ICIC Express Letters
Volume **17**, Number **3**, March **2023**
 ISSN 1881-803X
pp. **339–347**

DETECTION AND CLASSIFICATION OF MOVING VEHICLE AT NIGHT WITH VISIBLE CAMERA USING DEEP LEARNING MODEL

Ramadhani Dwi Susanti[1,*] and Suryadiputra Liawatimena[1,2]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science
[2]Computer Engineering Department, Faculty of Engineering
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
suryadi@binus.edu
*Corresponding author: ramadhani.susanti@binus.ac.id

Received February 2022; accepted May 2022

Abstract. *The number of vehicles in Indonesia significantly influences congestion. This problem can be solved by engineering traffic lights based on the average vehicle volume density. Furthermore, it is essential to determine the number and type of vehicles passing in an area. Vehicles could be detected and classified by using computer vision-based and deep learning approaches. Although many studies have examined vehicle detection and classification during the day, there is room for improvement at night because the low light condition results in poor image quality. Therefore, this study used YOLOv3 as a detection model and modification of the VGG16 with transfer learning and added a global average pooling layer as a classification model to detect and classify vehicles at night. A test on three video trials for detecting and classifying buses, trucks, cars, and motorcycles at night was conducted. The built system resulted in an average accuracy of 92.93%.*
Keywords: Object detection, Deep learning, Vehicle detection, Night vision, Computer vision

1. **Introduction.** TomTom, a renowned technology agency, has released the results of the TomTom Traffic Index 2019. The survey results showed that one city in Indonesia is among the 10 most congested cities globally [1]. Indonesia has about 138.56 million vehicles [2], significantly influencing congestion. Studies on vehicle detection and classification could generally be conducted using a computer vision approach based on deep learning [3]. One of the deep learning methods uses the Convolutional Neural Network (CNN). For instance, Tarmizi and Aziz developed a vehicle detection and calculation model with CNN modification to detect images with a 94.3% accuracy during the day but only 61.4% at night [4]. For specific vehicle detection at night with CNN in low light conditions, Cai et al. used images captured using a Far-Infrared (FIR) camera, resulting in an accuracy of 92.3% [5]. The cameras produce sufficient contrast between the object and the background images [6]. However, they are relatively expensive to be applied throughout Indonesia. Without an FIR camera, vehicle images taken at night cannot be identified because they hardly contrast with the background. Also, other problems such as vehicle lights and image noise occur in the background. The problems motivated this study because there are limited studies on object detection at night without using an FIR camera due to the challenges of managing images in low light conditions.

Vehicle detection and classification have the same principle as general object detection and classification. Lawal utilized YOLOv3 in detecting tomatoes and produced a precision of 97.4% [7]. Wagle and Ramachandran compared several CNN architectures on

DOI: 10.24507/icicel.17.03.339

tomato leaf classification and found that VGG16 is superior with an accuracy of 97.29% [8]. Moreover, Ye et al. utilized the VGG19 architecture on pest image detection. The classification accuracy of the modified VGG19 model increased from 85.83% to 99.99% [9]. Reddy and Juliet built a model for classifying malaria-infected cells using the ResNet-50 model, resulting in a validation accuracy of 95.4% [10]. The three studies on object classification utilized transfer learning and layer modification to build the models [8-10]. Therefore, this study used the YOLOv3 architectural approach to detect objects at night. Transfer learning also modified the VGG16, VGG19, and ResNet50 original models by taking several layer weights from the model that has been trained by ImageNet and then using it in the new model developed and added a global average pooling layer to classify vehicles at night. The model with high accuracy was tested on video with three different road conditions at night with low light using a visible camera.

Section 2 of this paper introduces the CNN algorithm, transfer learning, and global average pooling layer. It is followed by introducing the proposed model in Section 3, while Section 4 presents the experiment results. Section 5 provides a conclusion from the experiment conducted.

2. Theoretical Basis.

2.1. **Convolutional neural network.** The CNN architecture consists of several stages, whose inputs and outputs comprise arrays called feature maps. Each stage consists of the convolution, activation function, and pooling layers. Figure 1 shows a convolutional neural network architecture network [11].

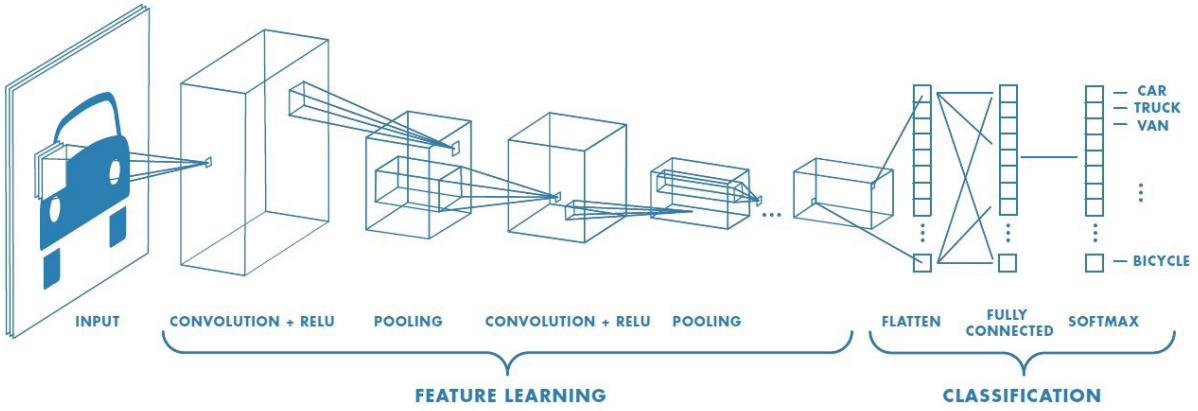

FIGURE 1. The architecture of convolutional neural network

The first stage in the CNN architecture is the convolution stage, conducted using a kernel of a specific size. The number of kernels used depends on the number of features produced. The activation function is performed using the Rectifier Linear Unit (ReLU) activation function, and then it goes through the pooling process. This process is repeated severally to obtain a sufficient map to proceed to the fully connected neural network that produces the output class [12].

2.2. **Transfer learning.** This approach uses a previously trained neural network and reduces the number of parameters by taking parts of the trained model to recognize the new ones [13]. In general, transfer learning needs a pre-trained model usually trained on large benchmark datasets to ensure it is excellent [14,15].

2.3. **Global average pooling layer.** It is often used to replace the fully connected layer in the classifier [16]. The model ends with a convolution layer that produces as many feature maps as the number of target classes. It applies unification with an average value to converting each feature map into one value. This approach improves model

performance by reducing overfitting because there are no parameters to learn in the global average pooling layer [17,18].

3. Methodology.

3.1. **Dataset acquisition.** Data were collected at night from several places in Surabaya, one metropolitan city in Indonesia. Images were taken from the vehicle's rare due to the light conditions of the headlights at night. The pictures were taken from the pedestrian bridge at the height of 6 meters at an angle of 50 degrees. The built model tested with a video at 720 pixels × 480 pixels was taken at 25 fps. The data used for the model training process was 300 images, consisting of buses, trucks, cars, and motorcycles, and 165 frames images containing a mixture of the four types of vehicles. Figure 2 shows pictures of the vehicles.

(a) Bus (b) Truck (c) Car (d) Motorcycle

FIGURE 2. Types of vehicle images

3.2. **Data preprocessing.** This aspect involved image preprocessing to improve quality [19] and augmentation to increase the data and add variety [20]. The intensity transformation technique through denoising images and contrast and brightness enhancement was used. Specifically, augmentation was conducted using geometric transformation through resizing and horizontal flipping.

- Intensity transformation: The image denoising process reduces noise [21], while contrast and brightness enhancement adjust the light intensity in the image [22]. For example, the pixel (i, j) is the intensity of the original pixel at the coordinates (i, j) and (i, j) is the intensity of the resulting pixel where $\alpha > 0$ is the gain parameter (contrast) and β is the bias parameter (brightness) and then the process contrast & brightness enhancement is defined by Equation (1). Figure 3 shows all the intensity transformation process.

$$g(i, j) = \alpha \cdot f(i, j) + \beta \tag{1}$$

(a) Original image (b) Denoising image (c) Contrast & brightness enhancement image

FIGURE 3. Intensity transformation image

- Geometrical transformation: The resizing process equalizes the input image by changing its horizontal and vertical resolution to 224 pixels × 224 pixels to be processed in the training model. The horizontal flipping process increases image variations because if the vehicle image is flipped horizontally, it is still identified as a vehicle.

3.3. **Proposed model.** In this paper, two different models were used in the detection process and in the classification process. YOLOv3 was used for vehicle detection, while the best-improved models between VGG16, VGG19, and ResNet50 for vehicle classification models were tested using several video conditions. Figure 4 shows the proposed architecture.

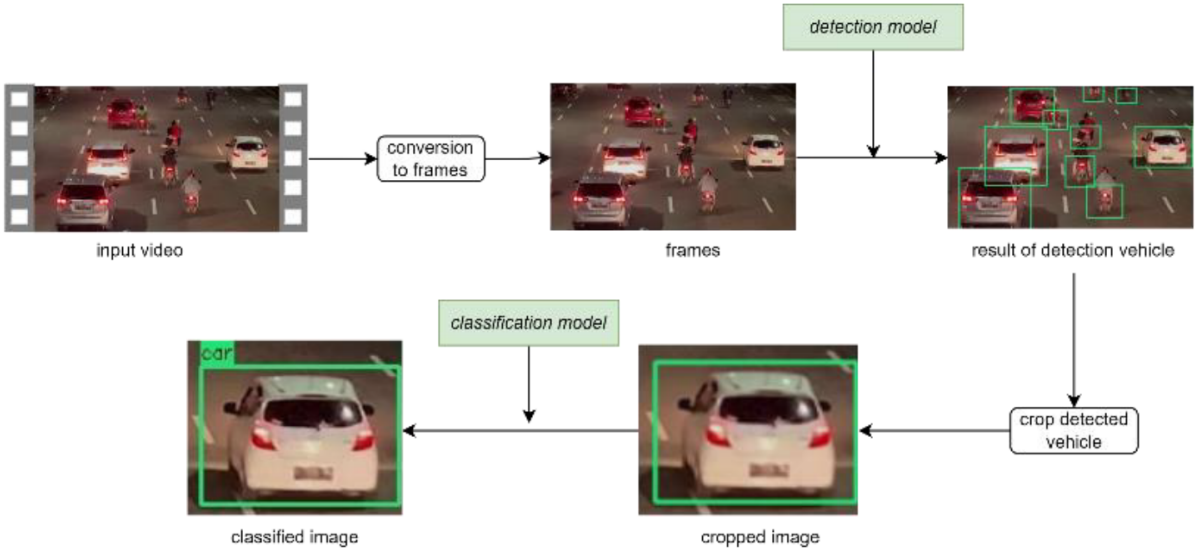

FIGURE 4. Overview vehicle detection and classification model

The video is converted into a frame or image and then a testing process is carried out using a detection model that has been formed from the previous YOLOv3 architecture and produces an image with a bounding box that serves to capture objects including vehicles. Then, image in the bounding box is cropped and the classification model is used to predict what type of vehicle the image belongs to and finally, a label is added in the form of the text of the type of vehicle for easy identification.

3.3.1. *Detection model architecture.* The detection model formation process was obtained from training image datasets using the YOLOv3 architecture to determine the presence of a vehicle object in an image. A pre-trained model with a Darknet-53 backbone was used to implement the YOLOv3 architecture. Figure 5 shows the system design scheme for vehicle detection.

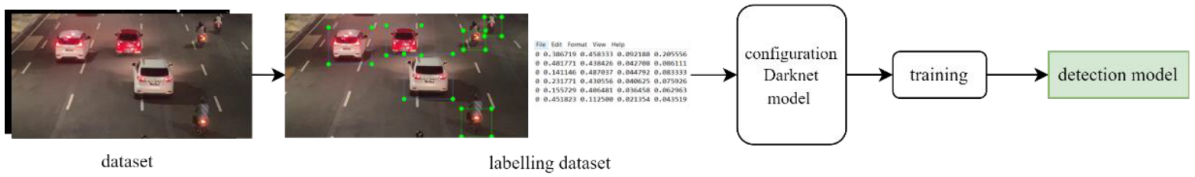

FIGURE 5. Vehicle detection with YOLOv3 architecture

The image datasets collected were labeled using a bounding box to produce a jpg file representing the image. The labeling also produced a txt file with information about the coordinates of the image bounding box location. After dataset collection and annotation, some Darknet model information was configured as the backbone. YOLOv3 was used to detect objects identified as vehicles and not classify them to shorten training time.

3.3.2. *Classification model architecture.* The classification model formation process was obtained from dataset training by comparing three CNN architectures: VGG16, VGG19,

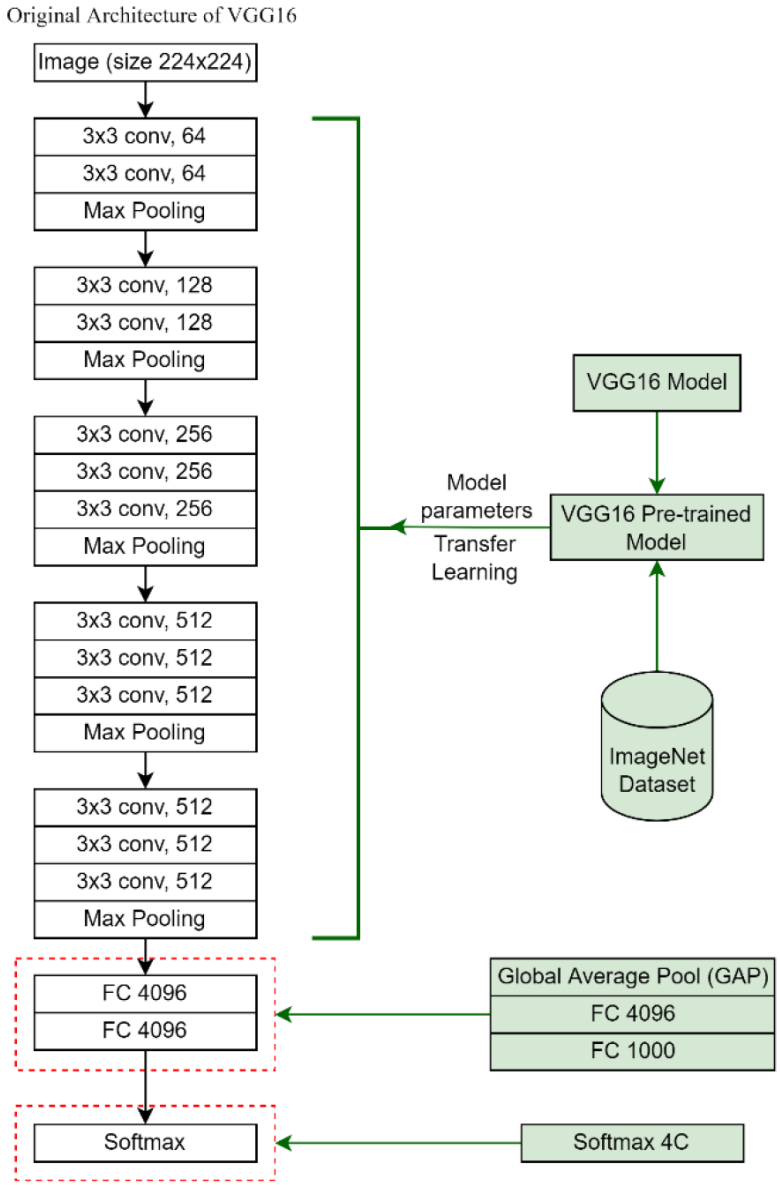

FIGURE 6. Proposed model architecture

and ResNet50. Figure 6 shows the proposed architectural modification. The VGG16 architecture was used to compare the baseline model based on its original version [23].

The proposed classification model was formed by utilizing transfer learning and modifying several layers. Vehicles were classified in four classes using transfer learning on the models trained using the ImageNet dataset. There is vehicle data even when the desired class is different but has the same characteristics. As a result, the weights tested on the model could be transferred. The model built reduces training time and produces lower generalization errors. Then layer was modified by adding a Global Average Pooling (GAP) layer before the first fully connected layer. This prevents overfitting by reducing the number of parameters in the model to speed up the training process. The four-label SoftMax classifier replaces the SoftMax classification layers based on the number of classes [24].

4. **Experiment Result.** The experimental process used Python 3.8.8 using the PyTorch framework and a desktop computer of Windows 10 (64 bit), 16GB RAM, 240GB SSD, Intel Core i5-10300H CPU @ 2.50GHz, and Nvidia GTX 1650 4GB GPU. The training process on forming the classification model was conducted using epoch 30, optimizer

adam, and learning rate 0.0001. The best model was selected based on the training and validation accuracy and the value generated at each epoch.

4.1. **Vehicle detection implementation results.** This section discusses the results of evaluating the implementation of the YOLOv3 architecture on 80% of the 165-frame images. The detection model was evaluated on 20% of images from 165 frames as validation data and divided into Types 1, 2, and 3 with quiet, moderate, and busy road conditions, respectively. The evaluation stage was based on the accuracy calculated through the number of detected vehicles divided by the actual number of vehicles.

An evaluation was conducted on the overall data validation by assessing the average accuracy obtained by each type through the vehicles detected divided by the actual number of vehicles. Table 1 presents the results, showing that conditions affect the detection model's performance. The model using YOLOv3 shows that vehicle detection at night performs better in quiet road conditions.

Table 1. Average accuracy of the YOLOv3 detection model

Type	Sum of images	Average accuracy
Type 1	14	97.42%
Type 2	11	93.09%
Type 3	8	88.87%

4.2. **Vehicle classification implementation results.** This study tested three baseline architectures and one proposed modified architecture for vehicle classification. Each architecture was tested with training and validation data to determine the accuracy of the results. The baseline architecture implementation evaluated the three basic architectures on the existing VGG16, VGG19, and ResNet50. This was conducted without changing the number of layers or the arrangement of the original versions. Each reference architecture was tested to evaluate the performance of the resulting model for training a classification model. Table 2 compares the implementation results of the three baseline architectures. Comparisons were conducted on the training and validation accuracy values at the epochs producing the best validation loss.

Table 2. Comparison of baseline classification architecture results

Model	Parameter weight	Training accuracy	Validation accuracy	Runtime
VGG16	134,276,932	94.20%	86.40%	9.50 s
VGG19	139,586,628	93.00%	86.20%	10.69 s
ResNet50	23,542,724	99.20%	84.70%	7.37 s

Table 2 shows that the VGG16 model produces the highest validation accuracy of 86.40%, with a runtime of 9.50 s. The ResNet50 model produces the lowest validation accuracy of 84.70%, with a runtime of 7.37 s. Although the ResNet50 training produces the highest accuracy, this could indicate overfitting. Therefore, this study focused more on the validation accuracy assessment. The experimental results show that VGG16 has a more complex architecture with a high validation accuracy value. However, the computation time is longer than the ResNet50 model, with fewer weight parameters. Therefore, this study proposed changing (VGG16) the architecture with the highest validation accuracy. Table 3 compares the implementation results of the VGG16 baseline architecture and the proposed model.

The comparison result shows that the proposed model produces a higher validation accuracy of 98.30% than VGG16 (baseline), with a validation accuracy of 86.40%. It has a better runtime process that decreases more than half the runtime process of the original VGG16.

Table 3. Comparison of classification architecture results

Model	Parameter weight	Training accuracy	Validation accuracy	Runtime
VGG16 (baseline)	134,276,932	94.20%	86.40%	9.50 s
VGG16 (transfer learning + global average pooling layer)	6,700,752	100%	98.30%	4.12 s

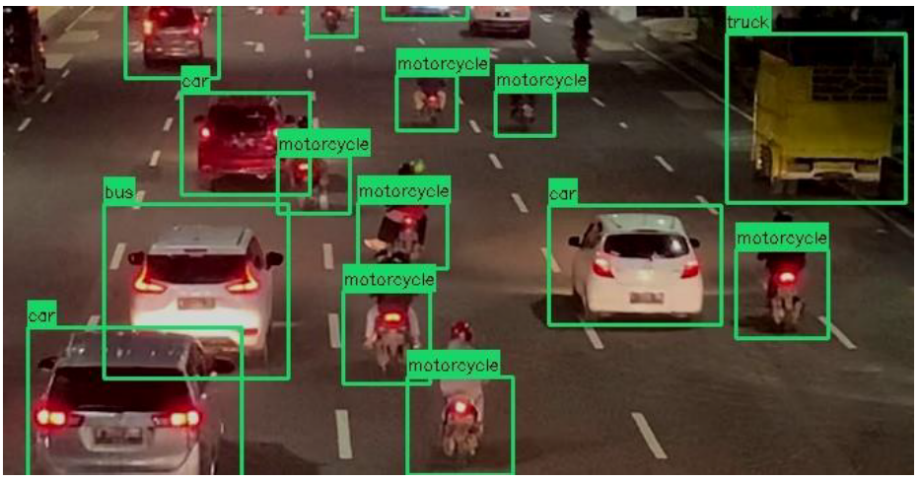

Figure 7. Example of detection and classification results on video

Table 4. Comparison of detection result and actual

Video	Detection result					Actual				
	Bus	Truck	Car	Motorcycle	Total	Bus	Truck	Car	Motorcycle	Total
Video 1	1	0	1	24	26	1	0	1	24	26
Video 2	0	1	12	13	26	0	1	12	14	27
Video 3	3	1	22	0	26	3	2	24	0	29

Table 5. Result of implementation of the proposed architecture

Video	Bus		Truck		Car		Motorcycle		Accuracy
	P	R	P	R	P	R	P	R	
Video 1	100%	100%	–	–	100%	100%	100%	100%	100%
Video 2	–	–	50%	100%	100%	91.66%	100%	92.85%	92.59%
Video 3	66.67%	66.67%	50%	50%	95.65%	84.61%	–	–	86.21%

4.3. **Implementation of the proposed architecture on video.** After building the detection model and determining the classification model, trials were conducted on video with various conditions taken in Surabaya, Indonesia. The architecture used in the video test evaluation process is YOLOv3 + Proposed Model (VGG16 transfer learning + global average pooling). The videos to be tested were divided into three types of conditions. Video 1 had relatively calm traffic conditions with a duration of 30 seconds, while Video 2 had moderate traffic conditions of 15 seconds. In contrast, the last video namely Video 3 had heavy traffic conditions with a duration of 10 seconds. Figure 7 shows an example of the model test results on the video. The implementation of the architecture used in detection and classification was evaluated. This involved comparing the number of detection and classification in the system and manual calculations in Table 4. The results calculated by the confusion matrix technique with precision (P) and recall (R) are shown in Table 5.

Table 5 shows that the overall average accuracy for detecting and classifying moving vehicles at night is 92.93%. However, the highest accuracy could reach 100% in light traffic conditions and drop to 86.21% under heavy traffic. Low traffic conditions result in better accuracy because the vehicles do not overlap. Overlap conditions make them difficult to be recognized by the engine.

5. **Conclusions.** This study proposed a solution to detect and classify vehicles at night. Several similar studies have used deep learning-based night vehicle detection, but they took relatively expensive infrared cameras. The resulting images also show differences between objects and backgrounds. In contrast, this study proposed a method to detect vehicles at night using a visible camera. The study performed several stages of image processing to improve image quality. The proposed model improved the performance of the original VGG16 model, with the validation accuracy increasing from 86.40% to 98.30%. Also, the combination of the YOLOv3 + Proposed Model (VGG16 transfer learning + global average pooling) model detects and classifies moving vehicles at night at an average accuracy of 92.93%.

Future studies could gather more complex datasets with different conditions and use an optimal design architecture model. They could choose the latest architectural base with better performance to detect and classify more than four classes of vehicles.

REFERENCES

[1] Indozone, *TomTom Traffic Index: Jakarta Ranks 10th on World's Most Congested Cities*, https://www.indozone.id, 2020.
[2] Transportologi, *How Many Indonesia's Vehicle in 2017?*, https://www.transportologi.org, 2019.
[3] J. Dorner, Š. Kozak and F. Dietze, Object recognition by effective methods and means of computer vision, *The 20th International Conference on Process Control (PC)*, DOI: 10.1109/PC.2015.7169962, 2015.
[4] I. A. Tarmizi and A. A. Aziz, Vehicle detection using convolutional neural network for autonomous vehicles, *International Conference on Intelligent and Advanced System (ICIAS),* 2018.
[5] Y. Cai, X. Sun, H. Wang, L. Chen and H. Jiang, Night-time vehicle detection algorithm based on visual saliency and deep learning, *Journal of Sensors*, vol.2016, pp.1-7, 2016.
[6] H. Kim, A knowledge-based infrared camera system for invisible gas detection utilizing image processing techniques, *Journal of Ambient Intelligence and Humanized Computing*, 2019.
[7] M. O. Lawal, *Tomato Detection Based on Modified YOLOv3 Framework*, Scientific Reports, 2021.
[8] S. A. Wagle and H. Ramachandran, A deep learning-based approach in classification and validation of tomato leaf disease, *Traitement du Signal*, vol.38, no.3, pp.699-709, 2021.
[9] H. Ye, H. Han, L. Zhu and Q. Duan, Vegetable pest image recognition method based on improved VGG, *Journal of Physics: Conference Series*, vol.1237, no.3, 2019.
[10] A. S. B. Reddy and D. S. Juliet, Transfer learning with ResNet-50 for malaria cell-image classification, *International Conference on Communication and Signal Processing (ICCSP)*, pp.945-949, 2019.
[11] S. Saha, *A Comprehensive Guide to Convolutional Neural Networks – The ELI5 Way*, https://towardsdatascience.com/, 2018.
[12] R. Yamashita, M. Nishio, R. K. G. Do and K. Togashi, Convolutional neural networks: An overview and application in radiology, *Insights into Imaging*, vol.9, pp.611-629, 2018.
[13] M. A. H. Abas, N. Ismail, A. I. M. Yassin and M. N. Taib, VGG16 for plant image classification with transfer learning and data augmentation, *International Journal of Engineering & Technology*, vol.7, no.4, pp.90-94, 2018.
[14] H. Pan, Z. Pang, Y. Wang, Y. Wang and L. Chen, A new image recognition and classification method combining transfer learning algorithm and MobileNet model for welding defects, *IEEE Access*, vol.8, pp.119951-119960, 2020.
[15] F. Zhuang et al., A comprehensive survey on transfer learning, *Proc. of the IEEE*, vol.109, no.1, pp.43-76, 2021.
[16] M. Lin, Q. Chen and S. Yan, Network in-network, *arXiv.org*, arXiv: 1312.4400v3, 2014.
[17] H. Ryu, J. Park and H. Shin, Classification of heart sound recordings using convolution neural network, *Computing in Cardiology Conference (CinC)*, pp.1153-1156, 2016.

[18] W. Zou, H. Lu, K. Yan and M. Ye, Breast cancer histopathological image classification using deep learning, *International Conference on Information Technology in Medicine and Education (ITME)*, pp.53-57, 2019.

[19] P. K. Bhaskar and S. Yong, Image processing based vehicle detection and tracking method, *International Conference on Computer and Information Sciences (ICCOINS)*, pp.1-5, 2014.

[20] C. Shorten and T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of Big Data*, vol.6, 60, DOI: 10.1186/s40537-019-0197-0, 2019.

[21] L. Fan, F. Zhang and H. Fan, Brief review of image denoising techniques, *Visual Computing for Industry, Biomedicine, and Art*, vol.2, no.7, 2019.

[22] L. Maurya, V. Lohchab, P. K. Mahapatra and J. Abonyi, Contrast and brightness balance in image enhancement using Cuckoo Search-optimized image fusion, *Journal of King Saud University – Computer and Information Sciences*, 2021.

[23] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv.org*, arXiv: 1409.1556v6, 2015.

[24] F. Gao, B. Li, L. Chen, Z. Shang, X. Wei and C. He, A softmax classifier for high-precision classification of similar ultrasonic signals, *Ultrasonics*, vol.112, 2021.