Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 2, April 2025, pp. 1202~1211
ISSN: 2302-9285, DOI: 10.11591/eei.v14i2.8805

Refining CNN architecture for forest fire detection: improving accuracy through efficient hyperparameter tuning

Kurniawan[1], Agus Perdana Windarto[2], Solikhun[3]
[1]Information Systems Study Program, Faculty of Science and Technology, Universitas Bina Darma, Palembang, Indonesia
[2]Master's Program, Computer Science Study Program, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
[3]Informatics Engineering Study Program, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia

Article Info

Article history:

Received Jun 6, 2024
Revised Sep 30, 2024
Accepted Oct 17, 2024

Keywords:

Artificial intelligence
Convolutional neural network
Deep learning
Forest fire detection
Neural network

ABSTRACT

Forest fire detection is one of the critical challenges in disaster mitigation and environmental management. This research aims to increase the accuracy of forest fire detection through improving the convolutional neural network (CNN) architecture. The main focus of research is on efficient hyperparameter tuning, which includes selecting and optimizing key parameters in CNN architectures such as convolutional layers, kernel size, number of neurons in hidden layers, and learning algorithms. By utilizing grid search techniques and heuristic-based optimization algorithms, the resulting CNN model shows significant improvements in detection accuracy compared to previous approaches. The evaluation was carried out using a pre-processed forest fire image dataset, and the results show that architectural refinement and appropriate hyperparameter tuning can substantially improve model performance. Evaluation results comparing two models, VGG16 and the proposed method, show significant improvements over the proposed method. The proposed method shows better capabilities with an accuracy of 95.31% and a precision of 97.22%. This research contributes to developing a more reliable and efficient forest fire detection system, which is expected to be used in real applications to reduce the impact of forest fires more effectively.

Corresponding Author:

Kurniawan
Information Systems Study Program, Faculty of Science and Technology, Universitas Bina Darma
Subdistrict Seberang Ulu I, Palembang, South Sumatra 30111, Indonesia
Email: kurniawan@binadarma.ac.id

1. INTRODUCTION

Forest fires are natural disasters that can cause major economic, ecological, and social losses [1]. The impacts are limited to loss of forest cover, increased greenhouse gas emissions, destruction of wildlife habitats, and threats to human life and property. As the frequency and intensity of forest fires increase due to global climate change, the need for fast and accurate early detection systems becomes increasingly urgent [2]-[5]. Forest fire detection systems aim to identify and monitor fires quickly and accurately, enabling immediate response to minimize losses. This system integrates various technologies and methods, including remote sensing using satellites, drones, temperature sensors, and monitoring cameras [6], [7]. Data is collected from satellite imagery such as moderate resolution imaging spectroradiometer (MODIS) and visible infrared imaging radiometer suite (VIIRS), drones equipped with infrared cameras, and monitoring cameras installed on towers or strategic points. This data is then processed using an image processing algorithm to detect fire signs such as temperature changes, smoke and flames. Machine learning algorithms, especially convolutional neural networks (CNN), are used to accurately analyze images and detect fires. In addition, predictive models are also used to estimate the spread of fires based on weather conditions and vegetation type [8]-[10]. Once a fire is detected, the system automatically sends an alert to firefighters and relevant authorities, including information on the fire's location, the burned area's size, and severity [11], [12].

Various algorithms are used in forest fire detection systems to increase accuracy and efficiency [13]. CNN is one of the main algorithms used to process image data, extracting important features such as edge and texture detection [10], [13]-[18]. Support vector machine (SVM) is a supervised learning algorithm that works by finding hyperplanes that separate data into different classes, which can be used to classify images [19]-[21]. Random forest, an ensemble learning method that uses multiple decision trees, improves classification accuracy by analyzing multi-source data such as imagery and weather data. The k-nearest neighbors (KNN) algorithm classifies data points based on the majority of their KNN so that it can compare unknown images with images that have been labelled [2], [22], [23]. Principal component analysis (PCA) reduces image dimensions before being applied to fire detection algorithms such as CNN or SVM, increasing computational efficiency. Finally, long short-term memory (LSTM), a type of recurrent neural network (RNN) suitable for processing sequential or time-series data, is used to analyze weather data [18], [24].

Previous research by [25] compared the performance of various 3D CNN-based deep learning architectures for hyperspectral image classification. This provides a deep understanding of the advantages and disadvantages of each architecture in dealing with hyperspectral image classification problems. However, it still needs to provide an in-depth analysis of the weaknesses of each architecture. In addition, this article also needs to provide information about the limitations and obstacles that may be encountered in using this architecture in a practical context. This article also needs to discuss the potential for developing or improving 3D CNN-based deep learning architectures for hyperspectral image classification in the future. Meanwhile, the next study [26] This study used a deep learning approach for the automatic diagnosis of pulmonary embolism on computed tomography pulmonary angiography (CTPA) with high sensitivity, specificity and accuracy, namely 0.80, 0.74, and 0.76 at the scanning level, and 0.93, 0.89, and 0.89 at the scanning level. Slice. The model also achieves area under the receiver operating characteristic curve (AUROC) of 0.85 and 0.94 at scan and slice levels, respectively. However, it has the disadvantage that the dataset used in the study is not publicly available due to ethical restrictions and the proprietary nature of the study. This may limit the ability of other researchers to replicate or validate research findings.

From the discussion that has been outlined, a problem can be seen. There is a need to increase the accuracy of forest fire detection using an efficient CNN architecture through appropriate hyperparameter tuning. Although CNNs have been proven to be effective in image analysis, the main challenge lies in selecting and optimizing key parameters in the CNN architecture, such as the number of convolutional layers, kernel size, and learning parameters [8], [27], [28]. With proper tuning, CNN models can avoid underfitting or overfitting, negatively impacting fire detection performance. Therefore, this research aims to develop a more efficient hyperparameter tuning method to achieve a more accurate and reliable model in detecting forest fires, which is crucial for disaster mitigation and environmental protection [29], [30].

Based on the literature study that has been described, it is realized that forest fire detection technology has developed rapidly, one of which is through satellite imagery and digital image processing. CNN has become a popular and effective method in image analysis, including forest fire detection. However, although CNNs have great potential, the main challenge is improving detection accuracy optimally. One approach to overcome this challenge is through enhancing the CNN architecture and efficient hyperparameter tuning [31]-[33]. Hyperparameters in CNNs, such as the number of layers, kernel size, and learning parameters, greatly influence the model performance. With proper tuning, CNN models can avoid underfitting or overfitting, negatively impacting detection accuracy. Therefore, this research focuses on the importance of an efficient hyperparameter tuning process to achieve a more accurate and reliable model in detecting forest fires. Through this research, it is hoped that better methods for tuning hyperparameters in CNN architectures can be developed, which will improve detection accuracy and computational efficiency. Thus, the resulting forest fire detection system can be widely applied and provide a fast response in mitigating fires, reducing losses incurred, and protecting the environment and human life.

2. METHOD

2.2. Method of collecting data

In researching to obtain data and information, the methods used in the data collection process are as follows: by using open source data from several countries, stock market data may be available for research in an open format (open source). You can explore these resources to create a research series with relevant data. Data from the Kaggale.com site. Image data collection: collecting image data from satellites, drones and monitoring cameras; data pre-processing: performing normalization, noise reduction, and image segmentation; dataset structuring: dividing the dataset into training and testing subsets and labelling the data. CNN model development.

2.2. Research design

This research uses a quantitative approach with experimental methods to explore the effect of architectural improvements by improving hyperparameters. This experimental design was chosen to test the hypothesis that there is an optimal hyperparameter architecture that can increase the accuracy of image recognition/prediction. Figure 1 illustrates the design of this research, which aims to enhance CNN architecture for recognizing forest fires. The research process is organized into several key stages: initially, the dataset preparation phase involves dividing the dataset into two subsets: 70% for training and 30% for testing. This division is performed randomly while ensuring a balanced proportion between fire and non-fire images to prevent bias. Each image is labeled as 'fire' or 'not fire' based on visual inspection and reference data, with experts conducting the manual labeling to guarantee accuracy and consistency. Figure 1 is the research framework.

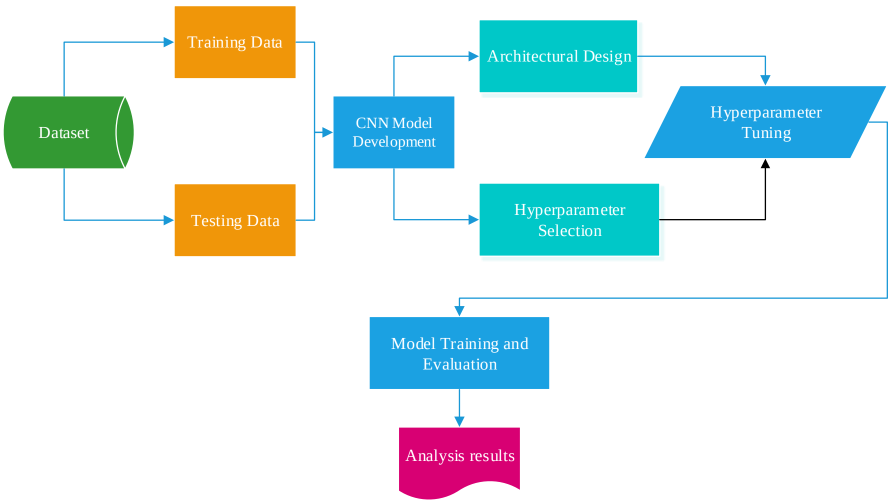


Figure 1. Research design

Next, in the CNN model development phase, the basic architecture of the CNN is designed by considering the number of convolutional layers, kernel size, pooling layers, and fully connected layers. This initial architecture is informed by a review of existing literature and best practices in image detection. The selection of key hyperparameters, such as learning rate, batch size, and the number of epochs, is guided by preliminary experiments and relevant studies.

Following this, hyperparameter tuning is conducted through a grid search, which systematically explores various combinations of hyperparameter values to identify the optimal set based on performance in the validation subset. Additionally, Bayesian optimization is applied to streamline the tuning process by navigating the hyperparameter space more efficiently. This probabilistic model helps to select promising hyperparameter combinations, reducing the number of necessary experiments. The model is then validated using a validation subset, with performance evaluated in terms of accuracy, precision, recall, and F1-score. This evaluation ensures the model can generalize well to new data beyond the training set.

Subsequently, during the model training and evaluation phase, the CNN model is trained using the training subset with the optimized hyperparameters. Training is executed with backpropagation and optimization algorithms like Adam or RMSprop. The model's performance is assessed on the test subset to gauge its overall accuracy and effectiveness post-training. This assessment includes a detailed analysis of performance metrics, such as the confusion matrix, to understand error distribution and performance across different classes.

Implementation and testing in the field mark the next step, where the trained model is integrated into a forest fire detection system for real-world testing. This involves the integration of the model with the hardware and software of the monitoring system. Real-time testing evaluates the system's performance in detecting fires under actual conditions, including trials across various locations and weather scenarios to ensure reliability. Based on field testing outcomes, adjustments are made to enhance system reliability and accuracy, potentially involving further tuning or adding new features.

Finally, data analysis is conducted to evaluate the overall effectiveness of the CNN architecture improvements and hyperparameter tuning. The analysis includes comparing these results with previous methods and identifying key factors that contributed to improved performance. The research findings are documented in a comprehensive report detailing the methodology, results, discussions, conclusions, and recommendations for future research. Additionally, the report provides practical implications of the findings and offers suggestions for implementation in the field.

2.3. Convolutional neural network algorithm

Train a CNN using different optimization optimizers to compare with different batch sizes. The author uses a popular deep learning framework like TensorFlow for this task. We present the CNN model optimization flowchart in Figure 2.

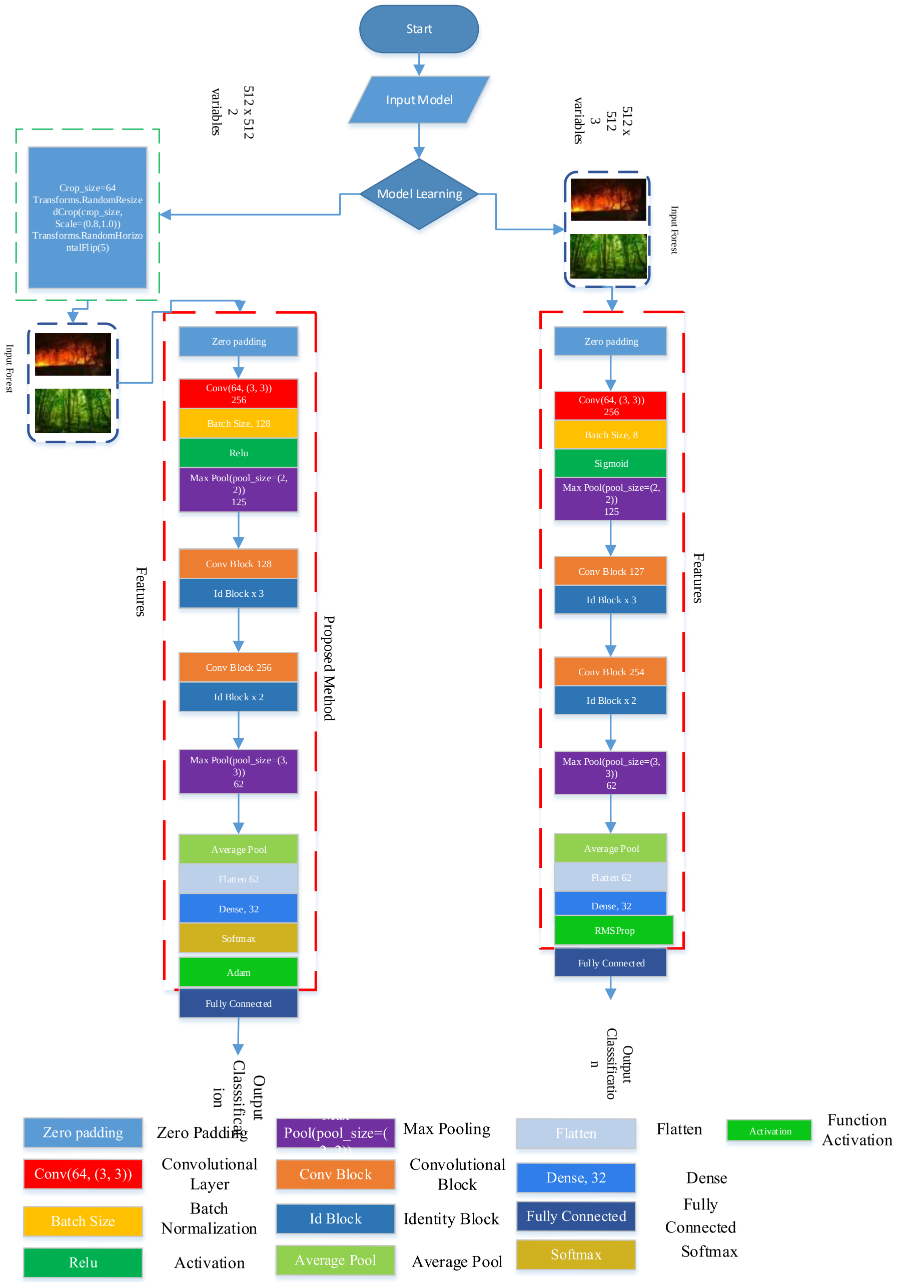


Figure 2. Comparison of standard CNN model with proposed model CNN

Figure 2 outlines the CNN model training process for forest fire detection through detailed flow diagrams, starting with the foundational steps. Initially, flowcharts conventionally begin with a "Start" symbol, which marks the initiation of the procedure. The process then advances to the model input stage, where training is applied to each model, beginning with the standard CNN model and moving to the proposed CNN model.

The training process is visually represented in Figure 2 where two distinct architectures both rooted in the same concept are illustrated using input images measuring 512×512 pixels. For the first model (depicted on the left), the process commences with model inputs, specifically a 512×512 pixel image with three color channels red green blue (RGB). The training process begins with zero padding, which adds zeros around the image border to preserve spatial dimensions. In the first convolutional block, a layer with 64 filters of size 3×3 is applied, followed by batch normalization to stabilize training and the rectified linear unit (ReLU) activation function to introduce non-linearity. Max pooling with a 2×2 window reduces the feature map size. Subsequent convolutional blocks increase the number of filters (e.g., 128 and 256) and are followed by pooling operations. Before reaching the fully connected layers, average pooling is applied, and the model incorporates a dense layer with 32 neurons, concluding with a softmax activation function for classification. The Adam optimization algorithm updates the model weights during training.

In contrast, the second model (illustrated on the right) also utilizes a 512×512 pixel image with RGB channels as input. The training process follows a similar structure, starting with zero padding and progressing through convolutional and pooling layers. However, this model integrates an identity block implementation with shortcut connections, enhancing the training process by speeding up convergence and mitigating the vanishing gradient problem. Like the first model, it employs average pooling before the fully connected layers and includes a dense layer with 32 neurons followed by a softmax layer. The RMSprop optimization algorithm is used to update the model weights.

A comparison of the two models reveals that both are based on the VGG16 architecture, featuring multiple convolutional and pooling layers. The primary distinction lies in the optimizer and the use of an identity block; the first model utilizes Adam without an identity block, while the second model employs RMSprop and incorporates an identity block with shortcut connections. Both models use average pooling before the fully connected layer and feature a dense layer with 32 neurons preceding the softmax layer for the final classification step. This figure shows how two VGG16-based CNN models are applied for forest fire detection, with some differences in optimization and the use of identity blocks. The results of this comparison will determine which model is more efficient and accurate in detecting forest fires based on the configuration used.

3. RESULTS AND DISCUSSION

At this stage, we explain the results obtained from research on improving the VGG16 CNN architecture for forest fire detection with hyperparameter optimization. The discussion will include an evaluation of model performance, a comparison between the two models tested, and an analysis of implementation and testing results in the field.

3.1. Research dataset

This test uses an image showing two images that will be trained to recognize and detect forest fires by improving the CNN architecture. In this test, two images train a CNN model to detect forest fires through architectural improvements (Figure 3). The first image shows a burning forest with clear flames, while the second image shows an unburned forest with lush green trees. These two images were resized to 512×512 pixels and normalized before being used in model training. Data augmentation techniques such as rotation and flipping are applied to increase dataset diversity and reduce overfitting. The model was then tested with a separate dataset to evaluate its performance in detecting forest fires, using accuracy, precision, recall and F1-score metrics. The results of this test aim to assess the effectiveness of CNN architecture improvements in increasing the accuracy and stability of forest fire detection. Figure 3 is a sample of 2 class data, namely burnt and unburnt forests.

Figure 3. Example of an image that will be practiced with 2 classes

3.2. Results

At this stage, image classification is carried out separately using the VGG16 architecture with VGG16+SMOTE data model and ResNet50+class weighted approach+SMOTE data. Figure 3 compares the accuracy results obtained by each architecture on training and validation data. The blue line shows the accuracy value for training data, while the orange line is for validation data. The Figure 4 classification training processes. In Figure 4, the accuracy and loss graph of the forest classification training process can be seen.

Figures 4(a) to (d) shows four line graphs representing the performance of the two methods over several periods during the training and testing phases for the CNN model. Each graph is labelled with the method and metric it represents. The graph shows the cost and score trends for the "proposed method" and "VGG16" methods. This method avoids averting using early stopping so that training does not exceed the training limit. This test uses two activations: the ReLU activation function with a learning rate of 0.001 for the proposed method and the Sigmoid learning rate activation function of 0.01 for the VGG16 model with parameters, namely batch size 128 for ReLU and 64 for the proposed model. For epoch, here we use early stop, where the process automatically stops when the model can't get any more accuracy and losses. The performance metrics obtained from evaluating the ReLU and sigmoid models are critically analyzed. Comparative analysis revealed potential differences in the model's ability to classify animal images accurately. Statistical tests were performed to determine the significance of these differences, thereby providing strong evidence to support the findings. The proposed method shows more stable and consistent cost and score metrics performance throughout the period compared to the VGG16 method.

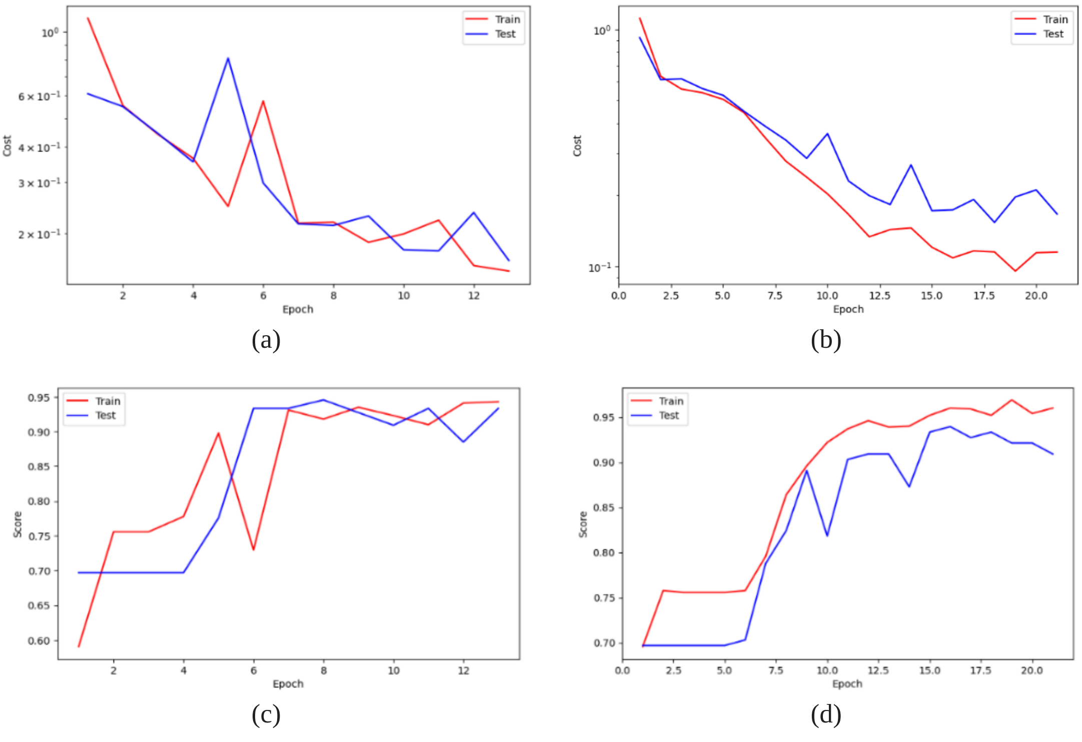


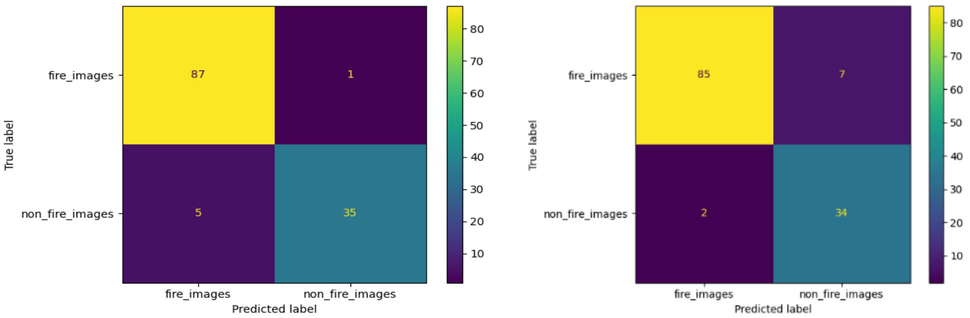
Figure 4. Accuracy and loss graph of forest classification training process; (a) cost propose method, (b) cost VGG16, (c) score propose method, and (d) score VGG16

3.2.1. Confusion matrix

Analysis of the confusion matrix can help in decision making to further refine the model. For example, if the model often makes false negative errors (fails to detect fires), then it may be necessary to adjust the threshold or use data augmentation techniques to improve fire detection. Confusion matrix is a very useful tool in evaluating and understanding the performance of classification models. By providing detailed insight into model prediction results, confusion matrices enable identification of specific errors, calculation of important evaluation metrics, and provide direction for further refinement. In the context of forest fire detection, the use of a confusion matrix helps ensure that the model is not only accurate but also reliable in detecting actual fires, so that it can provide a fast and effective response in emergency situations.

The confusion matrix in the training of the two models tested. In Figure 5, we can see the results of the confusion matrix. In Figure 5 the confusion matrix of the two methods tested.

Figure 5 contains two confusion matrix charts labelled “Figure 5(a) CM propose method” and “Figure 5(b) CM VGG16”. This matrix is used to evaluate the performance of the classification model. Each matrix has two rows and two columns, with the rows representing the actual labels and the columns representing the predicted labels. The actual labels are “fire_images” and “non_fire_images”. The colour coding of the boxes most likely indicates the number of images, with darker colours representing a greater number. The chart is used to visually compare the performance of two methods for classifying images as containing fire. In Figure 6, we can see the results of the classification report. Figures 6(a) and (b) shows the classification report of the two models being compared.



(a) (b)

Figure 5. Confusion matrix; (a) CM propose method and (b) CM VGG16

```
Confusion matrix :
 [[35  5]
 [ 1 87]]
Outcome values :
 35 5 1 87
Classification report :
              precision    recall  f1-score   support

           1       0.97      0.88      0.92        40
           0       0.95      0.99      0.97        88

    accuracy                           0.95       128
   macro avg       0.96      0.93      0.94       128
weighted avg       0.95      0.95      0.95       128
```

(a)

```
Confusion matrix :
 [[34  2]
 [ 7 85]]
Outcome values :
 34 2 7 85
Classification report :
              precision    recall  f1-score   support

           1       0.83      0.94      0.88        36
           0       0.98      0.92      0.95        92

    accuracy                           0.93       128
   macro avg       0.90      0.93      0.92       128
weighted avg       0.94      0.93      0.93       128
```

(b)

Figure 6. Classification report; (a) CM propose method and (b) CM VGG16

Of the two CMs, the proposed model has a better prediction level than the VGG16 model. You can see the results from precision, recall, F1 score, and accuracy. The prediction result image can be seen in Figures 7(a) and (b).

(a) (b)

Figure 7. Prediction results; (a) propose prediction method and (b) VGG16 prediction

Figure 7 shows that the actual label will be compared with the predicted label. Where if the actual label and predicted label show the same class, they will be coloured green, in other words, true positive, and if the actual and predicted labels show different classes, it will be false positive in red. Prediction errors can occur due to image factors such as test data, in this case in the form of unclear images, background influences such as nature, the similarity of colour and shape.

3.2.2. Model performance evaluation

The model evaluation results based on accuracy and loss during training and testing are explained. Table 1 compares the model performance of VGG16 and the proposed method (propose method) based on various evaluation metrics. VGG16 shows 92.97% accuracy, 82.93% precision, 94.44% recall, 88.31% F1-score, 84.35% FBeta, 83.67% Matthews correlation coefficient (MCC), 86.83% BM, and 80.63% marking coefficient (MK). Meanwhile, the proposed method has a higher accuracy of 95.31%, precision of 97.22%, recall of 87.50%, F1-score 92.10%, FBeta 95.75%, MCC 89.03%, BM 86.36%, and MK 91.79%. Although the recall of the proposed method is slightly lower, other metrics show superior performance compared to VGG16. With higher accuracy, precision, F1-score, FBeta, MCC, and MK, the proposed Method is more effective and reliable in detecting forest fires, balancing the reduction of false alarms with the detection of actual fire events. From the Table 1, the proposed method (propose method) performs better than the standard VGG16 model overall. Although the recall of the proposed Method is slightly lower, other metrics such as accuracy, precision, F1-score, FBeta, MCC, and MK show that the proposed method has superior performance. This means the proposed Method is more effective and reliable in detecting forest fires, with a better balance between reducing false alarms and detecting real fire events.

Table 1. Test results of the two CNN models

Method	Accuracy	Precision	Recall	F1-score	FBeta	MCC	BM	MK
VGG16	0.9297	0.8293	0.9444	0.8831	0.8435	0.8367	0.8683	0.8063
Propose method	0.9531	0.9722	0.8750	0.9210	0.9575	0.8903	0.8636	0.9179

3.3. Discussion

Hyperparameter optimization, through grid search and Bayesian optimization, significantly enhances model performance by identifying the best values for learning rate, batch size, and number of epochs, which are crucial for achieving optimal results. In comparing the VGG16 model with the proposed method, both models demonstrate effectiveness in detecting forest fires, yet the proposed method outperforms the standard VGG16. It achieves higher accuracy, precision, F1-score, FBeta, MCC, and MK, making it more effective and reliable, although it shows a slight decrease in recall values. This advantage is particularly important in real-world applications where minimizing false alarms is as critical as detecting fire events. Thus, the proposed method holds great promise for deployment in more efficient and dependable forest fire detection systems, emphasizing its potential for practical applications where precision and reliability are paramount.

4. CONCLUSION

This research aims to improve forest fire detection performance by enhancing the CNN architecture using efficient hyperparameter tuning. Evaluation results comparing two models, VGG16 and the proposed method, show significant improvements over the proposed method. With an accuracy of 95.31% and a precision of 97.22%, the proposed method shows a better ability to reduce false alarms than VGG16, which only achieved 92.97% and 82.93%, respectively. Although the recall is slightly lower, the F1-score of the proposed method remains higher, indicating a better balance between precision and recall. Other evaluation metrics such as FBeta, MCC, and MK also show significant improvements over the proposed method, making it a more reliable and efficient choice for forest fire detection with faster and more accurate response.

REFERENCES

[1] J. Choi and D. W. Shin, "Parallel Architecture Of CNN-Bidirectional Lstms For Implied Volatility Forecast," *Journal of Forecasting*, vol. 41, no. 6, pp. 1087–1098, 2022, doi: 10.1002/For.2844.
[2] S. Budilaksono *et al.*, "Comparison of Data Mining Algorithm: PSO-KNN, PSO-RF, and PSO-DT to Measure Attack Detection Accuracy Levels on Intrusion Detection System," *Journal of Physics: Conference Series*, vol. 1471, no. 1, p. 012019, 2020, doi: 10.1088/1742-6596/1471/1/012019.
[3] J. Chen and J. Wu, "The prediction of Chongqing's GDP based on the LASSO method and chaotic whale group algorithm–back propagation neural network–ARIMA model," *Scientific Reports*, vol. 13, no. 1, pp. 1–17, 2023, doi: 10.1038/S41598-023-42258-Z.
[4] V. Pandith, H. Kour, S. Singh, J. Manhas, and V. Sharma, "Performance Evaluation of Machine Learning Techniques for Mustard

Crop Yield Prediction from Soil Analysis," *Journal of Scientific Research*, vol. 64, no. 02, pp. 394–398, 2020, doi: 10.37398/Jsr.2020.640254.

[5] N. Habib and M. M. Rahman, "Diagnosis of corona diseases from associated genes and X-ray images using machine learning algorithms and deep CNN," *Informatics in Medicine Unlocked*, vol. 24, p. 100621, 2021, doi: 10.1016/J.Imu.2021.100621.

[6] A. P. Windarto, I. R. Rahadjeng, M. N. H. Siregar, and P. Alkhairi, "Deep Learning to Extract Animal Images With the U-Net Model on the Use of Pet Images," *Jurnal Media Informatika Budidarma*, vol. 8, no. 1, pp. 468–476, 2024.

[7] A. P. Windarto, T. Herawan, And P. Alkhairi, "Early Detection of Breast Cancer Based on Patient Symptom Data Using Naive Bayes Algorithm on Genomic Data," in *Artificial Intelligence, Data Science and Applications*, vol. 837, pp. 478–484, 2024, doi: 10.1007/978-3-031-48465-0_64.

[8] W. El-Shafai *et al.*, "Efficient classification of different medical image multimodalities based on simple CNN architecture and augmentation algorithms," *Journal of Optics*, vol. 53, no. 2, pp. 775–787, 2024, doi: 10.1007/S12596-022-01089-3.

[9] H. M. T. Khushi, "Performance analysis of state-of-the-art CNN architectures for brain tumour detection," *International Journal of Imaging Systems and Technology*, vol. 34, no. 1, 2024, doi: 10.1002/Ima.22949.

[10] S. Huang *et al.*, "Deep Learning-Based Clinical Wound Image Analysis Using a Mask R-CNN Architecture," *Journal of Medical and Biological Engineering*, vol. 43, no. 12, pp. 417–426, 2023, doi: 10.1007/S40846-023-00802-2.

[11] S. Defit, A. P. Windarto, and P. Alkhairi, "Comparative Analysis of Classification Methods in Sentiment Analysis: The Impact of Feature Selection and Ensemble Techniques Optimization," *Telematika*, vol. 17, no. 1, pp. 52–67, 2024, doi: 10.35671/telematika.v17i1.2824.

[12] P. Alkhairi and A. P. Windarto, "Classification Analysis of Back Propagation-Optimized CNN Performance in Image Processing," *Journal of Systems Engineering and Information Technology (JOSEIT)*, vol. 2, no. 1, pp. 8–15, 2023, doi: 10.29207/joseit.v2i1.5015.

[13] T. Anwar and H. Anwar, "LSNet: a novel CNN architecture to identify wrist fracture from a small X-ray dataset," *International Journal of Information Technology*, vol. 15, no. 5, pp. 2469–2477, 2023, doi: 10.1007/S41870-023-01311-W.

[14] S. S. L. Parvathi and H. Jonnadula, "An Efficient and Optimal Deep Learning Architecture using Custom U-Net and Mask R-CNN Models for Kidney Tumor Semantic Segmentation," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 6, 2022, doi: 10.14569/Ijacsa.2022.0130639.

[15] İ. Aydin, G. Budak, A. Sefer, and A. Yapar, "Recovery of impenetrable rough surface profiles via CNN-based deep learning architecture," *International Journal of Remote Sensing*, vol. 43, no. 15-16, pp. 5658–5685, 2022, doi: 10.1080/01431161.2022.2105177.

[16] B. Y. AlKhuraym, M. M. B. Ismail, and O. Bchir, "Arabic Sign Language Recognition using Lightweight CNN-based Architecture," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 4, 2022, doi: 10.14569/Ijacsa.2022.0130438.

[17] K. K. Mohammed, A. E. Hassanien, and H. M. Afify, "Classification Of Ear Imagery Database Using Bayesian Optimization Based On Cnn-Lstm Architecture," *Journal of Digital Imaging*, vol. 35, no. 4, pp. 947–961, 2022, doi: 10.1007/S10278-022-00617-8.

[18] S. Y. Xiong, E. F. M. Gasim, C. X. Ying, K. K. Wah, and L. M. Ha, "A Proposed Hybrid CNN-RNN Architecture for Student Performance Prediction," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 3, pp. 347–355, 2022.

[19] P. Alkhairi, W. Wanayumini, and B. H. Hayadi, "Analysis of the adaptive learning rate and momentum effects on prediction problems in increasing the training time of the backpropagation algorithm," *AIP Conference Proceedings*, vol. 3048, no. 1, p. 20049, 2024, doi: 10.1063/5.0203374.

[20] P. Alkhairi, E. R. Batubara, R. Rosnelly, W. Wanayaumini, and H. S. Tambunan, "Effect of Gradient Descent With Momentum Backpropagation Training Function in Detecting Alphabet Letters," *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, vol. 7, no. 1, pp. 574–583, 2023, doi: 10.33395/sinkron.v8i1.12183.

[21] A. P. Windarto, T. Herawan, and P. Alkhairi, "Prediction of Kidney Disease Progression Using K-Means Algorithm Approach on Histopathology Data," In *Artificial Intelligence, Data Science and Applications*, vol. 837, pp. 492–497, 2024, doi: 10.1007/978-3-031-48465-0_66.

[22] W. Sinhashthita and K. Jearanaitanakij, "Improving KNN Algorithm Based on Weighted Attributes by Pearson Correlation Coefficient and PSO Fine Tuning," *2020 - 5th International Conference on Information Technology (InCIT)*, Chonburi, Thailand, 2020, pp. 27-32, doi: 10.1109/InCIT50588.2020.9310938.

[23] A. A. Aburomman and M. Bin Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016, doi: 10.1016/J.Asoc.2015.10.011.

[24] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: a comparison between traditional and deep learning neural networks," *International Journal of Big Data Intelligence*, vol. 3, no. 1, pp. 28-37, 2016, doi: 10.1504/Ijbdi.2016.073903.

[25] H. Firat and D. Hanbay, "Comparison of 3D ESA-based deep learning architectures using hyperspectral images," *Journal Faculty of Engineering and Architecture Gazi University*, vol. 38, no. 1, pp. 521–534, 2023, doi: 10.17341/Gazimmfd.977688.

[26] P. Ajmera *et al.*, "A deep learning approach for automated diagnosis of pulmonary embolism on computed tomographic pulmonary angiography," *BMC Medical Imaging*, vol. 22, no. 192, pp. 1–9, 2022, doi: 10.1186/S12880-022-00916-0.

[27] Z. Boucetta, A. E. Fazziki, and M. E. Adnani, "Integration of Ensemble Variant CNN with Architecture Modified LSTM for Distracted Driver Detection," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13 no. 4, 2022, doi: 10.14569/Ijacsa.2022.0130452.

[28] H. Benradi, A. Chater, and A. Lasfar, "A combined method based on CNN architecture for variation-resistant facial recognition," *International Journal of Electrical and Computer Engineering Systems* (*IJECES)*, vol. 14, no. 9, pp. 993–1001, 2023, doi: 10.32985/Ijeces.14.9.4.

[29] L. Wei, Y. Wen, X. Liu, and H. Wang, "Hand gesture recognition based improved multi-channels CNN architecture using EMG sensors," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 1, pp. 643-656, 2022, doi: 10.3233/Jifs-212390.

[30] A. Tawte, S. Gonge, R. Joshi, P. Mulay, D. Vora, and K. Kotecha, "Detection of Pulmonary Embolism: Workflow Architecture and Comparative Analysis of the CNN Models," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 4, pp. 299–314, 2023, doi: 10.17762/Ijritcc.V11i4.6455.

[31] S. L. Ramaswamy and J. Chinnappan, "Review on positional significance of LSTM and CNN in the multilayer deep neural architecture for efficient sentiment classification," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 4, pp. 6077-6105, 2023, doi: 10.3233/Jifs-230917.

[32] A. A. A. Ali and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with

dropout," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3294–3300, 2022, doi: 10.1016/J.Jksuci.2021.01.012.
[33] U. Verma, P. Tyagi, and M. Kaur, "Single Input Single Head CNN-GRU-LSTM Architecture for Recognition of Human Activities," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 2, pp. 410–420, 2022, doi: 10.52549/Ijeei.V10i2.3475.

BIOGRAPHIES OF AUTHORS

Kurniawan is a Lector in the Information System Science Technology Faculty Universitas Bina Darma. He has been a lecturer since 2003 in the information systems study program, faculty of computer science. He is a writer who completed a Masters in Informatics Engineering and a Masters in Management at the Bina Darma University Postgraduate Program. Where he is the author/co-author of more than 2004 research publications. He can be contacted at email: kurniawan@binadarma.ac.id.

Agus Perdana Windarto born in Pematangsiantar on August 30 1986. He completed Masters in Computer Science and Doctoral (Ph.D.) degrees at Putra Indonesia University 'YPTK' Padang in 2014 and 2024. He has been an active lecturer at STIKOM Tunas Bangsa since 2012, teaching in the Information Systems program. His research focuses on artificial intelligence (decision support systems, expert systems, data mining, neural networks, fuzzy logic, deep learning, and genetic algorithms). Additionally, he has served as a reviewer for various nationally accredited journals (SINTA 2 - SINTA 6) and manages a community called "Pemburu Jurnal" at STIKOM Tunas Bangsa. He has won multiple research grant proposals from DIKTI (twice in 2018-2019), DIKTI Community Service Grant (once in 2019), PKM-P Grant (as a student advisor in 2018), and PKM-AI Grant (as a student advisor in 2019). He is also part of the Relawan Jurnal Indonesia (RJI) community in North Sumatra, the Data Science Indonesia Researchers Association (PDSI), the Forum of Higher Education Communities (FKPT), and is a co-founder of the Yayasan Adwitiya Basurata Inovasi (Yayasan Abivasi) foundation with fellow professors. He can be contacted at email: agus.perdana@amiktunasbangsa.ac.id or agus.perdana@abivasi.id.

Solikhun is an Associate Professor in the STIKOM Tunas Bangsa Informatics Engineering Undergraduate Study Program. In 2022 he will also become Chair of the STIKOM Tunas Bangsa Informatics Engineering Undergraduate Study Program. He is a writer who completed his Doctoral education at the Faculty of Computer Science and Information Technology, University of North Sumatra. Where he is the author/co-author of more than 2013 research publications. He can be contacted at email: solikhun@amiktunasbangsa.ac.id.