensorFlow is a

powerful and industrial tool that serves any deep learning models on mobile phones or microcontroller development boards. It has the following two major parts:

- TensorFlow lite converter: TensorFlow models are converted to a special, small-size format suitable for using on limited-memory embedded devices and could applying optimizations to satisfy more reduction in model size to realize fast real-time applications.
- TensorFlow lite interpreter: This invokes an appropriately converted TensorFlow lite model using the high-efficiency operations of a suitable kit.

TensorFlow lite converter's Python application programming interface (API) is used to obtain the proposed model of this paper. TensorFlow lite converter is also applied for optimizations to the model to reduce both the model size and consequently the time it takes to run but unfortunately it led to a little bit reduction in accuracy. Another drawback for this model is it used for single object detection only. To overcome these two drawbacks, YOLOv3 is the more suitable choice.

2.2. YOLOv3

YOLOv3 [16] object detection platform proposed in this paper is based on YOLOv1 detection networks. Some modifications to the loss function are performed leading to a more robust feature extractor network resulting in a multi-object detection algorithm. Therefore, this platform can now identify a greater variety of objects, ranging in size from large to tiny and in number from 1 to 10. Additionally, YOLOv3 is fast and enables short real-time inference with high FPS on GPU edge devices. Therefore, image categorization network became more advanced as compared to simple deep stacks of layers of the previous versions of YOLO. It included skip connections to aid activation in propagating through deep layers without reducing the gradient. Hence, the feature extractor of this work has successfully been expanded from 19 levels (in YOLOv1) to 53 levels as shown in Figure 1.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 1. Feature extractor of YOLOv3

2.3. Dataset

The first step in obtaining a face-mask detection system is the image collection. Images with/without masks are involved in the dataset [17], [18]. Images of 4,095 masked and unmasked individuals are used to create this dataset each of which is labelled, tagged, and pre-processed before training and testing. Pre-processing, done with MobileNet model, includes image scaling-down, array transformation, and labels' hot encoding. All images are resized to 224×224 pixels and then transformed into arrays utilizing loop function. Labeled data is obtained through hot encoding because the learning algorithms are unable to deal with labeled data directly. Training/testing is performed with 75\25% of the total data respectively.

2.4. Model development

Training of the model with training image generator, Darknet-53, model parameter addition, and compilation are done on Google Colab Tesla processors instead of our local computer to decrease training

time. Model storage for future testing and validation on Raspberry Pi 4 model B, 8 GB RAM fast GPU is performed. For mask detection from image/video streams, the proposed model is built with Python PyTorch package to conduct several tests. Mathematical for evaluating and measuring metrics of the proposed model are presented in the following (1) to (4) [19]:

$$Accuracy = \frac{[TP+TN]}{[TP+FP+TN+FN]} \quad (1)$$

$$Precision = \frac{[TP]}{[TP+FP]} \quad (2)$$

$$Recall = \frac{[TP]}{[TP+FN]} \quad (3)$$

$$F1 - Score = 2 \times \frac{[Precision \times Recall]}{[Precision + Recall]} \quad (4)$$

where FN is false negative, TP is true positive, FP is false positive, and TN is true negative [20]. TP values point to images that have been labelled as true and gave a true result as guessed by the model. Likely, TN images are those that have been classified as true but produced an incorrect output because of prediction. FP images have been classified as false so far yielded false positives because of prediction. FN images are classified as false so far turn out to be precise, producing false negatives. Precision is a measure for the number of predicted positive values. The recall reflects the ability of the classifier to indicate all positive cases, while the F1-score produces test accuracy. These performance metrics are considered since they realize the most precise measurements. Testing of the model is separated into steps to prove its accurate detections.

2.5. Hardware components

2.5.1. Raspberry Pi 4 model B and secure digital card

A new Raspberry Pi kit, recently introduced, is used to invoke recent AI models at small scalability, power consuming, speed, and cost [21]. This new version is Raspberry Pi 4 model B, with 8 GB memory shown in Figure 2. General purpose input output (GPIO) pins, camera serial interface (CSI) port, and two micro-HDMI terminals are installed in this microcomputer board which is powered by type C mini-USB. This led to various new detectors with various AI workloads to be realized. It works with 5V, which means it is low-power, energy efficient embedded device. Secure digital (SD) is a 128 GB memory card used for operating system download and for reading/writing large quantities of data.

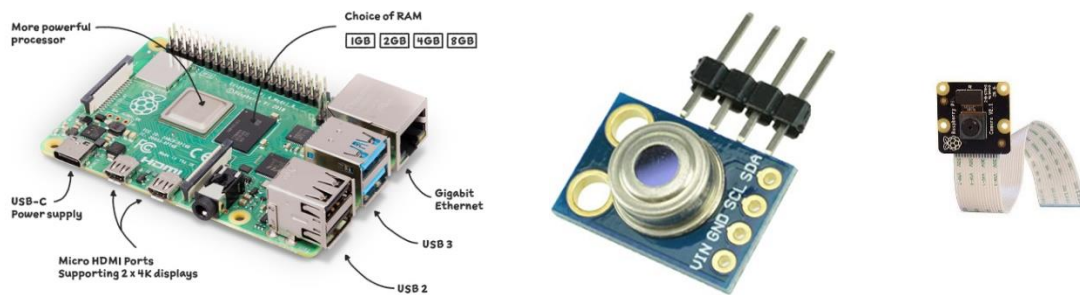


Figure 2. Various hardware components

2.5.2. MLX90614 Infrared thermopile sensor and IR camera Pi v2

MLX90614 sensor, manufactured by Melexis and shown in Figure 2, works on the principle of InfraRed thermopile sensor for temperature measurement and typically suited for contactless temperature measurement applications [22]. The sensor consists of two units embedded internally to give the temperature output: The sensing unit which has an infrared detector, followed by data computational unit. The sensor converts the computational analogue value into 17-bit digital value using analogue-to-digital converter (ADC) that can be accessed using I²C communication protocol. It measures an object (body) temperature in the range (-70 °C to 380 °C) with measurement resolution of 0.02 °C. After downloading the library and packages required to successfully interface the sensor to the Raspberry Pi, it is calibrated with respect to standard temperature measuring device and then tested successfully. The sensor is then integrated with a buzzer that rings when the temperature exceeds a threshold. For high-definition image/video capturing, camera Pi module

2 shown in Figure 2 [23] is used and interfaced to the Raspberry Pi board using the CSI port. It is an IR camera that can take images along with live videos and can be fully controlled programmatically.

2.6. The final setup

After creating and loading data into the dataset, the proposed classifier is trained based on TensorFlow and then YOLOv3. Google Colab [24] which has high-speed CPU and GPU is used to train and test the proposed model, resulting in an accuracy of 99% and 100% during training and testing respectively. The workflow of the training/testing phase is summarized in Figure 3(a). Afterward, the proposed development kit with its accessories is used to implement the trained model and starting real-time image/video capturing and mask/temperature detection. The face mask classification model will recover faces from image/video streams as needed. Real public faces are placed into the hidden CNN as an input to create the mask. The output of the CNN is a “mask” or “no mask” decision. In addition, the IR sensor is adopted to measure the body temperature and a “beep” ringtone will be heard if temperature exceeds a threshold of 37 °C. The workflow of the real-time validation phase is summarized in Figure 3(b). Another benefit of the developed system is its capability of displaying multiple persons (more than 10) in a single scene. Therefore, the system can be used easily in any crowded zone to discriminate “no mask” wearers.

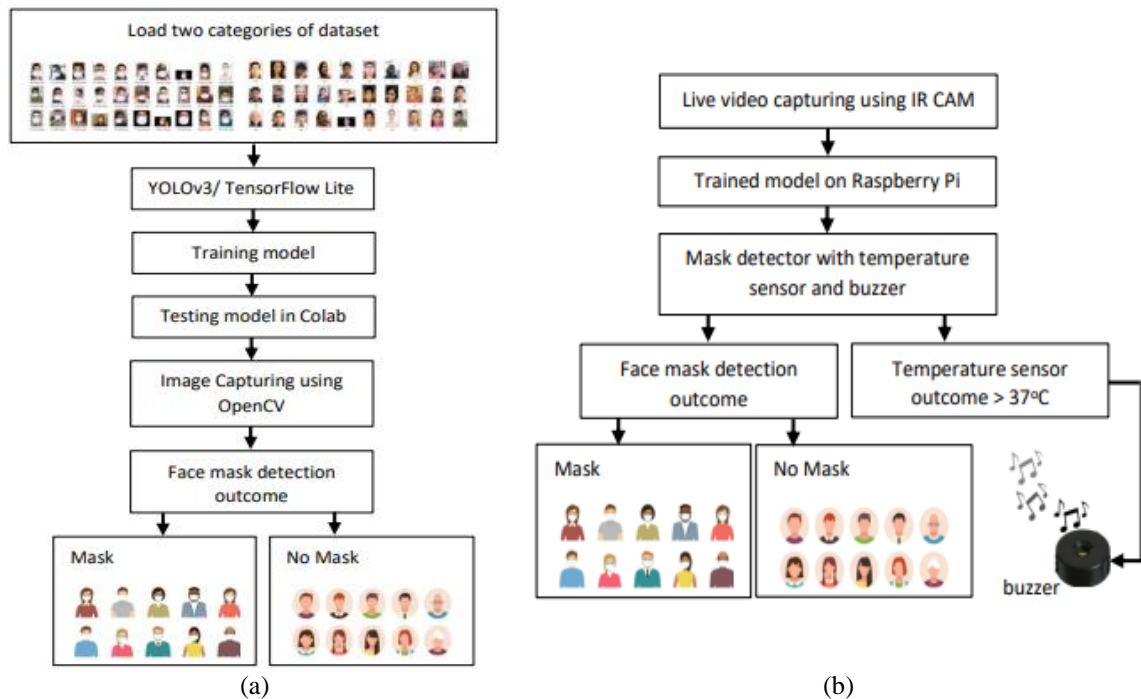


Figure 3. Workflow of the suggested system (a) training/testing model and (b) implementation model

3. EXPERIMENTAL RESULTS

The experimental results are divided into three phases: training, testing, and validation as explained in the following subsequent sections. The training is done with Google Colab to avoid any inconvenience with CPU and GPU specifications. The validation is done on Raspberry Pi embedded device selected for this purpose.

3.1. Training of the proposed model

The accuracy and loss of the proposed model are evaluated 20 times during training on Google Colab, but 10 values (for short) are shown in Table 1. This table is quite enough to prove that accuracy increases while loss decreases until reaching steady state. Table 2 displays the evaluation results of the proposed model during second phase. Macro average (MA) is used to determine F1 [25] to every label and gives the mean without considering the label's fraction in the dataset. The weighted average (WA) function considers the label's fraction in the dataset and determines F1 to every label. It is invoked with Google Colab to obtain the simulation result shown in Figure 4.

Table 1. Loss/accuracy vs. epochs

Epoch	Train_loss	Val_loss	Train_accuracy	Val_accuracy
1	0.35	0.15	0.88	0.99
3	0.11	0.08	0.97	0.99
5	0.08	0.07	0.97	0.99
7	0.07	0.05	0.97	0.99
9	0.05	0.05	0.98	0.99
11	0.05	0.05	0.98	1.0
13	0.05	0.04	0.99	1.0
15	0.04	0.04	0.99	1.0
17	0.04	0.04	0.99	1.0
19	0.03	0.03	0.99	1.0

Table 2. The proposed model evaluation

	Recall (%)	F1-score (%)	Precision (%)
No Mask	98	99	100
Mask	100	99	99
MA	99	99	99
WA	99	99	99
Accuracy		99	

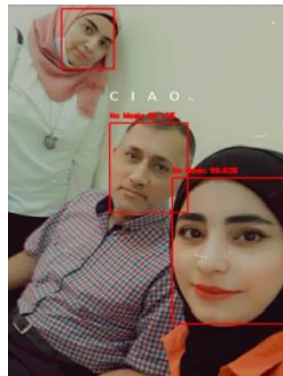


Figure 4. Predicting on Google Colab

3.2. Model implementation

After training/testing process, the model is implemented and validated on the proposed embedded device along with its IR sensor/buzzer and camera Pi. The overall hardware setup illustrating the relationship among the various hardware components is shown in Figure 5. The real-time validation results of the proposed surveillance system for multiple objects (persons) (using YOLOv3) and single object (person) (using TensorFlow Lite) are shown in Figures 6 and 7 respectively. For the sake of easy interaction with the designed system, a graphical user interface (GUI) is designed to provide a user-friendly environment for human machine interface (HMI) as shown in Figure 8. There are two buttons: multiple face mask detector operating with YOLOv3; and face mask and fever detector operating with TensorFlow lite platform.



Figure 5. The final setup

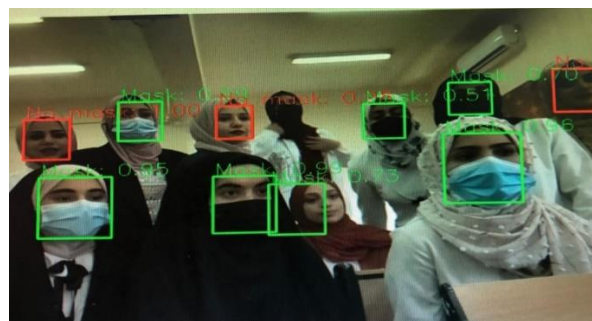


Figure 6. Validation the system for multiple persons using YOLOv3



Figure 7. Validation the system for single person



Figure 8. GUI for interaction with the system using TensorFlow lite

4. CONCLUSION

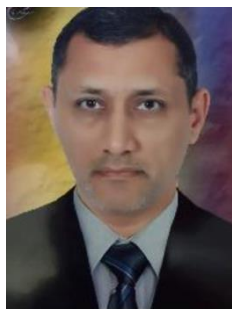
COVID-19 disease could no longer affects our everyday life if the RPi 4 model B, of 8 GB memory with YOLOv3 lightweight model is officially adopted. Consequently, this mask detection platform might be used in crowded zones. When the system is placed in any zone, it can be configured to capture either a live video stream or a pre-recorded video. These types of real-time detectors are beneficial in surveillance applications to detect and recognize masked faces and human temperature and produces sound notification when temperature exceeds a prespecified value. The obtained validation results proved 99.0% accuracy for training and 100% accuracy for testing. The processing time for YOLOv3 is about 10 FPS as compared to 4 FPS for TensorFlow Lite. Another benefit for YOLOv3 over TensorFlow Lite is the number of persons is 10 as compared to single person for TensorFlow lite. Therefore, TensorFlow Lite is well suited for low-cost applications with single individual and limited processing speed. For low-cost applications that need bulk monitoring with high speed, YOLOv3 is the choice. This work can be more developed if the whole software package is transferred to a mobile app the matter that make the application to be worldwide public app.




REFERENCES

- [1] World Health Organization, "WHO coronavirus (COVID-19) dashboard," *World Health Organization*, 2020. [Online], Available: <https://covid19.who.int/table> (2022, accessed Jun. 10).
- [2] World Health Organization, "Report of the WHO-China joint mission on coronavirus disease 2019 (COVID-19)," 2020. [Online]. Available: <https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report.pdf>
- [3] World Health Organization, "Statement for healthcare professionals: How COVID-19 vaccines are regulated for safety and effectiveness," *World Health Organization*, 2021. [Online], Available: <https://www.who.int/news/item/11-06-2021-statement-for-healthcare-professionals-how-covid-19-vaccines-are-regulated-for-safety-and-effectiveness> (2022, accessed Jun. 10).
- [4] A. H. Örnek, M. Çelik and M. Ceylan, "Mask Detection From Face Images Using Deep Learning and Transfer Learning," *2021 15th Turkish National Software Engineering Symposium (UYMS)*, 2021, pp. 1-4, doi: 10.1109/UYMS54260.2021.9659582.
- [5] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [7] S. Sethi, M. Kathuria, and T. Kaushik, "Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread," *Journal of Biomedical Informatics*, vol. 120, pp. 1–12, Aug. 2021, doi: 10.1016/j.jbi.2021.103848.
- [8] C. Jagadeeswari and M. U. Theja, "Performance evaluation of intelligent face mask detection system with various deep learning classifiers," *International Journal of Advanced Science and Technology*, vol. 29, no. 11, pp. 3074–3082, 2020.
- [9] A. Das, M. W. Ansari, and R. Basak, "COVID-19 face mask detection using TensorFlow, Keras and OpenCV," in *2020 IEEE 17th India Council International Conference (INDICON)*, Dec. 2020, pp. 1–5, doi: 10.1109/INDICON49873.2020.9342585.
- [10] T. S. Rao, S. A. Devi, P. Dileep, and M. S. Ram, "A novel approach to detect face mask to control covid using deep learning," *European Journal of Molecular and Clinical Medicine*, vol. 7, no. 6, pp. 658–668, 2020.
- [11] K. Suresh, M. B. Palangappa, and S. Bhuvan, "Face mask detection by using optimistic convolutional neural network," in *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, Jan. 2021, pp. 1084–1089, doi: 10.1109/ICICT50816.2021.9358653.
- [12] S. Sen and K. Sawant, "Face mask detection for COVID-19 pandemic using pytorch in deep learning," in *IOP Conference Series: Materials Science and Engineering*, Feb. 2021, pp. 1–10, doi: 10.1088/1757-899X/1070/1/012061.
- [13] R. Mohandas, M. Bhattacharya, M. Penica, K. van Camp, and M. J. Hayes, "On the use of deep learning enabled face mask detection for access/egress control using TensorFlow Lite based edge deployment on a Raspberry Pi," in *2021 32nd Irish Signals*




- and Systems Conference (ISSC), Jun. 2021, pp. 1–6, doi: 10.1109/ISSC52156.2021.9467841.
- [14] D.-L. Nguyen, M. D. Putro, and K.-H. Jo, "Facemask wearing alert system based on simple architecture with low-computing devices," *IEEE Access*, vol. 10, pp. 29972–29981, 2022, doi: 10.1109/ACCESS.2022.3158304.
- [15] I. S. B. Md Isa, C. J. Yeong, and N. L. A. bin Md S. Azyze, "Real-time traffic sign detection and recognition using Raspberry Pi," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 331–338, Feb. 2022, doi: 10.11591/ijece.v12i1.pp331-338.
- [16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [17] D. Matthias, C. Managwu, and O. Olumide, "Face mask detection application and dataset," *Journal of Computer Science and Its Application*, vol. 27, no. 2, Mar. 2021, doi: 10.4314/jcsia.v27i2.5.
- [18] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health*, vol. 19, pp. 1–6, Mar. 2021, doi: 10.1016/j.smhl.2020.100144.
- [19] S. Saypadith and S. Aramvith, "Real-time multiple face recognition using deep learning on embedded GPU system," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov. 2018, pp. 1318–1324, doi: 10.23919/APSIPA.2018.8659751.
- [20] A. A. Mamun, P. P. Em, and J. Hossen, "Lane marking detection using simple encode decode deep learning technique: SegNet," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 3032–3039, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3032-3039.
- [21] F. Mendoza-Cardenas, R. S. Leon-Aguilar, and J. L. Quiroz-Arroyo, "CP-ABE encryption over MQTT for an IoT system with Raspberry Pi," in *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2022, pp. 236–239, doi: 10.1109/CISS53076.2022.9751194.
- [22] Melexis, "MLX90614 family datasheet," *Melexis INSPIRED ENGINEERING*, 2019. [Online]. Available: <https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90614> (2022, accessed Jun. 09).
- [23] A. M., S. K., S. K. R., and Y. I., "Contactless temperature detection of multiple people and detection of possible corona virus affected persons using AI enabled IR sensor camera," in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2021, pp. 166–170, doi: 10.1109/WiSPNET51692.2021.9419439.
- [24] Tutorials Point Simply Easy Learning, *Colab tutorial point: Simple easy learning*. Hyderabad: Tutorials Point (I) Pvt. Ltd., 2019.
- [25] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F1 and macro-averaged F1 scores," *Applied Intelligence*, vol. 52, no. 5, pp. 4961–4972, Mar. 2022, doi: 10.1007/s10489-021-02635-5.

BIOGRAPHIES OF AUTHORS






Ali A. Abed    Assistant Professor at the University of Basrah, Department of Computer Engineering in Iraq. Received the B.Sc. & M.Sc. degrees in Electrical Engineering in 1996 & 1998 respectively. He received a Ph.D. in Computer & Control Engineering in 2012. His field of interest is Robotics, Computer Vision, and IoT. He is IEEE senior member, IEEE member in Robotics & Automation Society, IEEE member in IoT Community, member ACM. Currently supervising a research group developing deep learning platforms. He can be contacted at email: ali.abed@uobasrah.edu.iq.



Alaa Al-Ibadi    Lecturer at the University of Basrah/Iraq with a BSc. degree in electrical engineering in 2001, M.Sc. in Computer and Control Systems in 2005, and a PhD in multiple robot systems (soft robotics)-University of Salford 2019. He worked as assistant Lecturer at the University of Basrah, Department of Computer Engineering. His research interests: biologically inspired systems, soft robotics, pneumatic muscle actuators, multi-robot systems and intelligent control. He can be contacted at email: alaa.abdulhassan@uobasrah.edu.iq.



Issa Ahmed Abed    received BSc. and MSc. in Control and Computing from the University of Basrah, Iraq in 2002 and 2005, respectively. His Ph.D. in Control and Computing is received from University Tenaga Nasional, Malaysia in 2014. Currently, he is an Asst. Prof. and the head of control and automation department at Engineering Technical College Basrah, Southern Technical University, Iraq. Areas of interest: artificial intelligence, solar energy, control systems, robotics, and image processing. He can be contacted at email: issaahmedabed@stu.edu.iq.