A Comparative Analysis of Xception, CNN, and VGG16 Models for Fire Classification

Adrian Natanael Haryanto
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
adrian.haryanto@binus.ac.id

Imannuel Yabes
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
imannuel.yabes@binus.ac.id

Melisa Rachel Setiadi
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
melisa.setiadi001@binus.ac.id

***Abstract*—The study focuses on the performance of three deep learning models—Xception, CNN, and VGG16—for computer vision-based fire classification. The models are compared on a curated "FIRE Dataset" from Kaggle, emphasizing binary classification of fire and non-fire images. The architecture, training methodology, and performance metrics, including accuracy and F1 score, are thoroughly evaluated for each model. The findings show that the VGG16 model outperforms Xception and CNN, achieving an impressive accuracy of 98%. This research provides valuable insights into the field of computer vision-based fire detection and suggests avenues for future research.**

Keywords—computer vision, deep learning, VGG16, Xception, CNN, Fire Classification.

I. Introduction

Wildfires have become a common phenomenon throughout the world. A wildfire can cause problems of varying degrees. At worst, it can cause very high material losses [1]. Therefore, a solution is needed to overcome these fires. In this paper, a solution is given by using a computer vision approach with a comparasion between Xception, CNN, and VGG16 models which these three models will provide the results of each model.

Because fire detection has such profound effects on safety and catastrophe management, it is an important field for research. The ability to identify and categorize fire has significantly improved with the development of computer vision and deep learning. The objective of this work is to perform a comparative analysis of three different deep learning and computer vision based fire detection algorithms.

The first model trains the dataset using a basic feed-forward neural network after extracting features using the Xception architecture. The depth-wise separable convolutions in Xception, an extension of the Inception architecture, are well-known for providing better model parameter use. The feed-forward neural network further improves the model's capacity to learn from the spatial hierarchies included in the dataset.

Convolutional neural networks (CNNs) serve as the foundation for the second model that is being examined. With a focus on fire detection, this model aims to investigate the possibilities of CNNs. Customization makes it possible to include heuristics and domain-specific knowledge, which could enhance the model's functionality.

The third model makes use of the VGG16 architecture, which is well-known for being straightforward and effective in a range of computer vision applications. Because VGG16 is widely used and has been shown to be effective, it offers a solid baseline for comparison.

The fire dataset, which has two folders—a non-fire images folder and a fire images folder—was used for this study. Our goal is to assess these models' performance with regard to fire detection and classification through this comparative analysis. The knowledge gained from this research will be useful in improving computer vision-based fire detection systems moving forward, which will ultimately help with prompt and efficient fire incident response.

II. Literature Review

Many techniques have been developed to identify fire in photos and movies. A description of a few International Journal of Knowledge Based Computer Systems is provided in this section.

Fire detection is a critical area of research due to its significance in safeguarding lives and property. The application of deep learning models, specifically Xception, Convolutional Neural Networks (CNNs), and VGG16, has gained traction in recent years for addressing this challenge [2]. This section reviews relevant works in the literature that have explored these models for fire detection tasks.

Xception, an extension of the Inception architecture, has been leveraged for its powerful feature extraction capabilities. In the study conducted by [3], the authors employed Xception for fire detection. The research demonstrated that Xception effectively captured intricate patterns and spatial dependencies in fire images, providing a strong foundation for subsequent analysis.

Numerous studies have delved into the use of Convolutional Neural Networks (CNNs) for fire detection. [4] proposed a CNN architecture specifically tailored for fire detection tasks using a video. The model exhibited robust performance in learning spatial hierarchies of features, allowing it to discern fire patterns across diverse environmental conditions. This work establishes the effectiveness of CNNs in the context of fire detection.

VGG16, known for its simplicity and depth, has been explored in the realm of fire detection. The research conducted by [5] applied VGG16 to automatically learn hierarchical features from fire images. The study highlighted the adaptability of VGG16 to capture both low and high-level features crucial for accurate fire detection.

This work contributes to the understanding of VGG16's efficacy in fire-related tasks.

The comparative analysis of Xception, CNN, and VGG16 for fire detection is a key focus of this research. While studies such as [3, 4, 5] have individually explored these architectures, a comprehensive comparative study is lacking. This research seeks to bridge this gap by providing an in-depth analysis of the strengths and limitations of Xception, CNN, and VGG16 in the specific context of fire detection.

The fire dataset utilized in this research comprises two folders—fire images and non-fire images. [6] curated this dataset, ensuring a diverse collection of images representing both fire-related and non-fire scenarios. The dataset serves as a valuable resource for training and evaluating the performance of the proposed models in fire detection.

In conclusion, the related works outlined in this section provide a comprehensive overview of the application of Xception, CNN, and VGG16 in fire detection tasks. While existing studies have explored these models individually, the upcoming sections of this paper will present a detailed comparative analysis, shedding light on the strengths and weaknesses of each architecture within the specific context of fire detection.

III. Methodology

Method here are used for the research that will contain data preparation, data analytics, data pre-processing, data splitting, training, testing, evaluation, and result

A. *Dataset*

The research project utilized the "FIRE Dataset" from Kaggle, created by Ahmed Saied. The dataset was segregated into two folders. The first folder, named "fire_images," labeled as 1 comprised 755 images depicting outdoor fires, some of which exhibited substantial smoke. The second folder, "non-fire_images," labeled as 0 contained 244 nature images. The purpose of collecting this data was to train a model capable of distinguishing between images containing fire (fire images) and those representing regular scenes (non-fire images). Consequently, the primary objective of the study was binary classification.

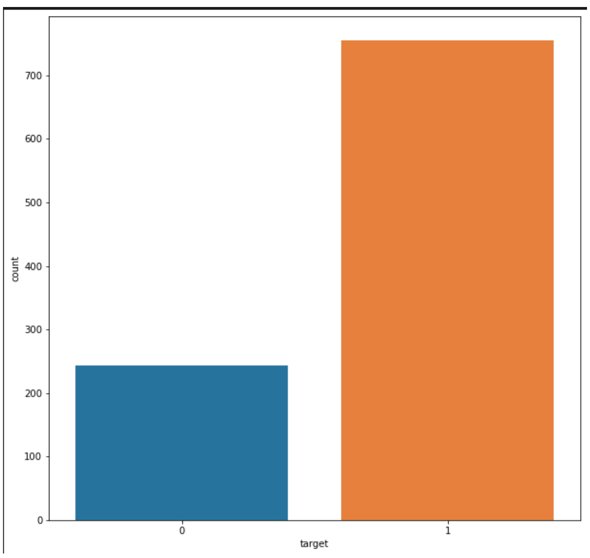

Fig. 1. Dataset Distribution

B. *Data Pre-processing*

First, the data was preprocessed by creating a mask to highlight the fire in the images so the model could detect the fire immediately and differentiate the fire from the environment. After that, the images were segmented to easily identify the fire. The last part is to sharpen the images. the purpose of sharpening images is to enhance the fine details, edges, and contours within the image. Sharpening techniques are applied to improve the visual quality of images, emphasize important features, and make images more suitable for subsequent analysis or interpretation.

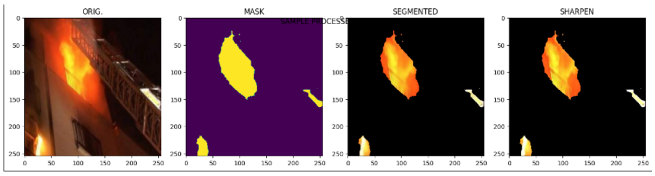

Fig. 2. Image pre-processing visualization

Next, the data were splitted into training and validation directories.. It splitted into 80: 20, the 80 is for training and the other 20 is for validation. In order to prevent overfitting, which occurs when a model performs well on the training set but is unable to generalize to new data, data splitting enables an objective assessment of a model's performance. By evaluating the model on a separate testing set, it can get a more realistic estimate of its effectiveness on new data.

C. *Model Evaluation*

In this research, to compare fire classification performance, three models were employed with the objective of evaluating their effectiveness. The three models implemented in this study are Xception, CNN, and VGG16. The use of these three models aims to determine the most efficient model in performing the fire classification task. This comparative analysis encompasses aspects such as accuracy, precision, recall, and other evaluation metrics to identify the model that yields the best results in the context of fire detection.

1. *CNN*

   for the CNN, the layer that is use are;
   Convolutional Layers: The model is then made up of multiple convolutional layers (Conv2D) with rectified linear unit (ReLU) activation functions. The first convolutional layer, which takes an input shape of (batch_Size, image_Size, image_Size, channels), has 64 filters/kernels with a size of (3, 3). It is followed by a max-pooling layer (MaxPooling2D), which has a pool size of (2, 2). There are two more pairs of convolutional and max-pooling layers that come next, with the third convolutional layer having 128 filters.
   Dropout Layers: These regularization layers (Dropout(0.1)) are added after specific convolutional layers in order to help prevent overfitting. Flatten Layer: Following the convolutional layers, the 3D output is flattened to a 1D vector as a prelude to the transition from convolutional layers to fully connected (dense) layers. **Dense (Fully Connected) Layers**: The

model consists of two fully connected (dense) layers with ReLU activation. The first dense layer has 64 units, and the last dense layer has a single unit with a sigmoid activation function. The sigmoid activation in the final dense layer indicates that the model is intended for binary classification tasks.

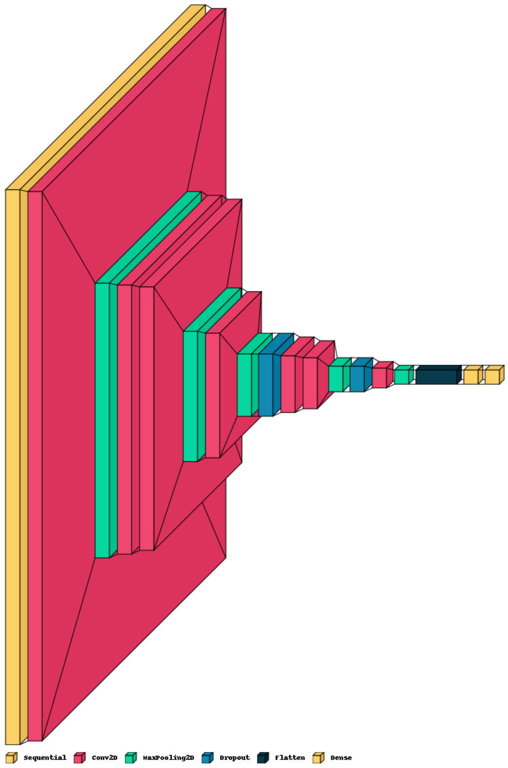

Fig. 3. CNN model visualization

2. *VGG16*

This model is supported by additional layers to support the resulting model.The output of the VGG16 base model is passed through a Flatten layer, which transforms the 4D output tensor (None, 7, 7, 512) into a 1D tensor (None, 25088). This is a common step when transitioning from convolutional layers to fully connected layers.A Dropout layer is applied after flattening with a dropout rate of 0, indicating no dropout is actually performed. Dropout layers are commonly used for regularization to prevent overfitting during training.A Dense layer with a single unit and a sigmoid activation function is added. This final layer is often used in binary classification problems, where the goal is to output a probability of belonging to a certain class.The Functional API is used to create the model, connecting the input of the VGG16 base model to the output of the final Dense layer.

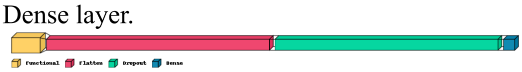

Fig. 4. VGG16 model visualization

D. *Performance Metrics*

a) Accuracy

Performance metrics is used to determine how well the model did on the test data based on the training data it was trained on. To calculate the performance metrics that were used in this models, we need to know what a confusion matrix is. Confusion matrix is used to represent each outcome of the model [7], it has 4 components:

- True Positives (TP) express the quantity of samples that were accurately classified as "positive."
- False Positives (FP) expresses the quantity of samples that were incorrectly classified as "positive."
- True Negatives (TN) expresses the quantity of samples that were accurately classified as "negative."
- False Negatives (FN) expresses the quantity of samples that were incorrectly classified as "negative."

b) F1 Score

F1 score serves as a performance metric that functions as a harmonic mean between accuracy and recall [7]. It can be considered the optimal choice among the three methods, as increased precision comes at the expense of recall, and vice versa. Therefore, maximizing the F1 Score implies the simultaneous maximization of accuracy and recall scores.

Mathematically is shown on the Fig.5.

$$F1\,Score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

or

$$F1\,Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Fig. 5. F1 Score Mathematical

IV. Result

In our research, we carried out this experiment through coding on Google Colab, utilizing a GPU with a RAM capacity of 12 GB.

TABLE I. Table of comparison

	Accuracy	F1
Baseline	87%	77%
CNN	96%	92%
VGG16	98%	97%

Based on Table 1, which contains a comparison of algorithms to assess the best algorithm for fire classification, we can observe that the VGG16 algorithm is the most effective in performing fire classification. It is evident from the table that the VGG16 algorithm achieves the highest accuracy among the algorithms, reaching 98%. This accuracy is considered quite high, indicating the superior performance of the VGG16 algorithm in the context of fire classification.

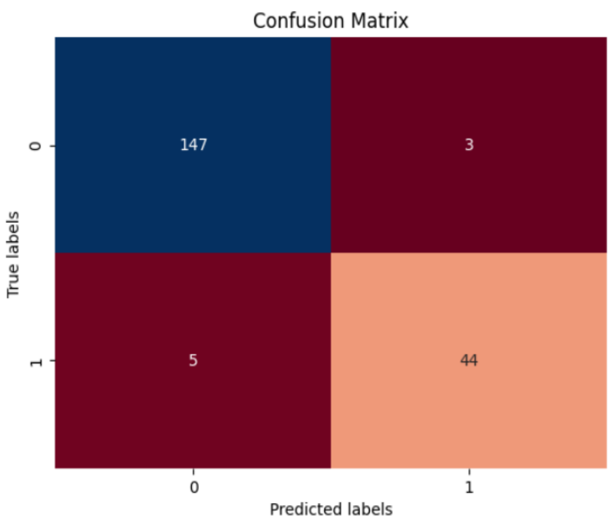

Fig.6. Confusion Matrix CNN

Figure 6 displays the confusion matrix of the CNN algorithm, where the CNN algorithm is the second-highest accuracy-generating algorithm after the VGG16 algorithm. The CNN algorithm achieves a substantial accuracy as well, reaching 96%.

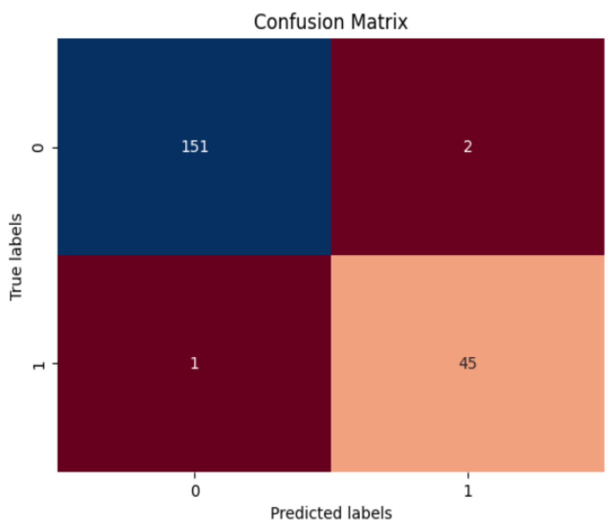

Fig.7. Confusion Matrix VGG16

Figure 7 shows the confusion matrix of the VGG16 algorithm, where the VGG16 algorithm is considered the best algorithm as it successfully achieves the highest accuracy, reaching 98%.

V. CONCLUSION

As we know, issues related to fire, such as wildfires, often occur worldwide, and if not prevented, they can lead to significant losses. Therefore, it is crucial to prevent such incidents, and in this context, we utilize a computer vision approach. Various computer vision algorithms are frequently employed to determine which one is most effective for fire detection[8]. From the results obtained, we found that the VGG16 algorithm is the most effective in fire detection, achieving an accuracy of 98%.This conclusion leads us to acknowledge that the VGG16 algorithm is considered quite efficient in performing fire detection, especially when compared to other algorithms, as their accuracy values are observed to be below that of the VGG16 algorithm. For instance, the CNN algorithm exhibits a commendable accuracy of 96%; however, this accuracy level falls short when compared to the VGG16 algorithm. Similarly, the Xception algorithm, among the three algorithms compared, is considered to have the lowest accuracy, providing an accuracy value of 87%. Certainly, when compared to the other algorithms, and notably the VGG16 algorithm, its accuracy value is noticeably lower.

It is evident that this research demonstrates satisfactory accuracy results using the VGG16 algorithm however, there is still much room for future works. For instance, expanding the dataset by using larger datasets, exploring other algorithms, developing ensemble-based models, and many more avenues for improvement can be pursued.

REFERENCES

[1] Daniel, E. R. (2023, January 12). Wildfire Smoke Detection with Computer Vision. arXiv.org. https://arxiv.org/abs/2301.05070

[2] Jin, C., Wang, W., Alhusaini, N., Zhao, S., Liu, H., Xu, K., & Zhang, J. (2023). Video fire detection methods based on deep learning: datasets, methods, and future directions. Fire, 6(8), 315. https://doi.org/10.3390/fire6080315

[3] Verlekar, T. T., & Bernardino, A. (2020). Video based fire detection using XCeption and Conv-LSTM. In Lecture Notes in Computer Science(pp. 277–285). https://doi.org/10.1007/978-3-030-64559-5_21

[4] S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," 42nd Annual Conference of the IEEE Industrial Electronics Society, 2016.

[5] Sathishkumar, V. E., Cho, J., Subramanian, M., & Naren, O. S. (2023). Forest fire and smoke detection using deep learning-based learning without forgetting. Fire Ecology, 19(1). https://doi.org/10.1186/s42408-022-00165-0

[6] FIRE Dataset. (2020, February 25). Kaggle. https://www.kaggle.com/datasets/phylake1337/fire-dataset/data

[7] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and Extreme Classification View project Modeling dependencies in latent topic models View project A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation," in Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain: Springer Berlin Heidelberg, 2005. doi: https://doi.org/10.1007/978-3-540-31865-1_25

[8] Geetha, S., Abhishek, C., & Akshayanat, C. S. (2020). Machine Vision Based Fire Detection Techniques: A survey. Fire Technology, 57(2), 591–623. https://doi.org/10.1007/s10694-020-01064-z