International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 14, No. 2, August 2025, pp. 645~652
ISSN: 2252-8776, DOI: 10.11591/ijict.v14i2.pp645-652

Human detection in CCTV screenshot using fine-tuning VGG-19

Firdaus Angga Dewangga, Abba Suganda Girsang
Department of Computer, BINUS Graduate Program, Master of Informatics, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Jun 28, 2024
Revised Nov 20, 2024
Accepted Dec 15, 2024

Keywords:

Fine-tuning
Human detection
Image classification
Transfer learning
VGG-19

ABSTRACT

Closed-circuit television (CCTV) systems have generated a vast amount of visual data crucial for security and surveillance purposes. Effectively categorizing security level types is vital for maintaining asset security effectively. This study proposes a practical approach for classifying CCTV screenshot images using visual geometry group (VGG-19) transfer learning, a convolutional neural network (CNN) classification model that works really well in image classification. The task in classification compromise of categorizing screenshots into two classes: "humans present" and "no humans present." Fine-tuning VGG-19 model attained 98% training accuracy, 98% validation accuracy, and 85% test accuracy for this classification. To evaluate its performance, we compared fine-tuning VGG-19 model with another method. The VGG-19-based fine-tuning model demonstrates effectiveness in handling image screenshots, presenting a valuable tool for CCTV image classification and contributing to the enhancement of asset security strategies.

Corresponding Author:

Firdaus Angga Dewangga
Department of Computer Science, BINUS Graduate Program, Master of Computer Science
Bina Nusantara University
Jakarta, 11480, Indonesia
Email: firdaus.dewangga@binus.ac.id

1. INTRODUCTION

Closed-circuit television (CCTV) serves as a situational crime prevention (SCP) method aimed at enhancing formal surveillance in specific areas. SCP emphasizes crime deterrence by minimizing opportunities for criminal acts and elevating the perceived likelihood of being caught [1]. Video surveillance, facilitated by CCTV systems, has proven to be an effective method for monitoring specific locations. Areas can be continuously monitored around the clock using video surveillance devices, with footage accessible as needed, complemented by burglar alarm systems [2]. Burglar alarm systems have been a fundamental component of home security systems for many years. However, implementing intruder behavior detection techniques presents challenges due to inherent limitations such as resource constraints and motion artifacts [3]. These challenges in modern surveillance systems result in a high number of false alarms [4]. To address this issue, we propose the utilization of deep learning algorithms to reduce false alarms in CCTV screenshots. In recent years, deep learning, particularly convolutional neural networks (CNNs), has achieved significant advancements in object classification and recognition [5].

CNN framework is utilized in numerous aspects of daily life, such as anomaly detection, natural language processing (NLP), computer vision, time-series prediction, image identification, drug development, video evaluation, health risk analysis, and recommendation systems. This allows for the encoding of image features into the architecture, making the structure especially effective for tasks involving images while

minimizing the parameters required to set up the model [6]. CNNs are a type of feedforward networks inspired from the structure of animal visual cortex [7]. They are variations of multilayer perceptrons, with neurons arranged in three dimensions, establishing a local connectivity pattern among nearby neurons and sharing the weights of learned filters. The architecture of visual geometry group (VGG-19) was influenced by AlexNet, a CNN presented in previous years' ImageNet competitions [8]. VGG-19 is a CNN thorough 19 layers, including 3 fully connected layers and 16 convolution layers, designed for images classification into 1,000 object categories [9]. An in-depth illustration of the VGG-19 architecture highlights the significance of the initial convolutional layers in capturing fundamental attributes such as shapes and edges, offering a comprehensive understanding of the model's early data progression from input to output [10].

The field of deep learning, pattern recognition, and human-computer interaction has garnered significant interest from research scientists, with a major focus on image classification [11]. This research also delves into image classification using transfer learning, which involves leveraging knowledge from another task that have used pre-trained model. Transfer learning significantly enhances learning performance by borrowing knowledge and label data from related domains and get extracted to assist a machine learning algorithm in achieving better performance in the target domain [12]. By transfer learning, we extract information patterns from screenshot images, improving our classification method and offering valuable insights to boost security management and asset preservation. By utilizing transfer learning and fine-tuning techniques, our model effectively categorizes images into two classes: human and non-human, reaching a remarkable 85% accuracy. This surpasses another method, such as VGG-16, which achieve 60% accuracy [13]. This demonstrates VGG-19's proficiency in image classification and its significant potential to enhance safety management strategies.

2. RESEARCH METHOD

A systematic six-step process was used in research method. First, an extensive survey is conducted to collect a diverse set of CCTV screenshot reports and relevant information. The gathered data then undergoes detailed preprocessing to improve its suitability and quality for analysis. Next, the dataset is carefully divided into training, validation, and test dataset, ensuring balanced representation. To simplify the process, each dataset is categorized into two classes: human and nonhuman. Model is then trained on the labeled training dataset, followed by a thorough validation of its performance. After training the model, we set and change hyperparameter of the model for fine-tuning and retrain the model with validation dataset. Lastly, an evaluation phase examines the model's precision in classifying and predicting human detection based on CCTV screenshots with test dataset. The conceptual framework of this research is illustrated in Figure 1.

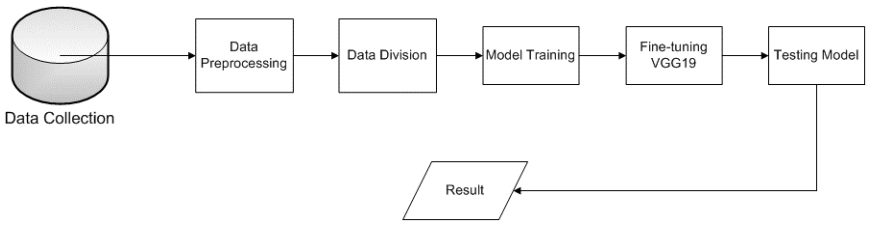

Figure 1. Conceptual framework human detection in CCTV screenshot using fine-tuning VGG-19

2.1. Data collection

The dataset size is considered a critical factor affecting the performance of model in machine learning [14]. Our study takes a practical approach to building a human detection classification model. We start by collecting historical real-life data and data from the Kaggle website, specifically the "Human Detection Dataset" by Verner [15], focusing on CCTV image screenshots related to human detection. We then refine the dataset to include only screenshots pertinent to human detection. Recognizing the importance of dataset size, our goal is to assess its practical influence on the accuracy of classification by our machine learning model for detecting humans in screenshots. The data collection process began by accessing the Kaggle website, specifically targeting 413 dataset clusters related to human detection. Through a thorough collection effort, 405 data points were gathered, resulting in a comprehensive dataset consisting of 2,820 instances. To further enhance the dataset, historical real-life data was added, bringing the total to 3,046 images. This enriched dataset serves as a strong foundation for in-depth analysis and exploration of various aspects of human detection, offering valuable insights for predictive decision-making.

2.2. Data preprocessing

A crucial preliminary step before employing machine learning techniques is using data preprocessing [16]. During data preparation, we undertake fundamental preprocessing tasks. This involves eliminating irrelevant CCTV screenshots to ensure image cleanliness and structural integrity. We also include CCTV images captured under various lighting conditions, ranging from low brightness at night to full brightness in daylight, thereby enhancing overall data consistency. Additionally, we filter out blurry, cropped, and irrelevant images, ensuring that our dataset is devoid of extraneous elements that could potentially disrupt our analysis.

A crucial part of the data analysis procedure is data processing, which often demands substantial effort and time [17]. In the preprocessing phase, the dataset was prepared using the image data generator library from TensorFlow. Each pixel value in the images was scaled to 1/255, ensuring uniformity across the dataset. Additionally, the images were resized to 150×150×3 dimensions, a step applied consistently to the training, validation, and test datasets.

2.3. Dataset division

In this phase of research, we partition dataset into three datasets: a training dataset, comprising 77% data, a validation dataset, constituting 19% data and test dataset with 4% data. The training dataset aids in the model's learning process and its performance to accurately classify image [18]. The accuracy of the model can be objectively evaluated using the validation dataset, providing insights into real-world classification results [19]. And test dataset will be used as testing the accuracy of the model in real world scenario.

This division is important for statistical analysis and machine learning, as the model used the training data to build and train the models, while the model used the validation data to encourage model's parameters in fine-tuning [20]. This partitioning ensures accuracy and integrity of the following modeling processes and analysis, resulting in reliable outcomes and insights. Each dataset containing two classes, training have human class (1,135 images) and nonhuman class (1,222 images), validation have human class (288 images) and nonhuman class (297 images), and test have human class (52 images) and nonhuman class (52 images). This data split gives a balanced representation for model training, validation and testing, enhancing the model's predictive capabilities and the robustness of the research findings.

2.4. Model training

In our training process, we utilize VGG-19, a CNN model trained on object detection and image classification data, which presents numerous benefits for processing image datasets [21]. With pretraining and fine-tuning, this model is capable of discerning intricate patterns and features within images [22]. The training process starts with transfer learning, utilizing pre-trained ImageNet weights and freezing all layers except the fully connected layer. The model is then trained on the training dataset with a learning rate of 0.001, batch size 64 and 25 epoch. After this, validation is performed using 20% split of the training dataset to evaluate the model's performance and to refine its ability in predicting human detection within screenshots. This validation helps in guiding the model's direction and improving its predictive accuracy.

The VGG-19 model, part of the VGG family, consists of 19 layers in total: 16 convolutional layers, 5 MaxPool layers, 1 fully connected layer, and 1 SoftMax layer. It involves 19.6 billion floating point operations (FLOPs). The VGG-19 model is simplified and made more effective by stacking 3×3 convolutional layers to increase its depth [23], [24]. In VGG-19, MaxPooling layers are used to reduce the size of volume, and only one fully connected (FC) layer is employed. The model obtains a 150×150×3 red, green, and blue (RGB) image as input. In CNN, there are four primary layers: convolution, pooling, rectified linear unit (ReLU), and fully connected to extract information from an image. The convolution layer uses several feature filters to perform convolution, comparing small sections of larger images to classify them. This process involves aligning the feature filter with image, multiplying corresponding pixels, summing these values, and dividing by the total number of pixels to obtain filtered image's final value. This process is reciprocated across image to obtain convolution output for each feature filter. ReLU, a rectified linear activation function, outputs the input directly if it is positive and zero otherwise. It is the default activation function for many neural networks due to its superior speed and performance. ReLU is applied to all rectified feature maps (feature images) to create the final output. During pooling, an initial window size is selected, and this window moves across image that have been filtered, giving maximum value from each section. Flatten layer then flattened the pooled feature map and sent the information to dense layer or fully connected layer. This layer performs the classification process and generates the final output using the SoftMax activation function. SoftMax, a type of probabilistic logistic regression, produces a probability distribution based on a given set of values. VGG-19 layers model are illustrated in Table 1.

Once the model generates predictions for human detection, we use test dataset to measure the accuracy of the transfer learning approach. This evaluation helps determine how well the model performs in real-world scenarios, providing insight into its practical effectiveness. Additionally, this process establishes a

baseline accuracy, which serves as a reference for further fine-tuning the model. By assessing the model in this way, we can identify areas for improvement and optimize its overall performance.

Table 1. Architecture layers VGG-19 model

Layer	Output shape
Input_1 (InputLayer)	([None, 150, 150, 3)]
Block1_conv1 (Conv2D)	(None, 150, 150, 64)
Block1_conv1 (Conv2D)	(None, 150, 150, 64)
Block1_pool (MaxPooling2D)	(None, 75, 75, 64)
Block2_conv1 (Conv2D)	(None, 75, 75, 128)
Block2_conv2 (Conv2D)	(None, 75, 75, 128)
Block2_pool (MaxPooling2D)	(None, 37, 37, 128)
Block3_conv1 (Conv2D)	(None, 37, 37, 256)
Block3_conv2 (Conv2D)	(None, 37, 37, 256)
Block3_conv3 (Conv2D)	(None, 37, 37, 256)
Block3_conv4 (Conv2D)	(None, 37, 37, 256)
Block3_pool (MaxPooling2D)	(None, 18, 18, 256)
Block4_conv1 (Conv2D)	(None, 18, 18, 512)
Block4_conv2 (Conv2D)	(None, 18, 18, 512)
Block4_conv3 (Conv2D)	(None, 18, 18, 512)
Block4_conv4 (Conv2D)	(None, 18, 18, 512)
Block4_pool (MaxPooling2D)	(None, 9, 9, 512)
Block5_conv1 (Conv2D)	(None, 9, 9, 512)
Block5_conv2 (Conv2D)	(None, 9, 9, 512)
Block5_conv3 (Conv2D)	(None, 9, 9, 512)
Block5_conv4 (Conv2D)	(None, 9, 9, 512)
Block5_pool (MaxPooling2D)	(None, 4, 4, 512)
Flatten (Flatten)	(None, 25088)
Dense (Dense)	(None, 256)
Dense (Dense)	(None, 2)

2.5. Fine-tuning VGG-19

After the initial training process, several parameters are adjusted to fine-tune the model. The fine-tuning process starts by unfreezing the last two layers, including the fully connected layer. Next, the learning rate is reduced to 10% of its original value, and data augmentation is applied to the validation dataset to enhance its diversity such as horizontal flip, rescaling, 30% rotation range, 20% width shift range, 20% height shift range, 20% zooming, and 20% shearing [25]. The model is then retrained using the augmented validation dataset with 128 batch size and 100 epochs. During this retraining phase, the model also undergoes validation through 20% split validation dataset to further assess its performance.

After fine-tuning the model and obtaining the prediction results, we proceed to evaluate its accuracy using a designated test dataset. This step is crucial because it allows us to assess how well the fine-tuned model performs when applied to unseen data, effectively simulating real-world conditions, and scenarios. By evaluating its performance on this independent dataset, we can identify any discrepancies between the model's training and testing capabilities. The primary goal of this evaluation is to ensure that the model is not only effective on the training data but also generalizes well to practical applications. This process provides insights into the model's robustness and reliability, which are essential for its deployment in real-world settings. Ultimately, accurate evaluation helps in building confidence in the model's predictive abilities and its potential impact in relevant fields.

2.6. Result and evaluation

In this section, the performance of the VGG-19 model, which has been tested, is compared with other models to gain deeper insights into its ability to classify images for human detection. The evaluation focuses on comparing performance metrics, particularly accuracy. Additionally, the final model presents a confusion matrix based on the test dataset, illustrating the true positive, true negative, false positive, and false negative predictions. The accuracy of the performance metrics is computed using the (1).

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Instances} \quad (1)$$

3. RESULTS AND DISCUSSION

3.1. Training model result

This phase, we will discuss the results of training VGG-19 transfer learning model. Researchers utilized the Python programming language to create the models, leveraging the TensorFlow library, which is

renowned for building deep learning models. Model development was conducted using the Jupyter Notebook IDE in Google Collaboratory. Despite some limitations, Google Collaboratory's GPU resources were adequate for this research. Creating the model in TensorFlow involves several parameters. VGG-19 model were trained with 2,097,922 trainable parameters and 20,024,384 non-trainable parameters. Last epoch or epoch 25 in this training resulting in 1.0000 in training accuracy and 1.0000 in validation accuracy. Last epoch also resulting in 0.0029 loss and 0.0045 validation loss. Training and validation result of the training model are illustrated in Figure 2.

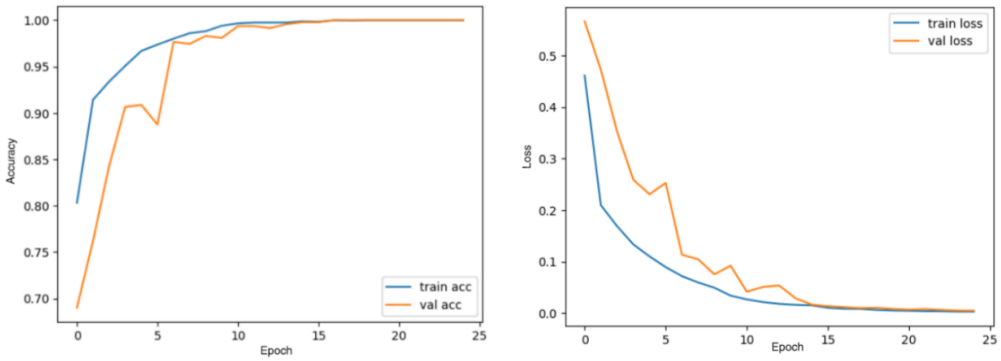

Figure 2. Accuracy and loss training VGG-19 model

After the model have prediction in human detection screenshot, then we test the prediction using test dataset with total of 52 human images and 52 nonhuman images with 64 batch size and false shuffle. For achieving an optimal classifier during training, evaluation metrics are essential [26]. The metrics for evaluation were accuracy, which were calculated by examining true positives, true negatives, false positives, and false negatives between the classes and prediction results of model. This testing resulting in 0.7596 or 75% accuracy with 45 true positives, 7 false positives, 18 false negatives and 34 true negatives.

3.2. Fine-tuning VGG-19 result

After training the model, we will discuss the results of fine-tuning VGG-19 model. Unfreezing 2 last layer in VGG-19 model or in block 4 layer creating 19,796,738 trainable parameters and 2,325,568 non-trainable parameters. The optimizer used was Adam with a learning rate of 0.0001, batch size 128, and the loss function was categorical cross-entropy. The fine-tuning training phase consisted of 100 epochs using validation dataset to evaluate model performance. Last epoch or epoch 100 in this training resulting in 0.9829 in training accuracy and 0.9828 in validation accuracy. Last epoch also resulting in 0.0440 loss and 0.0313 validation loss. Training and validation result of the fine-tuning model are illustrated in Figure 3.

After the model made predictions for human detection in the screenshots, we tested its performance using a test dataset consisting of 52 human images and 52 non-human images. The batch size used was 64, and the data was processed without shuffling. For evaluation, accuracy was measured by comparing the true positives, true negatives, false positives, and false negatives between the predicted results and the actual classes. The test yielded an accuracy of 85% (0.8557), with 50 true positives, 2 false positives, 13 false negatives, and 39 true negatives.

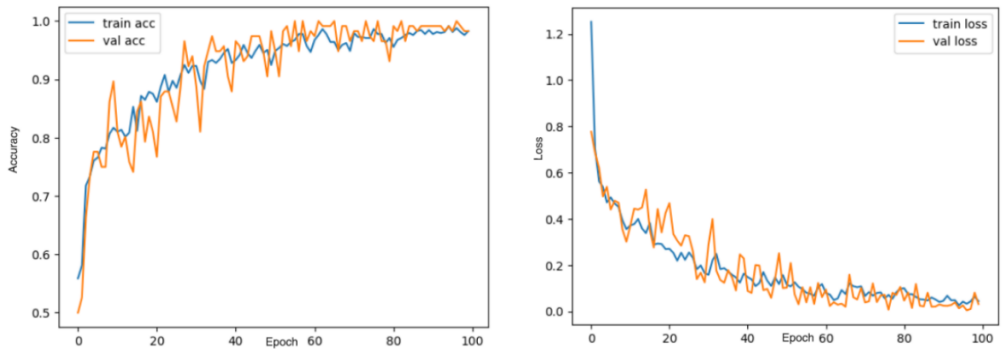

Figure 3. Accuracy and loss fine-tuned VGG-19 model

3.3. Evaluation

In our study, we measure performance finetuning VGG-19 model with another model for human image classification. The results show that VGG-19 achieved better accuracy rate 85% in testing, outperforming another method as shown in Table 2. This highlights the effectiveness of VGG-19 in classifying image categories, particularly within the context of CCTV screenshots. Although other method has its advantages in certain scenarios, VGG-19 outperforms it in managing the complexities and conditions of CCTV images, making VGG-19 ideal option for human detection in image classification process. Its deep learning capabilities allow it to catch specific details and nuances in CCTV images, resulting in improved accuracy and performance.

Table 2. Comparison with another method

Method	Accuracy
Fine-tuning VGG-19	85%
Transfer learning VGG-19	75%
VGG16 [13]	60%

4. CONCLUSION

This research proposed a different approach for human detection in screenshots, utilizing VGG-19, a customized object detection model for image classification. Achieving a remarkable 85% accuracy rate, this method surpasses other method in categorizing screenshots into two distinct classes, showcasing its proficiency in capturing image intricacies. This breakthrough represents a significant advancement in burglary prevention and security management, with potential to lessen fatalities, injuries, and economic losses associated with burglary incidents. Future research should focus on enhancing the model's capability to handle diverse image features. Furthermore, key goals include enhancing the classification system for more comprehensive object categorization and exploring real-time implementation for immediate human detection. Enhancing the dataset, collaborating with jurisdiction, and incorporating the model into decision support systems and detection reporting platforms are important actions to maximize its real-world effect and enhance safety in home surroundings.

ACKNOWLEDGEMENTS

We sincerely thank Mr. Konstantin Verner for generously providing important Human Detection Dataset via Kaggle. His dataset was crucial to our study, allowing us to conduct our research and achieve significant results in human detection classification. We deeply honor his willingness and support to share this important data collection, which has greatly advanced our comprehension of human detection in CCTV screenshots. We also widen our gratitude to the organizations and authors whose publications and research have been essential to our research. Their endowment has broadened our understanding and shaped the methodologies we utilized. Additionally, we value the variety of sources that provided essential insight and context for our research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Firdaus Angga Dewangga	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Abba Suganda Girsang		✓				✓				✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

ETHICAL APPROVAL

The research related to human use has been complied with all the relevant national regulations and institutional policies in accordance with the tenets of the Helsinki Declaration and has been approved by the authors' institutional review board or equivalent committee.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset.

REFERENCES

[1] E. L. Piza, B. C. Welsh, D. P. Farrington, and A. L. Thomas, "CCTV surveillance for crime prevention: a 40-year systematic review with meta-analysis," *Criminology and Public Policy*, vol. 18, no. 1, pp. 135–159, Feb. 2019, doi: 10.1111/1745-9133.12419.

[2] P. L. Chong, S. Ganesan, Y. Y. Than, and P. Ravi, "Designing an autonomous triggering control system via motion detection for IoT based smart home surveillance CCTV camera," *Malaysian Journal of Science and Advanced Technology*, pp. 80–88, Mar. 2023, doi: 10.56532/mjsat.v2is1.120.

[3] P. Netinant, A. Amatyakul, and M. Rukhiran, "Alert intruder detection system using passive infrared motion detector based on internet of things," in *ACM International Conference Proceeding Series*, Jan. 2022, pp. 171–175, doi: 10.1145/3520084.3520112.

[4] C. Muroi *et al.*, "Automated false alarm reduction in a real-life intensive care setting using motion detection," *Neurocritical Care*, vol. 32, no. 2, pp. 419–426, Apr. 2020, doi: 10.1007/s12028-019-00711-w.

[5] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon detection in real-time CCTV videos using deep learning," *IEEE Access*, vol. 9, pp. 34366–34382, 2021, doi: 10.1109/ACCESS.2021.3059170.

[6] A. A. Elngar *et al.*, "Image classification based on CNN: a survey," *Journal of Cybersecurity and Information Management*, p. PP. 18-50, 2021, doi: 10.54216/jcim.060102.

[7] M. Lagunas and E. Garces, "Transfer learning for illustration classification," *27th Spanish Computer Graphics Conference, CEIG 2017*, pp. 77–85, 2017, doi: 10.2312/ceig.20171213.

[8] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[9] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using VGG-19: Caltech-101 image data set," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3609–3620, 2023, doi: 10.1007/s12652-021-03488-z.

[10] D. Hindarto, N. Afarini, and E. T. Esthi H, "Comparison efficacy of VGG16 and VGG-19 insect classification models," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 3, pp. 189–195, 2023, doi: 10.33387/jiko.v6i3.7008.

[11] D. Jaswal, S. V, and K. P. Soman, "Image classification using convolutional neural networks," *International Journal of Scientific and Engineering Research*, vol. 5, no. 6, pp. 1661–1668, 2014, doi: 10.14299/ijser.2014.06.002.

[12] S. Tammina, "Transfer learning using VGG-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, p. p9420, 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[13] Srilaxmi, S. Kamath, and V. Mayya, "Automated human detection in images using deep learning," *Journal of Propulsion Technology*, vol. 44, no. 5, pp. 1897–1902, 2023.

[14] A. Althnian *et al.*, "Impact of dataset size on classification performance: an empirical evaluation in the medical domain," *Applied Sciences (Switzerland)*, vol. 11, no. 2, pp. 1–18, Jan. 2021, doi: 10.3390/app11020796.

[15] K. Verner, "Human detection dataset," *Kaggle*, 2022. https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset.

[16] Z. Xiao *et al.*, "Impacts of data preprocessing and selection on energy consumption prediction model of HVAC systems based on deep learning," *Energy and Buildings*, vol. 258, p. 111832, Mar. 2022, doi: 10.1016/j.enbuild.2022.111832.

[17] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, May 2017, doi: 10.1016/j.neucom.2017.01.078.

[18] B. Genç and H. Tunç, "Optimal training and test sets design for machine learning," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 2, pp. 1534–1545, Mar. 2019, doi: 10.3906/elk-1807-212.

[19] M. Rafało, "Cross validation methods: analysis based on diagnostics of thyroid cancer metastasis," *ICT Express*, vol. 8, no. 2, pp. 183–188, Jun. 2022, doi: 10.1016/j.icte.2021.05.001.

[20] V. Pradeep, A. B. Jayachandra, S. S. Askar, and M. Abouhawwash, "Hyperparameter tuning using Lévy flight and interactive crossover-based reptile search algorithm for eye movement event classification," *Frontiers in Physiology*, vol. 15, May 2024, doi: 10.3389/fphys.2024.1366910.

[21] M. A. Naufal and A. S. Girsang, "Traffic accident classification using IndoBERT," *International Journal of Informatics and Communication Technology*, vol. 13, no. 1, pp. 42–49, Apr. 2024, doi: 10.11591/ijict.v13i1.pp42-49.

[22] A. Stateczny, G. U. Kiran, G. Bindu, K. R. Chythanya, and K. A. Swamy, "Spiral search grasshopper features selection with VGG-19-ResNet50 for remote sensing object detection," *Remote Sensing*, vol. 14, no. 21, p. 5398, Oct. 2022, doi: 10.3390/rs14215398.

[23] Y. Zheng, C. Yang, and A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography," in *Computational Imaging III*, May 2018, p. 4, doi: 10.1117/12.2304564.

[24] L. Singh, R. R. Janghel, and S. P. Sahu, "An empirical review on evaluating the impact of image segmentation on the classification performance for skin lesion detection," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 40, no. 2, pp. 190–201, Mar. 2023, doi: 10.1080/02564602.2022.2068681.

[25] F. Chollet, "Deep learning with Python," *Manning Publication Co*, 2017. https://www.manning.com/books/deep-learning-with-python.
[26] S. Shedriko and M. Firdaus, "Comparison of adagrad, adadelta, and adam optimizers in image classification using deep learning (in Indonesian)," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 8, no. 1, p. 103, Aug. 2023, doi: 10.30998/string.v8i1.16564.

BIOGRAPHIES OF AUTHORS

Firdaus Angga Dewangga received his first degree from Bakrie University, Computer Engineering, Jakarta, in 2017. He is currently a senior IT staff in Crowe Indonesia. His responsibilities in this role encompass a wide range of tasks related to IT and application development. He continues to excel in this position, bringing his skills and expertise to contribute to the industry. He can be contacted at email: Firdaus.dewangga@binus.ac.id.

Abba Suganda Girsang is currently lecturer at master information technology at Bina Nusantara University, Jakarta. He obtained Ph.D. degree in the Institute of Computer and Communication Engineering, Department of Electrical Engineering and National Cheng Kung University, Tainan, Taiwan, in 2014. He graduated bachelor from the Department of Electrical Engineering, Gadjah Mada University (UGM), Yogyakarta Indonesia, in 2000. He then continued his master degree in the Department of Computer Science in the same university in 2006–2008. He was a staff consultant programmer in Bethesda Hospital, Yogyakarta, in 2001 and also worked as a web developer in 2002–2003. He then joined the faculty of Department of Informatics Engineering in Janabadra University as a lecturer in 2003-2015. He also taught some subjects at some universities in 2006–2008. His research interests include swarm intelligence, combinatorial optimization, and decision support system. He can be contacted at email: agirsang@binus.edu.