International Journal of Computer Sciences and Engineering
Vol.12, Issue.9, pp.01-08, September 2024
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

Research Article

A Comparative Study of CNN Models Built with TensorFlow and Theano for Forest Fire Detection

Nesiga A[1], Sahani S Shetty[2], Mythili Mahesh Velapakam[3*], Kiran Bailey[4], Geetishree Mishra[5]

[1,2,3,4,5]Dept. of Electronics and Communication, BMS College of Engineering, Bangalore, India

**Corresponding Author: mythili.ec19@bmsce.ac.in*

Received: 15/Jun/2024; **Accepted**: 20/Jul/2024; **Published**: 30/Sept/2024. **DOI**: https://doi.org/10.26438/ijcse/v12i9.18

Abstract: Over the past decade, forest fires have caused devastation in many areas of India, severely harming forest ecosystems, reducing biodiversity, and affecting the lives of populations that depend on the forests for their subsistence. Convolutional Neural Networks (CNNs), or ConvNets, represent a specialized deep learning architecture that extracts and learns patterns directly from data. CNNs are excellent at recognizing patterns in images, allowing them to identify objects, group similar items, and classify different categories with high precision. They can also be highly effective at classifying audio, time-series, and signal data. This work suggests creating a model that can be used to classify whether or not there is forest fire based on the images. In order to get better outcomes, the deep neural network component of the final model was developed from the VGG16 basic architecture. 5062 photos from open source sources, including both fire and no-fire conditions, were used to train the model. This paper presents a model developed using Keras with TensorFlow and Theano as the backend and the efficiency of the model was compared. The TensorFlow based model provided an accuracy of 97.6% and the Theano based model provided an accuracy of 97.54%. Even with limited resolution, the model with Keras and TensorFlow backend was able to categorise the majority of the random pictures given to it as Fire(1) and No Fire(0) class with better evaluation scores and less time.

Keywords: CNN, Deep Learning, VGG16, Forest Fires, Keras, TensorFlow, Theano, Image classification

1. Introduction

The environment has been significantly impacted in recent years by elements that are both caused by humans and climate change. Heat waves, droughts, floods, storms, and wildfires are a few of these occurrences. Forest fires have adversely impacted the natural balance of the planet, resulted in significant losses of human life and property, and attracted the attention of many nations. Recent years have seen a significant rise in both the frequency and severity of forest fires globally, driven largely by global warming, industrialization, and human activities. The effects of forest fires on forest ecosystems include changing the chemistry of the soil, eradicating microfauna, and flora, lowering water infiltration and percolation, increasing soil erosion, and deteriorating water quality, and changing the structure and composition of the plants.

Traditional methods for detecting forest fires, such as physical inspection, sensor-based systems, satellite remote sensing, and computer vision, each come with notable limitations. Detecting forest fires is particularly difficult due to the diverse forms, textures, and colours of flames. Various sensors, including smoke, gas, temperature, and humidity sensors, as well as integrated sensor systems, are commonly employed to identify fires. However, it has a short detecting range, is expensive to install, and needs to deal with challenging networking issues for the power supply. Traditional image processing techniques mainly rely on artificial characteristics, making them inapplicable on all forest conditions. India needs comprehensive forest fire mitigation techniques since 36% of its forests are vulnerable to regular forest fires and approximately 4% are severely vulnerable.

To tackle the challenge of detecting forest fires, deep learning technology is leveraged to identify and learn key features. Convolutional Neural Networks (CNNs) are especially well-suited for this task because they can efficiently process both visual and audio data, and handle complex multidimensional images directly. By improving image quality through denoising and reducing interference, CNNs excel at extracting features with remarkable precision.

Among the top models in computer vision is VGG16, a notable variant of CNN. This model is widely recognized for its strong performance in image classification and recognition. It operates as a feed-forward neural network and is trained using backpropagation and stochastic gradient descent, which optimizes its parameters for exceptional accuracy. The creators of VGG16 improved on previous designs by increasing the network’s depth and using very small (3 x 3) convolutional filters, leading to significant advancements over earlier models. They expanded the network to include 16 to 19 weight layers, resulting in approximately 138 million trainable parameters. Building on this foundation, our model enhances the detection network by deepening the fully connected (FC) layers, incorporating dropout for regularization, and employing the Adam optimizer for refined learning. We utilized the high-level Keras API with TensorFlow and Theano as backends to achieve the best possible accuracy for our model.

The rest of this paper is structured as follows: A summary of relevant research on convolutional neural networks, different architectures and their application in image classification for disaster management is given in Section 2. The suggested CNN VGG16-based forest fire detection system's methodology and architecture are described in Section 3. Section 4 includes the discussion on the complete model, the evaluation metrics, and a comparison between models with Theano and TensorFlow as backend. Section 5 discusses the results. Finally, Section 6 addresses any potential drawbacks, and suggests directions for further investigation.

2. Related Work

The related work for "A Comparative Study of CNN Models Built with TensorFlow and Theano for Forest Fire Detection" encompasses a range of studies that have explored forest fire detection techniques in various deep learning approaches.

Muhammad Aamir et al. [1] proposed an advanced multi-layered deep convolutional neural network specifically designed for identifying and classifying natural disasters, including forest fires. Their model is notable for its sophisticated integration of various filters and parameters, which enhances its ability to assess the intensity of disasters and improve detection accuracy. By utilizing multiple convolutional layers, the model effectively captures complex features and variations in disaster scenarios, leading to more precise detection and classification.

Federico Guede-Fernández et al. [2] introduced a deep learning-driven system focused on object detection within forest fire scenarios. Their approach is centred around classifying smoke columns, a critical indicator of fire presence. By employing object detection techniques and deep learning algorithms, their system provides accurate and reliable fire detection and monitoring. This method significantly enhances the capability to identify and track smoke columns, leading to improved fire detection and management.

Renjie Xu et al. [3] developed a novel forest fire detection system that integrates two distinct ensemble learning models—Yolov5 and EfficientDet. This combination leverages the strengths of both models to enhance the system's accuracy and reliability. Yolov5's robust object detection capabilities are paired with EfficientDet's efficient design, resulting in a system that offers improved performance in detecting and monitoring forest fires.

Diyana Kinaneva et al. [4] explored contemporary approaches to fire detection and control by utilizing cutting-edge technologies. Their method aims to refine detection capabilities and enhance response strategies, focusing on integrating advanced technological solutions to improve overall fire management effectiveness. This approach highlights the importance of leveraging modern technologies to advance fire detection and response systems.

Abdelmalek Bouguettaya et al. [5] reviewed techniques for early wildfire detection using unmanned aerial vehicles (UAVs) coupled with deep learning-based computer vision algorithms. Their focus is on detecting wildfires at an early stage to mitigate potential damage to human lives and forest ecosystems. The use of UAVs combined with deep learning enhances early detection capabilities and supports timely intervention to prevent or minimize fire damage.

Guoli Zhang et al. [6] presented a convolutional neural network model designed for predicting forest fire susceptibility in Yunnan Province, China. This model employs spatial prediction techniques to assess fire risks across different regions, providing valuable insights into areas prone to fires. The predictive capabilities of their model are instrumental in implementing preventive measures and enhancing fire management strategies.

Faroudja Abid et al. [7] conducted a thorough survey of machine learning algorithms applied to forest fire prediction and detection systems. Their review offers a comprehensive look at various approaches and their effectiveness, contributing to a deeper understanding of the techniques available for fire detection and prediction. This survey provides valuable insights into the strengths and limitations of different machine learning algorithms in the context of forest fires.

Mahreen Zainab et al. [8] analysed the implementation of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) on field-programmable gate arrays (FPGAs). Their focus was on optimization techniques to enhance model performance, comparing CNNs and RNNs in terms of efficiency and effectiveness when deployed on FPGA hardware. This comparison highlights the benefits and trade-offs of each approach, offering valuable insights into optimizing deep learning models for real-time applications.

Laith Alzubaidi et al. [9] proposed a holistic review of advancements in deep learning, aiming to provide a comprehensive overview of recent developments and innovations in the field. Their work seeks to offer an in-depth understanding of the latest enhancements and their impact on deep learning applications, contributing to a broader perspective on the progress and future directions in the field.

Connor Shorten et al. [10] focused on data augmentation as a crucial technique for addressing the challenge of limited data in deep learning. Their examination of various data augmentation strategies details methods to expand and enhance training datasets, ultimately improving the performance and accuracy of deep learning models. This work underscores the importance of effective data augmentation in developing robust and reliable models.

3. Theory

The research paper on "A Comparative Study of CNN Models Built with TensorFlow and Theano for Forest Fire Detection" involved the implementation of various python modules and APIs to develop and evaluate the proposed detection system.

3.1 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are among the most prominent deep learning models and have demonstrated remarkable success in computer vision tasks. Originally developed for image recognition, CNNs have since found applications in various fields, including object and face recognition, autonomous driving, and drone navigation. The architecture of a CNN typically consists of two main components: the feature extraction layers and the classification layers. Figure 1 illustrates a typical CNN architecture, highlighting these key components. The feature extraction part includes convolutional and pooling (or subsampling) layers. These layers analyse small sections of the image to extract essential features such as edges, shapes, or textures. Convolutional layers apply filters to the image to identify these features, while pooling layers reduce the spatial dimensions of the data, summarizing the detected features and enhancing the network’s efficiency.

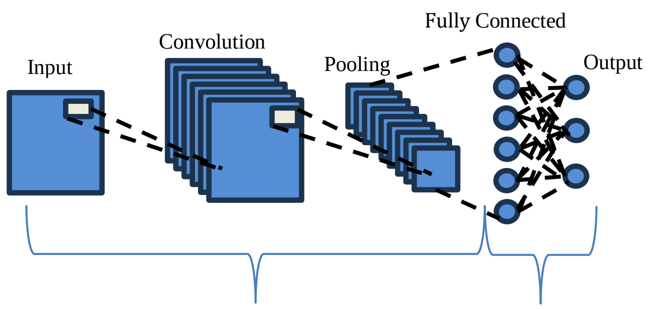

Figure 1: A typical CNN architecture

In the CNN architecture depicted in Figure 1, the first convolutional layer detects various features from the input image. This layer consists of multiple feature maps, each focusing on different aspects of the image. The subsequent pooling layer down samples the data, preserving the most significant features, which helps reduce computational complexity and mitigate overfitting. This process of alternating between convolution and pooling layers continues until the final pooling layer is reached. Following the feature extraction phase, the network transitions to the fully connected (FC) layers. These layers aggregate the features identified by the convolutional and pooling layers and perform the final classification. The output layer of the CNN delivers the final classification result based on the aggregated features.

A key challenge in deploying CNN models on embedded devices is their substantial requirement for computational resources and memory. While more complex CNN architectures can handle sophisticated tasks, they demand significant processing power and storage, which can be a limitation for embedded systems.

3.2 VGG 16

One of the greatest vision model architectures to date is the convolution neural network (CNN) architecture known as VGG16.

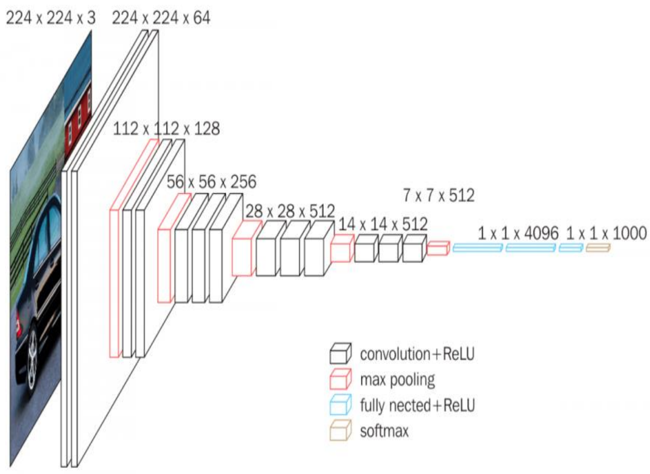

Figure 2: VGG16 architecture

VGG16, as depicted in Figure 2, is distinguished by its use of convolutional layers with 3x3 filters and a stride of 1, combined with max pooling layers featuring 2x2 filters and a stride of 2. Unlike models with numerous hyper-parameters, VGG16 maintains a straightforward and consistent design throughout. The network consists of a series of convolutional and max-pooling layers stacked in a uniform pattern. At the end of the network, there are two fully connected layers, which are followed by a softmax layer for producing the final output. The 16 in VGG16 refers to the total number of weight-bearing layers in the network. With approximately 138 million parameters, VGG16 is a large and robust network, capable of handling complex image recognition tasks.

3.3 Transfer Learning

Transfer learning (a technique in which knowledge learned from a task is re-used in order to boost performance on a related task) and hyperparameter tweaking were used to create the model, which was then trained on a dataset of normal and enhanced photos.

4. Experimental Method and Evaluation

The research paper on "A Comparative Study of CNN Models Built with TensorFlow and Theano for Forest Fire Detection" involved the implementation of various modules to develop and evaluate the proposed detection system.

4.1 Forest Fire Dataset
Forest geography photos were analysed to find the presence or absence of fire using the CNN VGG16-based algorithm. Images from various open-source sources were combined with publicly available datasets from Kaggle to create the dataset of 5062 images including different scenarios (Figure 3). The domain of forest fire management is still under exploration, hence the data available is limited.

Figure 3: Sample of forest fire dataset used

4.2 Dataset Preprocessing
Data preprocessing helps in converting the data into standard uniform format. The dataset created by us has images of varying dimensions and colour codes. Since the model design is based on VGG16, the input to it should be 224x224. Hence all the images in the dataset were reshaped to 224x224 dimension. The colour code of all the images were converted to RGB format. The final images were then stacked in the form of an array and their labels Fire or No Fire were defined accordingly.

4.3 CNN model
The model was developed using the VGG16 architecture. The FC layer network, which was created by experimenting with several hyperparameter combinations and choosing the one that provided the greatest accuracy, optimised learning, and quickest reaction time, performed the detection and classification.

We have developed a fully connected dense model by utilizing the VGG16 architecture, a distinguished pre-trained convolutional network, enhanced through Transfer Learning (TL). TL is a technique that enables us to apply knowledge acquired from solving one problem to a related task, thereby improving the model's effectiveness. By incorporating VGG16's pre-trained features, we can adapt its strengths to meet the specific demands of our project. The model architecture begins with the VGG16 base and is further refined by adding five dense layers, each followed by a dropout layer, culminating in a final layer dedicated to detection or classification, as illustrated in Figure 4.

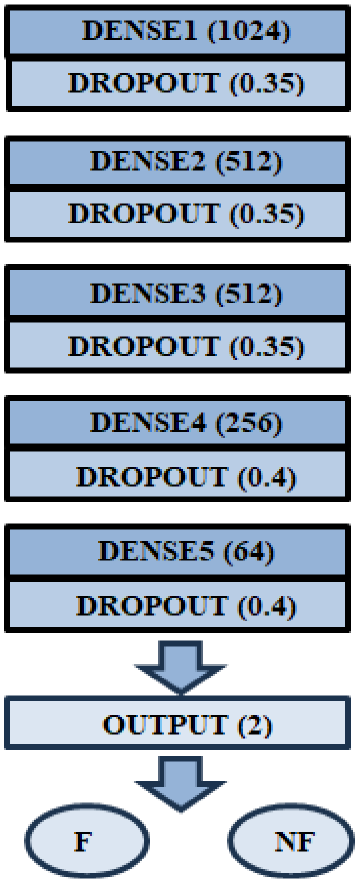

Figure 4: Architecture of customized model built on VGG16 base

- The CONV layer performs a dot product between two matrices, one of which is the kernel—a set of learnable parameters—and the other of which is the constrained region of the receptive field. VGG16 architecture uses CONV layers of different number of channels but 3x3 kernel. It totally has 13 CONV layers (Figure 2)
- The POOLING layer downsampled the input to reduce its dimensions. Max pooling is used here to retain the most prominent features during extraction. Total of 5 Max Pool layers are used (Figure 2)
- The Customized network consisted of 5 DENSE layers or fully connected layers with 1024, 512,512,256 and 64 neurons in every layer respectively.
- DROPOUT layers were added to avoid overfitting and boost accuracy. In order to avoid overfitting, the Dropout layer randomly sets input units to 0 at a frequency of rate at each step during training. The total of all inputs is maintained by scaling up non-zero inputs by 1/(1 - rate). The dropout rates for the first three layers are 0.35 and for the final two layers they are 0.4.
- The DENSE and CONV layers used ReLU for their activation.
- The final detection layer classified the images into two classes: FIRE and NO FIRE using softmax activation and binary cross entropy to get loss.

4.4 Training and Evaluation
The dataset is divided into train and test data using the after the model has been built. 80% of the data is used for training

and 20% for testing. Adam optimizer and binary cross entropy loss are used to train the model.

The actual class output, which is binary—either 0 or 1—is compared to the predicted probabilities using binary cross-entropy. To optimize the model, we use Adam, an alternative optimization technique that refines neural network weights through repeated iterations of "adaptive moment estimation." Unlike many other optimization methods, Adam enhances stochastic gradient descent, allowing it to solve non-convex problems more efficiently, with faster convergence and lower resource demands.

The concept of Early stopping is applied. It is a form of regularization technique that provides guidance as to how many iterations can be run before the learner or model begins to over-fit. The effectiveness of the resulting model is then assessed by applying it to fresh data.

A new image collected from other open-sources serves as the inference data, on which the model's classification accuracy is evaluated. Similar to training data, this data is similarly pre-processed before being supplied to the class prediction model. The training accuracy and loss over 10 epochs was monitored. The training was carried out in a Jupyter notebook using the PC's i5 core processor.

4.5 Comparison

The complete model was designed using high level Keras APIs. Keras needs a backend for its implementation. We created our model with Keras using TensorFlow as its backend and also Keras with Theano as its backend.

Theano is a Python library and optimising compiler for manipulating and analysing mathematical expressions, particularly those containing the matrix-values. TensorFlow is a software that machine learning programmers use to contribute to open source. Machine learning applications make use of the symbolic math library TensorFlow.

Implementation with TensorFlow is much simpler and more debug-friendly in terms of design, as TensorFlow supports all the features of the latest Keras version. In contrast, implementation with Theano is comparatively more complex, as it requires using downgraded versions of Keras (since versions above 2.4 use TensorFlow as the backend, and switching to Theano is not compatible) as well as other Python libraries like NumPy, Pandas, and OpenCV.

The two models were compared based on their accuracy, loss parameters and the time to train as well as response time. The results are discussed in the next section.

5. Results and Discussion

In this section, several parameters like accuracy, training loss, training time and response time on the inference data were monitored and tabulated (for training of 10 epochs) in Table 1.

The model with TensorFlow backend gave an accuracy of 97.6%. When the model was used for inference on fresh data of 380 images, it was able to classify images in a total time of 1693.43s .

The model with Theano backend gave an accuracy of 97.54%. When the model was used on the same inference data, it was able to classify images in total time of 3975.13s. The training loss and accuracy plots against epochs are shown in Figure 5 and Figure 6 respectively.

Table 1: Comparison of TensorFlow and Theano based models for Training accuracy, loss and timing

Criteria	Model TensorFlow	Model Theano
Accuracy	0.976	0.975
Loss	0.14	0.18
Training time	1745s/epoch	2258s/epoch
Response time	1693.43s	3975.13s

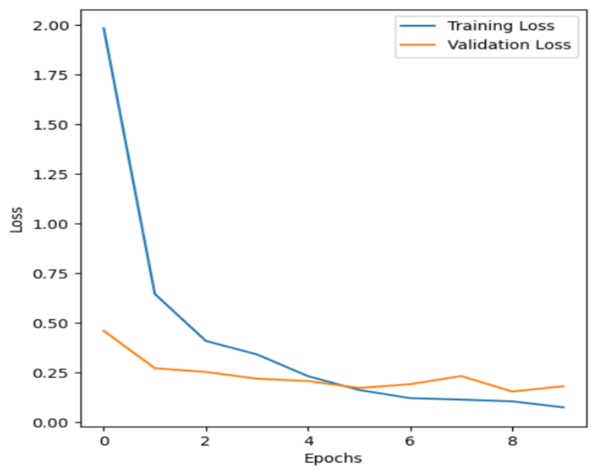

Figure 5: Loss vs Epochs for Theano model for 10 epochs

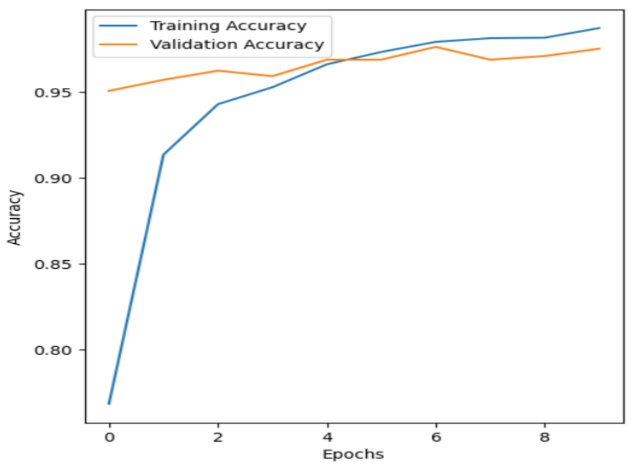

Figure 6: Accuracy vs Epochs for Theano model for 10 epochs

The inference data had 380 images with 190 images related to fire and no fire each. Figure 7 shows the confusion matrix for TensorFlow model and Figure 8 is the confusion matrix for Theano model. The confusion matrix for the Theano model reveals that the model correctly classified 184 fire images and 173 non-fire images out of the 380 images tested. However, it misclassified 23 images in total, with 17 false positives and 6 false negatives. This slightly higher number of misclassifications, compared to the TensorFlow model, indicates a marginally lower performance in distinguishing between fire and non-fire images.

When comparing the confusion matrices of the TensorFlow and Theano models, it becomes evident that the TensorFlow model performs better, particularly in reducing the number of false positives. The TensorFlow model misclassified 20 images (16 false positives and 4 false negatives), whereas the Theano model misclassified 23 images (17 false positives and 6 false negatives). This indicates that while both models are effective, the TensorFlow model offers a slight edge in accuracy and precision, particularly in correctly identifying non-fire images.

The differences in the confusion matrices also suggest that the TensorFlow model is slightly better at avoiding false positives, which is critical in applications where false alarms could have significant consequences. This could be attributed to TensorFlow's more efficient handling of complex data or its superior optimization techniques.

		Actual	
		Fire	No Fire
Predicted	Fire	186	16
	No Fire	4	174

Figure 7: Confusion matrix for TensorFlow model

		Actual	
		Fire	No Fire
Predicted	Fire	184	17
	No Fire	6	173

Figure 8: Confusion matrix for Theano model

The statistics of the working of both the modules (in percentage) on this data is presented in Table 2.

Table 2: Comparison based on evaluation metrics

Metric	Model TensorFlow	Model Theano
Accuracy	94.74	93.94
Sensitivity	97.89	96.84
Specificity	91.16	91.05
Precision	92.08	91.54
F measure	94.89	94.11

Hence, from the results we can conclude that both the models are well trained but Keras with TensorFlow backend gives better results with respect to both time and evaluation or classification compared to Keras with Theano backend.

6. Conclusion and Future Scope

Many researchers have tried to employ various deep learning techniques to identify forest fires. Deep learning systems are still unable to predict forest fires with sufficient accuracy and in a timely manner. In conclusion, this study suggested a multilayered deep convolutional neural network for the detection and classification of natural catastrophes to overcome these issues.

The overall suggested model uses an image dataset to identify and categorise whether there is or is not a forest fire (i.e., there is).The model was trained on an i5 core CPU and created using Python libraries. The architecture and weights of the trained model were retained.

This model's primary goal is to quickly and accurately identify and categorise forest fires with a high rate of accuracy.

Due to its multilayered nature, the suggested model outperformed other cutting-edge techniques in terms of accuracy. The model was 97.61% accurate with a 14.36% loss. The suggested approach performs much better at detecting and categorising forest fires.

In future, this model can be developed to include sensor data as annotations in image for better detection. Also, the model can be modified to account for different degrees of fire for better management and prevention of disaster.

Data Availability

The data used in this research paper are from forest fire datasets available on Kaggle. We have also considered many images from Google which have low resolutions in order to train and improve the accuracy of our model better. The authors are committed to promoting transparency and reproducibility in research and will make reasonable efforts to accommodate data requests within the limitations imposed by data privacy and confidentiality.

Conflict of Interest

The authors whose names are listed in the paper certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Funding Source

There is no funding for this research paper.

Authors' Contributions

All three authors contributed equally through their sound knowledge of convolutional neural networks and implementation using Keras to develop the whole design of the model. Nesiga A's contribution includes preparation of the training, testing and inference datasets by collecting several images with different resolution, angle, scenarios etc. She also contributed to the data pre-processing. Sahani S Shetty's contribution includes setting up and installing the python modules to ensure compatibility with Tensorflow and Theano. She also contributed in debugging the model designs and in the monitoring of the training of the models. Mythili Mahesh Velapakam's contribution included calculating the final evaluation metrics for both the models based on the confusion matrix obtained. She also contributed in creating the first draft of this manuscript and organizing the results.

Acknowledgements

We express our heartfelt gratitude to our guide Dr. Kiran Bailey and reviewers, Mrs.Sujatha K, Dr. Poornima.G for their invaluable guidance and support. Special thanks to Dr. Geetishree Mishra who helped us throughout, leading to the successful completion of the project. We also extend our thanks to the department of Electronics and Communication, BMS College of Engineering, authors of relevant literature, friends, and family for their contributions, resources, assistance, and encouragement throughout this research on " A Comparative Study of CNN Models Built with TensorFlow and Theano for Forest Fire Detection".

References

[1] M. Aamir *et al.*, "Natural Disasters Intensity Analysis and Classification Based on Multispectral Images Using Multi-Layered Deep Convolutional Neural Network," *Sensors*, Vol.**21**, No.8, pp.**2648**, **2021**.

[2] F. Guede-Fernández, L. Martins, R. V. de Almeida, H. Gamboa, and P. Vieira, "A Deep Learning Based Object Identification System for Forest Fire Detection," *Fire*, Vol.**4**, No.4, pp.**75**, **2021**.

[3] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A Forest Fire Detection System Based on Ensemble Learning," *Forests*, Vol.**12**, No.2, pp.**217**, **2021**.

[4] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev, "Early Forest Fire Detection Using Drones and Artificial Intelligence," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp.**1060-1065**, **2019**.

[5] Abdelmalek Bouguettaya, Hafed Zarzour, Amine Mohammed Taberkit, Ahmed Kechida, "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms," Signal Processing, Vol.**190**, pp.**108309**, **2022**.

[6] Zhang, G., Wang, M., & Liu, K., "Forest Fire Susceptibility Modeling Using a Convolutional Neural Network for Yunnan Province of China," Int J Disaster Risk Sci, Vol.**10**, pp.**386–403**, **2019**.

[7] Abid, F., "A Survey of Machine Learning Algorithms Based Forest Fires Prediction and Detection Systems," Fire Technol, Vol. **57**, pp.**559–590**, **2021**.

[8] M. Zainab, A. R. Usmani, S. Mehrban, and M. Hussain, "FPGA based implementations of RNN and CNN: a brief analysis," In 2019 International Conference on Innovative Computing (ICIC), pp.**1-8**, **2019**.

[9] Alzubaidi, L., Zhang, J., Humaidi, A. J., et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," J Big Data, Vol.**8**, pp.**53**, **2021**.

[10] Bogacz, M., and A. Qouneh, "Convolution Neural Network on BeagleBone Black Wireless for Machine Learning Applications," 2022 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, USA, pp.**1-4**, **2022**.

[11] Shorten, C., and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, Vol.**6**, No.1, pp.**1-48**, **2019**.

[12] N. T. Toan, P. Thanh Cong, N. Q. Viet Hung, and J. Jo, "A deep learning approach for early wildfire detection from hyperspectral satellite images," 2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA), pp.**38-45**, **2019**.

[13] D. Rashkovetsky, F. Mauracher, M. Langer, and M. Schmitt, "Wildfire Detection From Multisensor Satellite Imagery Using Deep Semantic Segmentation," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol.**14**, pp.**7001-7016**, **2021**.

[14] M. Park, D. Q. Tran, D. Jung, and S. Park, "Wildfire-Detection Method Using DenseNet and CycleGAN Data Augmentation-Based Remote Camera Imagery," Remote Sensing, Vol.**12**, No.**22**, pp.**3715**, **2020**.

[15] K. Govil, M. L. Welch, J. T. Ball, and C. R. Pennypacker, "Preliminary Results from a Wildfire Detection System Using Deep Learning on Remote Camera Images," Remote Sensing, Vol.**12**, No.1, pp.**166**, **2020**.

[16] Shruthi G, Disha Bhat, Gagana H, Dimple M K, Kavitha, "*Forest Fire Detection Using Convolutional Neural Networks*", International Journal of Computer Sciences and Engineering, Vol.**7**, Special Issue.**14**, pp.**323-325**, **2019**.

AUTHORS PROFILE

Nesiga A is currently a graduate student at Department of Electronics and Communication Engineering at BMS College of Engineering, Bengaluru, India (Affiliated to Visvesvaraya Technological University (VTU), Belagavi, India). Her areas of interest include web development, VLSI, data engineering, and cloud infrastructure. She is very enthusiastic in acquiring more knowledge about the fields of machine learning and AI.

Sahani S Shetty is currently a graduate student at Department of Electronics and Communication Engineering at BMS College of Engineering, Bengaluru, India (Affiliated to Visvesvaraya Technological University (VTU), Belagavi, India). Her major interests include Digital design, VLSI, Machine learning and Deep Learning. She is an eager observer and has curiosity and interest in exploring new technologies especially in the VLSI and DL domains.

Mythili Mahesh Velapakam is currently a graduate student at Department of Electronics and Communication Engineering at BMS College of Engineering, Bengaluru, India (Affiliated to Visvesvaraya Technological University (VTU), Belagavi, India). Her areas of interest include learning new programming languages, web development and Artificial Intelligence as a whole. She is an enthusiastic learner and wants to explore all the new domains.

Dr. Kiran Bailey is a seasoned academician with over 25 years at BMS College of Engineering, specializing in curriculum design, lab setup, and industry collaboration. With more than 35 publications and several patents, she currently supervises four Ph.D. scholars and seeks government funding for research projects. She is also instrumental in establishing the Network Embedded Lab and Incubation Center to promote student innovation.

Dr. Geetishree Mishra is a seasoned academician with over 12 years of experience in Electronics and Communication Engineering. She has 7 years of industry experience with C-DOT switching systems and server management software. Her research focuses on RTOS scheduling, and she has received awards for her technical papers. She has programming skills in Data Science and Machine Learning, has set up advanced labs, and co-authored a textbook on Embedded Systems. She is a VTU-recognized PhD supervisor and holds a Postgraduate degree in AI and Machine Learning from the University of Texas at Austin (UTA).

