International Journal of Advances in Applied Sciences (IJAAS)
Vol. 13, No. 4, December 2024, pp. 991~1003
ISSN: 2252-8814, DOI: 10.11591/ijaas.v13.i4.pp991-1003

Real-time smoke and fire detection using you only look once v8-based advanced computer vision and deep learning

Shakila Rahman[1], Syed Muhammad Hasnat Jamee[1], Jakaria Khan Rafi[1], Jafrin Sultana Juthi[1], Sajib Abdul Aziz[1], Jia Uddin[2]
[1]Department of Computer Science and Engineering, Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
[2]Department of Artificial Intelligence and Big Data, College of Endicott, Woosong University, Daejeon, Republic of Korea

Article Info

Article history:

Received Feb 26, 2024
Revised Aug 29, 2024
Accepted Sep 4, 2024

Keywords:

Computer vision
Fire detection
Image processing
Object detection model
Smoke detection
You only look once v8

ABSTRACT

Fire and smoke pose severe threats, causing damage to property and the environment and endangering lives. Traditional fire detection methods struggle with accuracy and speed, hindering real-time detection. Thus, this study introduces an improved fire and smoke detection approach utilizing the you only look once (YOLO)v8-based deep learning model. This work aims to enhance accuracy and speed, which are crucial for early fire detection. The methodology involves preprocessing a large dataset containing 5,700 images depicting fire and smoke scenarios. YOLOv8 has been trained and validated, outperforming some baseline models- YOLOv7, YOLOv5, ResNet-32, and MobileNet-v2 in the precision, recall, and mean average precision (mAP) metrics. The proposed method achieves 68.3% precision, 54.6% recall, 60.7% F1 score, and 57.3% mAP. Integrating YOLOv8 in fire and smoke detection systems can significantly improve response times, enhance the ability to mitigate fire outbreaks, and potentially save lives and property. This research advances fire detection systems and establishes a precedent for applying deep learning techniques to critical safety applications, pushing the boundaries of innovation in public safety.

Corresponding Author:

Jia Uddin
Department of Artificial Intelligence and Big Data, College of Endicott, Woosong University
Daejeon 34606, Republic of Korea
Email: jia.uddin@wsu.ac.kr

1. INTRODUCTION

The main smoke detection plays a vital role in mitigating the risks associated with fires, industrial accidents, and natural disasters. Traditional methods, which predominantly utilize heat or optical sensors, often suffer from delayed responses and limitations in accuracy, particularly in challenging environments where factors such as steam or airborne particles can lead to false alarms or missed detections. The advent of computer vision and machine learning (ML) technologies has paved the way for more sophisticated smoke detection systems. Among these, you only look once (YOLO)V8 [1] model stands out as a state-of-the-art solution for real-time detection. This model, part of the YOLO series, operates on a single-stage detection principle, offering substantial improvements in speed and accuracy over its predecessors and two-stage counterparts like faster region-based convolutional neural network (R-CNN) [2], [3], Mask R-CNN [4], and other segmentation-based techniques [5]–[7]. Notably, YOLOv8's deep architecture ensures robust performance across diverse conditions, including poor lighting and complex backgrounds. The evolution of YOLO models from versions 4 through 7 has been marked by significant enhancements in detection capabilities, processing speed, and user accessibility. Starting with YOLOv4 [8], which was notable for its ability to identify a wide array of objects across a vast dataset, each subsequent version has introduced improvements in speed, accuracy, and functionality, culminating in YOLOv7's unparalleled efficiency [9] and YOLOv8's advanced features [10], like multi-object tracking and optimized performance for challenging detections. Research literature underscores the effectiveness of YOLO models in smoke detection. For instance, studies [11] and [12] demonstrate the application of YOLOv5 and YOLOv6, respectively, in detecting smoke with high accuracy, leveraging temporal information, and enhancing spatial features to minimize false positives.

This work presents a comprehensive evaluation of a smoke detection system based on the YOLOv8 model. By training the model on a custom dataset comprising smoke and non-smoke images, and employing metrics such as precision, recall, and accuracy, the study aims to offer an advanced solution for fire safety. The system's flexibility and efficiency make it applicable in real-world scenarios, including fire detection, industrial safety, environmental monitoring, and potentially revolutionizing the development of smoke detection technologies. The proposed framework is unquestionably flexible and can be utilized in true circumstances, such as fire detection systems, industrial safety systems, and environmental monitoring systems. Additionally, it enables the invention of services and products such as smoke alarms and smoke detection cameras. YOLOv8 is a powerful smoke detection model that is well-suited for real-world applications. It is fast, accurate, efficient, and versatile. YOLOv8 is fast enough to be used for real-time smoke detection; it can detect smoke in videos and images as they are being captured. This is important for applications such as smoke detection systems in buildings and forests. YOLOv8 is also accurate, even in challenging lighting conditions and complex backgrounds, including dark rooms, foggy forests, and busy streets. YOLOv8 is efficient and requires relatively few resources to run, which makes it suitable for use on a variety of devices, including smartphones, embedded systems, and servers. YOLOVv8 is versatile and can be used to detect a wide variety of smoke types and sizes, such as fire detection systems, industrial safety systems, and environmental monitoring systems. The proposed YOLOv8-based smoke detection technology has the potential to significantly improve both environmental and public safety.

The main contribution of the proposed work is, i) First, a large dataset containing 5,700 images has been built. Then the collected image data is pre-processed using various image processing tools such as augmentation, resizing, annotating, or labeling; ii) After that, the pre-processed data is processed which prepares the data for training. Then the processed data is trained using the YOLOv8-based model; and iii) Finally, the results of the proposed YOLOv8-based model performance are compared with other baseline models, such as YOLOv7, YOLOv5, MobileNetv2, and ResNet32, where YOLOv8 outperforms them in all cases.

2. LITERATURE REVIEW

Smoke and fire detection using YOLOv8 reviews a diverse range of studies showcasing the evolving capabilities and applications of the YOLO algorithm in smoke and fire detection. The studies highlight significant advancements in speed, and efficiency across various contexts, including urban environments, marine engine rooms, and forest areas. Key contributions involve the integration of lightweight models like Light-YOLOv5, accomplishing an adjustment between exactness and computational effectiveness for real-time applications. The literature indicates a trend toward more sophisticated, efficient, and versatile YOLO-based smoke and fire detection systems, with potential applications in emerging technologies like the internet of things (IoT) and cloud computing for enhanced emergency management systems.

While advancements in YOLO-based smoke detection algorithms have shown significant progress, key areas like adaptability to diverse and challenging environments, computational efficiency for resource-constrained settings, and specificity to varying types of smoke and fire scenarios need further research and development. This ongoing refinement and testing are crucial for the practical and widespread application of these technologies in real-world scenarios. YOLO-F [13] shows high accuracy using YOLOv3 and YOLOv4 for flame detection. A notable shortcoming is the lack of testing in diverse fire scenarios, particularly in environments with varying light and smoke intensities. This raises questions about its adaptability to real-world, diverse fire situations. In research [14], a multi-task learning-based forest fire detection model (MTL-FFDET) introduces a multi-task approach, improving accuracy and reducing false detections. Despite these advancements, the model's complexity might hinder its deployment in resource-constrained environments, and its performance in detecting very small or distant fire sources has not been thoroughly evaluated. In research [15], common weed and crop dataset (CWC)-YOLOv5s used for marine engine room smoke detection demonstrate good precision and recall rates. However, the model's performance in environments with varying smoke densities and types, as well as its effectiveness in detecting smoke at the early stages of fire, are areas that need further exploration. Improved YOLOv7 [16] is applied for coal mine detection that shows effectiveness in challenging environments. Its application, however, is limited to specific scenarios (coal mines), and its adaptability to other smoke detection contexts like forest fires or urban settings is not addressed. In research [17], a smart fire detection system (SFDS) based on YOLOv8 is applied for smart cities that integrate with IoT and cloud computing, offering high precision. The potential limitation is the dependence on robust network infrastructure for real-time data processing, which could be a challenge in less developed areas. In research [18], Enhanced YOLOv7 unmanned aerial vehicles (UAV)-based forest fire smoke detection focuses on aerial imagery. The model's dependency on the quality of UAV imagery and its performance in different weather conditions are potential limitations.

In research [19], YOLOv8- multinational corporations (MNC) is used for smoking behavior detection and demonstrates improved accuracy. The algorithm's specific focus on smoking behavior limits its broader applicability to general fire or smoke detection scenarios. CNN-based smoke detection system [20] incorporates attention mechanisms and improved upsampling. The system's performance in detecting smoke from different sources and in various lighting conditions is not fully explored. In research [21], a characteristics-based fire detection system called YOLO-v4 and ViBe is introduced for fire detection under electric fields that demonstrates high accuracy. However, the algorithm's performance in varied environmental conditions beyond electric fields and its scalability to different fire detection scenarios remain untested. In research [22], YOLOv4 combined with MobileNetV3 is applied for fire detection which reduces the computational burden. The shortcoming lies in the algorithm's performance on less powerful hardware and its adaptability to larger, more complex scenes. In research [12], YOLOv6-based fire detection in smart cities demonstrates YOLOv6's efficiency, but the study lacks extensive evaluation in different urban settings, especially in environments with diverse lighting and occlusion challenges. In researchers [23]–[25], improved YOLOv5 is proposed by targeting better accuracy for small targets. A potential limitation is its performance in detecting fires in diverse agricultural environments and varying weather conditions. In research [26], a forest fire classification and detection model (FCDM) is proposed that improves the detection of various forest fire types but may face challenges in differentiating fires in dense forest areas and under varying environmental conditions. In research [27], they compare YOLO models for smoke detection, but there is a lack of in-depth analysis of each model's performance in rapidly changing environmental conditions. ReSTiNet [28] showcases the application of Tiny-YOLO-based CNN architecture. The model's applicability to fire or smoke detection scenarios is not discussed, which limits its relevance to this specific review.

The review identifies common shortcomings across these studies. One significant challenge is the adaptability of these algorithms to diverse and dynamically changing environments, such as varying weather conditions and landscapes. Additionally, the computational efficiency of more complex models in resource-constrained settings remains a concern. There is also a need for further research to improve the models' effectiveness in detecting different types and stages of fires and smoke, especially in early detection scenarios.

Overall, the literature indicates a strong trend towards more sophisticated, efficient, and versatile YOLO-based smoke and fire detection systems. These systems show promise in not only enhancing current fire detection capabilities. They are also integrating with emerging technologies such as IoT and cloud computing for smarter, more responsive emergency management systems.

3. RESEARCH METHODOLOGY

In this section, procedures for carrying out the study are covered. A framework for the detection and localization of smoke and fire zones was developed using the YOLOv8 model. The architecture automatically extracted many smoke and fire aspects from an input image in order to identify and precisely localize the potential location.

Figure 1 presents a process flowchart for all methods used to detect smoke, fire, and both smoke and fire, which aids in a deeper understanding of the paper's concept. First of all, collect data from different sources, which is presented in detail in subsection 3.1. Then the collected data is pre-processed which is explained in subsection 3.2 with proper diagrams and explanations. After that, the pre-processed data is processed and divided into two phases. In phase 1, the training part is described. Here, a label map is created and the YOLOv8 model is configured. Then the data is trained with the proposed YOLOv8 model. Finally, the desired training graphs from the training process are extracted. In phase 2, the testing part is explained. After completing the training process, the data set is set for testing. In this case, first, the parameters are adjusted and feed data for accurate predictions. Then, we must go through some steps (fire proposed return on investment (RoI) polling and softmax). After that, we get output with a bounding box while evaluating the test data.

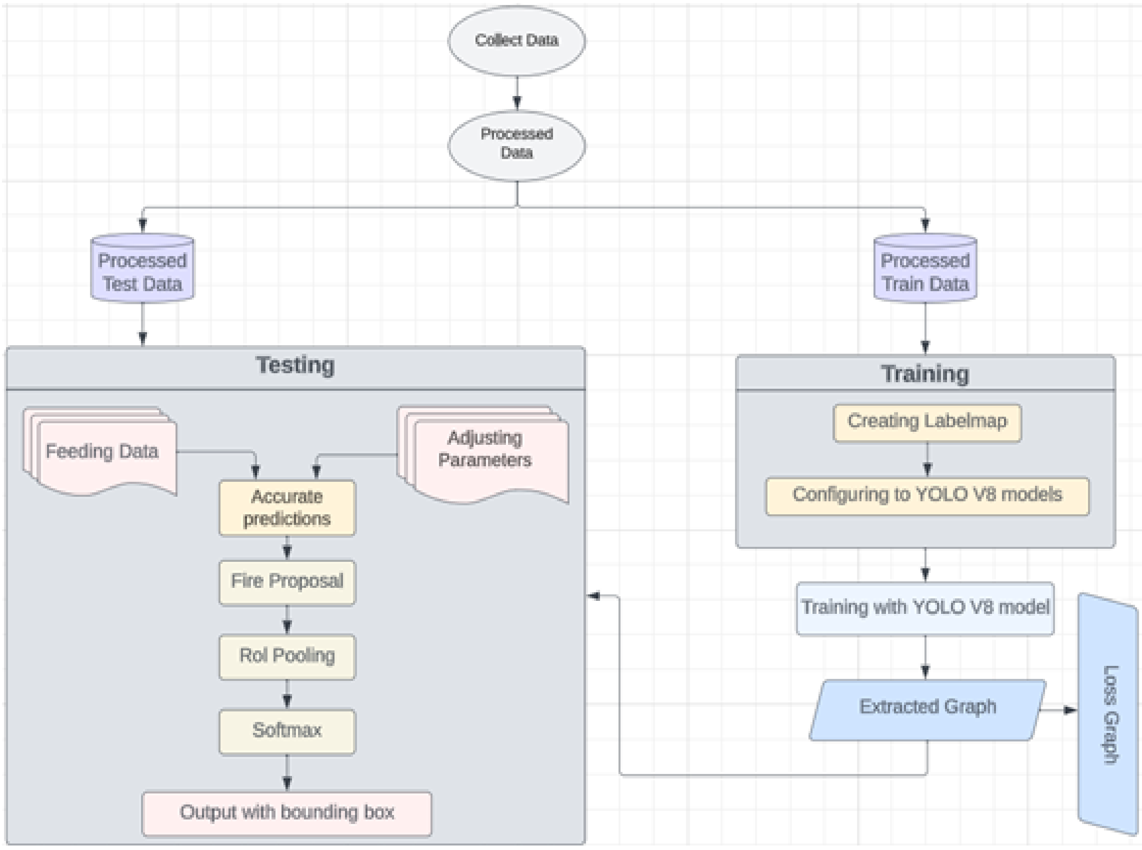


Figure 1. Workflow architecture diagram for the whole detection process

3.1. Image acquisition

Information collection is a vital step that includes gathering data from differing, substantial sources. Within the setting of this situation, in this ponder, the information is first collected from diverse online sources, which are as if they were pictures. These online sources can be websites, social media platforms, online databases, or any other significant online store. To supplement the dataset, other photos from the web are also collected in this dataset. Once the vital information and pictures are collected from online sources, a custom dataset is formed. This compilation comprises diverse images showcasing day and night fires, aerial views, fixed shot fires, mountain, surface, trunk, and canopy fires, as well as natural forest images with disturbances. Table 1 shows the measurements of the picture dataset, where we have categorized picture datasets into three sorts (fire pic, smoke pic, and both fire and smoke) for the proposed framework.

Table 1. Dataset statistics

Category	Train	Test	Validation	Total
Fire	2,379	670	350	3,399
Smoke	665	190	95	950
Fire and Smoke	946	270	135	1,351
Total	3,990	1,130	580	5,700

A dataset serves as one of the most profitable for preparing machine learning models, conducting investigations, or performing examinations. The custom dataset is curated based on the necessities and targets of the extent, guaranteeing that it includes a different run of significant information to support the planning objectives. Figure 2 appears in a few sample picture datasets. In Table 1, it is evident that the total number of fire-related images is 3,399. These images are partitioned for different purposes, with 70% allocated for training, 10% for validation, and the remaining 20% designated for testing. Similarly, we used an equal percentage of images for both smoke and fire. The number of pictures is 950 and 1,351, respectively. This distribution ensures a comprehensive and balanced dataset for the development and evaluation of models related to fire detection or analysis.

Figure 2. Samples of raw image data

3.2. Data pre-processing

3.2.1. Augmentation

Data augmentation is a commonly employed technique aimed at enhancing dataset diversity and variability, ultimately enhancing the execution and flexibility of ML models. We have collected our information from different sources with distinctive sizes and resolutions. That's why we apply information enlargement methods in a few pictures to make the information set more reasonable. We have connected distinctive information enlargement procedures for the pictures, as taken after. In case 1, we applied crop and rotational augmentation techniques. Since we have collected our data set from different sources, we have to crop some data and sometimes apply rotation to our data to label the shapes. This method involves augmenting specific images by randomly applying rotations within a range of -20 to +20 degrees. This augmentation technique aims to mimic real-world scenarios where objects can appear in various orientations, thereby making the dataset more representative. Exposing the model to rotated images during training improves its capacity to memorize and generalize, supporting superior execution when experiencing pivoted pictures amid deduction or testing stages. In case 2, there were some images in our data set that lacked brightness, which is why we had to make a brightness adjustment. We used some random brightness adjustments in the data augmentation step. The percentage of the adjustment was between +20 to -20. This method proved instrumental in improving the visibility and definition of dimly lit images, rendering them more conducive for analysis and facilitating their suitability for model training purposes. In case 3, we used the flip technique. Here, we selected some random data and then just flipped it in a different position. We used this technique to enhance our data set.

3.2.2. Image resizing

The process of resizing an image involves standardizing the dimensions of all images within a dataset, ensuring uniformity in their shapes. In this context, the entire custom dataset, comprising various images, underwent resizing procedures to achieve dimensions of 640×640 pixels. Resized images are shown in Figure 3. Then, the resized data is a set of annotations.

Figure 3. Samples of resized image data

3.2.3. Image labeling or annotating

First, roboflow labeling software was used to choose the necessary scaled image. Following that, the smoke or fire region of the picture data was highlighted and labeled as ‘smoke’ and ‘fire’ using the options entitled ‘bounding box tool’ and ‘polygon tool’. The annotation gets saved automatically, and there is an export option that creates the TXT file, which provides information in-depth on the selected smoke and fire region. Figure 4 shows a flowchart for labeling the resized image data and data pre-processing steps. Figure 4(a) shows the flowchart for labeling the images.

The aggregate of the information pre-processing steps is clearly delineated in Figure 4(b) and labeled portions of sample data are shown in Figure 5. Upon concluding these steps, the prepared information experienced division, apportioning 70% for preparation, 20% for testing, and 10% for validation purposes to encourage demonstration preparation. Then the pre-processed data is set for data processing steps.

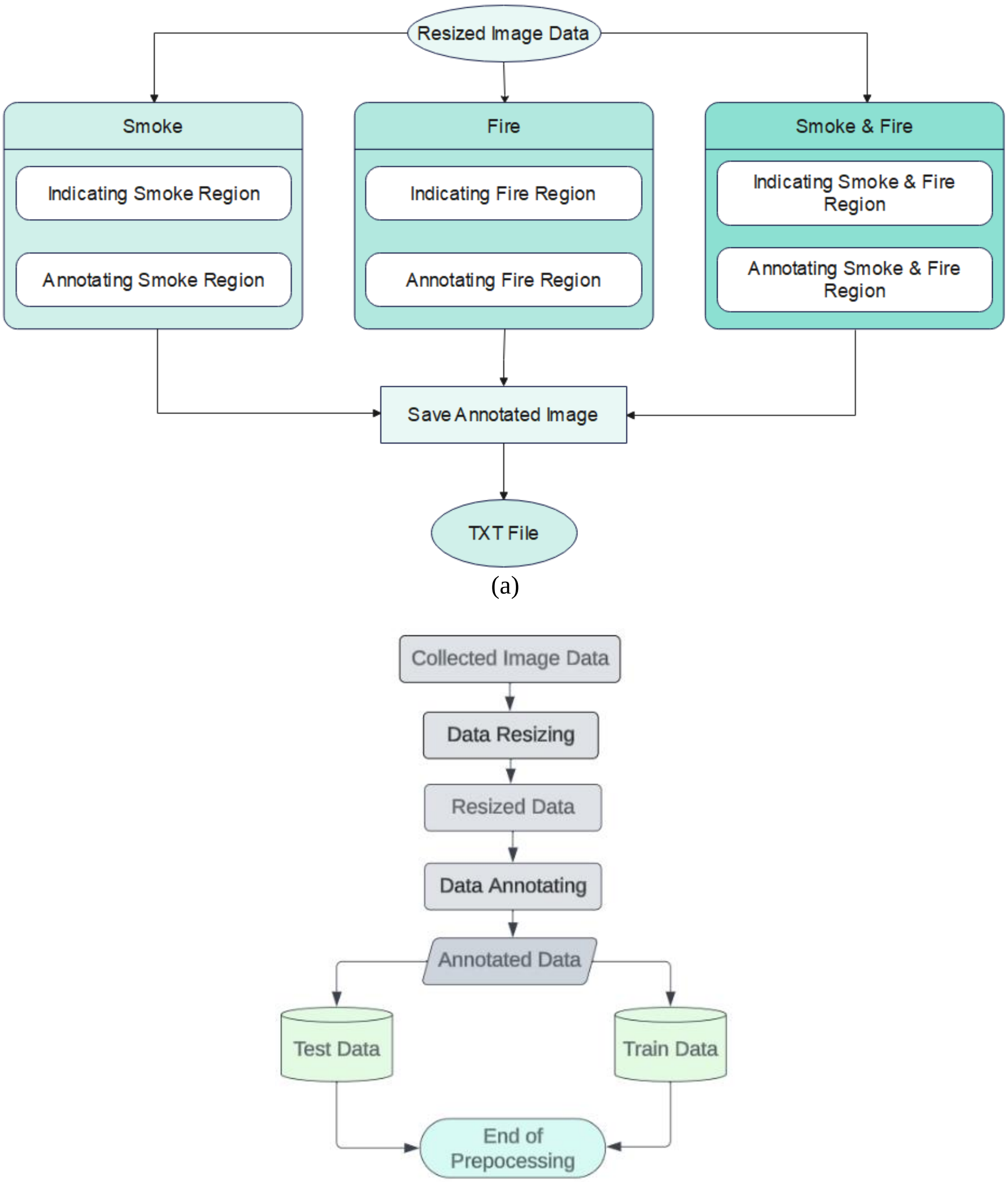


Figure 4. Flowchart for (a) labeling the resized image data and (b) data pre-processing steps

Figure 5. Samples of labeled image data

3.3. Data processing

In this data processing context, only one step is considered for generating a TXT file. Considering that, a file of plain text was created for exporting data easily and importing in a structured manner. Then the processed data is set for model training steps.

4. RESULTS AND DISCUSSION

Numerous pivotal training configurations and hyperparameters are intricately woven into the process of training the YOLOv8-based model for detecting fire and smoke. This section provides a comprehensive overview, delving into the hyperparameters meticulously utilized throughout the training regimen. During the training phase, an epoch count of 100 is set, leveraging the stochastic gradient descent (SGD) optimizer and the common objects in context (COCO) pre-trained model. To curtail overfitting and streamline the training trajectory, an early stopping mechanism is integrated into the model. This intervention halts the training prematurely when improvement is absent over the last 50 epochs, effectively preventing unnecessary computation cycles. Employing an early stopping technique with a patience value of 50 implies that if no discernible enhancement occurs for 50 consecutive epochs, the training process will cease automatically. It is worth noting that future experiments could explore fine-tuning the patience value to achieve even greater optimization.

Moreover, meticulous attention is paid to other pivotal parameters. Parameters such as batch size, learning rate, and weight decay values are calibrated at 16, 0.01, and 0.001, respectively, in a concerted effort to amplify model optimization. These nuanced adjustments fortify the model's ability to discern fire and smoke instances with heightened accuracy and robustness, warranting a thorough exploration of potential refinements in forthcoming experiments.

4.1. Model evaluation

The evaluation metrics employed in this paper to assess the model's performance included precision (P), recall (R), average precision (AP), mean average precision (mAP), F1 score, parameters, floating point operations (FLOPs), and frames per second (FPS). AP represents the area under the precision-recall (PR) curve, while mAP signifies the average AP across different categories. The formulas used for these metrics are outlined as (1)-(3).

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \tag{1}$$

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \tag{2}$$

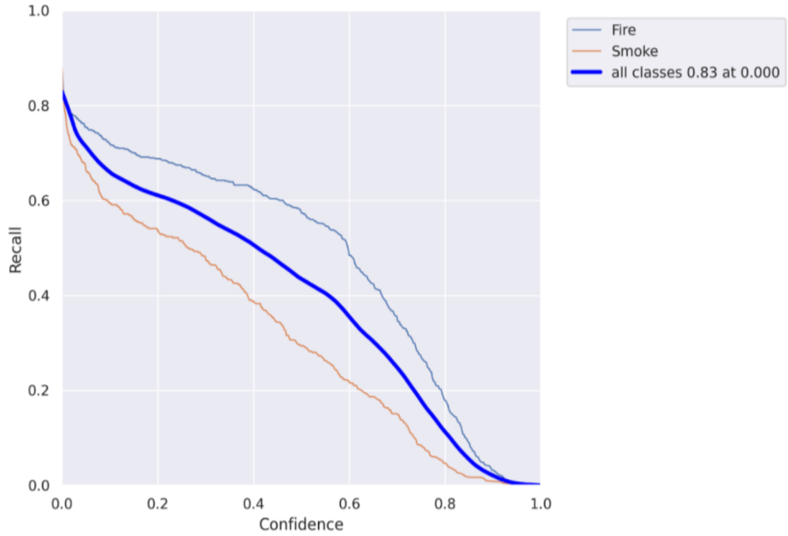
$$\text{mAP} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i-1}}^{\mathbf{n}} \int_{\mathbf{0}}^{\mathbf{1}} \mathbf{p(R)\ dR} \tag{3}$$

True positive (TP) signifies accurate classification of a sample as positive, while false positive (FP) denotes incorrect classification of a sample as positive. False negative (FN) signifies the misclassification of a sample as negative. 'n' represents the count of categories. FLOPs are a measure of computational complexity, indicating the number of computations performed by a model. FPS stands for frames per second, representing the rate at which frames are transmitted.

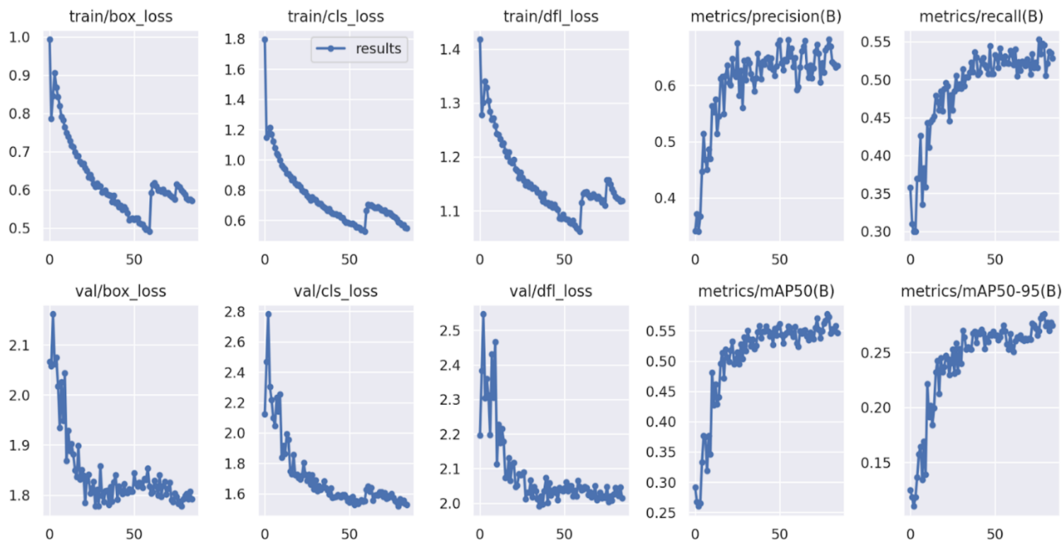
The YOLOv8 model was evaluated for fire and smoke detection within a data collection system. The evaluation was conducted using Python on a platform equipped with CUDA 12.0 and NVIDIA-SMI 525.85.12, employing a GTX 1650 GPU with 4 GB of VRAM and 12 GB of RAM. The model comprises 225 layers and 11138309 parameters, showcasing efficient computation with a GFLOP value of 28.7. Various metrics were employed to gauge the model's efficacy in detecting fire and smoke.

4.2. Analysis of results

Figure 6 shows the YOLOv8-based recall confidence curve and training graph with 85 epochs. The YOLOv8-based recall confidence curve is shown in Figure 6(a) for individual fire and smoke classes, where all class values are around 0.83 at 0.0. Then, the YOLOv8-based training graphs with 85 epochs are depicted in Figure 6(b), showing that the best results are obtained at training step 80 for the proposed scheme. Thus, the decision to train for 100 epochs is based on the observed performance. The early stopping mechanism was not activated because there was not much difference between the maximum epoch and the best result steps. Further, the proposed YOLOv8 exhibits a recall of 54.6%, a precision value of 68.3%, and a mAP of 57.3% when trained with 85 epochs. For all classes, the recall confidence threshold value is 0.83 at 0.000. The graph, recall vs. confidence curve merged smoothly.



(a)



(b)

Figure 6. YOLOv8-based (a) recall confidence curve and (b) training graph with 85 epochs

4.2.1. Confusion matrix

A confusion matrix is a fundamental tool in machine learning for evaluating the performance of a classification algorithm which is particularly useful in understanding the strengths and weaknesses of a model, aiding in the assessment of precision, recall, and overall accuracy. The confusion matrix from Figure 6(a) reveals insightful information about the model's performance. It defines three classes: fire, smoke, and background. TP indicates correctly identified instances, while FP represents misclassifications. For fire, the model correctly identifies 70% (TP) as fire but confuses 30% (FN) with the background. Similarly, smoke detection achieves 56% (TP) accuracy, but misclassifies 43% (FN) as background. It is important to note that the values within the matrix range from 0.01 to 0.70, offering further insights into specific confidence levels for each classification. This analysis highlights areas for improvement, such as potentially reducing false negatives to enhance overall detection accuracy.

Analyzing the training curve in Figure 6(b), we can see that the better result is achieved on 82 iterations and continues up to 85. The evaluation of the proposed fire-related phenomena detection instances, employing the YOLOv8 model, extends to a comparative analysis with other established object detection models such as YOLOv7 [9], YOLOv5 [25], Mobilenet-v2 [29], and ResNet-32 [30]. The performance metrics are meticulously assessed across all classes with varying iteration steps, as detailed in Table 2. The mAP@0.5 is tracked during the training phase, indicating the model's ability to learn on the validation set, with a higher value denoting superior learning.

Additionally, the F1score, calculated through a precise formula, reveals that the YOLOv8-based model consistently outperforms others, boasting an F1score and mAP@0.5 values of 60 and 57.3%, respectively. A detailed examination of model complexities in Table 2 further highlights YOLOv8's superiority, particularly over YOLOv7, which exhibits the highest number of trainable parameters, potentially compromising generalization capacity.

Furthermore, Figure 7 offers a comprehensive view of the confusion matrix diagram at the 85th iteration, showcasing the model's performance limitations with an accuracy of 90.45%. This matrix visually dissects predicted and actual classes, with notable accuracy peaks observed for the fire class (70%) and a corresponding dip for the smoke class (56%). The YOLOv8-based model, however, reveals promising potential for real-time fire and smoke detection in diverse scenarios, emphasizing its efficacy in still images, videos, and camera feeds through rigorous testing and analysis.

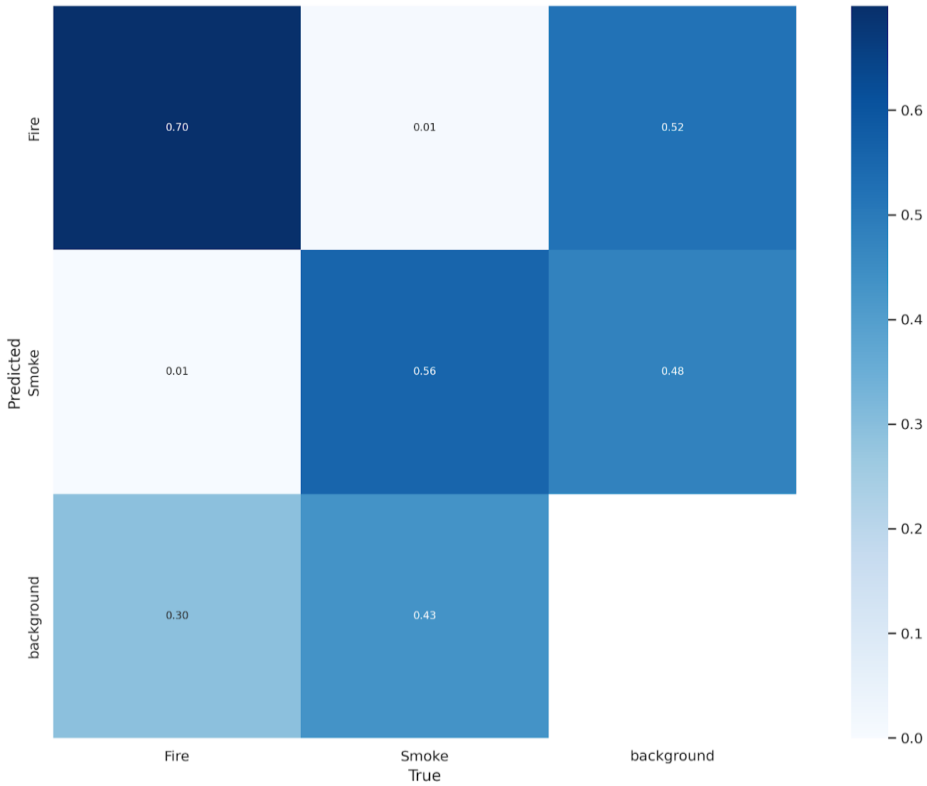


Figure 7. Confusion matrix diagram for 85 epochs

Table 2. Testing execution of YOLOv8 with YOLOv7, YOLOv5, MobileNet V2 and ResNet-32

Model	Epoch	Class	Trainable Parameters	F1 Score	mAP@0.5
Proposed YOLO v8	50	All	11.13M	0.587	0.555
Proposed YOLO v8	85	All	11.13M	0.607	0.573
YOLO v7 [9]	50	All	37.2M	0.430	0.372
YOLO v7 [9]	85	All	37.2M	0.458	0.391
YOLO v5 [25]	50	All	7.2M	0.487	0.426
YOLO v5 [25]	85	All	7.2M	0.487	0.426
MobileNet- v2 [29]	50	All	3.4M	0.359	0.324
MobileNet- v2 [29]	85	All	3.4M	0.365	0.335
ResNet- 32 [30]	50	All	0.47M	0.267	0.245
ResNet- 32 [30]	85	All	0.47M	0.276	0.255

4.3. Visualization

The model's training spanned 100 epochs, signifying a complete iteration through the entire training dataset in each epoch. The training process involved iterative updates to the model's parameters based on calculated losses and gradients. The decision to conclude the training at 85 epochs aligns with the configuration settings, with optimal results emerging around the 80th step. The non-engagement of the early stopping mechanism is justified, as its activation criteria weren't met within the specified 85 epochs after achieving the best result. Consequently, the training process halted at 85 iterations, consuming approximately 3 hours. This duration, subject to variations based on computational resources and hardware, underscores the resource-intensive nature of the training. The performance of the object detection model peaked at 85 epochs and then started to decline. This is a common phenomenon called overfitting, which occurs when the model memorizes the training data too well and loses its ability to generalize to new data.

The individual detection accuracies for YOLOv8, YOLOv7, and YOLOv5 in 85 training steps are shown in Figure 8, considering all the classes. In the case of the first row, the detection accuracies are 90 and 51% for the YOLOv8, and YOLOv5 models, but we see that YOLOv7, ResNet-32, and MobileNet-v2 cannot detect any fire portion. Similarly, for the YOLOv8 model, we see in the second row’s first picture that the detection accuracy for smoke is 64% and for fire is 66%, We also see that for YOLOv7 and ResNet-32 there was no detection for this picture, but for the same picture, the YOLOv5 and MobileNet-v2 models can detect only the fire portion, which is 53 and 52%, respectively. For the YOLOv8 model and MobileNet-v2, we see in the third row’s first and last pictures that the detection accuracy for smoke is 75 and 59%, respectively, but for the same picture, the YOLOv7, YOLOv5, and ReNet-32 models cannot detect any smoke region. Lastly, here we also see the detection accuracies are 79, 60, 47, 27, and 49% for YOLOv8, YOLOv7, YOLOv5, ResNet-32, and MobileNet-v2 models for the fourth row. Thus, for all classes in most cases, the detection accuracy is higher in the proposed YOLOv8-based smoke and fire detection model compared to the other YOLOv7 and YOLOv5 models with 85 epochs. Among all the models that are being trained with our custom dataset, the YOLOv8 gave the best performance metrics along with detection accuracy in all aspects. The model is trained with different epochs; among them, the training curve of the YOLOv8 50 Epoch is slightly better, but in terms of all other evaluation metrics, the YOLOv8 85 Epoch model performs best, and the performance started to decrease in the 100 and 150 epochs for overfitting.

Proposed YOLOv8 YOLOv7 [9] YOLOv5 [25] ResNet-32 [30] MobileNet-v2 [29]

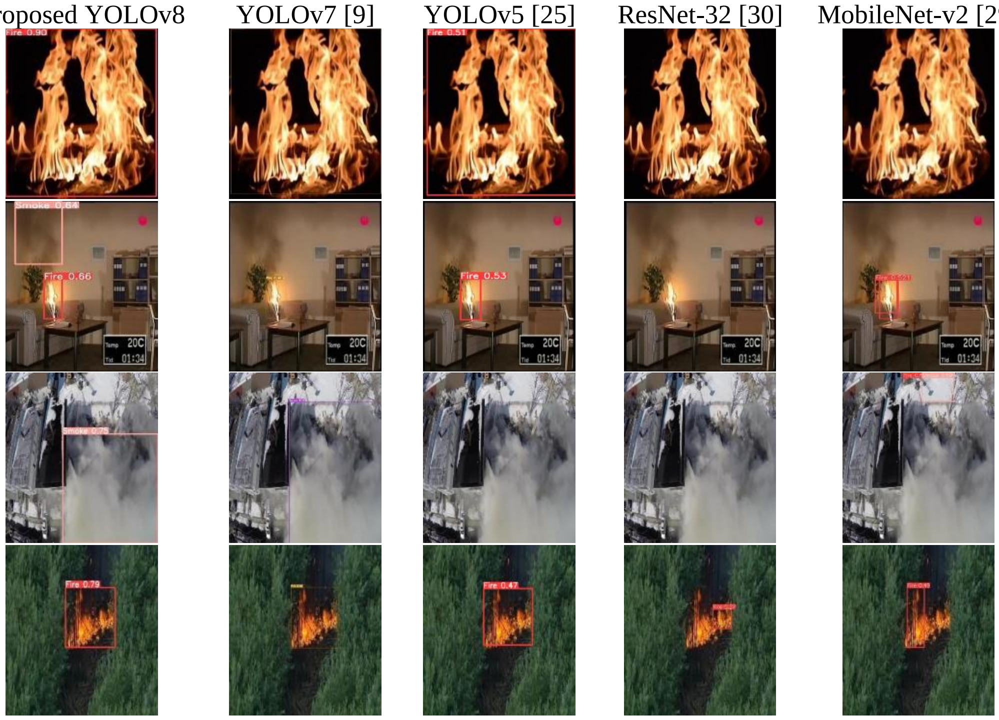


Figure 8. Sample detected images

5. CONCLUSION

This study presents a significant leap in the field of smoke and fire detection by leveraging the capabilities of the YOLOv8 deep learning model. Demonstrating a marked improvement over traditional detection methods and previous iterations of the YOLO model, YOLOv8 distinguished itself through superior accuracy, speed, and dependability. With a training dataset of 5700 images, the system showcased notable performance metrics: precision of 68.3%, recall of 54.6%, F1 score of 60.7%, and mAP of 57.3%. These outcomes highlight YOLOv8's robust potential in practical settings, facilitating quicker and more flexible responses to emergent fire and smoke situations, which is vital for averting extensive calamities. Nevertheless, this research acknowledges existing constraints. It identifies the necessity for a broader examination of the variability of environmental factors, fire and smoke properties, and the overall system's adaptability. Future research directions could include the expansion of the training dataset to encompass instances of smoke under various obstructions, different lighting and weather conditions, and a range of fire origins. Further, exploring synergies between the model and additional sensing technologies, alongside integration with real-time monitoring systems, could significantly boost both the efficacy and reliability of detection systems.

ACKNOWLEDGEMENTS

This research is funded by Woosong University Academic Research 2024.

REFERENCES

[1] S. Rahman, J. H. Rony, J. Uddin, and M. A. Samad, "Real-time obstacle detection with YOLOv8 in a WSN using UAV aerial photography," *J. Imaging*, vol. 9, no. 10, p. 216, Oct. 2023, doi: 10.3390/jimaging9100216.

[2] R. Siddiqua, S. Rahman, and J. Uddin, "A deep learning-based dengue mosquito detection method using faster R-CNN and image processing techniques," *Ann. Emerg. Technol. Comput.*, vol. 5, no. 3, pp. 11–23, Jul. 2021, doi: 10.33166/AETiC.2021.03.002.

[3] S. B. Hasan, S. Rahman, M. Khaliluzzaman, and S. Ahmed, "Smoke detection from different environmental conditions using faster R-CNN approach based on deep neural network," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 325 LNICST, 2020, pp. 705–717. doi: 10.1007/978-3-030-52856-0_56.

[4] P. D. K. He, G. Gkioxari and R. Girshick, "Mask R-CNN," in *IEEE international conference on computer vision*, IEEE, 2017, pp. 2961–2969. [Online]. Available: openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html

[5] A. Khondaker, A. Khandaker, and J. Uddin, "Computer vision-based early fire detection using enhanced chromatic segmentation and optical flow analysis technique," *Int. Arab J. Inf. Technol.*, vol. 17, no. 6, pp. 947–953, Nov. 2020, doi: 10.34028/iajit/17/6/13.

[6] R. A. Khan, J. Uddin, and S. Corraya, "Real-time fire detection using enhanced color segmentation and novel foreground extraction," in *4th International Conference on Advances in Electrical Engineering, ICAEE 2017*, IEEE, Sep. 2017, pp. 488–493. doi: 10.1109/ICAEE.2017.8255405.

[7] R. A. Khan, J. Uddin, S. Corraya, and J.-M. Kim, "Machine vision-based indoor fire detection using static and dynamic features," *Int. J. Control Autom.*, vol. 11, no. 6, 2018, doi: 10.14257/ijca.2018.11.6.09.

[8] H. Zheng, J. Duan, Y. Dong, and Y. Liu, "Real-time fire detection algorithms running on small embedded devices based on MobileNetv3 and YOLOv4," *Fire Ecol.*, vol. 19, no. 1, p. 31, May 2023, doi: 10.1186/s42408-023-00189-0.

[9] H. Du, W. Zhu, K. Peng, and W. Li, "Improved high speed flame detection method based on YOLOv7," *Open J. Appl. Sci.*, vol. 12, no. 12, pp. 2004–2018, 2022, doi: 10.4236/ojapps.2022.1212140.

[10] S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y. I. Cho, "An improved wildfire smoke detection based on YOLOv8 and UAV images," *Sensors (Basel).*, vol. 23, no. 20, 2023, doi: 10.3390/s23208374.

[11] H. Xu, B. Li, and F. Zhong, "Light-YOLOv5: a lightweight algorithm for improved YOLOv5 in complex fire scenarios," *Appl. Sci.*, vol. 12, no. 23, 2022, doi: 10.3390/app122312312.

[12] S. Norkobil Saydirasulovich, A. Abdusalomov, M. K. Jamil, R. Nasimov, D. Kozhamzharova, and Y. I. Cho, "A YOLOv6-based improved fire detection approach for smart city environments," *Sensors*, vol. 23, no. 6, 2023, doi: 10.3390/s23063161.

[13] K. Xu, Y. Xu, Y. Xing, and Z. Liu, "YOLO-F: yolo for flame detection," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 37, no. 1, 2023, doi: 10.1142/S0218001422500434.

[14] K. Lu, J. Huang, J. Li, J. Zhou, X. Chen, and Y. Liu, "MTL-FFDET: a multi-task learning-based model for forest fire detection," *Forests*, vol. 13, no. 9, 2022, doi: 10.3390/f13091448.

[15] Y. Zou *et al.*, "Smoke detection of marine engine room based on a machine vision model (CWC-YOLOv5s)," *J. Mar. Sci. Eng.*, vol. 11, no. 8, 2023, doi: 10.3390/jmse11081564.

[16] S. Sun, H. Ma, K. Wang, C. Wang, Z. Wang, and H. Yuan, "Research on the detection method of coal mine roadway bolt mesh based on improved YOLOv7," *Electron.*, vol. 12, no. 14, 2023, doi: 10.3390/electronics12143050.

[17] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, 2023, doi: 10.1007/s00521-023-08809-1.

[18] S. Y. Kim and A. Muminov, "Forest fire smoke detection based on deep learning approaches and unmanned aerial vehicle images," *Sensors*, vol. 23, no. 12, 2023, doi: 10.3390/s23125702.

[19] Y. Hu *et al.*, "Fast forest fire smoke detection using MVMNET," *Knowledge-Based Syst.*, vol. 241, 2022, doi: 10.1016/j.knosys.2022.108219.

[20] H. Yin, M. Chen, W. Fan, Y. Jin, S. G. Hassan, and S. Liu, "Efficient smoke detection based on YOLO v5s," *Mathematics*, vol. 10, no. 19, 2022, doi: 10.3390/math10193493.

[21] S. Zhao, B. Liu, Z. Chi, T. Li, and S. Li, "Characteristics based fire detection system under the effect of electric fields with improved YOLO-v4 and vibe," *IEEE Access*, vol. 10, pp. 81899–81909, 2022, doi: 10.1109/ACCESS.2022.3190867.

[22] X. Xie, K. Chen, Y. Guo, B. Tan, L. Chen, and M. Huang, "A flame-detection algorithm using the improved YOLOv5," *Fire*, vol. 6, no. 8, 2023, doi: 10.3390/fire6080313.

[23] Y. Zhou, M. Wu, Y. Bai, and C. Guo, "Flame detection with pruned and knowledge distilled YOLOv5," *Proc. 2021 5th Asian Conf. Artif. Intell. Technol. ACAIT 2021*, pp. 780–785, 2021, doi: 10.1109/ACAIT53529.2021.9731227.

[24] J. Li and X. Lian, "Research on forest fire detection algorithm based on improved YOLOv5," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 3, pp. 725–745, 2023, doi: 10.3390/make5030039.

[25] L. Zhang, J. Li, and F. Zhang, "An efficient forest fire target detection model based on improved YOLOv5," *Fire*, vol. 6, no. 8, 2023, doi: 10.3390/fire6080291.

[26] Q. Xue, H. Lin, and F. Wang, "FCDM: an improved forest fire classification and detection model based on YOLOv5," *Forests*, vol. 13, no. 12, 2022, doi: 10.3390/f13122129.

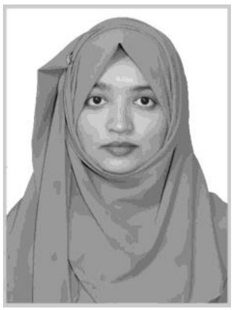
[27] Y. Al-Smadi *et al.*, "Early wildfire smoke detection using different YOLO models," *Machines*, vol. 11, no. 2, 2023, doi: 10.3390/machines11020246.

[28] S. S. Sumit, D. R. Awang Rambli, S. Mirjalili, M. M. Ejaz, and M. S. U. Miah, "ReSTiNet: on improving the performance of Tiny-YOLO-based CNN architecture for applications in human detection," *Appl. Sci.*, vol. 12, no. 18, 2022, doi: 10.3390/app12189331.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.

BIOGRAPHIES OF AUTHOR



Shakila Rahman is a lecturer in the Department of Computer Science and Engineering, Faculty of Science and Technology, American International University-Bangladesh (AIUB). She received an M.Sc. in AI and Computer Engineering from the University of Ulsan, South Korea, and completed a B.Sc. in Computer Science and Engineering from IIUC, Bangladesh. Her research interests include machine learning, artificial intelligence, image processing, optimization algorithms, and wireless sensor networks. She can be contacted by email: shakila.rahman@aiub.edu.

Syed Muhammad Hasnat Jamee is a recent graduate student in the Department of Computer Science and Engineering, Faculty of Science and Technology, American International University-Bangladesh, with a steady academic performance. His research interests include machine learning and deep learning. He can be contacted by email: jamee.syed.42918@gmail.com.

Jakaria Khan Rafi is a recent graduate student in the Department of Computer Science and Engineering, Faculty of Science and Technology, American International University-Bangladesh, with a steady academic performance. His research interests include machine learning and deep learning. He can be contacted by email: jakariakhan135@gmail.com.

Jafrin Sultana Juthi is a recent graduate student in the Department of Computer Science and Engineering, Faculty of Science and Technology, American International University-Bangladesh, with a steady academic performance. Her interests include machine learning and deep learning. She can be contacted by email: juthijafrin0@gmail.com.

Abdul Aziz Sajib is a recent graduate student in the Department of Computer Science and Engineering, Faculty of Science and Technology, American International University-Bangladesh, with a steady academic performance. His research interests include machine learning and deep learning. He can be contacted by email: abdulazizsajib@gmail.com.

Jia Uddin is an Assistant Professor, at the Department of Artificial Intelligence and Big Data, College of Endicott, Woosong University, Daejeon, South Korea. He received a Ph.D. in Computer Engineering from the University of Ulsan, South Korea, an M.Sc. in Electrical Engineering (Specialization: Telecommunications), from the Blekinge Institute of Technology, Sweden, and a B.Sc. in Computer and Communication Engineering, from the International Islamic University Chittagong, Bangladesh. He was a visiting faculty at the School of Computing, Staffordshire University, United Kingdom, Telkom University, Indonesia, and the University of Foggia, Italy. He was an Associate Professor at the Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh. His research interests are Industrial Fault Diagnosis, Machine Learning/Deep Learning prediction, and detection using multimedia signals. He can be contacted by email: jia.uddin@wsu.ac.kr.