

Article

Optimizing Deep Learning Models for Fire Detection, Classification, and Segmentation Using Satellite Images

Abdallah Waleed Ali * and Sefer Kurnaz

Electrical and Computer Engineering (ECE), Engineering Technical, Altinbas University, 34217 Istanbul, Turkey; sefer.kurnaz@altinbas.edu.tr

* Correspondence: abodwaleed806@gmail.com or essamkarkh@gmail.com

Abstract: Earth observation (EO) satellites offer significant potential in wildfire detection and assessment due to their ability to provide fine spatial, temporal, and spectral resolutions. Over the past decade, satellite data have been systematically utilized to monitor wildfire dynamics and evaluate their impacts, leading to substantial advancements in wildfire management strategies. The present study contributes to this field by enhancing the frequency and accuracy of wildfire detection through advanced techniques for detecting, classifying, and segmenting wildfires using satellite imagery. Publicly available multi-sensor satellite data, such as Landsat, Sentinel-1, and Sentinel-2, from 2018 to 2020 were employed, providing temporal observation frequencies of up to five days, which represents a 25% increase compared to traditional monitoring approaches. Sophisticated algorithms were developed and implemented to improve the accuracy of fire detection while minimizing false alarms. The study evaluated the performance of three distinct models: an autoencoder, a U-Net, and a convolutional neural network (CNN), comparing their effectiveness in predicting wildfire occurrences. The results indicated that the CNN model demonstrated superior performance, achieving a fire detection accuracy of 82%, which is approximately 10% higher than the best-performing model in similar studies. This accuracy, coupled with the model's ability to balance various performance metrics and learnable weights, positions it as a promising tool for real-time wildfire detection. The findings underscore the significant potential of optimized machine learning approaches in predicting extreme events, such as wildfires, and improving fire management strategies. Achieving 82% detection accuracy in real-world applications could drastically reduce response times, minimize the damage caused by wildfires, and enhance resource allocation for firefighting efforts, emphasizing the importance of continued research in this domain.



Academic Editor: Grant Williamson

Received: 6 November 2024

Revised: 1 December 2024

Accepted: 1 January 2025

Published: 21 January 2025

Citation: Ali, A.W.; Kurnaz, S. Optimizing Deep Learning Models for Fire Detection, Classification, and Segmentation Using Satellite Images. *Fire* **2025**, *8*, 36. <https://doi.org/10.3390/fire8020036>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: wildfire; active fire detection; deep learning; semantic segmentation

1. Introduction

Forest fire research is crucial for understanding the origins of forest fires and determining reasons for further investigation. Forest fires remain highly uncontrollable events that inflict significant disruption on entire ecosystems, necessitating examination through remote sensing technology. The motivation for this research stems from the imperative to protect forests against the devastating impact of forest fires. Human activities profoundly influence biological resources, contributing to the deterioration of biodiversity. Effective forest management tools are essential for safeguarding biodiversity. With its diverse flora and fauna, India faces the challenge of forest degradation resulting from fires and other activities, posing a threat to animal habitats [1]. Legislation has been enacted to protect

wildlife, plants, animals, birds, and everything associated with wildlife, emphasizing the need to preserve the country's biological and environmental safety. Article 8 of the Convention on Biological Diversity emphasizes the establishment and management of protected areas, promoting resource efficiency, and ensuring the protection and restoration of ecosystems to safeguard a nation's biodiversity [2].

The detection and monitoring of wildfires have gained significant attention due to their impact on the environment, economy, and public safety. The literature provides a comprehensive overview of various methods and technologies employed for wildfire detection, with a focus on advancements in remote sensing and deep learning approaches. Recent data from the Forest Survey of India [1] indicate a significant increase in forest fire incidents across India, with a 2.7-fold rise compared to previous years [2]. This trend underscores the need for more effective monitoring and early detection systems. The rise in wildfire occurrences has been linked to climatic factors, land-use changes, and human activities. Traditional methods of fire detection, such as ground-based observations and manual reporting, have limitations in terms of coverage and response time, necessitating the adoption of advanced technologies like remote sensing. Remote sensing has emerged as a crucial tool for monitoring forest fires, offering broad coverage and the ability to detect fires in inaccessible regions. Platforms such as Landsat-8, which provides high-resolution imagery and thermal data, have been widely utilized in wildfire detection and monitoring [3].

The use of satellite imagery allows for real-time tracking and assessment of fire dynamics, aiding in decision-making for firefighting and resource allocation. Optical remote sensing techniques, which involve the use of satellite or aerial imagery to monitor changes in vegetation and surface temperatures, have shown promise in early fire detection. Barmpoutis et al. [4] reviewed various optical remote sensing systems, highlighting their effectiveness in identifying fire hotspots and smoke plumes, which are early indicators of wildfires. However, challenges remain, including false alarms due to cloud cover or other atmospheric conditions. Recent advancements in deep learning have led to significant improvements in the accuracy and speed of wildfire detection. Several studies have explored the application of deep neural networks (DNNs) and convolutional neural networks (CNNs) for wildfire detection using various data sources, including satellite imagery, UAV footage, and remote camera feeds [5].

Toan et al. [6] proposed a deep learning approach utilizing hyperspectral satellite images for early wildfire detection, which demonstrated high accuracy in identifying fire-prone areas. Similarly, Lee et al. [7] employed deep neural networks with UAV-based imagery, achieving effective fire detection in remote regions. The use of UAVs provides an additional advantage of flexibility and high-resolution data, though challenges such as limited battery life and flight range persist. Deep learning techniques have also been employed for segmenting wildfire regions in satellite images [8]. Khryashchev and Larionov [9] applied deep learning algorithms for segmentation tasks, demonstrating the capability to accurately delineate fire-affected areas. Ganesan et al. [10] compared various segmentation methods for forest fire regions in high-resolution satellite images, further emphasizing the importance of precise segmentation in early fire detection. Deep-learning-based segmentation models, such as U-Net and Mask R-CNN, have shown substantial improvements over traditional methods by leveraging hierarchical feature extraction to capture complex fire patterns [11]. Wang et al. [11] utilized deep learning techniques for early forest fire region segmentation, reporting significant advancements in accuracy. Machine learning algorithms have been utilized not only for detecting fires but also for predicting their spread and behavior. Priya and Vani [12] developed a deep-learning-based classification system for identifying different types of fire occurrences in satellite images. Predictive modeling

techniques, such as Bayesian networks and random forests, have been used to forecast fire spread, incorporating variables like wind speed, temperature, and vegetation type [13,14]. Khakzad [13] modeled wildfire spread using a dynamic Bayesian network, demonstrating its effectiveness in capturing complex interactions between variables in wildland-industrial interfaces. Similarly, Sayad et al. [14] introduced a new dataset and employed machine learning methods to enhance wildfire prediction capabilities. Combining data from multiple sources, such as satellite imagery, UAVs, and ground-based sensors, can significantly improve the accuracy of wildfire detection systems. Govil et al. [15] reported preliminary results from a wildfire detection system that integrated deep learning algorithms with remote camera images, demonstrating promising outcomes in early fire identification. The fusion of multi-sensor data allows for a more comprehensive analysis of fire events, enabling better situational awareness and resource management [16,17]. Despite the advancements in wildfire detection technologies, several challenges persist. The reliability of remote sensing-based systems can be affected by environmental factors such as cloud cover and atmospheric disturbances [18]. Moreover, deep learning models require extensive training datasets and computational resources, which may limit their deployment in real-time applications. Future research should focus on developing more robust algorithms that can operate under varying conditions and integrating emerging technologies such as hyperspectral imaging and LiDAR for improved fire detection capabilities [19]. Table 1 below summarizes the main contributions and limitations in this field.

Table 1. Summary of wildfire detection methods in the literature.

Authors	Method	Contributions	Limitations
Forest Survey of India [1]	Analysis of forest fire incident data	Provided an overview of forest fire incidents in India, highlighting a 2.7-fold increase	Limited to data analysis without proposing detection methods
Barmpoutis et al. [4]	Review of optical remote sensing systems	Identified effective early indicators (hotspots, smoke plumes) for fire detection	Prone to false alarms due to atmospheric disturbances
Toan et al. [6]	Deep learning with hyperspectral satellite images	Demonstrated high accuracy in early detection of fire-prone areas	Requires extensive computational resources for hyperspectral data
Lee et al. [7]	Deep neural networks with UAV-based imagery	Achieved effective fire detection in remote areas	Limited by UAV battery life and flight range
Khryashchev and Larionov [9]	Deep-learning-based image segmentation	Showed accurate delineation of fire-affected regions	Performance can degrade in low-quality or noisy images
Priya and Vani [12]	Deep learning classification system for satellite images	Developed a system to classify various types of fire occurrences	High dependency on labeled datasets for training
Govil et al. [15]	Integration of deep learning and remote camera images	Enhanced detection accuracy through multi-sensor data fusion	Performance affected by environmental conditions (e.g., weather, lighting)
Khakzad [20]	Predictive modeling using Bayesian networks	Modeled fire spread dynamics using variables like wind speed and temperature	Requires extensive input data to maintain model accuracy
de Almeida Pereira et al. [21]	Active fire detection in Landsat-8 imagery using deep learning	Created a large-scale dataset for training deep learning models for fire detection	Sensitive to cloud cover and other obstructions in satellite imagery
Sayad et al. [19]	Machine learning with a new dataset for wildfire prediction	Introduced a new dataset for predictive modeling, improving forecast accuracy	May require further validation across different geographical regions

This present study aimed to utilize publicly available large-scale multi-sensor satellite data to develop and implement advanced algorithms for the accurate detection, classification, and segmentation of fire outbreaks. The primary focus was to create an automated fire detection algorithm that leverages satellite imagery to enhance recall, precision, and accuracy while minimizing false alarm rates (commission errors) and maintaining efficient processing times.

The present study involves designing a machine learning architecture aimed at achieving near-real-time capabilities. To evaluate the effectiveness of our wildfire risk assessment model, we compared it with other advanced models, with the goal of training a system capable of accurately assessing wildfire risk across diverse landscapes.

This paper investigates the wildfire risk through the lens of an image segmentation task, wherein the model assesses the susceptibility of each pixel to fire rather than providing a classification for the entire image, resulting in an image mask that delineates areas prone to fires within the region of interest. The primary objectives of this research are as follows:

- To develop and optimize deep learning models for the detection, classification, and segmentation of wildfires using multi-sensor satellite images, focusing on improving real-time prediction capabilities.
- To evaluate the performance of various deep learning architectures, including convolutional neural networks (CNNs), U-Net, and autoencoders, in accurately predicting wildfire risk.
- To reduce false alarms in wildfire detection by implementing advanced loss functions and optimizing model parameters.
- To assess the scalability and applicability of these models in fire-prone regions across different temporal and spatial scales, enhancing wildfire management strategies.

The remainder of this article is organized as follows: Section 2: Materials provides an overview of the data and tools used in this study. It describes the sources and characteristics of the wildfire data and discusses the deep learning techniques employed for fire detection and segmentation. Subsections include details on the wildfire datasets used and the specific deep learning architectures implemented. Section 3: Proposed System introduces the system architecture and methodology used for wildfire detection. It details the dataset preparation process, followed by the architectural design of the three deep learning models: autoencoder, U-Net, and convolutional neural network (CNN).

This section also discusses the loss functions used in model training to optimize performance. Section 4: Experimental Results presents the outcomes of various experiments conducted to optimize the CNN model and assess its variability across different configurations.

This section provides insights into the tuning of hyperparameters and the model's response to different training conditions. Section 5: Results offers a comprehensive evaluation of the models' performance, comparing the CNN, U-Net, and autoencoder architectures. It includes analyses of feature importance, spatial dependence in the data, and the overall accuracy of the proposed system. Section 6: Discussion addresses the implications of the findings, highlights the limitations of the current approach, and provides recommendations for future work. This section also discusses potential improvements to the system and alternative methodologies. Section 7: Conclusion summarizes the key contributions of the study and emphasizes the importance of integrating deep learning techniques with satellite imagery for enhancing wildfire detection and management.

2. Materials

This section provides an overview of the materials used in the study, including the data sources for wildfire detection and the deep learning techniques employed for modeling and prediction.

2.1. Wildfires

Wildfire data were collected from publicly available satellite datasets, including multispectral imagery from the Landsat, Sentinel-1, and Sentinel-2 missions. These satellites were chosen due to their fine spatial, temporal, and spectral resolutions, which are suitable for detecting and monitoring fire-related phenomena such as active fires, burned areas, and smoke. The satellite data covered the period from 2018 to 2020 and included regions known for frequent wildfire occurrences. Data were acquired from the Google Earth Engine (GEE) platform, which provides access to a vast repository of geospatial datasets. However, due to gaps in satellite data and the absence of fire instances in some areas, the data were not globally uniform. To address this, specific samples were selected based on known wildfire events, using the date and location of the incidents to guide the extraction of relevant imagery. This ensured a comprehensive dataset that represents various fire conditions and intensities. To supplement the satellite data, ground-truth information on fire occurrences was obtained from government agencies and wildfire monitoring services. These records helped validate the model predictions by providing reference points for assessing the accuracy of the deep learning models. Additionally, environmental factors such as vegetation type, land cover, and meteorological conditions were considered to improve the accuracy of the models in predicting fire-prone areas.

2.2. Deep Learning

Deep learning techniques were employed to develop models capable of detecting, classifying, and segmenting wildfires from satellite images. Three primary architectures were tested: convolutional neural networks (CNNs), U-Net, and autoencoders. These models were chosen for their proven effectiveness in image analysis tasks, such as semantic segmentation and feature extraction.

- **Convolutional Neural Networks (CNNs):** The CNN architecture consisted of multiple layers designed to progressively extract spatial features from the satellite images. Each layer applied a set of filters to the input data, followed by batch normalization and activation functions to improve the model's learning capability. The CNN model in this study contained approximately 587,177 trainable parameters, making it suitable for handling complex wildfire detection tasks.
- **U-Net Architecture:** U-Net is a popular model for image segmentation that utilizes an encoder-decoder structure. In this study, the encoder reduced the spatial dimensions of the input data while increasing the number of feature channels, while the decoder expanded the dimensions to reconstruct a segmentation map. The U-Net architecture was tested with different configurations to find an optimal balance between accuracy and computational efficiency.
- **Autoencoders:** Autoencoders are unsupervised learning models that encode the input data into a compressed representation and then decode it back to the original form. In this study, the autoencoder was used as a baseline model for wildfire detection, employing a simpler architecture compared to CNNs and U-Nets. The autoencoder's ability to learn compact feature representations was evaluated to determine its effectiveness in identifying fire-prone regions.

To optimize the performance of these models, several loss functions, including Dice loss, binary cross entropy (BCE), and focal loss, were tested. The Dice loss function, which measures the overlap between the predicted and ground truth segmentation, was selected as the primary loss function due to its superior performance in handling class imbalance. Hyperparameters such as learning rate, batch size, and the number of epochs were fine-tuned to achieve the best possible accuracy for each model.

The combination of satellite data and deep learning techniques enabled the development of an advanced system for wildfire detection, capable of real-time monitoring and prediction. The models were trained using a dataset that represented a wide range of fire conditions, which allowed for the evaluation of the models' robustness across different scenarios.

The CNN architecture comprises multiple blocks, each integrating convolutional, batch normalization, and activation layers, as depicted in Figure 1. The U-Net engineering for semantic division utilizes convolutional layers to keep up with the first size of the picture while continuously expanding the number of channels. The architecture consists of four blocks, with the filter counts for each block set as 40, 60, 30, and 1, respectively. The activation functions used in the inner blocks were tanh, while the remaining blocks utilized the ReLU activation function. The final layer employed a sigmoid activation function to transform the outputs of the preceding layers into a single filter. The entire model encompasses 587,177 parameters.

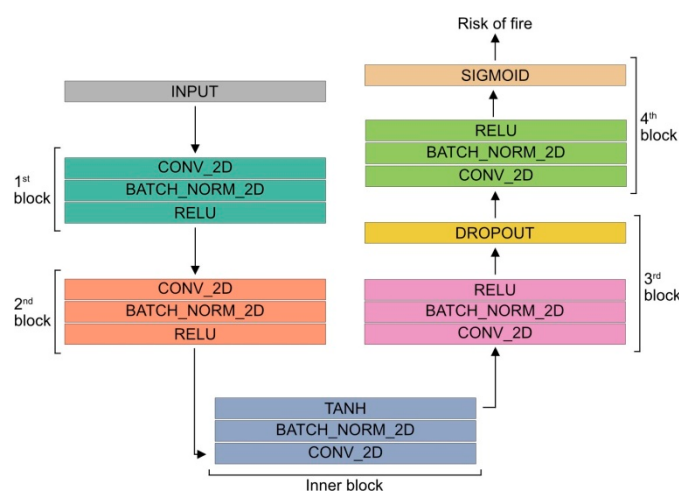


Figure 1. Architecture of the CNN.

3. Proposed System

The proposed system aims to enhance wildfire detection, classification, and segmentation by integrating advanced deep learning techniques with satellite-based Earth observation data. Given the increasing frequency and intensity of wildfires around the world, timely and accurate fire prediction has become a critical need for effective disaster management and mitigation. Traditional methods relying on spectral indices and manual analysis are often limited by fixed thresholds and environmental variability, making it challenging to achieve consistent accuracy across different regions and fire conditions. To address these limitations, this study introduces a system that leverages deep learning models—specifically convolutional neural networks (CNNs), U-Net architectures, and autoencoders—to process multispectral satellite imagery for improved fire detection and segmentation. The system is designed to automatically identify fire-prone areas and classify various fire-related phenomena, such as active fire fronts, burned areas, and smoke, with a high degree of accuracy. By optimizing the architectures and fine-tuning key hyperparameters, the proposed approach seeks to outperform traditional methods and existing models in terms of both speed and precision.

The system's architecture is built to handle the complexities of real-time wildfire monitoring, integrating various pre-processing, training, and prediction modules. It includes data acquisition from multiple satellite sensors, model training using ground-truth data for validation, and post-processing techniques to refine the output segmentation maps. The following subsections provide a detailed description of each component of the system,

including the data preparation workflow, model architectures, training procedures, and evaluation metrics used to assess the system's performance.

3.1. Dataset

The NASA Earth Observing System (EOS) is a comprehensive program designed to study Earth's climate, ecosystems, and atmosphere using a series of advanced satellites and ground-based observations. It focuses on collecting long-term, consistent, and global data for understanding and monitoring environmental and climate processes.

Utilizing Google Earth Engine (GEE), a diverse set of images has been extracted from various regions worldwide from 2018 to 2020. The study emphasized fire seasons in Africa, Asia, Australia, Europe, South America, and the United States (Figure 2) with broad geographical and historical coverage, while accounting for considerations related to missing data. In this study, 20 remote sensing data sources were explored, aiming for a comprehensive representation. Table 2 presents details on the spatial and temporal resolution of these features. Figure 2 below illustrates global patterns and trends in forest loss and associated dynamics across different regions from 2000 to 2020.

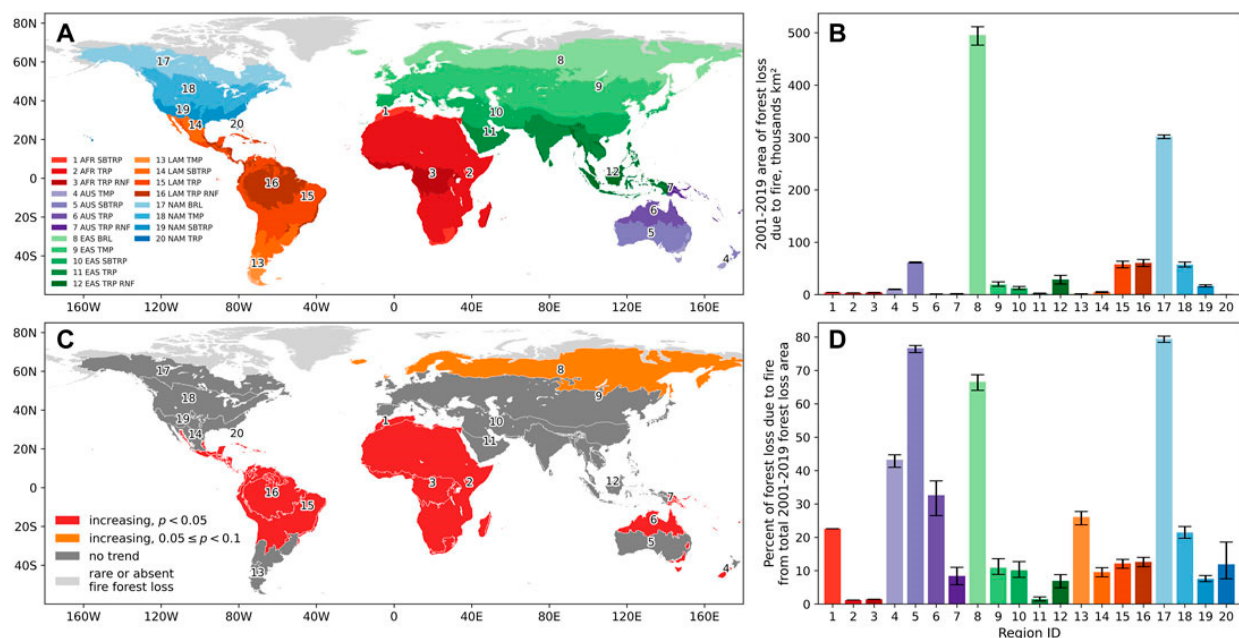


Figure 2. Spatial distribution of training and validation samples, where (A) presents a map categorizing regions based on the rate of forest loss, with color coding representing various loss levels, and numbered regions corresponding to those analyzed in Panel (B). Panel (B) shows a bar chart detailing the total forest loss for each region (Region IDs) with error bars indicating variability or uncertainty in the data. Panel (C) provides a map highlighting trends in forest dynamics, such as increasing or decreasing rates, statistical significance (e.g., $p < 0.05$), and areas with no trend, while gray regions indicate minimal or absent forest coverage. Correspondingly, Panel (D) presents a bar chart summarizing these trends for each region, indicating whether forest conditions are improving or worsening over time.

Figure 3 and Table 2 provide a comprehensive visualization of environmental variables derived from remote sensing and meteorological data, including topographical parameters (elevation), vegetation indices (LAI, FAPAR, NDVI), climatic factors (land surface temperature, humidity, precipitation, wind components, and air pressure), and soil properties (soil temperature). The figure also includes histograms for variables such as LAI, FAPAR, land surface temperature, soil temperature, precipitation, and humidity, offering statistical insights into their distributions. Additionally, evapotranspiration, fire occurrences, and land

cover classifications are represented, highlighting the diversity of datasets used for environmental modeling and analysis. These datasets are critical for studying land-atmosphere interactions, ecological dynamics, and disaster monitoring.

Table 2. Image features and their spatial and temporal resolutions.

Image Feature	Spatial Resolution	Temporal Resolution
Elevation	30 m/pixel	1 year
HistoryLAI	5566 m/pixel	10 years
HistoryFAPAR	5566 m/pixel	10 years
LST	11,132 m/pixel	daily
HistoryLST	4638 m/pixel	5 days
Soiltemperature	11,132 m/pixel	daily
Historysoiltemperature	11,132 m/pixel	5 days
Dailyprecipitations	5566 m/pixel	daily
Historyprecipitations	5566 m/pixel	5 days
Airpressure	11,132 m/pixel	daily
Winducomponent	11,132 m/pixel	daily
Windvcomponent	11,132 m/pixel	daily
Dailyhumidity	11,132 m/pixel	daily
Historyhumidity	11,132 m/pixel	5 days
DailyLAIhigh	11,132 m/pixel	daily
DailyLAIlow	11,132 m/pixel	daily
DailyNDVI	463 m/pixel	daily
8 daysEvapotranspiration	500 m/pixel	8 days
Historyfire	1000 m/pixel	1 year
Landcover	500 m/pixel	daily

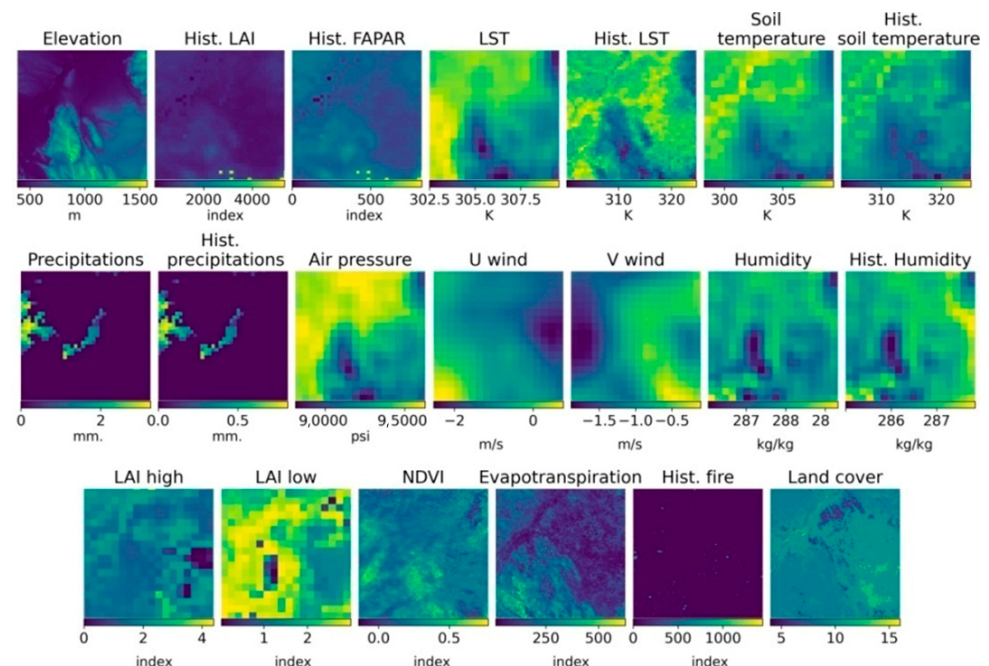


Figure 3. Test by preparing a set of remote-detecting information sources.

3.2. Autoencoder Architecture

The initial architecture for assessing wildfire risk is an autoencoder. The autoencoder comprises six essential blocks, with the initial three being convolutional layers that logically decrease the number of channels, really compacting the picture. The subsequent three layers then increase the number of filters, expanding the image. The filter counts for each respective convolutional layer are 40, 20, 5, 5, 20, and 40. All max-pooling and upsampling layers employed a 2 by 2-pixel kernel. The activation function used across all layers was

tanh, except for the final layer, which employed a sigmoid activation function to convert the outputs of preceding layers into a single filter. The entire model encompassed 36,596 parameters. Figure 4 illustrates the architecture of the autoencoder designed to predict fire risk. The model begins with an input layer, followed by three successive blocks of convolutional layers (CONV_2D), activation functions (TANH), and pooling layers (MAX_POOL_2D) to extract and compress spatial features. The compressed representation is then passed through symmetric decoding layers comprising upsampling layers (UPSAMPLING_2D), activation functions (TANH), and convolutional layers (CONV_2D) to reconstruct the data. Finally, a sigmoid activation function in the output layer maps the decoded features to a fire risk probability. This structure demonstrates the use of autoencoders for feature extraction, dimensionality reduction, and reconstruction, tailored for fire risk prediction.

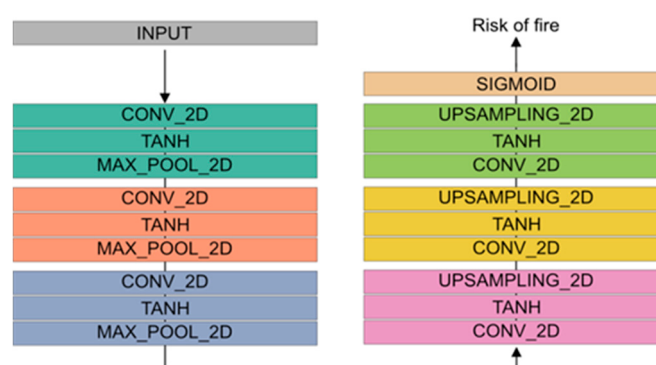


Figure 4. Architecture of the autoencoder.

3.3. U-Net Architecture

The subsequent model utilized was a U-Net, a notable engineering model exceptionally respected for its presentation in tasks like biomedical segmentation. The U-Net architecture consists of five down blocks for compression and four up blocks for expansion. The number of filters for the down blocks ranges from 64 to 1024, while for the up blocks, the filter counts range from 512 to 64. The final layer utilizes a sigmoid activation function to convert the outputs into a single filter.

Convolutional Neural Network Architecture

In the context of this research, the convolutional neural network (CNN) was employed as a deep learning model for predicting fire risk and segmenting wildfire regions in satellite images. The CNN architecture used in this study was designed to optimize the performance in semantic segmentation tasks by leveraging multiple layers to extract relevant features from the input data.

The architecture comprises several sequential blocks, each integrating convolutional layers, batch normalization, and activation functions to progressively refine the feature representation. As depicted in Figure 5, the CNN architecture utilized in this study consists of four main blocks. The filter counts for each block were set to 40, 60, 30, and 1, respectively, to adjust the network's capacity for learning different levels of features. This arrangement was chosen to gradually increase the number of feature maps, enabling the model to capture more complex patterns related to wildfire occurrences. For activation functions, the architecture used the hyperbolic tangent (tanh) function in the inner blocks to maintain the range of the output while allowing negative values, which can help in capturing intricate details. The remaining blocks employed the Rectified Linear Unit (ReLU) activation function, known for its efficiency in deep learning tasks due to reduced likelihood of vanishing gradients. The final layer used a sigmoid activation function to transform the outputs into a single filter, representing the probability of each pixel belonging to the fire risk class. The overall CNN model encompasses 587,177 trainable parameters. This parameter

count was selected to achieve a balance between model complexity and computational efficiency, making the architecture suitable for real-time wildfire detection tasks while maintaining a high level of accuracy.

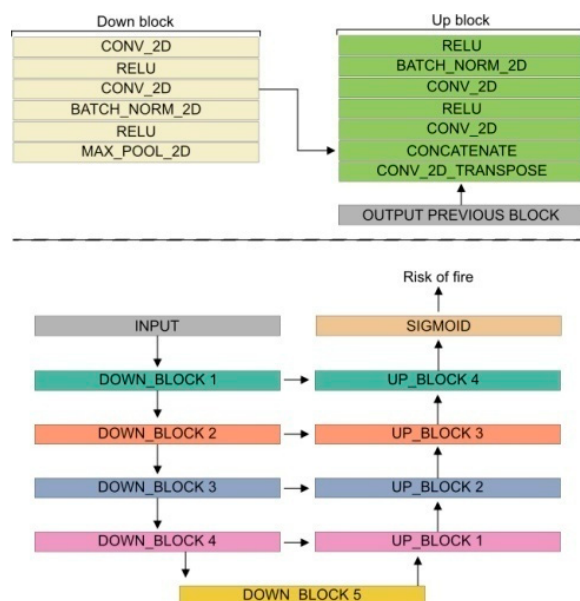


Figure 5. U-Net architecture for semantic segmentation.

U-Net Engineering for Semantic Segmentation

The U-Net architecture, which was also tested in this study, is designed specifically for image segmentation tasks. It uses an encoder–decoder structure where convolutional layers in the encoder path reduce the spatial dimensions while increasing the number of channels. The U-Net maintained the original image size by utilizing convolutional layers with padding, allowing for the retention of spatial information throughout the network. In the decoder path, the network expands the spatial dimensions while reducing the number of channels, enabling the reconstruction of a segmented output that matches the input size. This process facilitates the precise localization of fire risk areas in the satellite images. The comparison between the CNN and U-Net models revealed that while both architectures performed well in segmenting wildfire regions, the CNN architecture demonstrated superior performance with fewer parameters, making it more suitable for real-time applications.

Loss Function

Several loss functions were tested in the U-Net model for semantic segmentation, including mean squared error (MSE), binary cross entropy (BCE), and focal loss. These loss functions are commonly used in machine learning for various tasks, such as regression (MSE), binary classification (BCE), and class imbalance handling (focal loss). However, they yielded unsatisfactory results in this study, indicating their lack of suitability for the specific task of wildfire segmentation. The primary goal of the optimization model in this context was to minimize the discrepancy between the predicted segmentation map and the ground truth, thus maximizing the accuracy of the segmentation. The optimization problem can be expressed as

$$\min_{\theta} L(P, G; \theta), \quad (1)$$

where L represents the loss function, P denotes the predicted segmentation, G is the ground truth segmentation, and θ is the set of parameters (weights) of the model. The choice of an appropriate loss function plays a crucial role in effectively optimizing the model.

After experimenting with different loss functions, Dice loss, derived from the Dice coefficient, demonstrated optimal performance and was consequently implemented across the three previously presented architectures. The Dice loss function is particularly suited for segmentation tasks as it measures the overlap between the predicted and ground truth segmentation. It is defined as

$$\text{Dice Loss} = 1 - (2 \sum p_i g_i) / (\sum p_i^2 + \sum g_i^2), \quad (2)$$

where p_i and g_i represent the predicted and ground truth values, respectively, at each pixel i , and N is the total number of pixels. The Dice loss is effective for addressing class imbalance, which is common in semantic segmentation tasks where the target class (wildfire) is often much smaller than the background.

Boundary Conditions:

For the optimization problem, the boundary conditions were set such that the values of the predicted segmentation probabilities P fall within the range $[0, 1]$, ensuring valid probability outputs. Additionally, the Dice loss function ensures that the output is continuous and differentiable, which is a necessary condition for gradient-based optimization methods used in training the deep learning models.

4. Experimental Results

4.1. Convolutional Neural Network Optimization

A grid search method was employed to optimize the hyperparameters of the CNN architecture, as illustrated in Figure 6. The grid search systematically explored combinations of hyperparameters to identify the configuration yielding the best performance, as shown in Table 3. Key hyperparameters tuned during this process included the learning rate, batch size, dropout rate, number of filters, kernel size, and activation functions for each convolutional layer. The implementation was conducted in Python using TensorFlow on Google Colab, utilizing a single Tesla T4 GPU for accelerated computation. The dataset was preprocessed by normalizing pixel values to the range $[0, 1]$ and applying data augmentation techniques such as random rotations, flips, and zooms to enhance model robustness against overfitting. The dataset was split into training (70%), validation (20%), and testing (10%) subsets. During training, the Adam optimizer was employed due to its efficiency in handling sparse gradients, with an initial learning rate explored in the range of 1.0×10^{-4} to 1.0×10^{-2} . A learning rate scheduler was implemented to reduce the rate by a factor of 0.1 if the validation loss plateaued for more than five epochs. Batch sizes of 16, 32, and 64 were evaluated to balance memory usage and convergence speed. Dropout rates between 0.2 and 0.5 were tested to mitigate overfitting. The CNN architecture consisted of three convolutional layers, each followed by ReLU activation and max-pooling layers, to progressively reduce spatial dimensions while capturing critical features. Fully connected layers at the end of the network were regularized using L2 regularization to prevent overfitting. The final output layer used a sigmoid activation function to predict the binary presence or absence of wildfire. Training was conducted for 50 epochs, with early stopping triggered if validation loss failed to improve for 10 consecutive epochs. Each grid search configuration was evaluated using the F1-score, accuracy, and precision metrics on the validation set to ensure balanced performance across all criteria. The best-performing configuration achieved an 82% detection accuracy on the test set, with additional metrics reported in Table 4. Figure 5 illustrates the formula for the Dice Coefficient, a metric used to evaluate the overlap between predicted segmentation (y_{pred}) and the ground truth (y_{true}) in image analysis tasks, where the numerator represents twice the area of overlap between the predicted and true regions, and the denominator is the sum of the areas of the

predicted and true regions. The visualization uses color coding to enhance understanding: the green region represents the predicted segmentation (y_{pred}), the pink region represents the ground truth segmentation (y_{true}), and their overlap is highlighted to show the intersection between the two. This metric ranges from 0 (no overlap) to 1 (perfect overlap), making it ideal for evaluating segmentation models.

$$\text{Dice Coefficient} = \frac{2 \times (y_{pred} \cap y_{true})}{y_{pred} \cup y_{true}}$$

Figure 6. Graphical representation of the Dice coefficient.

Table 3. Search space for the CNN's parameters.

Parameter	Values
No. inner blocks	{0, 2, 4, 6, 8}
No. Incremental filters	{10, 20}
Dropout	{0, 0.1, 0.3, 0.5}
Learning Rate	{0.01, 0.001, 0.0001}

Table 4. Convolutional neural network parameter optimization.

Model	Fire Ratio	No-Fire Ratio	Dice Coefficient	IOU
Model 1	0.82	0.87	0.0059	0.0029
Model 2	0.80	0.86	0.0059	0.0029

The expression “steady channels” alludes to a philosophy where the number of channels per convolutional layer increases by a proper sum (‘x’) in each progressive layer. To determine the optimal boundary configuration, multiple cycles were conducted over a span of 30 years, with each iteration lasting between 15 and 60 min.

4.2. Convolutional Neural Network Variability

In Figure 7, the thresholding process is used to binarize the predicted output by converting values greater than 0.5 to 1 (indicating fire presence) and values below 0.5 to 0 (indicating no fire). However, this binary approach does not explicitly address cases where the predicted value equals exactly 0.5. To resolve this ambiguity, a consistent approach is required: values equal to 0.5 can be treated as 1 (indicating fire) to ensure that edge cases are not excluded. This choice should be made based on the desired sensitivity of the model, particularly for low-intensity fire detection.

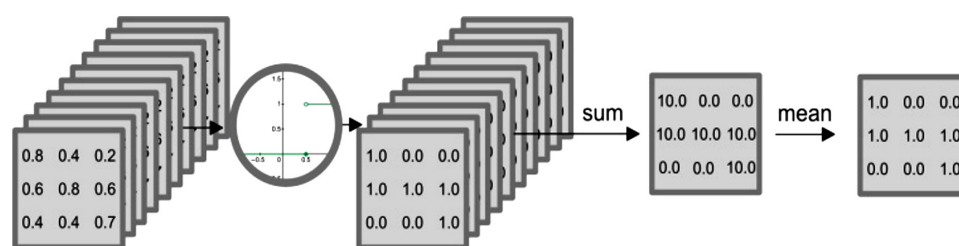


Figure 7. Calculation of convolutional neural network variability from multiple model outputs.

4.2.1. Limitations of the Binary Thresholding Approach

The use of a fixed threshold may lead to significant limitations in modeling fire events with low intensity or uncertain predictions. By setting a threshold at 0.5, the model may overlook important fire instances that fall below this value but still carry relevance for understanding fire dynamics and risks. Low-intensity fires, or fires in their early stages, may produce prediction values that are not high enough to meet the defined threshold, resulting in their exclusion from the detection process.

To address this limitation, alternative approaches can be considered, such as the following:

- **Soft Thresholding:** Instead of a strict binary threshold, use a probabilistic approach where the output value indicates the confidence level of fire presence. This would allow for varying degrees of detection sensitivity based on specific use cases or risk levels.
- **Multi-Class Classification:** Extend the model to predict multiple levels of fire intensity, rather than just binary fire/no-fire classes. This would help in distinguishing between different levels of fire risk and capturing low-intensity fires that may not meet the standard threshold.
- **Adaptive Thresholding:** Implement an adaptive threshold based on the distribution of prediction values in each image. This approach could dynamically adjust the threshold to improve detection in scenarios with varying data quality or different fire conditions.

4.2.2. Recommendations for Improving Fire Detection

To ensure that important fire events are not overlooked, incorporating one or more of the aforementioned approaches could improve the model's ability to detect low-intensity fires while maintaining accuracy for more significant fire occurrences. Furthermore, conducting sensitivity analysis on the chosen threshold would help optimize the model's performance across different types of fires and environmental conditions.

5. Results

Every one of the outcomes introduced in this segment was figured on the approval set, which has been painstakingly built to consider test cross-over and is autonomous of the preparation set. The model's performance on unseen data was assessed using the validation set ensuring its ability to generalize effectively. Figure 8 depicts five distinct regions where the model predicts rapidly spreading fires. These visual examples provide insights into the model's performance and its ability to detect wildfires in different scenarios. Analyzing predictions on the validation set helps identify the model's strengths and weaknesses, enabling further improvements and fine-tuning.

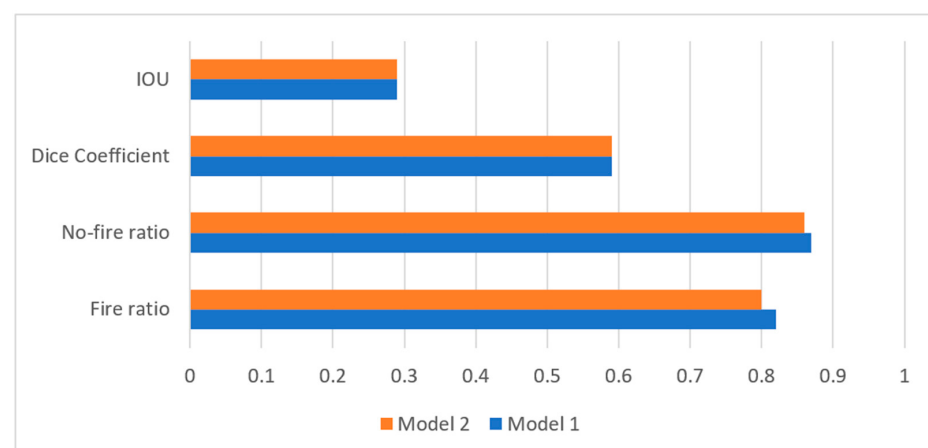


Figure 8. Convolutional neural network parameter optimization.

5.1. Convolutional Neural Network Optimization

The CNN architecture has been carefully designed to achieve near-real-time capabilities. A grid search strategy was employed to optimize the model, using the parameters outlined in Table 3. This detailed optimization process fine-tunes the model's performance, enhancing its effectiveness in real-time applications. By conducting a grid search, different combinations of parameters were systematically tested to identify the optimal configuration. This approach helps to identify the parameter values that yield the best results in terms of accuracy, efficiency, and overall performance. For ease of interpretation, the results (Table 4) were presented with a focus on the varying number of filters. This optimization process enhances the efficiency and responsiveness of the CNN, aligning it with the desired goal of near-real-time performance.

For down-to-earth contemplations, the runs were separated according to the number of channels, explicitly somewhere in the range of 10 and 20 channels. Model 1 was designed with boundaries: 6 blocks, 10 gradual channels, and a dropout of 0.3. Then again, Model 2 was portrayed by boundaries: 4 blocks, 20 gradual channels, and a dropout of 0.5. Initial observations showed that models with no internal blocks and zero dropouts tend to underfit, leading to their exclusion from further in-depth analysis. As a result, an exponentially decaying learning rate, starting at 0.0008, was applied to all optimization runs. Figure 9 illustrates the predictions generated by a Convolutional Neural Network (CNN) for wildfire detection, overlaid with ground truth fire locations. The red regions represent the ground truth data for actual fire occurrences, serving as the benchmark for evaluating the model's predictions. The background color map shows the CNN's prediction confidence, with a gradient scale ranging from 0 (low confidence) to 1 (high confidence), indicated by colors transitioning from light blue (low) to dark blue (high). Each panel displays a different spatial scenario, showing how the CNN identifies regions at risk of fire. Accurate predictions occur when high-confidence areas (darker blue) align with the red ground truth regions, while mismatches highlight either false positives or false negatives. This visualization demonstrates the model's spatial performance and its ability to generalize fire risk predictions across varying environmental conditions.

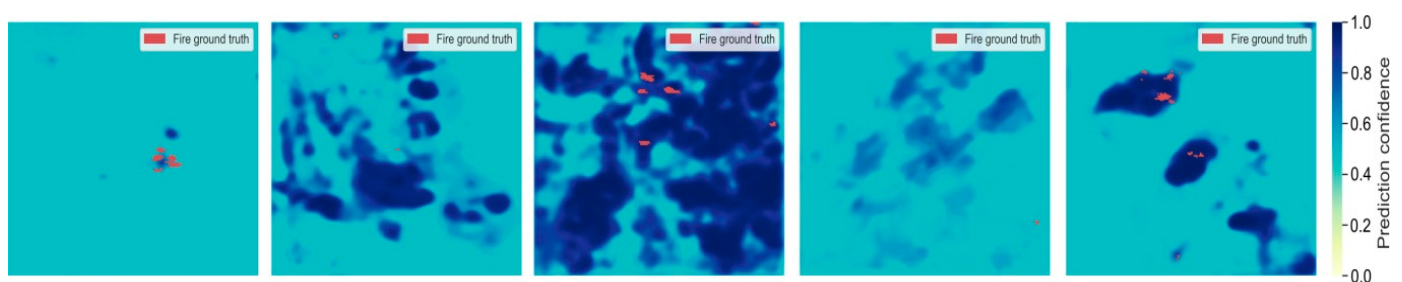


Figure 9. Predictions from the convolutional neural network: fire ground truth and model output.

5.2. Convolutional Neural Network Variability

Ten different CNN models were generated and evaluated using the validation dataset. Figure 10 presents a summary of the findings. This approach ensures that the graph reflects a diverse range of outcomes from various CNN instances, offering a comprehensive view of the model's performance across different runs.

As shown in Figure 10, roughly 20% of the fire examples anticipated by the model were mislabeled, and a comparative circumstance was noticed for the no-fire cases. The bar plot shows that examples are predominantly distributed between the extremes, with fewer cases in the center. This insight provides a quantitative understanding of the model's performance, highlighting areas where misclassifications occur and indicating the distribution of prediction errors across different classes.

In Figure 11, the model's confidence in predicting wildfire risk is visualized using a color gradient from white (low prediction confidence) to dark blue (high prediction confidence), with red areas indicating detected fire instances. The large predicted areas with varying confidence levels reflect the model's uncertainty in certain regions. While broader regions of predicted risk may appear to encompass fire instances, this does not necessarily imply high accuracy.

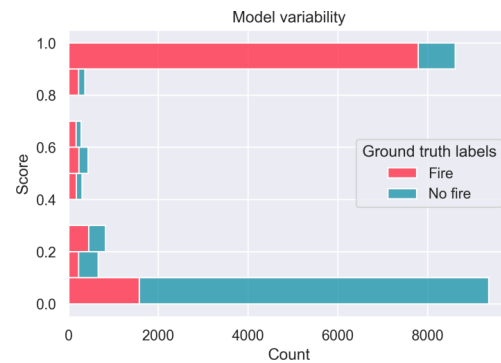


Figure 10. Variability of model predictions: fire and non-fire instances.

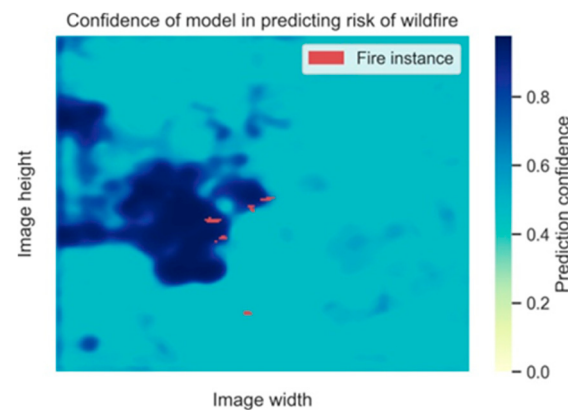


Figure 11. Illustration of model results: computed sample and method.

The actual accuracy of the model should be measured using quantitative metrics, such as the Dice coefficient, Intersection over Union (IoU), or precision–recall values, rather than solely relying on visual inspection. To improve the accuracy assessment, we can refine the prediction threshold to exclude areas with low confidence, focusing only on regions with high prediction certainty. This would reduce the overestimation of predicted fire areas and provide a more precise evaluation of the model's performance. Additionally, including metrics that consider the spatial overlap between the predicted fire regions and actual ground truth data can offer a clearer measure of the model's true accuracy.

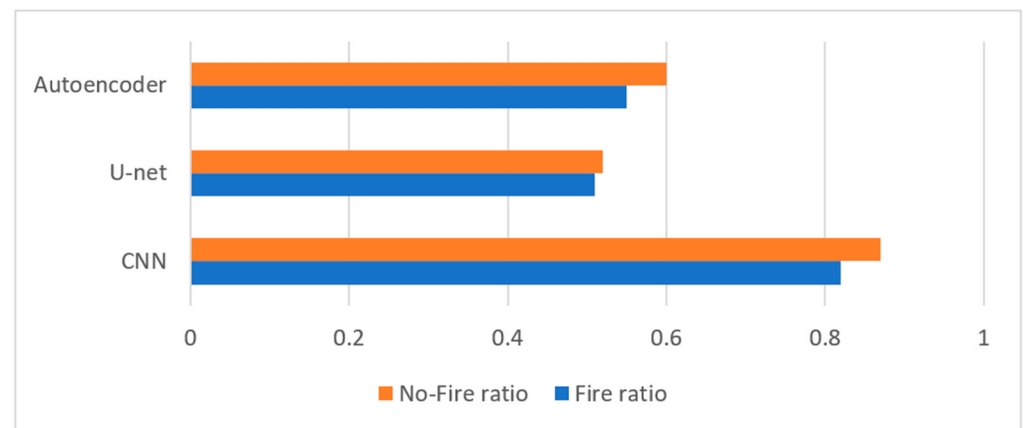
5.3. Comparing Architectures

Following the optimization of the autoencoder and U-Net, both models exhibited unsatisfactory results with the Adam optimizer. Consequently, the Stochastic Gradient Descent (SGD) optimizer was employed for both models. The best presentation of the autoencoder and U-Net models was accomplished with learning paces of 0.1 and 0.01, separately. After their development, all three models were evaluated in this section, with the results presented in Table 5. This comparative analysis provides insights into the performance of the CNN, autoencoder, and U-Net models in the context of wildfire prediction.

Table 5. Comparison of convolutional neural network (CNN), U-Net, and autoencoder models for wildfire risk prediction.

Models	Fire Ratio	No-Fire Ratio	Dice Coefficient	IOU
CNN	0.82	0.87	0.0059	0.0029
U-Net	0.51	0.52	0.0037	0.0018
Autoencoder	0.55	0.60	0.0036	0.0018

Figure 12 provides a comparison of the performance of three models—Convolutional Neural Network (CNN), U-Net, and Autoencoder—in predicting wildfire risk. The bar chart uses two metrics: the ratio of ‘No-Fire’ predictions (orange bars) and ‘Fire’ predictions (blue bars). Each model’s performance is represented by the relative proportions of these ratios, indicating their ability to differentiate between fire and no-fire scenarios. The CNN shows a higher ‘No-Fire’ ratio compared to the ‘Fire’ ratio, indicating it is more conservative in predicting fire risk. The U-Net model has a more balanced distribution between ‘Fire’ and ‘No-Fire’ ratios, suggesting a potentially better trade-off between sensitivity and specificity. The Autoencoder shows the closest parity between the ‘Fire’ and ‘No-Fire’ ratios, potentially highlighting its suitability for detecting subtle patterns related to fire risk.

**Figure 12.** Comparison of convolutional neural network (CNN), U-Net, and autoencoder models for wildfire risk prediction.

An examination of Table 5 uncovers that the CNN model outperformed the U-Net and the autoencoder, accomplishing a fire proportion of 0.82. Moreover, the autoencoder demonstrated slightly better performance compared to the U-Net concerning the fire ratio and the no-fire ratio. This detailed exploration of the comparative results will shed light on the capabilities of each model and offer valuable insights into their effectiveness in predicting wildfire occurrences. By delving deeper into these findings, it becomes possible to identify the specific areas where each model excels and where improvements could be made, ultimately enhancing the accuracy and reliability of wildfire prediction.

5.4. Feature Analysis

The results (Figure 13) for the fire ratio metric showcase how alterations to the input features impact the model’s predictions. This analysis provides valuable insights into the model’s sensitivity to changes in input features and highlights the potential variability in predictions under different conditions.

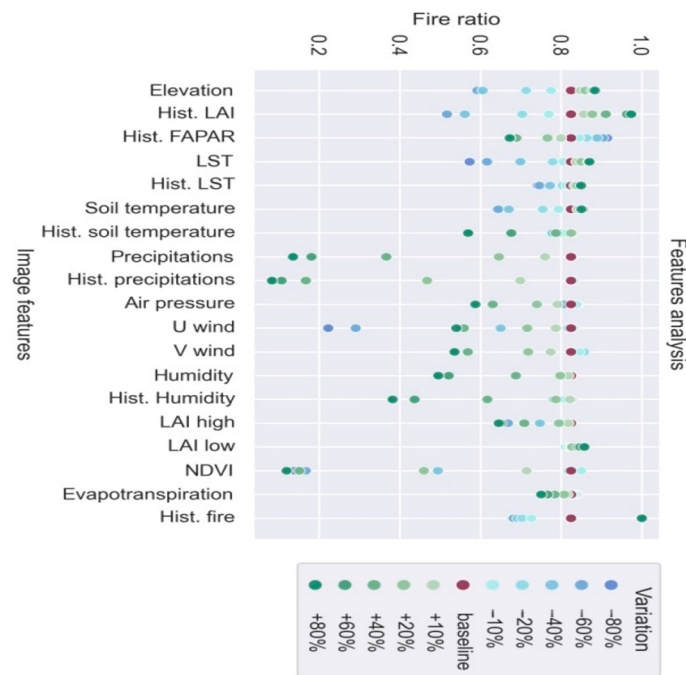


Figure 13. Analysis of scatterplot: fire ratio variation in each feature.

Upon analyzing Figure 13, it became evident that there is no direct proportionality between a percentage increase or decrease in a feature and the corresponding change in the fire ratio. To assess the relevance of these features, an iterative process was employed to remove them from the dataset.

5.5. Spatial Dependence

The results of this inference provide insights into how well the model generalizes to different geographical regions and its ability to make predictions on a broader scale (Figure 14). The results contribute to a comprehensive understanding of the model's performance in diverse geographical contexts.

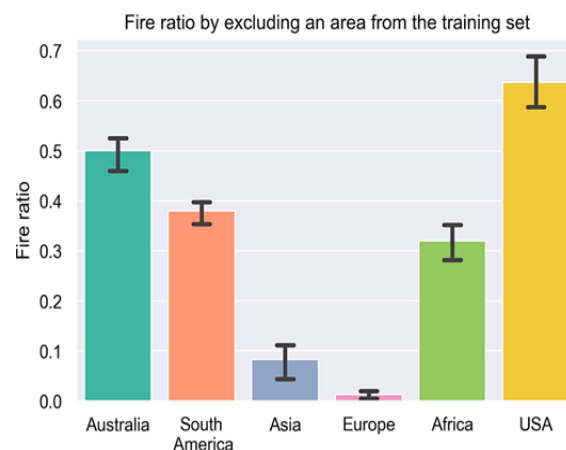


Figure 14. Analysis of bar plot: fire ratio of trained convolutional neural networks with confidence intervals.

Upon analyzing Figure 14, it became apparent that the exclusion score for different areas is not uniform. The highest score, close to previous estimates, is achieved for the US region, with a score of 0.64. In contrast, the lowest scores were observed for the Asian and European regions, with scores of 0.08 and 0.01, respectively. The Australian, South American, and African regions had fire ratios of 0.50, 0.38, and 0.32, respectively. These

results highlight the model's varying performance across different geographical areas, offering valuable insights into its generalization capabilities.

6. Discussion

While traditional spectral indices, such as the Normalized Burn Ratio (NBR) and Fire Radiative Power (FRP), are indeed effective for rapid detection and monitoring of various fire-related phenomena (e.g., smoke, active fires, burned areas, and fire severity), they have limitations in accurately predicting fire occurrences in complex scenarios. Spectral indices rely heavily on fixed thresholds and specific spectral characteristics, which may not be reliable in all cases, especially under varying atmospheric conditions, land cover types, or mixed fire severity levels. The motivation for developing deep learning models for fire detection lies in their ability to learn complex patterns in the data without relying on predefined thresholds. These models can integrate multiple sources of information (e.g., multispectral data from different satellite sensors) and account for non-linear relationships that are difficult to capture using spectral indices alone. Additionally, deep learning models can adapt to new data over time, potentially improving their predictive capabilities as more labeled data become available. To ensure practical applicability, the proposed models, particularly the CNN that achieved an 82% detection accuracy, can be integrated into existing wildfire management systems in several ways. First, these models can be embedded into satellite-based early warning platforms, providing real-time fire detection alerts to firefighting teams and disaster management authorities. Second, they can be combined with geographic information systems (GISs) to create detailed fire risk maps, aiding resource allocation and decision-making. Third, the adaptability of deep learning models makes them suitable for integration with drone- or ground-based sensor networks, allowing for real-time fire monitoring and dynamic updates in high-risk areas. Finally, partnerships with wildfire management agencies could facilitate the deployment of these models into operational workflows, such as automated resource planning and evacuation strategies. Such integrations would not only enhance the speed and accuracy of fire detection but also improve the overall efficiency and effectiveness of wildfire response systems. Continued research and collaboration will be essential to address challenges like computational resource demands and model generalization across diverse geographies.

To validate the effectiveness of the proposed models, including the convolutional neural network (CNN), U-Net, and autoencoder, a detailed comparison with traditional approaches was performed. The CNN model, which achieved a fire detection accuracy of 82%, demonstrated improved robustness in identifying fire-prone areas compared to spectral indices under diverse conditions. The model was designed to optimize performance by incorporating advanced techniques such as batch normalization, dropout layers, and regularization, which contribute to reducing overfitting and improving generalization to new data.

6.1. Comparative Analysis and Model Performance

Table 4 compares the three deep learning models used in this study, highlighting their strengths and limitations. The CNN outperformed the U-Net and autoencoder, achieving a fire ratio score of 0.82 and a no-fire ratio score of 0.87. This performance advantage can be attributed to the CNN's ability to capture more intricate spatial features due to its higher number of parameters (16 times more than the autoencoder). Incorporating dropout and batch normalization layers into the CNN further enhanced its regularization capabilities, contributing to its superior performance.

The U-Net model's relatively average performance in fire detection can be linked to its architectural design. The use of Conv2DTranspose layers for image upscaling, while

useful in some segmentation tasks, may not be as effective in capturing fine-grained details necessary for accurate fire detection compared to the CNN. In contrast, the autoencoder's simpler upscaling method using UpSampling2D, along with its fewer parameters, limited its ability to learn complex patterns, resulting in poorer performance compared to the other models.

6.2. Justification for Model Development over Traditional Approaches

While spectral indices remain valuable tools for initial fire detection, the optimized deep learning models offer advantages in adapting to various environmental conditions and integrating multi-sensor data. The ability of these models to generalize across different datasets and handle varying data quality makes them suitable for large-scale applications. However, the use of deep learning models does not exclude spectral indices; rather, these approaches can complement each other. By integrating spectral-index-based pre-processing steps with deep learning models, the combined approach could provide a more accurate and faster detection pipeline.

The findings from this study indicate that the CNN, with its optimized architecture and regularization techniques, offers a viable alternative to traditional spectral index methods, especially in complex scenarios where conventional approaches struggle. Continued research is needed to further improve the integration of spectral indices and machine learning techniques, ensuring that the developed models can outperform existing methods consistently.

7. Conclusions

The advancement of machine learning techniques, combined with the availability of satellite data, presents a significant opportunity to develop models capable of predicting extreme events and potentially preventing disasters. The research findings indicate that the CNN architecture achieves a balance between various metrics and the number of weights to be learned. Optimization tasks have tested different parameters, and experiments suggest that this architecture is better suited for wildfire detection than U-Net or autoencoder models. Interestingly, the U-Net architecture performed poorly, suggesting that alternative architectures may yield better outcomes. However, it is important to address the inherent limitations of satellite-based wildfire detection systems. The delay in data acquisition and processing means that fires must already be significantly ignited to be detected from space, reducing the system's real-time applicability for rapid response. Moreover, such delays could result in scenarios where fires either are already extinguished or have spread too far to be effectively controlled. The geographical challenges associated with accessing remote or rugged terrains further compound the difficulty of firefighting efforts based on this information. Experiments on the model's variability revealed that it correctly classifies fire risk approximately 80% of the time. While this underscores the potential of machine learning models in predicting extreme events, it also highlights the need for continued research to improve real-time data processing, sample selection, algorithm design, and model performance. Integrating satellite-based models with ground-based sensors and predictive analytics might help mitigate some of these limitations and improve the practical relevance of these systems in disaster management.

Author Contributions: Methodology, A.W.A.; Validation, A.W.A.; Investigation, S.K.; Resources, S.K.; Data curation, A.W.A.; Writing—original draft, A.W.A.; Supervision, S.K.; Project administration, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Forest Survey of India. India State of Forest Report 2021. Dehradun. 2021. Available online: <https://fsi.nic.in/forest-report-2021-details> (accessed on 25 March 2022).
2. Forest Survey Report 2021: Forest Fire Counts Up 2.7 Times. Available online: <https://www.downtoearth.org.in/forests/forest-survey-report-2021-forest-fire-counts-up-2-7-times-81123> (accessed on 25 March 2022).
3. Landsat 8 | Landsat Science. Available online: <https://landsat.gsfc.nasa.gov/satellites/landsat-8/> (accessed on 25 March 2022).
4. Barmapoutis, P.; Papaioannou, P.; Dimitropoulos, K.; Grammalidis, N. A review on early forest fire detection systems using optical remote sensing. *Sensors* **2020**, *20*, 6442. [\[CrossRef\]](#)
5. Singh, S.; Biwalkar, A.; Vazirani, V. Chapter 3: Clinical Decision Support Systems and Computational Intelligence for Healthcare Industries. In *Knowledge Modelling and Big Data Analytics in Healthcare: Advances and Applications*; Mehta, M., Passi, K., Chatterjee, I., Patel, R., Eds.; Taylor & Francis: Boca Raton, FL, USA, 2021; pp. 37–63.
6. Toan, N.T.; Thanh Cong, P.; Viet Hung, N.Q.; Jo, J. A deep learning approach for early wildfire detection from hyperspectral satellite images. In Proceedings of the 2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA), Daejeon, Republic of Korea, 1–3 November 2019; pp. 38–45. [\[CrossRef\]](#)
7. Lee, W.; Kim, S.; Lee, Y.-T.; Lee, H.-W.; Choi, M. Deep neural networks for wild fire detection with unmanned aerial vehicle. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 8–10 January 2017; pp. 252–253. [\[CrossRef\]](#)
8. Siddiqui, M.H.; Albahli, H.; Nawaz, R. Hybrid deep learning model with reinforcement learning for forest fire detection and prediction. *Future Internet* **2023**, *15*, 61. [\[CrossRef\]](#)
9. Khryashchev, V.; Larionov, R. Wildfire segmentation on satellite images using deep learning. In Proceedings of the 2020 Moscow Workshop on Electronic and Networking Technologies (MWENT), Moscow, Russia, 11–13 March 2020; pp. 1–5. [\[CrossRef\]](#)
10. Ganesan, P.; Sathish, B.S.; Sajiv, G. A comparative approach of identification and segmentation of forest fire region in high resolution satellite images. In Proceedings of the 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), Coimbatore, India, 29 February–1 March 2016; pp. 1–6. [\[CrossRef\]](#)
11. Wang, G.; Zhang, Y.; Qu, Y.; Chen, Y.; Maqsood, H. Early forest fire region segmentation based on deep learning. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 6237–6241. [\[CrossRef\]](#)
12. Govil, K.; Welch, M.L.; Ball, J.T.; Pennypacker, C.R. Preliminary Results from a Wildfire Detection System Using Deep Learning on Remote Camera Images. *Remote. Sens.* **2020**, *12*, 166. [\[CrossRef\]](#)
13. Khakzad, N. Modeling wildfire spread in wildland-industrial interfaces using dynamic Bayesian network. *Reliab. Eng. Syst. Saf.* **2019**, *189*, 165–176. [\[CrossRef\]](#)
14. Sayad, Y.O.; Mousannif, H.; Al Moatassime, H. Predictive modeling of wildfires: A new dataset and machine learning approach. *Fire Saf. J.* **2019**, *104*, 130–146. [\[CrossRef\]](#)
15. de Almeida Pereira, G.H.; Fusioka, A.M.; Nassu, B.T.; Minetto, R. Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study. *SPRS J. Photogramm. Remote Sens.* **2021**, *178*, 171–186. [\[CrossRef\]](#)
16. Latifah, A.L.; Shabrina, A.; Wahyuni, I.N.; Sadikin, R. Evaluation of random forest model for forest fire prediction based on climatology over Borneo. In Proceedings of the 2019 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Tangerang, Indonesia, 23–24 October 2019; pp. 4–8. [\[CrossRef\]](#)
17. Collins, L.; Griffioen, P.; Newell, G.; Mellor, A. The utility of Random Forests for wildfire severity mapping. *Remote Sens. Environ.* **2018**, *216*, 374–384. [\[CrossRef\]](#)
18. Heidari, H.; Keshtkar, M.; Moazzeni, N.; Jafari, M.; Azadi, H. Wildfire severity zoning through Google earth engine and fire risk assessment: Application of data mining and fuzzy multi-criteria evaluation in Zagros forests, Iran. *Res. Sq.* 2021; preprint. [\[CrossRef\]](#)
19. Bar, S.; Parida, B.R.; Pandey, A.C. Landsat-8 and Sentinel-2 based Forest fire burn area mapping using machine learning algorithms on GEE cloud platform over Uttarakhand, Western Himalaya. *Remote Sens. Appl. Soc. Environ.* **2020**, *18*, 100324. [\[CrossRef\]](#)

20. Zhai, C.; Zhang, S.; Cao, Z.; Wang, X. Learning-based prediction of wildfire spread with real-time rate of spread measurement. *Combust. Flame* **2020**, *215*, 333–341. [[CrossRef](#)]
21. Jindal, R.; Kunwar, A.K.; Kaur, A.; Jakhar, B.S. Predicting the dynamics of forest fire spread from satellite imaging using deep learning. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 344–350. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.