Evolution of Information, Communication and Computing Systems (EICCS)
https://publisher.uthm.edu.my/bookseries/index.php/eiccs

Wildfire Detection Using Convolutional Neural Network

Anang Suryana[1*], Abd Kadir Mahamad[2] and Sharifah Saon[2]

[1]Faculty of Engineering, Computer and Design (FECD),
Nusa Putra University, INDONESIA

[2]Faculty oF Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, MALAYSIA

*Corresponding Author

Abstract:
This project focuses on leveraging deep learning techniques for the detection of wildfires, emphasizing their severe threat to property, human life, and ecosystems. Timely and accurate wildfire detection is crucial for effective response and mitigation. Using convolutional neural networks (CNNs), the study introduces a novel approach that eliminates the need for manual feature engineering by directly learning from diverse data sources, including satellite photos, aerial images, and real-time video feeds. Unlike traditional methods that rely heavily on predefined rules or manual data analysis, the CNN-based model automatically identifies complex patterns, such as fire fronts and smoke plumes, leading to improved detection accuracy. The project achieves a final accuracy of 96.36% and a loss of 0.1013 in fire segmentation, with training accuracy at 95.12% and a loss of 0.1418. During validation, the model reaches a final accuracy of 94.54% with a loss of 0.2612 in fire classification. The outcomes demonstrate the potential of deep learning in improving wildfire response plans, and early warning systems, and reducing devastation. These results suggest significant advancements over existing methods, providing a robust foundation for early wildfire detection and response, potentially revolutionizing wildfire management and reducing damage

Keywords:
Wildfire, Convolutional Neural Networks, deep learning, aerial images satellite images

1. Introduction

Deep learning algorithms are a subset of machine learning algorithms based on developing computer models with several processing layers that enable learning at various levels of abstraction [1]. Deep learning algorithms have been investigated for use in the detecting and monitoring of wildfires using aerial and satellite photography. Deep neural networks can be taught to recognize patterns and feature typical of wildfires, such as smoke plumes, fire fronts, and heat signatures, by training them on large data sets of wildfire and non-wildfire photos. Many studies have been undertaken to handle wildfire challenges since the emergence of machine learning. However, the findings and outcomes were quite case-specific. Most investigations have concentrated on specific localities with distinct technique settings, making it difficult for others to replicate the results [2].

*Corresponding author: anang.suryana@nusaputra.ac.id

Forest fires, also known as "wildfires," can and do occur naturally and perform a variety of critical functions in ecosystems. Wildfires are a pervasive and vital component of the Earth's ecosystem, occurring globally throughout all twelve months of the year [3]. However, wildfires are one of the world's most destructive natural disasters. They contribute to global warming, ruin property, cause massive economic losses, and eventually result in the loss of human and animal life and the destruction of communities [4]. The current research reveals an estimated worldwide annual burnt area of roughly 420 million hectares [5].

In recent years, forests, farms, residential areas, wildlife habitats, and ecosystems have all suffered significant damage because of wildfires. According to reports from Jabatan Bomba dan Penyelamat Malaysia, there were 1,753 fires caused by wildfires in 2021 [6] compared to 2002 in 2020 [7]. The worst wildfires in Malaysia experienced happened in Sabah in 1982 – 1983 when over a million hectares of natural forest burned in Sabah. At the same time, numerous fires affected Borneo and 3.2 million in Kalimantan [8]. As a result, millions of dollars are spent annually on fire management efforts that aim to reduce or stop wildfires' negative effects. To effectively manage wildfires, it is essential to comprehend them and improve our ability to predict them. Some of these areas include emergency response, ecosystem management, land-use planning, and climate adaptation, to name a few [9].

In general, wildfires can occur due to a combination of natural and human factors. Natural causes may include lightning strikes, volcanic eruptions, or spontaneous combustion in peatlands. Human activities that can lead to wildfires include agricultural burning, land clearing through slash-and-burn methods, irresponsible disposal of cigarettes or open flames, and accidental ignition from industrial activities or campfires.

2. Literature Review:

In recent years, there has been significant progress in the use of artificial intelligence and machine learning for wildfire detection. Various techniques have been explored, ranging from the use of satellite imagery to ground-based sensor networks and advanced AI models.

- **Satellite Imagery-Based Detection:** Satellite imagery has been a popular method for wildfire monitoring, utilizing sensors like MODIS and VIIRS for heat signature analysis. Recent studies have leveraged machine learning models such as Convolutional Neural Networks (CNNs) for processing satellite data, achieving notable improvements in detection accuracy. For example, research by [10] demonstrated how CNNs could identify wildfires with higher precision by extracting spatial features from thermal infrared data. However, satellite-based methods are sometimes limited by resolution and cloud cover, which can impede visibility.
- **Ground-Based Sensor Networks:** Sensor networks, including heat and smoke sensors, have been deployed in high-risk areas to provide early warnings. Research by [11] explored the integration of wireless sensor networks (WSNs) for wildfire detection, emphasizing the real-time capabilities and low latency of these systems. While effective for localized monitoring, these methods face challenges such as high installation and maintenance costs, as well as limited coverage compared to satellite-based solutions.
- **Hybrid Approaches:** Combining satellite imagery with ground-based sensor networks has been shown to enhance detection reliability. A study by [12] integrated data from drones, satellites, and ground sensors using AI algorithms, achieving comprehensive wildfire monitoring. These hybrid approaches are especially useful in areas where one method alone might be insufficient, such as in dense forests or regions with frequent cloud cover.
- **Advanced AI Techniques:** Machine learning models, particularly deep learning architectures, have transformed wildfire detection capabilities. Beyond traditional CNNs, more recent research has focused on techniques like recurrent neural networks (RNNs) for spatiotemporal data analysis and transfer learning for model optimization. For instance, [13] demonstrated how transfer learning could expedite model training while maintaining high accuracy, which is crucial for real-time applications.

3. Methodology

A lot of planning and discussion is carried out before the start of this work. A general overview of the work is established. To ensure that the work matches the flow, a great deal of preparation and planning is required before it can begin. It also ensures that the project runs smoothly. This work's time management is meticulously planned, and a flowchart is used to create the process. The flowchart in Figure 1 illustrates the process of using a Convolutional Neural Network (CNN) to classify wildfire images while Figure 2 contains the block diagram.

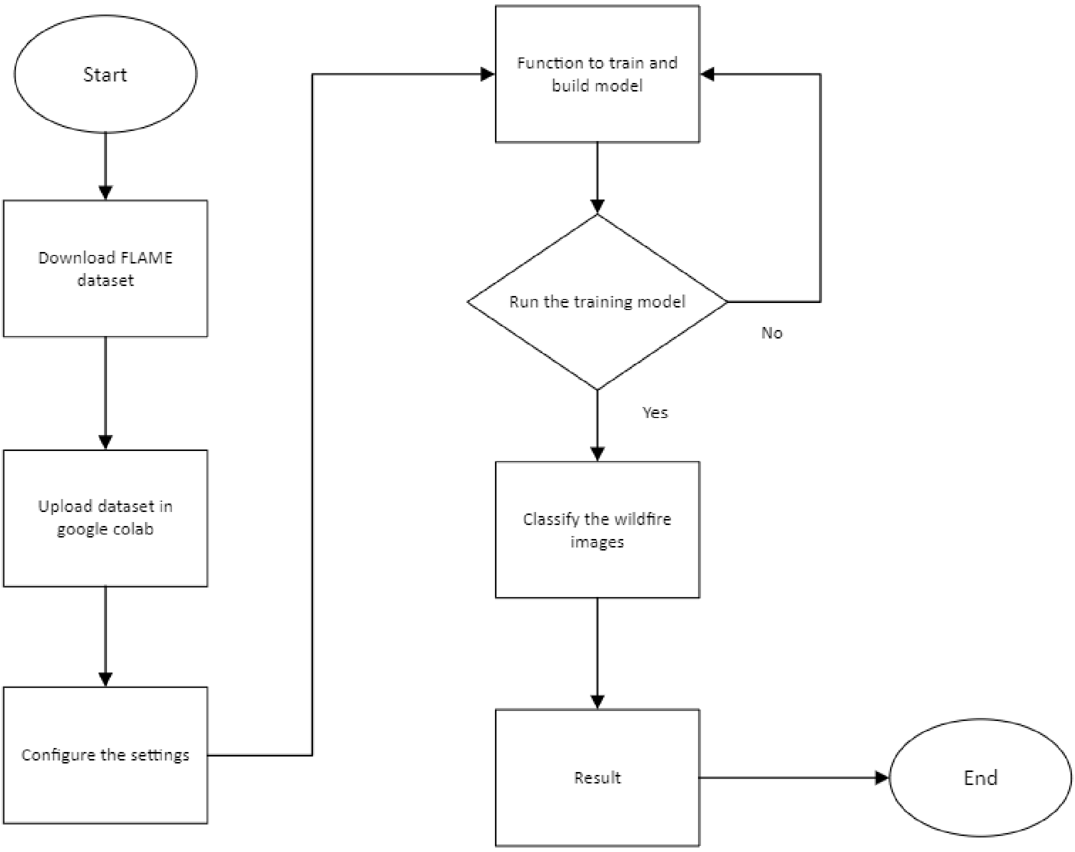


Figure 1: The flowchart of process flow by using Google colaboratory

Figure 1 shows a flowchart of how to use a Convolutional Neural Network (CNN) to classify wildfire images. The flowchart has the following steps.

i. Download the FLAME dataset, which is a collection of images of wildfires and non-wildfires from different sources.
ii. Upload the dataset to Google Colab, which is a cloud-based platform that allows you to run Python code and access machine learning libraries.
iii. Train and compile the model using CNN, which means setting up the network architecture, the loss function, the optimizer, and the metrics.
iv. Configure the settings, which means choosing the parameters such as the number of epochs, the batch size, the learning rate, and the validation split.
v. Evaluate the model, which means checking how well the model performs on the validation data and comparing the accuracy with the desired threshold.
vi. If the evaluation is satisfactory, plot the graph for the training model, which shows the change in accuracy and loss over the epochs.

vii. If the evaluation is not satisfactory, adjust the settings and retrain the model until the desired accuracy is achieved.
viii. Classify the wildfire images, which means using the trained model to predict whether a new image contains a wildfire or not.
ix. Display the result, which shows the predicted label and the confidence score for each image.

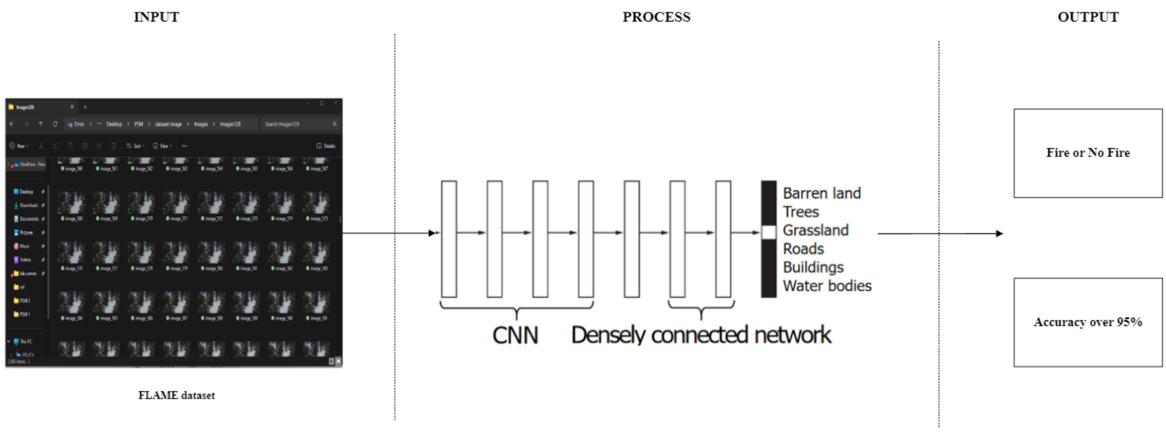


Figure 2: Block diagram for wildfire detection using deep learning

a) Input:-

i. The dataset uses the aerial imagery of the target area captured by drones.
ii. Additional contextual data, such as topographical information.

b) Process:-

i. Image processing techniques like resizing normalization and enhancement to improve the quality of dataset.
ii. CNN is a deep learning algorithm commonly used for image recognition tasks.
iii. The architecture typically consists of multiple convolutional layers, pooling layers and fully connected layers
iv. Each layer extracts relevant features and learns representation from the dataset
v. Annotated training dataset comprising labelled images of wildfire and non-wildfire regions.
vi. The CNN model is trained using this dataset, where the weights and biases of the network are adjusted to minimize the prediction error.
vii. The trained model is to used to predict wildfire regions in unseen or real-time data.

c) Output:-

i. Classify fire against no fire using a convolutional neural network (CNN) approach.
ii. Can improve accuracy of fire detection.

4. Results

In this section, this project has developed a convolutional neural network (CNN) model that has demonstrated good accuracy in identifying wildfire patterns in the dataset. Good accuracy in detecting fire hotspots and differentiating from non-fire got by training the model on a large dataset of labelled images. The deep learning approach has demonstrated its potential for automated and real-time wildfire detection, paving the way for further advancements and refinement in the system to enhance its robustness and effectiveness in wildfire prevention and response.

4.1 Fire Segmentation

Figure 3 shows the result evaluation for the performance accuracy of fire segmentation. The model's training accuracy steadily improves over epochs, reaching 95.93% by the last epoch. An epoch refers to one complete pass through the entire training dataset during the training phase of the machine learning model. The training process consists of multiple epochs, where the model iteratively updates its parameters to minimize the difference between predicted outputs and actual ground truth labels. Training loss decreases significantly, indicating that the model is learning effectively. Validation accuracy also improves consistently, reaching 96.36% on the last epoch. The model performs well on the test set, achieving a test accuracy of 96.36% and a low test loss of 0.1013. There is a slight gap between training and validation accuracy, but it doesn't seem to be a significant overfitting issue. The model appears to be effective for wildfire detection, as evidenced by high accuracy and low loss on both validation and test sets.

```
loading data...
train:324000
 test:81000
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 26, 26, 8)         296

 max_pooling2d_2 (MaxPoolin  (None, 13, 13, 8)         0
 g2D)

 conv2d_3 (Conv2D)           (None, 11, 11, 64)        4672

 max_pooling2d_3 (MaxPoolin  (None, 5, 5, 64)          0
 g2D)

 flatten_1 (Flatten)         (None, 1600)              0

 dense_2 (Dense)             (None, 16)                25616

 dense_3 (Dense)             (None, 6)                 102

=================================================================
Total params: 30686 (119.87 KB)
Trainable params: 30686 (119.87 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Start learning...
Epoch 1/10
486/486 [==============================] - 29s 59ms/step - loss: 0.7954 - accuracy: 0.7069 - val_loss: 0.3252 - val_accuracy: 0.8747
Epoch 2/10
486/486 [==============================] - 28s 58ms/step - loss: 0.2778 - accuracy: 0.8828 - val_loss: 0.2340 - val_accuracy: 0.8993
Epoch 3/10
486/486 [==============================] - 25s 52ms/step - loss: 0.2142 - accuracy: 0.9119 - val_loss: 0.1756 - val_accuracy: 0.9326
Epoch 4/10
486/486 [==============================] - 27s 55ms/step - loss: 0.1880 - accuracy: 0.9229 - val_loss: 0.1558 - val_accuracy: 0.9380
Epoch 5/10
486/486 [==============================] - 27s 56ms/step - loss: 0.1662 - accuracy: 0.9360 - val_loss: 0.2202 - val_accuracy: 0.9128
Epoch 6/10
486/486 [==============================] - 27s 55ms/step - loss: 0.1470 - accuracy: 0.9433 - val_loss: 0.1408 - val_accuracy: 0.9465
Epoch 7/10
486/486 [==============================] - 28s 58ms/step - loss: 0.1313 - accuracy: 0.9503 - val_loss: 0.1579 - val_accuracy: 0.9333
Epoch 8/10
486/486 [==============================] - 26s 53ms/step - loss: 0.1218 - accuracy: 0.9540 - val_loss: 0.1011 - val_accuracy: 0.9626
Epoch 9/10
486/486 [==============================] - 26s 53ms/step - loss: 0.1136 - accuracy: 0.9564 - val_loss: 0.0985 - val_accuracy: 0.9663
Epoch 10/10
486/486 [==============================] - 27s 56ms/step - loss: 0.1090 - accuracy: 0.9593 - val_loss: 0.1013 - val_accuracy: 0.9636
Start evaluation...
Test loss: 0.10125929117202759
Test accuracy: 0.9636213779449463
```

Figure 3: Result evaluation for the performance accuracy of fire segmentation

```
[[ 426    0    0    3  138    3]
 [    1 2662    1  109    1    0]
 [    0    2 2066  100    0    0]
 [    0   93   10 1751    2    0]
 [    5    0    0   11  280    3]
 [    0    0    0    0    0 4483]]
```

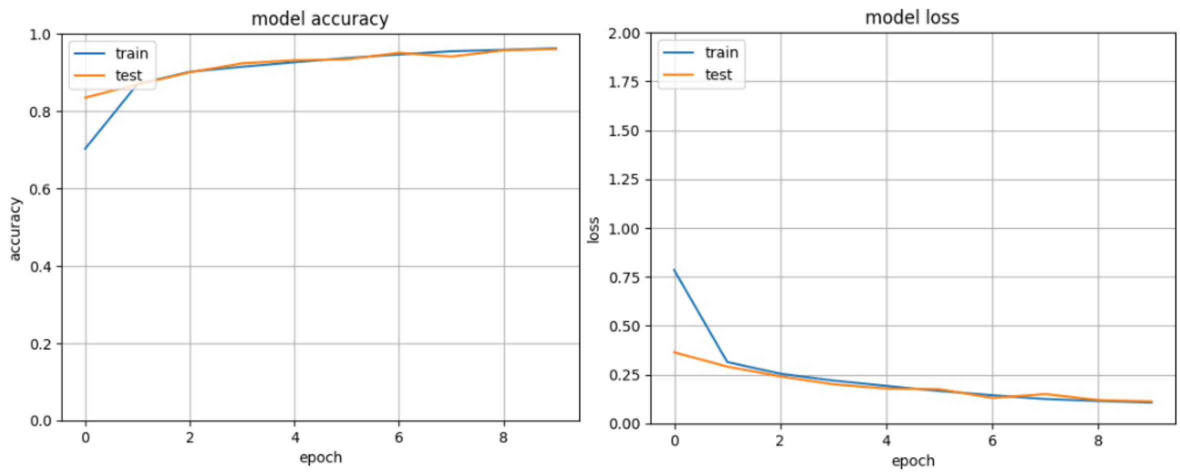
Figure 4: Confusion matrix

	precision	recall	f1-score	support
0	0.99	0.75	0.85	570
1	0.97	0.96	0.96	2774
2	0.99	0.95	0.97	2168
3	0.89	0.94	0.91	1856
4	0.67	0.94	0.78	299
5	1.00	1.00	1.00	4483
accuracy			0.96	12150
macro avg	0.92	0.92	0.91	12150
weighted avg	0.96	0.96	0.96	12150

Figure 5: Result for each class

Figure 4 shows the confusion matrix provides a detailed breakdown of the model's predictions across different classes. Diagonal elements represent the true positive counts for each class. Figure 5 shows the class which is precision, recall, F1-score, and support. Precision is the proportion of true positive predictions among all positive predictions. Recall is the proportion of true positive predictions among all actual positives. F1-score is the harmonic mean of precision and recall. Support is the number of actual occurrences of each class. Classes 1, 2, and 5 have high precision, recall, and F1-score, indicating accurate and reliable predictions. Class 0 has high precision but lower recall, suggesting potential room for improvement in detecting instances of this class. Class 4 has a relatively lower precision and F1-score, indicating some difficulty in correctly identifying instances of this class.

The model achieves an overall accuracy of 96%, which is a good performance metric. The weighted average precision, recall, and F1-score are all high, indicating a well performing model across all classes. Macro average considers each class equally, providing an overall performance metric. Weighted average considers the number of instances of each class, giving more weight to larger classes. The model demonstrates strong performance, especially in classes 1, 2, and 5. Further investigation and potential improvement may be needed for class 0 and class 4 to enhance the model's overall performance.



(a) **(b)**

Figure 6: (a) Accuracy values for the training and test sets; (b) Loss values for the training and test sets

Figure 6 illustrates the trends observed during the training and validation phases. As the epochs progressed, both training and validation losses decreased, indicating that the model was effectively learning to distinguish between fire and non-fire images. The gap between training and validation accuracy was minimal, suggesting that the model was not significantly overfitting. This is a critical observation, as it demonstrates the model's ability to generalize well to new data.

Key Insights:

- The steady improvement in accuracy over epochs indicates that the CNN architecture and training strategy were well-tuned for this application.
- The low and gradually decreasing loss values suggest that the model's predictions became increasingly accurate as training progressed.
- A slight variation in validation loss during certain epochs (e.g., a spike at Epoch 6) may point to minor overfitting, which could be addressed with additional regularization techniques or further data augmentation.

4.2 Fire Classification

Figure 7 shows the results performance training model. There are two classes are involved: 'fire' and 'No_fire. A weighted loss is applied during training, with a higher weight assigned to the 'No_fire' class (1.37) compared to the 'fire' class (0.79). A total of 39,375 files are found, with 31,500 files used for training and 7,875 files for validation. The model is trained for 10 epochs. The training accuracy steadily increases, reaching 95.12% by the final epoch. The validation accuracy fluctuates but generally stays high, reaching 94.54% by the end. Training loss decreases over epochs, indicating that the model is learning the patterns in the data. Validation loss shows some variation, and in Epoch 6, there’s a spike, suggesting potential overfitting or a need for adjustments.

```
--------- Training ---------
Weight for class fire : 0.79
Weight for class No_fire : 1.37
Found 39375 files belonging to 2 classes.
Using 31500 files for training.
Found 39375 files belonging to 2 classes.
Using 7875 files for validation.
Epoch 1/10
1969/1969 [==============================] - 266s 134ms/step - loss: 0.2793 - accuracy: 0.8963 - val_loss: 0.2395 - val_accuracy: 0.9331
Epoch 2/10
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.
  saving_api.save_model(
1969/1969 [==============================] - 264s 134ms/step - loss: 0.1950 - accuracy: 0.9313 - val_loss: 0.2096 - val_accuracy: 0.9420
Epoch 3/10
1969/1969 [==============================] - 260s 132ms/step - loss: 0.1791 - accuracy: 0.9370 - val_loss: 0.1808 - val_accuracy: 0.9457
Epoch 4/10
1969/1969 [==============================] - 263s 133ms/step - loss: 0.1674 - accuracy: 0.9393 - val_loss: 0.2894 - val_accuracy: 0.9077
Epoch 5/10
1969/1969 [==============================] - 263s 134ms/step - loss: 0.1595 - accuracy: 0.9436 - val_loss: 0.0945 - val_accuracy: 0.9670
Epoch 6/10
1969/1969 [==============================] - 259s 132ms/step - loss: 0.1563 - accuracy: 0.9443 - val_loss: 0.3744 - val_accuracy: 0.8337
Epoch 7/10
1969/1969 [==============================] - 257s 130ms/step - loss: 0.1518 - accuracy: 0.9470 - val_loss: 0.1264 - val_accuracy: 0.9550
Epoch 8/10
1969/1969 [==============================] - 258s 131ms/step - loss: 0.1464 - accuracy: 0.9479 - val_loss: 0.1045 - val_accuracy: 0.9578
Epoch 9/10
1969/1969 [==============================] - 274s 139ms/step - loss: 0.1448 - accuracy: 0.9486 - val_loss: 0.0848 - val_accuracy: 0.9783
Epoch 10/10
1969/1969 [==============================] - 259s 132ms/step - loss: 0.1418 - accuracy: 0.9512 - val_loss: 0.2612 - val_accuracy: 0.9454
```

Figure 7: Result performance for training model

4.3 Classify the wildfire images

The data have been trained with images stored in the Training folder. It contains two labels:

- Fire128: The data labeled as 'Fire'
- No_Fire128: The data labeled as 'NoFire'40

The size of the image is 128x128. Ground truth data will be used. For this project, 3 fire and no fire sample images have been used.

Table 1: Test result for classifying fire sample images

Image	Result
 f1	- This fire image is 13.10% fire and 86.90% no Fire - The result from the test simulation in Google Colab

Image	Result
 f2	- This fire image is 98% fire and 2% no fire - The result from test simulation in Google Colab
 f3	- This fire image is 13.76% fire and 86.24%% no fire - The result from test simulation in Google Colab

The results for individual images revealed that the model had occasional difficulty distinguishing between fire and non-fire images, especially in scenarios where environmental conditions complicated the detection (e.g., heavy cloud cover or reflective surfaces).

- Image f1: Classified as 86.90% "No Fire," likely due to minimal visible flames and obscuring elements.
- Image f2: Accurately identified as 98% "Fire," showcasing the model's strength in detecting clear fire patterns.
- Image f3: Incorrectly classified with 86.24% "No Fire," emphasizing the need for more robust training data in complex conditions.

Table 2: Test result for classifying no fire sample images

Image	Result
 n1	- This no fire image is 10.76% fire and 89.24% no fire. - The result from test simulation in Google Colab
 n2	- This no fire image is 2.39% fire and 97.61% no fire. - The result from test simulation in Google Colab
 n3	- This no fire image is 88.68% fire and 11.32% no fire. - The result from test simulation in Google Colab

The results for individual images revealed that the model had occasional difficulty distinguishing between fire and non-fire images, especially in scenarios where environmental conditions complicated the detection.

- Image n1: Classified as 89.24% "No Fire" and 10.76% "Fire." The model correctly identified this image as non-fire, but the small percentage of misclassification suggests the presence of features that slightly resembled fire, such as bright reflections.
- Image n2: Accurately classified as 97.61% "No Fire" with only 2.39% misclassification. This high confidence indicates the model's reliability in distinguishing clear non-fire scenarios.
- Image n3: Misclassified as 88.68% "Fire" and 11.32% "No Fire." This error highlights the model's sensitivity to certain non-fire features, such as sunlight reflections or dense clouds, which were incorrectly interpreted as fire-like patterns.

5. Discussion

While the model shows promising results, there are several challenges in real-world implementation, particularly in a context like Malaysia:

- Data Quality and Variability: Satellite and aerial images may vary in quality due to weather conditions, cloud cover, and dense vegetation. In Malaysia, frequent rain and high humidity could impact image clarity, making it difficult for the model to distinguish between fire and other natural phenomena like mist or heavy rain.
- Dense Vegetation and Peatlands: Malaysia has unique environmental factors, such as tropical rainforests and peatlands, which may present additional difficulties. Fires in peatlands are hard to detect as they can burn underground, making it challenging for traditional optical and infrared sensors to identify hotspots.
- Resource and Infrastructure Limitations: Implementing a real-time system requires high-quality satellite feeds and reliable internet connectivity, which might not be consistently available in remote areas. Additionally, integrating the model with existing emergency systems could pose logistical and operational challenges.
- False Alarms and Public Safety: Given the potential consequences of false positives (unnecessary evacuations) and false negatives (delayed responses), careful calibration of the system is crucial. In Malaysia, where fire management is crucial for palm oil plantations and forest reserves, any inaccuracies could have significant economic and social repercussions.

6. Conclusion

The use of deep learning models, such as convolutional neural networks (CNNs), has enabled the project to achieve high accuracy in detecting wildfires from various data sources, including satellite imagery, aerial images, and real-time video feeds. The models have proven to be capable of learning complex patterns and features straight from raw data, eliminating the requirement for human feature engineering and enhancing adaptability to various wildfire conditions. In this project, the accuracy for fire segmentation achieves 96.36% with the higher F1-score achieving 1.00, recall achieving 1.00, and precision achieving 1.00. While accuracy for fire classification achieves 94.54%. The outcomes of the project might have a big impact on how wildfires are managed and prevented. By detecting wildfires at their early stages, authorities can swiftly mobilise firefighting resources, implement evacuation plans, and coordinate effective response strategies. This can reduce the number of fatalities, property losses, and negative effects that wildfires bring on the ecosystem.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: https://doi.org/10.1038/nature14539

[2] Y. J. Chew, S. Y. Ooi, Y. H. Pang, and K.-S. Wong, “A Review of Forest Fire Combating Efforts, Challenges and Future Directions in Peninsular Malaysia, Sabah, and Sarawak,” Forests, vol. 13, no. 9, p. 1405, Sep. 2022, doi: https://doi.org/10.3390/f13091405.

[3] W. Bond and J. Keeley, “Fire as a global ‘herbivore’: the ecology and evolution of flammable ecosystems,” Trends in Ecology & Evolution, vol. 20, no. 7, pp. 387–394, Jul. 2005, doi: https://doi.org/10.1016/j.tree.2005.04.025.

[4] L. Giglio et al., “Active fire detection and characterization with the advanced spaceborne thermal emission and reflection radiometer (ASTER),” Remote Sensing of Environment, vol. 112, no. 6, pp. 3055–3063, Jun. 2008, doi: https://doi.org/10.1016/j.rse.2008.03.003

[5] S. C. P. Coogan, F.-N. Robinne, P. Jain, and M. D. Flannigan, “Scientists’ warning on wildfire — a Canadian perspective,” Canadian Journal of Forest Research, vol. 49, no. 9, pp. 1015–1023, Sep. 2019, doi: https://doi.org/10.1139/cjfr-2019-0094.

[6] Statistic KPKT 2021, Strategic Planning and International Division Ministry of Housing and Local Government.

[7] Statistic KPKT 2020, Strategic Planning and International Division Ministry of Housing and Local Government.

[8] GFMCadmin, “Malaysia: Forest Fire Situation (IFFN No. 25 – July 2001) – GFMC.” https://gfmc.online/iffn/country/my/my_2.html

[9] P. Jain, S. C. P. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan, “A review of machine learning applications in wildfire science and management,” Environmental Reviews, vol. 28, no. 4, pp. 478–505, Dec. 2020, doi: https://doi.org/10.1139/er-2020-0019.

[10] Ghali, Rafik, and Moulay A. Akhloufi. 2023. "Deep Learning Approaches for Wildland Fires Using Satellite Remote Sensing Data: Detection, Mapping, and Prediction" *Fire* 6, no. 5: 192. https://doi.org/10.3390/fire6050192

[11] Dampage, U., Bandaranayake, L., Wanasinghe, R., Kottahachchi, K., & Jayasanka, B. (2022). Forest fire detection system using wireless sensor networks and machine learning. Scientific reports, 12(1), 46. https://doi.org/10.1038/s41598-021-03882-9

[12] Mzid N, Cantore V, De Mastro G, Albrizio R, Sellami MH, Todorovic M. The Application of Ground-Based and Satellite Remote Sensing for Estimation of Bio-Physiological Parameters of Wheat Grown Under Different Water Regimes. *Water*. 2020; 12(8):2095. https://doi.org/10.3390/w12082095

[13] Wiame Benzekri, Ali El Moussati, Omar Moussaoui and Mohammed Berrajaa, “Early Forest Fire Detection System using Wireless Sensor Network and Deep Learning” International Journal of Advanced Computer Science and Applications(IJACSA), 11(5), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110564