International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Comparison of Car Parking Space Using Pre-Trained Models and Computer Vision Technique

Abdul Haris Rangkuti*, Albert Enrico, Andros Clarence Chen, Leonardo, Stanley Wisely

Submitted: 25/04/2023 **Accepted :** 27/06/2023 **Accepted:** 08/07/2023

Abstract*:* Most public areas have parking areas to make it easier for customers to park their cars. With advances and developments in technology, parking lots can smartly solve the problem of finding a parking space that is still empty or cannot be filled. To find out which parking lots are empty or already filled, an algorithm or method is needed and with the help of Computer Vision techniques. In this study, some data were collected and used several types of trained models to find the most accurate results. The results of the research are in two scenarios, namely the parking lots are still empty and those have been filled them. During with empty and filled scenarios where the parking space is empty or has been filled with cars conditions. Based on experiments using the Convolutional neural network (CNN) model, an average percentage of accuracy, precision, recall, and F1 Score are above 90%. Meanwhile, there are several pre-training models in occupied parking lot scenarios such as VGG19, Densenet121, Resnet V2 50 and EfficientNetB7 which can be used for development purposes due to their excellent accuracy. The research results contribute to a better understanding of selecting accurate pre-trained model results for use in development and turnkey applications.

Keywords**:** *Parking areas, cars, CNN, pre-training models, accuracy, precision, recall, F1 Score*

1. Introduction

Parking is a traffic management problem that is still being solved. Parking itself means the vehicle stops / or in the sense that the vehicle is not moving stop for a while. Parking is becoming a major problem due to increasing traffic and volume vehicles resulting in increased demand for parking space demand for the area certain areas, such as business areas / or areas that have activities. So parking becomes a big problem for DKI Jakarta which until now has not been resolved [1]. Nowadays, the parking problem has become one of the major issues in urban cities since there are limited parking spots [8]. Since the rapidly increasing number of cars, finding empty spots makes it more of a significant issue [17].

Most of the public spaces in this world as well as in Indonesia have parking spaces for their customers to park their motorcycle or car. Parking spaces have different layouts. There are small and large parking spaces as can be seen in Figure 1. Small parking spaces are often found at pharmacies or minimarkets. While large parking spaces can be found at shopping malls or vacation spots. In a small parking space, it is easy for customers to know whether there are available parking spots or not. On the contrary, it is hard for customers to know whether there are available parking spots or not in a large parking space [9]. Other than that, bad management of parking space can be seen from the unused parking spots. This absolutely causes an ineffectiveness of the parking area [20]. This problem needs to be solved because it has led to substantial traffic causing air pollution, time and energy waste, and even the fuel itself. Therefore, innovative solutions and smart parking systems should be implemented for more sustainable future cities [10]. In addition, the advancement of technology provides opportunities in making parking systems intelligent with computers and sensors [18].

** Corresponding Author Email: rangku2000@binus.ac.id*

Fig 1. Drone's view of a large parking space courtesy of PKLot Dataset [2]

We proposed a solution for detecting available parking spaces by using Computer Vision technique which is in the field of Artificial Intelligence with Computer Vision, data can be collected from the environment and complete specified tasks [16]. The data itself will be granted when a car enters a parking space. It will recognize vehicle number, which track the car is parked in, and update parking space information [19]. We decided to use a TensorFlow to detect cars from a top-down view at a parking space since TensorFlow provides us the utilities to build our own model. With this model, we can conclude how many empty and occupied parking spots and where those empty parking spots are using additional resources if needed.

TensorFlow is a product of the Google Brain team under Google. It is an open-source software library to create machine learning models for desktop, mobile, web, and cloud. Such a model is created by performing data training and inference of deep neural networks. Some examples of applications which use the TensorFlow model are face recognition, object recognition, optical character recognition (OCR), and many others [2].

The objectives of the research are:

A. To create and develop a reliable model for doing car parking space classification, occupied or empty.

B. To determine which pre-trained model is best to use for car parking space classification on a training dataset.

This research is focused on detecting a car in a parking spot inside a parking space. It is done by creating a TensorFlow model that is able to classify an image of a parking spot. Such a model can then be used in a parking space application to provide information if a parking spot is occupied or empty.

The output of this research are:

A. To carry out a model that can be implemented to an external application, such as a car parking space application.

B. To help customers cars on large parking spaces to easily and quickly find available parking spots.

2. Literature Review

Along with the increasing difficulty of finding a parking space, more and more smart parking system developments are being carried out to find efficient solutions [6]. There are various solutions that can be done, such as there is a study to provide an empty parking slot via SMS notifications. This research uses the Internet of Things to implement a parking management system, it uses sensors like IR sensors to collect data which will later be analyzed and processed to produce output that can minimize fuel use when looking for a parking space [3]. Other studies also use the Internet of Things method using sensors to track available parking space and show it on an Android application, both of the studies use sensors to collect data wirelessly [4]. It shows that in order to detect or track an empty parking slot there are needed something to detect it like a sensor. But IoT also has a drawback such as it's costly to install sensors in each parking slot [7]. In other studies, the input from sensors is replaced by computer vision or machine vision. Computer vision uses machine learning algorithms to detect empty parking spaces from aerial images or videos of parking lots. The system reports each parking slot, whether it's occupied or not. We read studies that review other computer vision for smart parking systems [7]. One of them is using Convolutional Neural Networks. These experimental results based on the training set show how robust the system can be when the prediction has to take place in a different [5]. Computer vision can be connected with IoT systems, for example using Haar cascade algorithm to detect vehicles with a Raspberry Pi camera. Then the system sends the data through a Wi-Fi module to a Firebase cloud platform. [7, 11]

3. Materials and Methods

To achieve the purpose of this research, we used Google Colab development environment with GPU runtime, and to decide using the Python 3.8.16 as the base programming language with several libraries, such as TensorFlow 2.9.2, Keras 2.9.0, scikit-learn 1.0.2, and a couple more utility libraries.

Based on a similar previous work [12], our research is divided into 5 main steps for the result of performing parking spot classification, which are:

- Dataset collection and image selection.
- Preprocessing input images.
- Creating and training the model.
- Comparison of pre-trained models.
- Classification of images.

For better visualization, all of the steps for image classification and choosing the best pre-trained model can be seen in Figure 2.

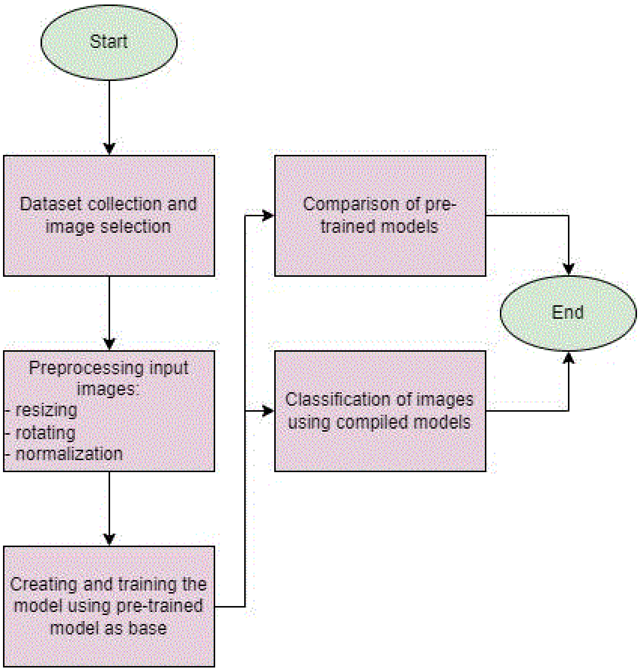

Fig 2. The stages of car parking space image classification

3.1. Dataset Collection and Image Selection

For our first step, we performed a dataset collection. In this research, we used the already available PKLot dataset. While it provides both the full image of parking spaces and each individual parking spot, we decided to only use the images for each individual parking spot (it is named PKLotSegmented). The dataset we used provides 81.406 training images and 42.384 testing images of parking spots. Training images are collected from the UFPR04 and UFPR05 dataset from 3 weather conditions, sunny, rainy, and cloudy. Testing images are collected from the PUC dataset from 3 weather conditions as well, sunny, rainy, and cloudy. This way of data collection is done to prevent the model from using the same image for testing and training which may alter the testing results.

In Figure 3 illustrated in creating the CNN model which decided to use 5500 images, 5000 training images and 500 testing images from the dataset for saving time and resources. All of the collected images have a file extension of *.jpg with varying resolutions. Occupied parking space sample images can be seen in Figure 4. On the other hand, empty parking space sample images can be seen in Figure 5. The training and testing images have been classified beforehand into 2 different classes, occupied and empty.

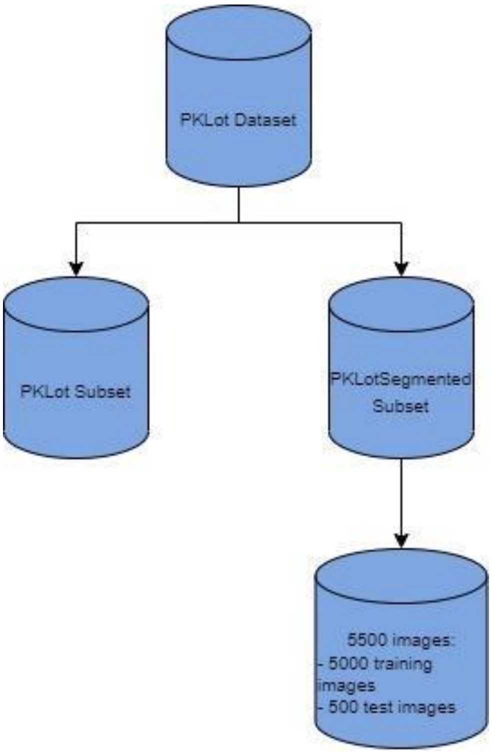

Fig 3. Dataset collection and image selection

Fig 4. Occupied parking spaces

Fig 5. Empty parking spaces

3.2. Preprocessing Input Images

Every image from the dataset is resized into the resolution of width 37 pixels and height 49 pixels. All of the training images are then rotated with a range of 90 degrees, horizontally flipped, vertically flipped, and apply the preprocessing function from TensorFlow ResNet 50 (preprocess input) [13]. This is done to provide various orientation of the parking spot in the image and prevent skewing interference. All of the pixels in the testing images are converted from the range of [0, 255] to [0, 1]. This is done for normalization. In general, the preprocessing images can be seen in figure 6.

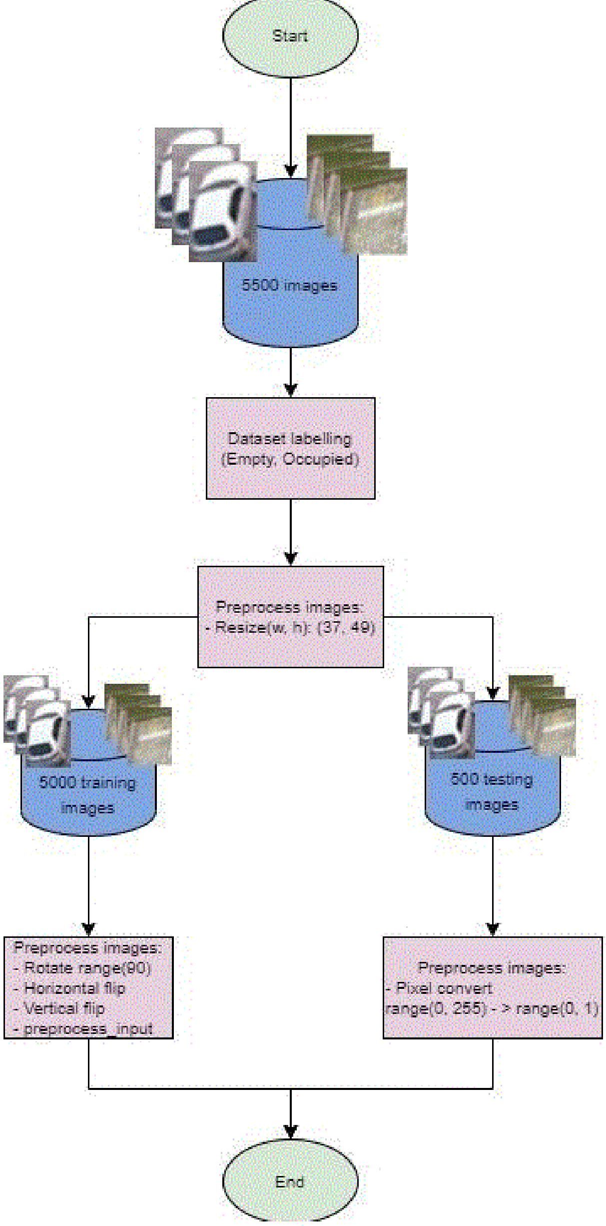

Fig 6. Preprocessing the training and testing images

In figure 6.0 explains the initial process starting from equalizing the size of each image (resize) including making labels on 5000 images as training images and 500 images as test images. In general, all images experience a rotational change of 90 degrees both horizontally and vertically.

3.3. Creating and Training the Model

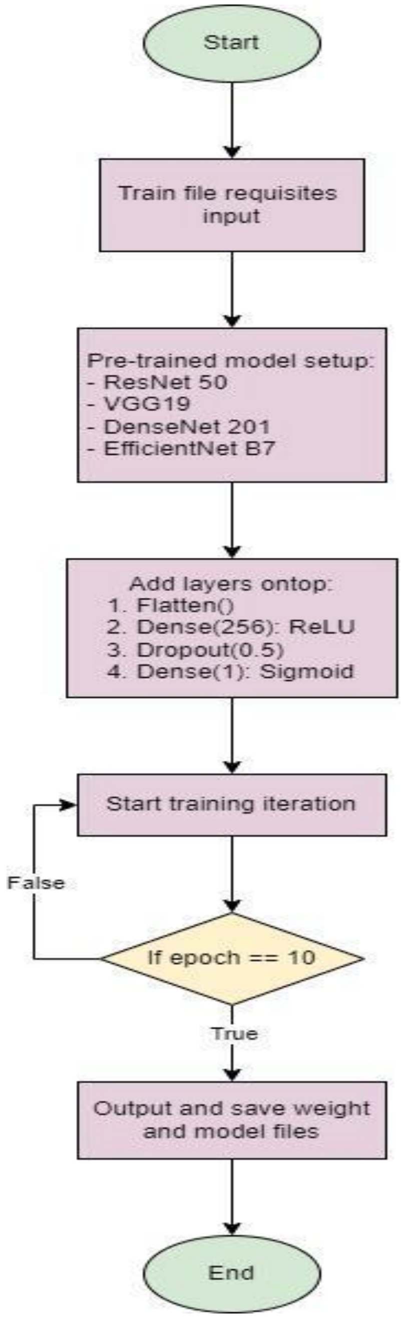

Fig 7. Training stages in parking space recognition

Figure 7 is the stages of creating the model. We initially used several pre-trained models. To get the optimal accuracy this study used 4 different pre-trained models, ResNetV2 50, VGG19, DenseNet201, and EfficientNetB7. Those models are then locked to prevent the layers from getting trained and altered by our dataset. On top of the pre-trained model, we also added several layers. First, we flatten the layers to reduce dimensionality. Then, we created a dense Rectified Linear Unit, ReLU (function can be seen in Figure 8) activation layer with 256 neurons. On top of that, we used a dropout layer with 50% activation to reduce model overfitting. Finally, we ended with another dense sigmoid (function can be seen in Figure 9) activation layer with 1 neuron for the final output of the model. Easier visualization of the model layers can be seen in Figure 10.

This TensorFlow model training starts with compiling the model with SGD optimizer that uses a learning rate of 0.00001 and momentum of 0.9. It also uses the binary accuracy metric to measure its accuracy with the training set. We only used 10 epochs for model training to save time and prevent overfitting.

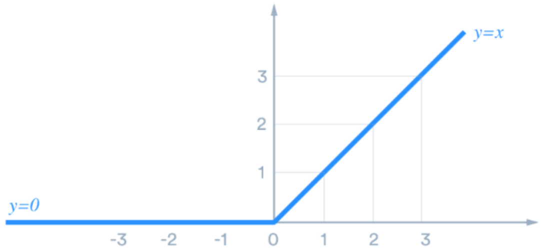

Fig 8. Rectified Linear Unit (ReLU) function graph

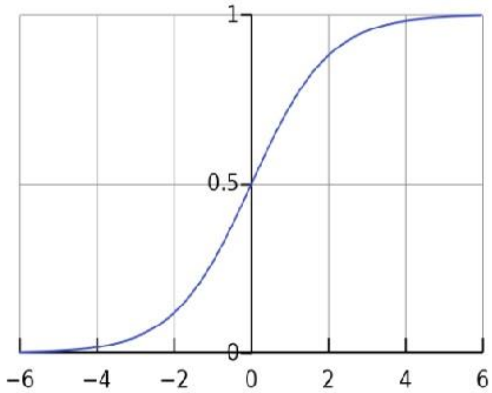

Fig 9. Sigmoid function graph

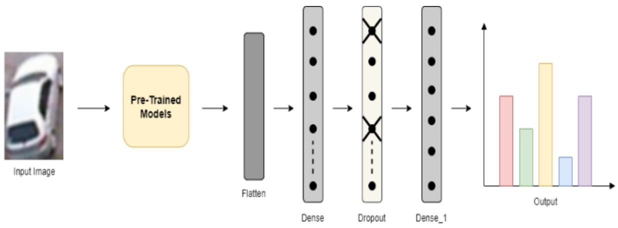

Fig 10. Layer Arrangement in Model

3.4. Comparison of Pre-trained Models

For each created model, we used several metrics to determine which model is best. The most deciding factor which we used is accuracy using the built-in argument from TensorFlow. Additionally, we also used metrics such as, precision [14][15], recall [14][15], and F1-score [15] with Eq. (1), Eq. (2), and Eq. (3) respectively. They are calculated by using the classification_report function provided by the scikit-learn library. More information on the comparisons can be found in the chapter, Result and Discussion.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Where:

- True Positive= total of positive samples that the model correctly classified
- False Positive = total of negative samples that the model mistakenly classified as positive samples
- False Negative = total of positive samples that the model could not classified

3.5. Classification of Images

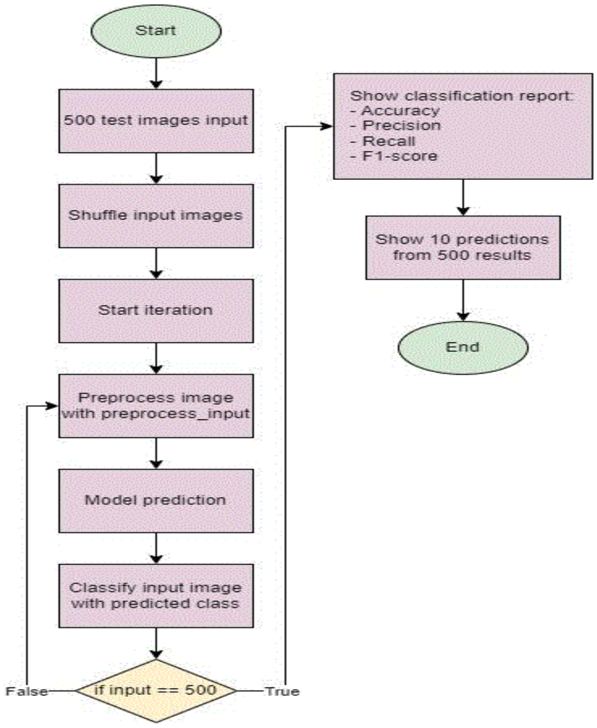

Fig 11. Testing stages for the 500 testing images

Figure 11 is the testing stages for the 500 testing images created earlier from the PKLot Segmented dataset. Every testing image is shuffled and preprocessed by the preprocessing input function from TensorFlow. Later in the result chapter of this research, we also used our own testing images outside of the PKLotSegmented dataset. There are 9 testing images as can be seen in Figure 12.

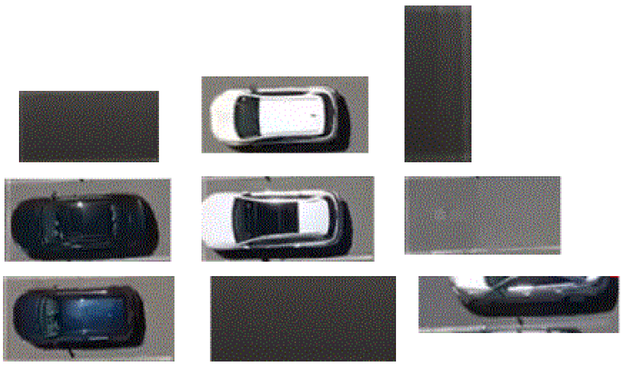

Fig 12 As many testing images outside of PKLotSegmented dataset

In Figure 12 inform after creating and compiling the model, it is then used to make predictions on some test images. It is then used to classify test images into 2 classes, occupied and empty. If the model's prediction value is larger than 98% or 0.98 then the image is classified as occupied. Otherwise, the image is classified as empty. The input test images were divided into 2 folders, occupied and empty to classify their correct classes. The model then predicts all of the test images from the 2 folders and classifies them according to the prediction values.

Table I and Table II provides a classification example of the 500 testing images. The filename has a pattern of /<correct-class>/<file-name>.jpg. Each file is then run through the model which provides a class prediction with their respective prediction value.

TABLE I. CLASSIFICATION EXAMPLE WITH DENSENET201

Data No.	Filename (correct class)	Class (predicted)	Predictions (decimals)
1	/Occupied/*.jpg	Occupied	0.99746156
2	/Empty/*.jpg	Empty	0.06568048
3	/Occupied/*.jpg	Empty	0.9066474
4	/Occupied/*.jpg	Empty	0.5285996
5	/Occupied/*.jpg	Empty	0.90957785
…	…	…	…
495	/Empty/*.jpg	Empty	0.5098256
496	/Occupied/*.jpg	Empty	0.9551314
497	/Empty/*.jpg	Empty	0.037587404
498	/Empty/*.jpg	Empty	0.024425676
499	/Empty/*.jpg	Empty	0.63913816

TABLE II. CLASSIFICATION EXAMPLE WITH VGG19

Data No.	Filename (correct class)	Class (predicted)	Predictions (decimals)
1	/Empty/*.jpg	Empty	0.051002555
2	/Empty/*.jpg	Empty	3.8244867e-05
3	/Occupied/*.jpg	Occupied	0.9999976
4	/Occupied/*.jpg	Occupied	0.9999964
5	/Occupied/*.jpg	Occupied	1.0
…	…	…	…
495	/Occupied/*.jpg	Occupied	0.9999999
496	/Empty/*.jpg	Empty	1.108218e-06
497	/Empty/*.jpg	Empty	0.0011742158
498	/Empty/*.jpg	Empty	0.0010310786
499	/Occupied/*.jpg	Occupied	0.99999964

4. Result and Discussion

We have tested several pre-training models to check which is the best pre-training model in terms of accuracy, recall, precision, and F1-score.

4.1. Empty Scenario

The condition of the parking lot is divided into two, namely occupied and empty. In the empty section the system has the highest percentage in the recall section where the results of the three pre-trained models are the same above 90%.

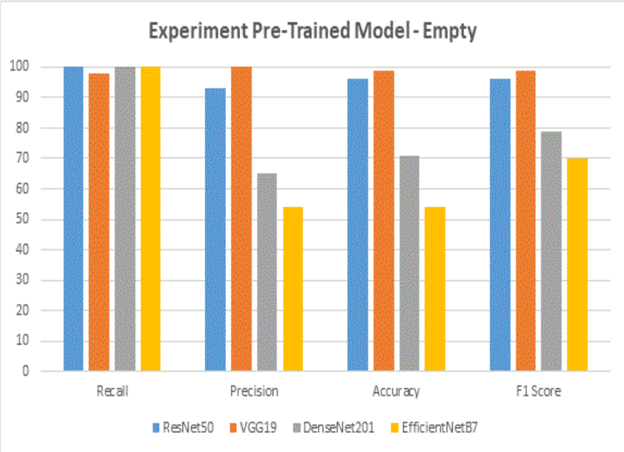

Fig 8. Chart Pre-Trained Model - Empty

4.2. Occupied Scenario

In the occupied section the system has the highest percentage in the precision section where the results of the three pre-trained models are the same above 90% and some almost 100%.

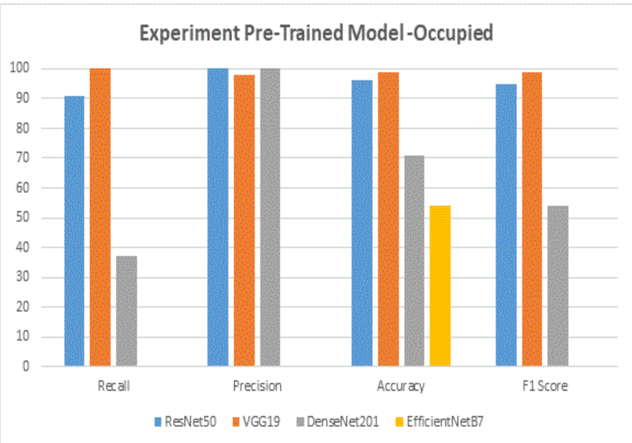

Fig 9. Chart Pre-Trained Model – Occupied

A. ResNet50 Model

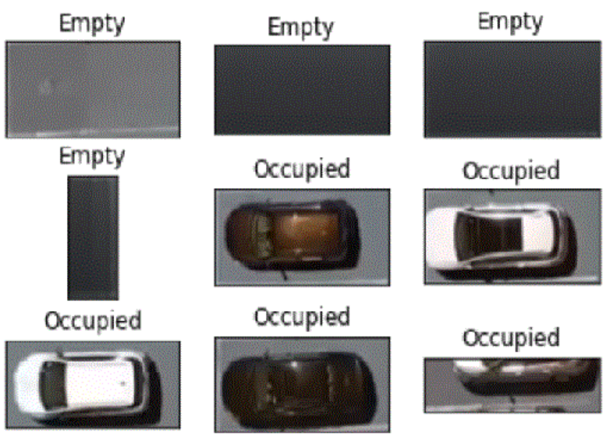

Fig 10. ResNet50 pre-trained model results

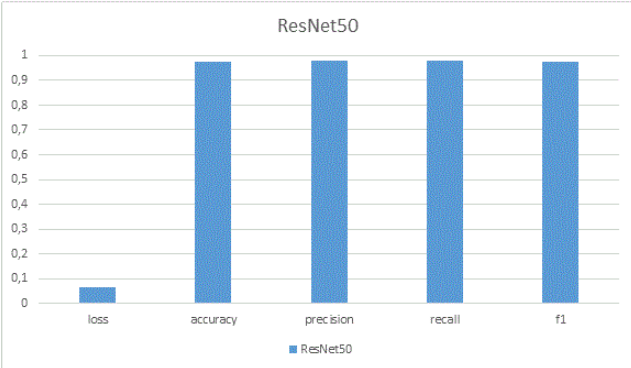

Fig 11. ResNet50 pre-trained model epoch results

Figure 11 illustrates the results of the experiment using the ResNetV2 50 pre-training model. Based on this experiment, are illustrated that the results of starting from accuracy, Recall, Precision and F1 have an average accuracy above more than 90%, with a loss of less than 5%. So, it is illustrated by using the Resnet V2 50 model that has very good performance. In addition, for images with a plain background, which is a parking space, it is identified as an empty class. Conversely, images with a top view of the car are identified as class occupied in either full or semi-view. This model illustrates very good results for determining the condition of empty and occupied parking lots

B. VGG19

Fig 12. VGG19 pre-trained model results

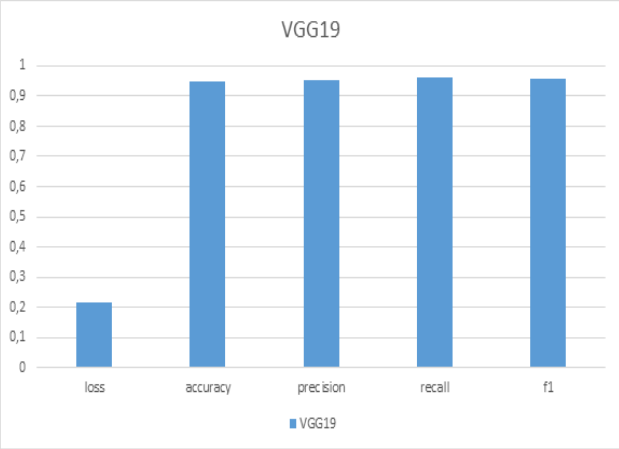

Fig 13. VGG19 pre-trained model epoch results

Figure 13 illustrates the results of the experiment using the VGG 19 pre-training model. Based on this experiment, are illustrated that the results of starting from accuracy, Recall, Precision and F1 have an average accuracy above more than 92%, with a loss of morethan 20%. So, it is illustrated by using the VGG 19 model has very good performance. In addition, for images with a plain background, which is a parking space, it is identified as an empty class. Conversely, images with a top view of the car are identified as class occupied in either full or semi-view. This model illustrates very good results for determining the condition of empty and occupied parking lots

C. DenseNet201

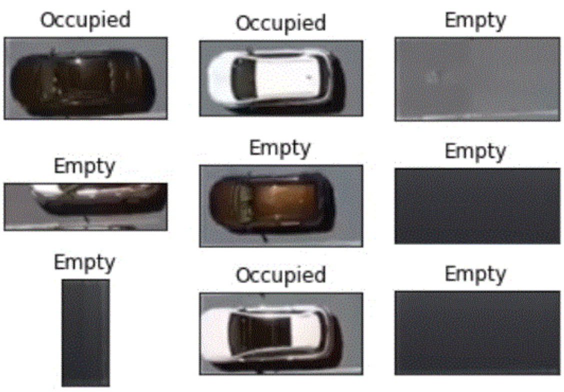

Fig 14. DenseNet201 pre-trained model results

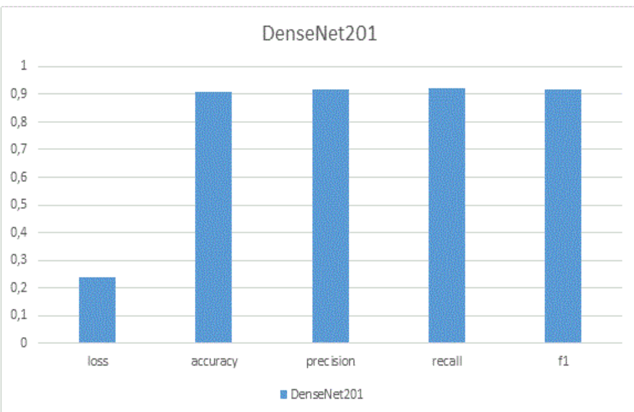

Fig 15. DenseNet201 pre-trained model epoch results

Figure 14 illustrates the results of the experiment using the DenseNet201 pre-training model. Based on this experiment, are illustrated that the results of accuracy starting from accuracy, Recall, Precision and F1 have an average accuracy above more than 90%, with a loss of more than 20%. So, it is illustrated by using the DenseNet201 model has good performance only. In addition, for images with a plain background, which is a parking space, it is identified as an empty class. Conversely, images with a top view of the car are identified as class occupied in either full or semi-view. This model illustrates good results for determining the condition of empty and occupied parking lots

D. EfficientNetB7

Fig 16. EfficientNetB7 pre-trained model results

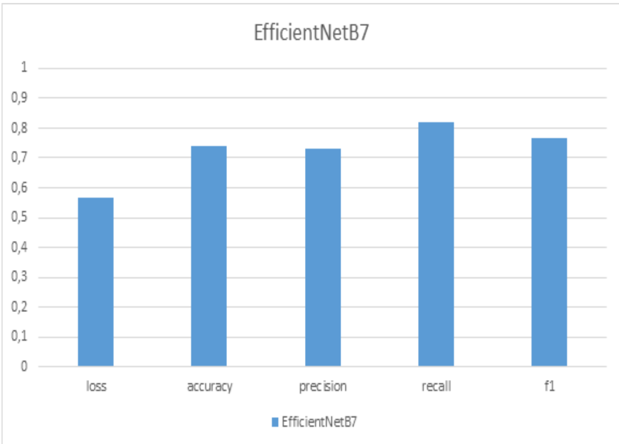

Fig 17. EfficientNetB7 pre-trained epoch model results

Figure 17 illustrates the results of the experiment using the EfficientNetB7 pre-training model. Based on this experiment, are illustrated that the results of starting from accuracy, Recall, Precision and F1 Score have an average accuracy between 70% - 80%, with a loss of more than 55%. So, it is illustrated by using the EfficentNetB7 model has good enough performance only. In addition, for images with a plain background, which is a parking space, it is identified as an empty class. Conversely, images with a top view of the car are identified as class occupied in either full or semi-view. This model illustrates good enough results for determining the condition of empty and occupied parking lots

5. Conclusion

- In order to identify empty or occupied parking spaces in large areas, we proposed a solution using Computer Vision and the help of TensorFlow to build our own model. For the research itself, we used Google Colab development environment with GPU runtime. Then, we used Python 3.8.16 as the base programming language with several libraries such as TensorFlow 2.9.2, Keras 2.9.0, scikit-learn 1.0.2, and other utility libraries.

- Our research is divided into 5 main steps for the result of performing parking spot classification from dataset collection and image selection, preprocessing input images, creating and training model, comparison of pre-trained models, until the last one is classification of images. In the last step, predictions are made on some test images to classify test images into 2 classes which are occupied and empty. If the model's prediction value is larger than 98% or 0.98 then the image is classified as occupied. Otherwise, the image is classified as empty.

- We have also tested several pre-training models to check which is the best pre-training model in terms of accuracy, recall, precision, and F1-score. With the empty scenario where the condition of the parking lot is empty, the system has the highest percentage in the recall section where the results of the three pre-trained models are the same above 90%. Meanwhile, with the occupied scenario, we conclude that ResNet50 and VGG19 can be used as an accurate model for application development or other purposes. It is because both of the pre-trained models can distinguish the empty and occupied scenario.

Acknowledgements

This work was supported by Binus University. The authors wish to thank Abdul Haris Rangkuti as the overall supervisor who provided insight and expertise that greatly assisted the research.

Author contributions

Abdul Haris Rangkuti: Supervisor of Software System and Paper Review. **Albert Enrico:** Methodology and Software. **Andros Clarence Chen:** Methodology and Software. **Leonardo:** Visualization, Investigation, Writing-Reviewing and Editing. **Stanley Wisely:** Writing-Reviewing and Editing.

Conflict of interest

The submitted research was done by Albert Enrico, Andros Clarence Chen, Leonardo, and Stanley Wisely as a student in Bina Nusantara University. The research is supervised under Abdul Haris Rangkuti as part of a research project in Bina Nusantara University. Every author has agreed to accept all the terms and conditions in conducting this research. Hence, the authors declare no conflict of interest.

References

[1] Sandra, C. F. K. (2020). KEBIJAKAN STRATEGI PARKIR (Studi Kasus: Ibu Kota Metropolitan Jakarta). In Prosiding Seminar Intelektual Muda (Vol. 2, No. 1).

[2] Fedorak, Bohdan. "Parking Lot Dataset." Kaggle, 2019. Accessed 6 January 2023.

[3] TensorFlow. "TensorFlow." https://www.tensorflow.org. Accessed 6 January 2023.

[4] R., E., Seth, J., Ashritha, P., Namith, R., "Smart Parking System using IoT", *International Journal of Engineering and Advanced Technology (IJEAT)* ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019.

[5] Elsonbaty, A. A., Shams, M. Y.., "The Smart Parking Management System", *International Journal of Computer Science & Information Technology (IJCSIT)* Vol 12, No 4, August 2020.

[6] Nyambal, J., Klein, R., "Automated parking space detection using convolutional neural networks," 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), 2017.

[7] Shawly, T., Alsheikhy, A., Said, Y., Lahza, H., "An Effective Approach for Smart Parking Management", *Ingénierie des Systèmes d'Information* Vol. 27, No. 5, October, 2022, pp. 783-789.

[8] Abidin, M. Z., Pulungan, R., "A Systematic Review of Machine-Vision-Based Smart Parking Systems", *Scientific Journal of Informatics* Vol. 7, No. 2, November 2020.

[9] Shin J.-H., Jun H.-B.A study on smart parking guidance algorithm Transportation Research Part C: Emerging Technologies 201444299-317

[10] Janak Parmar, Pritikana Das, Sanjaykumar M. Dave, "Study on demand and characteristics of parking systems in urban areas", 2014

[11] Al-Turjman, Fadi & Malekloo, Arman. Smart Parking in IoT-enabled Cities: A Survey. Sustainable Cities and Society. 49. 101608. 10.1016/j.scs. 2019.

[12] Hwang, K., Lee J, M., Jung, I, H. Smart Parking Management System Using AI, *Webology* Vol. 19, No 1, January 2022.

[13] Rangkuti, Abdul H. "Content based batik image classification using wavelet transform and fuzzy neural network." *Journal of Computer Science* Vol. 10, No. 4, 2013, pp. 604-613.

[14] TensorFlow. "tf.keras.applications.resnet50.preprocess_input." https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/preprocess_input. Accessed 24 January 2023.

[15] Olson, David L., and Dursun Delen. "Advanced Data Mining Techniques." *Springer* Vol. 1, 2008, p. 138. ISBN 3-540-76916-1.

[16] Taha, Abdel A., and Allan Hanbury. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool." *BMC Medical Imaging* Vol. 15, No. 29, 2015, pp. 1-28. doi:10.1186/s12880-015-0068-x. PMC 4533825. PMID 26263899.

[17] Namdari, Himan & Nanda, Devak & Yuan, Xiaohui. Progress in Computer Vision: Object Recognition. 2022. 10.1007/978-3-030-84729-6_5

[18] Taylor, Onate & Sochima, Ezekiel & Emmah, Victor. (2021). Smart Vehicle Parking System Using Computer Vision and Internet of Things (IoT). *European Journal of Information Technologies and Computer Science*. 1. 11-16. 10.24018/compute.2021.1.2.12.

[19] Li, Yongyi & Mao, Hongye & Yang, Wei & Guo, Shuang & Zhang, Xiaorui.. Research on Parking Space Status Recognition Method Based on Computer Vision. Sustainability. 2022. 15. 107. 10.3390/su15010107.

[20] Jung, In & Lee, Jae & Hwang, Kitae. Smart Parking Management System Using AI. *Webology* 2022. 19. 4629-4638. 10.14704/WEB/V19I1/WEB19307.

[21] Bibi, Nazia & Majid, Muhammad & Dawood, Hassan & Guo, Ping. (2017). Automatic Parking Space Detection System. 10.1109/ICMIP.2017.4.

[22] Patil, A. ., & Govindaraj, S. K. . (2023). Enhanced Deep Learning Models for Efficient Stroke Detection Using MRI Brain Imagery. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 191–198. https://doi.org/10.17762/ijritcc.v11i3.6335

[23] Morzelona, R. (2021). Human Visual System Quality Assessment in The Images Using the IQA Model Integrated with Automated Machine Learning Model . Machine Learning Applications in Engineering Education and Management, 1(1), 13–18. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/5

Biographies

Abdul Haris Rangkuti is a lecturer and a researcher at the School of Computer Science in Bina Nusantara University. He received the B.S. degree in Management Informatics and the M.Sc and Ph.D Degree in Computer Science both from Institut Pertanian Bogor and University of Gadjah Mada in 1996, 2006 and 2020. His research interests are multimedia processing (image, audio, video processing), computer vision, and medical disease recognition. He was involved in international projects such as ADB and Word Bank as system analyst and team leader. He carried out some projects from department/minister as Analyst system or Team Leader.

Albert Enrico was born in Bandung, West Java, Indonesia on 23rd October 2002. He is currently studying in Bina Nusantara University (Bandung, West Java, Indonesia) for his Honors, Bachelor of Science in computer science, completed in 2024. His research interests are image processing and machine learning. While his expertise includes and is not limited to android development, website development, and Internet of Things.

Andros Clarence Chen was born in Sukabumi, West Java, Indonesia on 6th January 2003. He is currently studying in Bina Nusantara University (Bandung, West Java, Indonesia) for his Honors, Bachelor of Science in computer science, completed in 2024. His research interests are machine learning and Internet of Things. While his expertise includes and is not limited to competitive programming, android development.

Leonardo was born in Bandung, West Java, Indonesia on 26th June 2002. He is currently studying in Bina Nusantara University (Bandung, West Java, Indonesia) for his Honors, Bachelor of Science in computer science, completed in 2024. His research interests are machine learning and Internet of Things. While his expertise includes and is not limited to website development especially for frontend, mobile development, and UI/UX.

Stanley Wisely was born in Bandung, West Java, Indonesia on 25th July 2002. He is currently studying in Bina Nusantara University (Bandung, West Java, Indonesia) for his Honors, Bachelor of Science in computer science, completed in 2024. His research interests are embedded systems and Internet of Things. While his expertise includes and is not limited to android development and website development.