JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 2, Juni 2023, Pp. 656-663

IMPROVED REAL-TIME HOUSE FIRE DETECTION SYSTEM PERFORMANCE WITH IMAGE CLASSIFICATION USING MOBILENETV2 MODEL

Muhammad Faris*[1], Endro Ariyanto[2], Yogi Anggun Saloko Yudo[3]

1. Universitas Telkom, Indonesia
2. Universitas Telkom, Indonesia
3. Universitas Telkom, Indonesia

Article Info

Kata Kunci: klasifikasi, citra, deteksi, kebakaran, mobilenetv2

Keywords: classification, image, detection, fire, mobilenetv2

Article history:
Received 27 January 2023
Revised 3 February 2023
Accepted 1 March 2023
Available online 1 June 2023

DOI :
https://doi.org/10.29100/jipi.v8i2.3803

* Corresponding author.
Corresponding Author
E-mail address:
muhammadfarisld@student.telkomuniversity.ac.id

ABSTRAK

Permasalahan pada sistem pendeteksi kebakaran berbasis mikrokontroler Ardunio dengan sensor api dan asap adalah jarak deteksi. Misalnya pada penelitian lain disebutkan bahwa jarak maksimal deteksi kebakaran pada dua buah kertas yang dibakar adalah 140 cm. Ini berarti jika titik kebakaran berada pa-da jarak lebih jauh maka sistem tidak dapat mendeteksi adanya kebakaran secara dini, tentu hal ini akan bermasalah jika digunakan pada ruangan yang lebih luas. Berdasarkan permasalahan tersebut, dibutuh-kan sistem yang dapat mendeteksi kebakaran pada ruangan yang luas. Sebuah metode yang dapat digunakan adalah deteksi menggunakan klasifikasi citra. MobileNetV2 adalah salah satu model untuk klasifikasi atau mendeteksi sebuah objek secara real time pada suatu citra. Pada penelitian ini model dibangun menggunakan Edge Impulse berdasarkan library TensorFlow dan Keras. Sistem akan menggunakan laptop dengan GPU Nvidia GeForce MX130, kamera smartphone beresolusi 48MP, dan library OpenCV untuk proses klasifikasi citra, serta Telegram untuk mengirimkan notifikasi kebakaran melalui library Requests. Hasil pengujian yang didapatkan pada ban motor berukuran 80/90 yang diba-kar, jarak deteksi terjauh yang paling optimal adalah 7 meter dengan akurasi 99,91%. Sedangkan pen-gujian pada dua lembar kertas yang dibakar, jarak deteksi terjauh yang paling optimal adalah 3 meter dengan akurasi 99,75%. Response time rata-rata yang didapat sangat bervariasi mulai dari 74.5 ms hingga 117.1 ms, yang mana tergantung pada koneksi jaringan internet.

ABSTRACT

The problem with the Ardunio microcontroller-based fire detection system with fire and smoke sensors is the detection distance. For example, in another research, it was stated that the maximum distance for fire detection on two pieces of paper that were burned was 140 cm. This means that if the fire point is at a far-ther distance, the system cannot detect a fire early, of course, this will be problematic if used in a wider room. Based on these problems, a system is needed that can detect fires in large rooms. A method that can be used is detection using image classification. MobileNetV2 is a real-time model for classifying or detect-ing an object in an image. In this study, the model was built using real-time based on the TensorFlow and Keras libraries. The system will use a laptop with an Nvidia GeForce MX130 GPU, a 48MP resolution smartphone camera, and the OpenCV library for the image classification process, as well as Telegram for sending fire notifications via the Re-quests library. The test results obtained on burnt 80/90 motorcycle tires, the most optimal detection distance is 7 meters with an accuracy of 99.91%. While testing on two sheets of paper that are burned, the most optimal detection distance is 3 meters with an accuracy of 99.75%. The average response time obtained varies greatly from 74.5 ms to 117.1 ms, which depends on the internet network connection.

I. INTRODUCTION

Based on fire data on the DKI Jakarta Sectoral Statistics Portal in 2020, out of 1,505 fire cases, the highest cases were house fires [1]. When a fire occurs, people usually only realize when the fire has started to grow. Sometimes fires even occur when people are not at home [2]. The problem with conventional fire detection systems that use fire and smoke sensors is distance detection. For example, in research [3], in this study using an Ardunio microcontroller-based system and fire and smoke sensors, it was stated that the maximum distance between two pieces of paper being burned is 140 cm. This means if the fire point is at a greater distance then the system cannot detect a fire, of course this will be problematic if used in a wider room. Based on these problems, an automation system is needed that can detect fires early in a large room. A method that can be used is fire detection with image classification. There is a model for image classification developed by Google called MobileNetV2. This model has a well-known Convolutional Neural Network architecture which is quite light [4]. Although quite light, this model has comparable accuracy to heavier models.

In research [4], the MobileNetV2 model can detect objects up to a distance of 5 meters, with an average accuracy value of 100%, even at a distance of 25 meters an average accuracy of 89.5% is still obtained. In research [5], the comparison results in mask detection showed that MobileNetV2 had a classification accuracy of 98% and 99% respectively in datasets 1 and 2, while DCNN had a classification accuracy of 97% in both datasets. In research [6], from a small data set, the MobileNetV2 model has a classification accuracy of 96%. In research [7], the MobileNetV2 model obtained up to 85% accuracy outperforming ResNet50V2, InceptionV3, and InceptionResNetV2 in terms of accuracy and efficiency. In research [8], testing on 168 of 192 sets of normal, early, and late stage SSC images were correctly diagnosed by the MobileNetV2 model. The accuracy obtained as a whole reaches 87.5%. These findings indicate that when classifying normal, early, and late SSC skin images, the MobileNetV2 architecture is more precise and effective than CNN. In research [9], the MobileNetV2 model was used for fruit classification, on three different datasets, respectively, this model was able to achieve a stable classification accuracy of 95.75, 96.7 and 96.23%. This is higher than other methods such as Light-CNN, Fruit-CNN, and CNN-Augmentation. In research [10], the accuracy of the classification of MobileNetV2 vehicle types is higher than Alexnet. Alexnet's accuracy is 93.81 and 96.19%, better than the accuracy of the VGGNet mini in the previous study, which was 73%. In research [11], the performance of this deep learning-based pedestrian detection system obtained a maximum confidence level of 90% in daytime imagery conditions, 60.40% of the images can be detected perfectly. For evening imagery conditions, a maximum confidence level of 85% can detect 62.25% of the image perfectly. In research [12], the MobileNetV2 architecture can predict the position of objects both day and night, with up to 52% precision for objects at close or far distances. In research [13], the accuracy of human detection with MobileNetV2 from three different datasets is 98.0%, 82.0%, and 97.00%. The detection results are more accurate than MobileNetV1 pre-trained, which achieved 80.25% accuracy when tested with Pascal VOC2012.

In this research, the system will use the MobileNetV2 model to detect fires in real-time. It is hoped that the use of this model can increase the distance and response time of the fire detection system. The model is built using Edge Impulse based on the TensorFlow and Keras libraries. The system will use a laptop with an Nvidia GeForce MX130 GPU, a smartphone camera with 48MP resolution, and the OpenCV library for the image classification process, as well as Telegram for sending fire notifications via the Requests library. This research will focus on analyzing the accuracy, distance, and response time of the system in detecting fire images using the MobileNetV2 model and comparing it with the Arduino microcontroller-based fire detection system in research [3].

II. GUIDELINES FOR MANUSCRIPT PREPARATION

A. Dataset

In this study, two types of datasets were used with the label "fire" on fire images and "non_fire" on neutral images or non-fire images. There is a dataset for training totaling 3033 images, with 1702 fire images and 1331 neutral images. There is also a dataset for testing totaling 327 images, with 123 fire images and 208 neutral images. The dataset was obtained from the Kaggle website for users named Ahmed Saied [22] and Christofel Ganteng [23]. A few examples of the dataset can be seen in Figures 1 and 2.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 2, Juni 2023, Pp. 656-663

Figure 1. *Fire Dataset*

Figure 2. *Non-fire Dataset*

B. MobileNetV2

A lightweight convolutional neural network also called MobileNetV1 was introduced by Google in 2017 with a focus on mobile and embedded devices. The foundation of the MobileNet architecture is depthwise separable convolution. Standard 2D convolution, which directly processes all input channels to produce one output channel, also involves depth dimensions (channels). With depthwise convolution, the input and filter images are divided into various channels, and each input channel is then concatenated with the corresponding filter channel. These output channels are then stacked back after the filtered output channels have been created. Inseparable depthwise convolution, the stacked output channels are then combined into a single channel by filtering them using a 1x1 convolution, also known as pointwise convolution. Deep separable convolutions produce the same results as conventional convolutions, but are more effective because they use fewer parameters [14].

By adding Inverted Residuals and Linear Bottlenecks, MobileNetV2 has been improved over the base of MobileNetV1. Deep convolution cannot change the number of channels, so the number of features that can be extracted is limited by the number of input channels. This problem is solved with Inverted Residuals. The ReLU activation function with the Inverted Residuals block accelerates learning, reduces gradient dispersion, and improves model stability. MobileNetV2 replaces the ReLU function in the last layer of Inverted Residuals with a linear activation function to reduce information loss because ReLU transformations can result in a significant loss of low-dimensional feature information. The Linear Bottlenecks layer contains inputs and outputs between models, shortcuts between bottlenecks are similar to residual connections on conventional CNNs as they allow for faster training times and better accuracy [15]. How MobileNetV2 works can be seen in Figure 3.

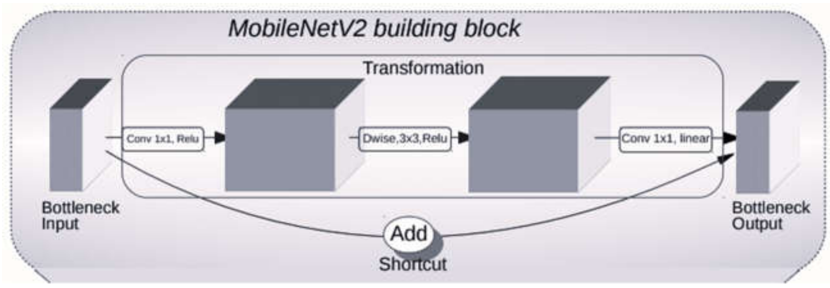

Figure 3. *MobileNetV2*

C. Edge Impulse

Edge Impulse is a state-of-the-art platform designed to deliver machine learning solutions embedded in edge applications. Edge Impulse offers the most practical method to collect data from various mobile platforms, it can also use internal or external sensors. Edge Impulse also provides multiple versions of deep learning models that can be used without a lot of coding skills, and helps with data analysis, model design, and testing. Moreover, it offers options for customized design and model data [16].

D. TensorFlow

TensorFlow is a software library for implementing machine learning algorithms as well as an interface for expressing those algorithms, which is open-source. On a wide variety of systems, including mobile devices such as smartphones and tablets to large-scale distributed systems such as GPU cards, computations expressed using TensorFlow can be executed with little or no modification. This system is adaptable and can be used to express various algorithms, including deep learning inference algorithms. It has been used for research and for implementing machine learning systems in many fields of computer science and other fields, one of which is the Internet of Things [17].

E. Keras

Fully functional deep learning models can be built using Keras, a high-level neural network API written in Python, in just 15 lines of code. It has a larger user base and support and is very easy to use because it is written in Python. Because Keras is very flexible and fast at developing DL models while remaining a high-level API, Keras is easy to use. Consequently, working with Keras is a unique framework. It also adds flexibility as it supports a variety of other frameworks as a back-end, allowing one to use different low-level APIs for different use cases if necessary. The most popular way to use Keras is with TensorFlow as a back end (TensorFlow serves as a low-level back end of Keras high-level Deep Learning API) [18].

F. OpenCV

The free cross-platform software library for computer vision and machine learning is called Open Source Computer Vision (OpenCV). It was first created by Intel in 2000. This library is available under the open-source BSD license and can be used for both business and academic purposes. Currently, around 2,500 optimized algorithms are used for image classification, detecting and recognizing human faces and different objects, classifying human actions in videos, following moving objects, etc [19].

G. Transfer Learning

Using existing models that have been trained to solve related problems is known as transfer learning in deep learning. Instead of starting from scratch, the learning process uses pre-trained models from large-scale data sets with common objects thereby making it faster & more accurate to adjust and adapt to new tasks [20]. Transfer learning takes advantage of the similarities between the source and target domains and uses them as a springboard to accelerate knowledge acquisition in the target domain [21]. It is faster than building a model from scratch, uses less computing power, and allows model training with only a small amount of data [20]. The transfer learning workflow can be seen in Figure 4.

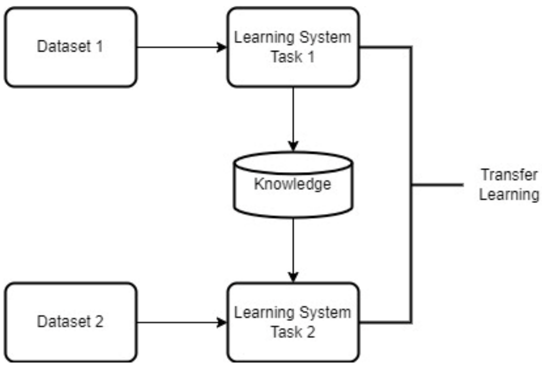

Figure 4. *Transfer Learning*

III. Result and Discussion

A. System Overview

The system can detect fires with image classification using the MobileNetV2 model in real time. The trained model is built via the Edge Impulse platform. The training process uses two types of images, namely fire images which are labeled "fire", and neutral images which are labeled "non_fire". The classification process uses a laptop and the OpenCV library. The system can also send notifications of fires via Telegram, and is equipped with a buzzer for fire alarms. The following features/functionality of the system include :

1. The system is able to detect fire through image classification.
2. The system is able to send fire notifications.
3. The system is able to turn on fire alarm

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 2, Juni 2023, Pp. 656-663

B. System Block Diagram

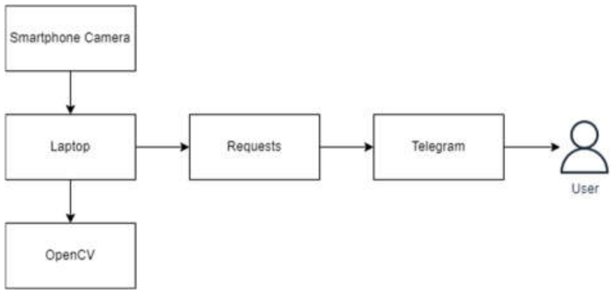

Figure 5. *System Block Diagram*

In Figure 5 it is explained that the system uses a laptop which is used for the image classification process through the OpenCV library, and a smartphone as a camera, which is also the input of the image. Then Telegram sends fire notifications to users via the Requests library.

C. System Workflow

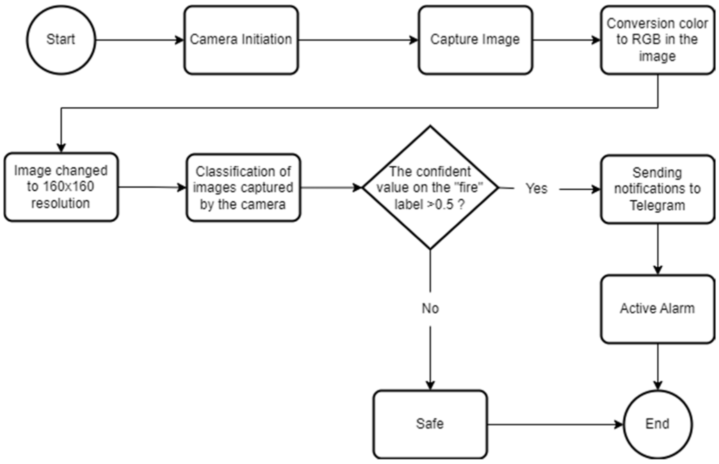

Figure 6. *System Workflow*

In Figure 6 it is explained that first the camera is initiated, then the camera will capture the image. Then do the color conversion to RGB in the image. Furthermore, the image size will be changed according to the image resolution at the time of training, namely 160x160 pixels. Next is to classify the image captured by the camera, if the value of the confident label "fire" is more than 0.5, then a fire notification is sent to Telegram, but if not, then it is considered safe.

D. Functional Testing

1. The system is able to detect fire through image classification.

In the fire detection function, if no fire is detected, the text that comes out is "Aman", and if a fire is detected, the text that comes out is "Kebakaran". The following are the results of testing the fire detection function which can be seen in Figure 7.

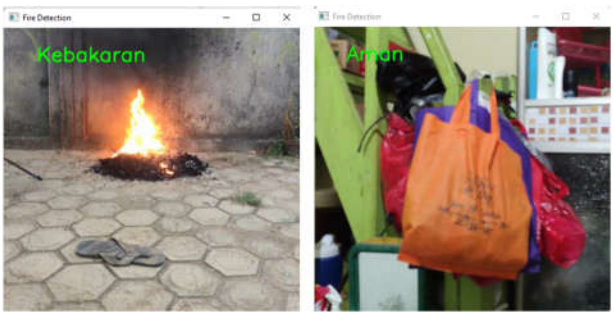

Figure 7. *Fire Detection Test Result*

2. The system is able to send fire notifications.

In sending fire notifications, Telegram bots are used. When the system detects a fire, the system will send a text notification to the user's Telegram. Following are the results of testing the Telegram notification sending function which can be seen in Figure 8.

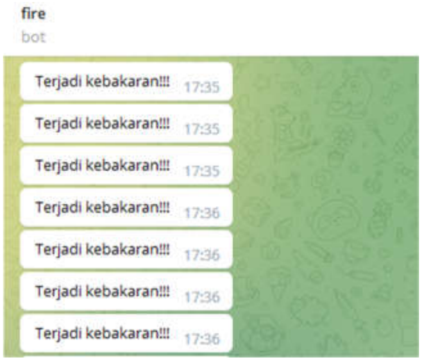

Figure 8. *Fire Notifiaction*

3. The system is able to turn on fire alarm

The fire alarm function uses the Playsound library. When the system detects a fire, an alarm will sound.

E. Fire Detection Test Result

The test was carried out in two different scenarios, the first scenario was carried out by burning two 80/90 motorbike tires. The test was carried out 10 times for each distance from 1 meter to 10 meters from the fire source in multiples of 1 meter. The following are the results of fire detection testing in the first scenario which can be seen in Table I.

Table I. Fire Detection Test Result

Range (meters)	Number of Trials	Average Accuracy	Fire Detected	Number of Telegram Notifications Received
1	10	99,99 %	10	10
2	10	99,99 %	10	10
3	10	99,99 %	10	10
4	10	99,99 %	10	10
5	10	99,91 %	10	10
6	10	99,60 %	10	10
7	10	97,84 %	10	10
8	10	76,55 %	9	9
9	10	57,91 %	2	2
10	10	-	Not Detected	-

The accuracy value is obtained from the confident value multiplied by 100, not from how many times the system has successfully detected fire. In the first scenario, system accuracy from a distance of 1 to 4 meters is consistent at 99.99%, then starts to decrease slowly at a distance of 5 meters to 7 meters. There is a drastic decrease in accuracy at a distance of 8 to 9 meters. At a distance of 10 meters, the system can no longer detect a fire.

In scenario two, a comparison of the detection distance was carried out with research [3]. the test was carried out in the same way as in the study, namely by burning two pieces of paper. The test results show that the fire detection system using image classification with the MobileNetV2 method can detect fire at a maximum distance of up to 300 cm. This is further compared to the Ardunio microcontroller-based fire detection system with fire and smoke sensors in research [3] which is only 140 cm.

Then when compared with other similar studies using MobileNetV2 with different objects, the data used is the most optimal detection distance, which can be seen in Table II.

Table II. Comparison With Other Research

Research	Object	Study Case	Range (meters)	Average Accuracy
	Fire	Fire detection	8	76.55%
Research [4]	Human	Human detection	25	89.5%
Research [13]	Human	Human body shape classification	2	82%

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 2, Juni 2023, Pp. 656-663

F. Fire Detection Fault Test Results

This test is carried out to find out errors in the fire detection system when it detects objects or objects that are similar in color to fire. The test was carried out 10 times for each object, there were 3 objects tested. The following are the results of the fire detection error test which can be seen in Table III.

Table III. Fire Detection Fault Test Results

Object	Number of Trials	Fire Detected
Tote Bag	10	0
Long pants	10	0
Cutlery Rack	10	0

Of the 10 tests on three orange colored objects, none of the objects were detected as fires. An example of an object shape can be seen in Figure 9.

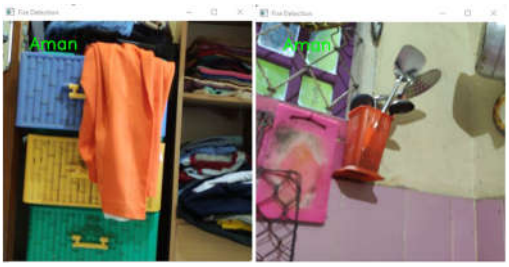

Figure 9. *Fire Detection Fault Test*

G. Response Time Test Results

In this test, what is considered is the response time of the system, starting from performing image classification to sending notifications to Telegram. The test is carried out simultaneously with the fire detection test. Because at a distance of 10 meters, the system cannot detect a fire, data at that distance is not entered. The followings are the results of the response time test and comparison with conventional fire detection systems which can be seen in Table IV.

Table IV. Response Time Test Results

Range (meters)	Number of Trials	Average Response Time (millisecond)
1	10	101
2	10	101.2
3	10	100.2
4	10	94.1
5	10	97.4
6	10	99
7	10	117.1
8	10	99.3
9	10	74.5

The system response time for each distance varies greatly. The longest average response time is at a distance of 7 meters, and the fastest is at a distance of 9 meters. But keep in mind that at a distance of 8 and 9 meters the number of detections is less than 10 times, and the internet connection also affects the system response time.

IV. CONCLUSION

In this research a fire detection system using image classification was successfully created using the MobileNetV2 model. The system can detect fires and send notifications via Telegram. The test results in the first scenario with two motorcycle tires measuring 80/90 showed that the system was able to detect fires from a distance of 1 meter to 9 meters, but at the farthest distance the average accuracy was low, namely 57.91%. The most optimal distance is 7 meters with a high accuracy of 99.91%. The average response time obtained varies greatly from 74.5 ms to 117.1 ms which depends on the internet network connection. Then testing in the second scenario with two pieces of paper, the fire detection system in this study can detect fires at a maximum distance of up to 300 cm. This is farther than the fire detection system in research [3] which is only 140 cm.

REFERENCES

[1] Unit Pengelola Statistik, "Kejadian Kebakaran di DKI Jakarta Tahun 2020," 2021. [online] Available: <https://statistik.jakarta.go.id/kejadian-kebakaran-di-dki-jakarta-tahun-2020/> [Accessed 28 November 2021].

[2] Simbolon, C.G., Hanuranto, A.T. dan Novianti, A., 2020. Desain Dan Implementasi Prototipe Pendeteksi Dini Kebakaran Gedung Menggunakan Algoritma fuzzy logic Berbasis Internet Of Things (IOT). *eProceedings of Engineering*, 7(2), p. 3532

[3] Madhar, M., 2018. Rancang Bangun Sistem Monitoring Deteksi Dini Kebakaran Dengan Fitur Gps Berbasis Website. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *2*(1), pp.367-372.

[4] Edel, G. and Kapustin, V., 2022, July. Exploring of the MobileNet V1 and MobileNet V2 models on NVIDIA Jetson Nano microcomputer. *In Journal of Physics: Conference Series* (Vol. 2291, No. 1, p. 012008). IOP Publishing.

[5] Hussain, D., Ismail, M., Hussain, I., Alroobaea, R., Hussain, S. and Ullah, S.S., 2022. Face Mask Detection Using Deep Convolutional Neural Network and MobileNetV2-Based Transfer Learning. *Wireless Communications and Mobile Computing*, 2022.

[6] Dai, W., Dai, Y., Hirota, K. and Jia, Z., 2020. A Flower Classification Approach with MobileNetV2 and Transfer Learning. In Proceedings of the 9th International Symposium on *Computational Intelligence and Industrial Applications (ISCIIA2020)*, Beijing, China (Vol. 31).

[7] Indraswari, R., Rokhana, R. and Herulambang, W., 2022. Melanoma image classification based on MobileNetV2 network. *Procedia Computer Science*, 197, pp.198-207.

[8] Akay, M., Du, Y., Sershen, C.L., Wu, M., Chen, T.Y., Assassi, S., Mohan, C. and Akay, Y.M., 2021. Deep learning classification of systemic sclerosis skin using the MobileNetV2 model. *IEEE Open Journal of Engineering in Medicine and Biology*, 2, pp.104-110.

[9] Shahi, T.B., Sitaula, C., Neupane, A. and Guo, W., 2022. Fruit classification using attention-based MobileNetV2 for industrial applications. *Plos one*, 17(2), p.e0264586.

[10] Rahman, S., Titania, A., Sembiring, A., Khairani, M. and Lubis, Y.F.A., 2022. Analisis Klasifikasi Mobil Pada Gardu Tol Otomatis (GTO) Menggunakan Convolutional Neural Network (CNN). *Explorer*, 2(2), pp.54-60.

[11] Nufus, N., Ariffin, D.M., Satyawan, A.S., Nugraha, R.A.S., Asysyakuur, M.I., Marlina, N.N.A., Parangin, C.H. and Ema, E., 2021, December. Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi. *In Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO)* (Vol. 3, pp. 123-134).

[12] Sakti, W.W., Abhiyaksa, M. and Arif, R., 2022. FOD Detection Camera Pada Object Landasan Bandara. SKYHAWK: *Jurnal Aviasi Indonesia*, 2(1), pp.11-14.

[13] Dewi, I.A., 2019. Deteksi Manusia menggunakan Pre-Trained MobileNet untuk Segmentasi Citra Menentukan Bentuk Tubuh. *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, 4(1), pp.65-79.

[14] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[15] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[16] Mihigo, I.N., Zennaro, M., Uwitonze, A., Rwigema, J. and Rovai, M., 2022. On-Device IoT-Based Predictive Maintenance Analytics Model: Comparing TinyLSTM and TinyModel from Edge Impulse. *Sensors*, 22(14), p.5174.

[17] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

[18] Moolayil, J., Moolayil, J. and John, S., 2019. *Learn Keras for deep neural networks* (pp. 33-35). Birmingham: Apress.

[19] Culjak, I., Abram, D., Pribanic, T., Dzapo, H. and Cifrek, M., 2012, May. A brief introduction to OpenCV. *In 2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730). IEEE.

[20] Młodzianowski, P., 2022. Weather Classification with Transfer Learning-InceptionV3, MobileNetV2 and ResNet50. In *Conference on Multimedia, Interaction, Design and Innovation* (pp. 3-11). Springer, Cham.

[21] Kaya, A., Keceli, A.S., Catal, C., Yalic, H.Y., Temucin, H. and Tekinerdogan, B., 2019. Analysis of transfer learning for deep neural network based plant classification models. *Computers and electronics in agriculture*, 158, pp.20-29.

[22] A. Saied, "FIRE Dataset," 2020. [online] Available: <https://www.kaggle.com/datasets/phylake1337/fire-dataset/> [Accessed 14 January 2023].

[23] C. Ganteng, "Fire Detection Dataset," 2020. [online] Available: < https://www.kaggle.com/datasets/christofel04/fire-detection-dataset/> [Accessed 14 January 2023].