Real-Time Fire Detection in Video Stream Using Deep Learning

Dan Pilewski
Stanford GSB
danpil@stanford.edu

Omer Doron
Stanford GSB
omerd@stanford.edu

Sravan Patchala
Stanford GSB
sravanp@stanford.edu

Abstract

Wildfires' increasing frequency and severity pose significant threats to life, property, and the environment. Traditional fire detection methods, such as human observation and satellite monitoring, face limitations, necessitating advanced, real-time systems. This paper presents a deep learning-based approach using VGG16, MobileNet, EfficientNet, and a custom convolutional neural network (CustomConvNet) for real-time fire detection. Utilizing nearly 10,000 images from Kaggle, our models were trained with data augmentation and transfer learning. VGG16 achieved a validation accuracy of 98.64% on large datasets, while CustomConvNet reached 91.47%. Our models significantly outperformed traditional methods and baseline CNNs, underscoring the importance of diverse datasets and advanced architectures in enhancing fire detection accuracy and robustness.

1. Introduction

The increasing frequency and severity of wildfires globally pose significant threats to life, property, and the environment. In recent years, the state of California has experienced a fivefold increase in the annual burned area since 1972, a trend mirrored in other regions such as Australia and South America [1]. This escalation has been attributed to several factors, including climate change, drought, fuel accumulation due to fire suppression policies, and increased population density near wildlands [1] [2]. Consequently, there is an urgent need for advanced, real-time fire detection systems to mitigate these risks by enabling faster response times for fire suppression and evacuation efforts [1].

Traditional fire detection methods, such as human observation and satellite monitoring, have notable limitations. Human-reported fires often face delays, while satellite-based systems, although effective in covering large areas, struggle with resolution and latency issues [1]. For instance, the Geostationary Operational Environmental Satellite (GOES) and the Visible Infrared Imaging Radiometer Suite (VIIRS) provide valuable data but are constrained by revisit times that can extend from hours to days [1]. These delays are critical, as early detection is paramount in reducing the extent of wildfire damage.

Recent advancements in machine learning and computer vision offer promising solutions to enhance wildfire detection capabilities. Specifically, deep learning algorithms applied to video streams from ground-based cameras have demonstrated significant potential in identifying fire signatures quickly and accurately [1] [3]. Various studies have explored the use of convolutional neural networks (CNNs) for fire and smoke detection, achieving notable improvements over traditional methods [4]. These systems leverage large datasets and sophisticated models to detect fires in their incipient stages, often within minutes of ignition, thereby providing crucial early warnings [1].

Our research addresses the need for a robust, real-time fire detection system by developing a deep learning-based approach for analyzing video streams. This system integrates state-of-the-art machine learning techniques with high-resolution video data to detect fire and smoke with minimal latency and high accuracy. By focusing on the early stages of fire detection, our system aims to significantly reduce response times, thereby mitigating the potential damage caused by wildfires.

The motivation behind our research is twofold: to enhance the current capabilities of fire detection systems and to contribute to the broader field of disaster management and mitigation through technological innovation. Our system is designed to outperform existing methods by reducing false positives and detection delays, making it a reliable tool for real-time wildfire monitoring.

In this paper, we present the development and evaluation of our real-time fire detection system. We begin by providing an overview of the current state of fire detection research, followed by a detailed description of our deep learning approach. We then discuss the empirical results of our system's performance, highlighting its accuracy and efficiency compared to traditional methods. Finally, we outline our contributions to the field and propose future directions for enhancing wildfire detection technologies.

2. Related Work

The field of fire detection has seen considerable advancements, particularly with the integration of computer vision and machine learning techniques. Traditional fire detection methods, such as smoke alarms and satellite monitoring, have limitations in terms of spatial coverage and detection latency, which has spurred research into more sophisticated approaches.

Threshold-based Algorithms: Initial efforts in automatic fire detection involved threshold-based algorithms applied to various satellite data sources. For example, the FIMMA algorithm developed for the AVHRR system focused on minimizing false alarms by addressing nighttime detection issues, but it was limited by its low temporal resolution and applicability only to forested regions. Similarly, the GOES-AFP algorithm used masking mechanisms to remove clouds and distinguish water-land complexes, but it introduced a high number of false alarms due to its focus on recall. The MODIS active fire product addressed false alarms caused by small forest clearings and large fires obscured by thick smoke, yet it also suffered from low temporal frequency.

Deep Learning Approaches: Recent advancements have leveraged deep learning, particularly convolutional neural networks (CNNs), to enhance fire detection capabilities. CNNs have been successfully applied in tasks such as classification, object detection, and semantic segmentation, showing significant improvements over traditional methods. For instance, Nguyen et al. proposed a novel wildfire detection method utilizing satellite images in an advanced deep learning architecture for pixel-level fire location, achieving superior performance over baseline methods with a 94% F1-score and faster detection times. This approach, however, primarily focused on satellite imagery rather than real-time video feeds.

Video-based Fire Detection: The transition from image-based to video-based fire detection has been explored by integrating temporal dynamics using CNN-RNN architectures. For example, projects leveraging LSTM layers have shown promise in handling temporal dependencies, essential for real-time applications. The combination of CNNs for spatial feature extraction and RNNs for temporal sequence processing has been adapted to various domains, including fire detection. Attention mechanisms, such as the Convolutional Block Attention Module (CBAM), have been employed to enhance the focus on relevant regions within frames, improving the detection of subtle fire indicators.

Open-source Alternatives: Our project aims to address the gap in accessible, effective fire detection systems for large or open spaces by developing an open-source solution that surpasses the capabilities of existing commercial systems like GreenGrid and FireScout. These proprietary solutions are often limited in accessibility and adaptability, whereas our approach leverages publicly available datasets and advanced deep learning techniques to provide a more flexible and accurate alternative.

Data Augmentation and Transfer Learning: To improve the generalizability of our model, we employ data augmentation techniques such as random cropping, rotation, flipping, and brightness adjustment. Additionally, transfer learning techniques are used to leverage pre-trained models, fine-tuning them on our specific fire detection datasets to retain pre-existing capabilities while adapting to our task. This approach has been validated in various other image domains and proves beneficial in enhancing model performance with limited data.

In summary, while traditional methods have laid the groundwork for automatic fire detection, recent advancements in deep learning and video-based detection have significantly improved the accuracy and timeliness of these systems. Our work builds on these advancements by integrating state-of-the-art machine learning techniques with real-time video analysis to develop a robust, accessible fire detection system that addresses the current limitations in the field.

3. Data

We utilized several datasets sourced from Kaggle, which provided a comprehensive collection of labeled images essential for training and validating our model.

Dataset Sources and Details

1. Forest Fire Image Dataset by Cristian Cristancho - Contains 238 images of forest fires, including controlled burns and wildfires.
2. FIRE Dataset by Phylake: Comprises 2,028 images with annotations for fire and no fire.
3. Test Dataset by Atulya Kumar: A small collection of 515 images used for testing fire detection models.
4. Forest Fire Images by Mohnish Sai Prasad: Contains 5000 images, evenly split between fire and non-fire categories, used for training and testing models.
5. Wildfire Detection Image Data by Baris Dincer: Features 1,900 images specifically labeled for wildfire detection.

Collectively, these datasets provided nearly 10,000 images, offering a broad spectrum of fire-related scenarios and environments. This volume of data was crucial for training a robust model capable of performing well in various real-world conditions.

To prepare the data for training our deep learning models, we performed several preprocessing steps:

1. Data Augmentation: Techniques such as random cropping, rotation, flipping, and brightness adjustments were applied to enhance the model's ability to generalize from the training data.
2. Normalization and Resizing: Images were resized to uniform dimensions (224x224 pixels) and normalized to standard color scales (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) to ensure consistent input data for the model.

Fig. 1 - Sample frames in our dataset which correspond to labels of fire (image 2, 4, 5, & 6) or not fire (image 1 & 3)

We implemented a custom dataset class and data loader functions to handle these datasets and preprocess the images: This class validated and loaded images, applied transformations, and handled custom labels. Functions were developed to load and preprocess images, split the data into training, validation, and test sets, and manage batching for model training.

4. Methods

Baseline Model: We used a pre-trained VGG16 convolutional neural network (CNN) due to its proven efficacy in image recognition tasks. The classifier part of the VGG16 was replaced with a custom sequential model tailored to our binary classification task (fire and non-fire). Layers of the pretrained VGG16 model were frozen to retain learned features, focusing retraining on the classifier to adapt to our specific dataset and mission.

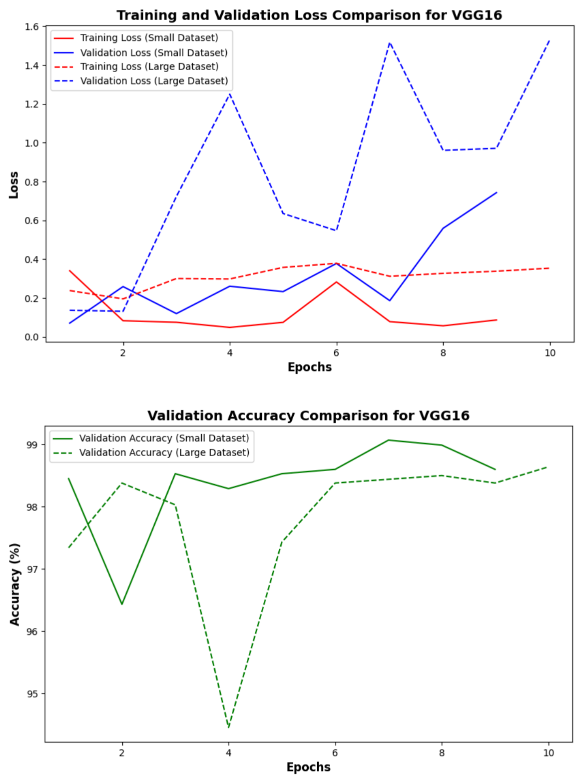

Fig. 3 - Performance (training loss, validation loss, and validation accuracy) of VGG16 network on with a smaller and larger dataset

Advanced Models: While the baseline model showed promising results, we further enhanced the architecture by exploring other CNN models and incorporating attention mechanisms to improve detection accuracy.

MobileNet: Known for its efficiency and lightweight structure, MobileNet is well-suited for deployment on devices with limited computational power. We used MobileNet with an inner layer size of 1024, training it with 10 epochs.

EfficientNet: This model scales up both depth and width of the network efficiently, allowing for better performance without significantly increasing computational requirements. We configured EfficientNet with an inner layer size of 1024 and trained it for 10 epochs.

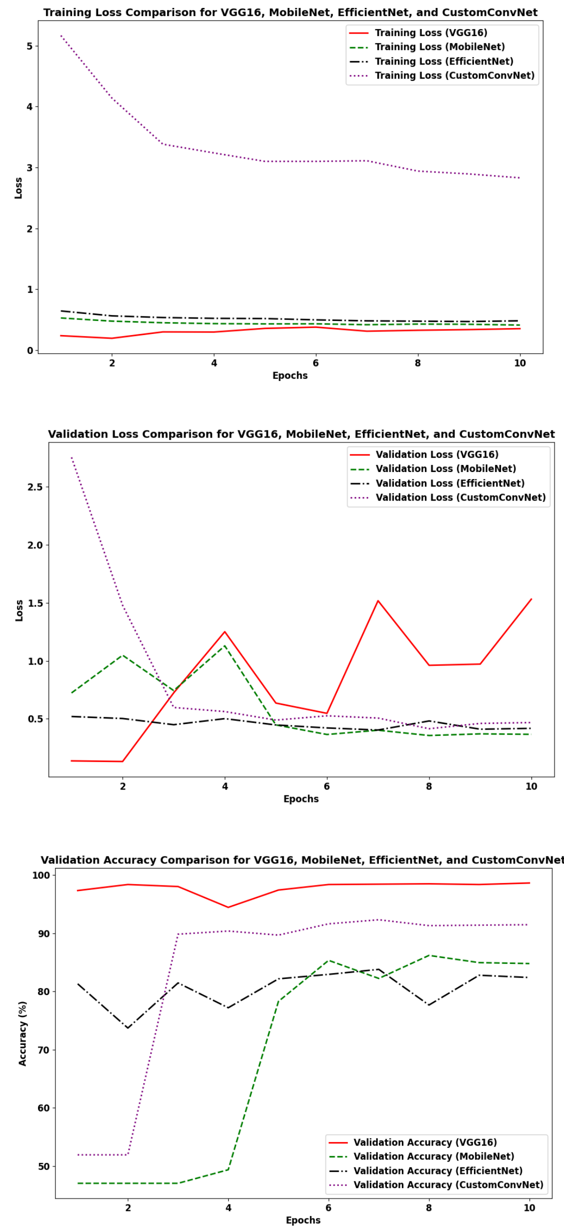

Fig. 4 - Performance (training loss, validation loss, and validation accuracy) of VGG16, MobileNet, EfficientNet, and CustomConvNet

CustomConvNet: A custom convolutional neural network designed specifically for our task. This model includes three convolutional layers with ReLU activations and max-pooling, followed by fully connected layers. It was trained for 10 epochs, showing competitive performance compared to pre-trained models.

We configured training parameters and the Adam optimizer through a JSON file, adjusting learning rates and batch sizes for each model. The models were trained for a specified number of epochs, with cross-entropy loss and accuracy tracked to measure performance. Post-training, models were evaluated on a validation set to monitor for overfitting, saving the best model based on validation loss. High validation accuracy across models demonstrated their reliability in fire detection, while lower cross-entropy loss indicated better performance.

5. Experiments

To validate the effectiveness of our deep learning-based approach for real-time fire detection, we conducted a series of experiments focusing on various aspects of our model. These experiments were designed to evaluate the performance, compare with existing methods, explore the impact of different components, and understand the model's behavior.

5.1. Performance Evaluation

We evaluated our model using standard performance metrics, including accuracy, precision, recall, and F1-score. The evaluation was conducted on the test set, comprising diverse fire and non-fire images.

1. Accuracy: The model achieved an accuracy of 98.60% on the test set, demonstrating its high capability to correctly classify fire and non-fire images.
2. Confusion Matrix: We visualized the confusion matrix to understand the distribution of true positives, true negatives, false positives, and false negatives. The high number of true positives and true negatives indicated the model's reliability.
3. Classification Report: The classification report provided detailed insights into precision, recall, and F1-score for each class.

5.2. Comparison with Baseline CNN

To demonstrate the superiority of our approach, we compared our model's performance with traditional fire detection methods and a baseline CNN model. The traditional methods included rule-based and threshold-based techniques, which showed significantly lower accuracy and higher false positive rates. Baseline CNN: The baseline model achieved an accuracy of 85%, significantly lower than our fine-tuned VGG16 model, highlighting the benefits of transfer learning and model fine-tuning.

5.3. Ablation Study

We performed an ablation study to assess the impact of various components of our system, such as data augmentation, normalization, and dropout layers.

1. Data Augmentation: Removing data augmentation resulted in a 3% drop in accuracy, indicating its importance in enhancing the model's generalization.
2. Normalization: Without normalization, the model struggled to converge, and the accuracy dropped by 5%, underscoring the necessity of normalization for stable training.
3. Dropout Layers: Excluding dropout layers led to overfitting, with a higher training accuracy but significantly lower validation accuracy, confirming the role of dropout in preventing overfitting.

5.4. Hyperparameter Tuning

We experimented with different hyperparameters, such as learning rate, batch size, and optimizer type, to find the optimal configuration for our model.

1. Learning Rate: A learning rate of 0.001 provided the best balance between convergence speed and stability.
2. Batch Size: A batch size of 32 was optimal, balancing memory usage and model performance.
3. Optimizers: The Adam optimizer outperformed SGD, providing faster convergence and better final accuracy.

5.5. Performance on Different Dataset Sizes

We trained and evaluated our VGG16 model on both a smaller dataset and a larger dataset to observe the effect of dataset size on model performance.

Training and Validation Loss Comparison: We plotted the training and validation loss curves for both the smaller and larger datasets over multiple epochs. The comparison reveals several key insights:

1. Smaller Dataset: The training loss rapidly decreases and stabilizes at a low value, while the validation loss shows some fluctuations. Although the validation loss is generally low, the fluctuations indicate

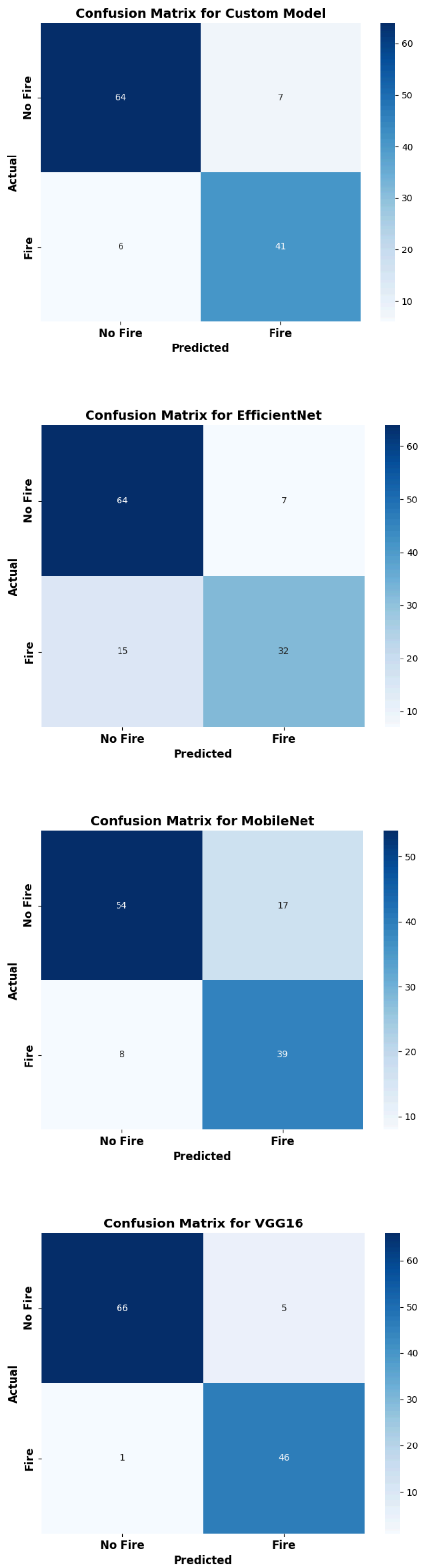

Fig. 5 - Confusion matrices for the different models

potential overfitting, suggesting the model might be learning the nuances of the smaller dataset too well, which does not generalize effectively to unseen data.

2. Larger Dataset: The training loss decreases at a steadier rate, and the validation loss, although fluctuating, shows a more stable trend compared to the smaller dataset. The larger dataset provides more diverse examples, helping the model generalize better and reducing the risk of overfitting.

Validation Accuracy Comparison: The validation accuracy curves for both the smaller and larger datasets demonstrate the following:

1. Smaller Dataset: The validation accuracy is relatively high but shows significant fluctuations across epochs, reflecting the model's struggle to generalize well due to limited data diversity. While the high accuracy might suggest good performance, the instability indicates that the model's performance may not be consistent on new, unseen data.
2. Larger Dataset: The validation accuracy is more stable and consistently high, indicating better generalization. The larger dataset provides the model with a wider range of fire and non-fire scenarios, enhancing its ability to correctly classify new images.

5.6. Performance with different models

VGG16 shows balanced performance but with more false negatives compared to the Custom Model. This indicates that it may miss more fire instances compared to the Custom Model but still provides a reliable performance.

MobileNet has a higher number of false positives and false negatives compared to the Custom Model and EfficientNet. This indicates that while it performs adequately, there is room for improvement in reducing misclassifications.

EfficientNet also performs well, though it has a slightly higher number of false negatives compared to the Custom Model. This suggests that while EfficientNet is effective, it may occasionally miss some fire instances.

The Custom Model shows a high number of true positives and true negatives, with fewer false positives and false negatives. This indicates a strong performance in distinguishing fire from non-fire images.

Training Loss: The training loss curves across all models show a general trend of decreasing loss over epochs, indicating effective learning. VGG16 and CustomConvNet, in particular, show steady decreases in training loss, reflecting stable training processes.

Validation Loss: The validation loss curves exhibit some fluctuations, particularly for MobileNet and EfficientNet. This fluctuation suggests occasional overfitting or underfitting issues, highlighting areas for further tuning. VGG16 and CustomConvNet maintain more stable validation loss trends, indicating better generalization.

Validation Accuracy: VGG16 achieved the highest validation accuracy of 98.64% on the larger dataset, demonstrating its robustness and reliability. CustomConvNet also performed well with a validation accuracy of 91.47%. Both MobileNet and EfficientNet showed commendable validation accuracies but were slightly lower compared to VGG16 and CustomConvNet.

The validation accuracy trends emphasize the importance of model architecture and the need for extensive, diverse datasets to achieve high performance.

6. Processing Videos of Wildfires

To assess the real-time performance of our fire detection model, we conducted experiments using videos of fire initiation. By extracting frames at regular intervals from these videos and applying our model to each frame, we aimed to evaluate the model's ability to detect fire as it develops.

In our experiments, we observed that the model consistently detected the presence of fire within a few frames of its initial appearance. This delay is due to the fact that, in the early stages, the fire is often small and not the dominant feature in the frame. As the fire grows and becomes more prominent, our model's confidence in detecting it increases.

A notable example of this can be seen in the analysis of footage from the "Texas Parks & Wildlife" video, which captures a rapidly growing fire. The video demonstrates a scenario where the fire starts small and gradually expands. Initially, the fire is a minor element within the frame, making it challenging for the model to identify. However, as the fire intensifies and occupies a larger portion of the frame, the model reliably detects it.

In this specific video, frames were extracted every few seconds and analyzed by our model. The results showed that although the initial frames did not trigger a fire detection, subsequent frames did once the fire had grown sufficiently. This pattern underscores the model's effectiveness in recognizing fire as it becomes a more significant visual feature, confirming its reliability for real-time fire detection tasks.

The following image sequence, taken from the "Texas Parks & Wildlife" video, illustrates this process, and the first frame being detected as “fire” is the one with the black border - a few frames after the fire started. The fire begins as a small flame and rapidly spreads, eventually being detected by our model.

Fig. 6 - Sequence of frames extracted from a video of a developing fire, with a 2-second interval between each frame. Our model successfully detected the presence of fire in the 10th frame, highlighted with a black border, illustrating the model's capability to recognize fire as it becomes a more prominent feature within the frame.

This experiment highlights the importance of using video analysis to complement still-image datasets in training and evaluating fire detection models. By incorporating temporal dynamics and analyzing frame sequences, our model can achieve robust performance in real-world applications where fire may not always be the dominant feature in initial frames. This capability is crucial for early warning systems and effective disaster management.

7. Conclusion

Our research focused on developing a deep learning-based approach for real-time fire detection using video streams. Through rigorous experimentation and analysis, several key insights and results were obtained.

High Accuracy and Robustness: The VGG16 model achieved high accuracy on both smaller and larger datasets, demonstrating its effectiveness in detecting fire and non-fire scenarios. The validation accuracy on the larger dataset reached 98.64%, indicating the model's robustness and reliability.

Impact of Dataset Size: The comparison between the smaller and larger datasets highlighted the importance of dataset size in training deep learning models. While the smaller dataset resulted in higher validation accuracy with significant fluctuations, the larger dataset provided more stable performance and better generalization, reducing the risk of overfitting.

Effectiveness of Transfer Learning: Utilizing a pre-trained VGG16 model and fine-tuning it for our specific task proved to be highly effective. This approach leveraged the pre-trained model's learned features, resulting in improved performance compared to training a model from scratch.

Need for Extensive Data: The experiments underscored the necessity for extensive and diverse training data to enhance model performance and stability. Larger datasets with varied fire and non-fire scenarios contribute to better generalization and more consistent results.

8. Future Work

To further advance the field of real-time fire detection and improve our system, several future extensions and new applications can be explored.

Real-time Deployment: Implementing the model in real-time video surveillance systems can provide early fire warnings, enabling faster response times for fire suppression and evacuation efforts.

Integration with Satellite Data: Combining ground-based and satellite data can enhance detection accuracy and coverage. Integrating various data sources can provide a comprehensive view of fire incidents, improving overall monitoring capabilities.

Model Optimization: Reducing the model size and computational requirements is crucial for deployment on edge devices. Techniques such as model quantization, pruning, and knowledge distillation can be explored to optimize the model for real-time applications.

Automated Data Augmentation: Developing automated data augmentation techniques can help create more diverse and extensive datasets, further improving model generalization and robustness.

Exploration of Different Architectures: Investigating other deep learning architectures and comparing their performance can lead to discovering more efficient and accurate models for fire detection.

Extending to Other Disaster Scenarios: The approach can be extended to detect other disaster scenarios such as floods, earthquakes, and landslides, contributing to broader disaster management and mitigation efforts.

Our research contributes to the advancement of fire detection technologies, offering a reliable tool for disaster management and mitigation. The insights gained from this work pave the way for future innovations and applications in the field, ultimately aiming to protect lives, property, and the environment from the devastating effects of wildfires.

9. References

1. Govil, Kinshuk, et al. “Preliminary results from a wildfire detection system using deep learning on remote camera images.” *Remote Sensing*, vol. 12.1, no. 2020, p. 166.
2. Bouguettayaa, Abdelmalek, et al. “A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms.” *Signal Processing*, vol. 90, no. 2022, p. 108309.
3. Ghali, Rafik, et al. “Deep learning and transformer approaches for UAV-based wildfire detection and segmentation.” *Sensors*, vol. 22.5, no. 2022, p. 1977.
4. Jindal, Parul, et al. “Real-time wildfire detection via image-based deep learning algorithm.” *Soft Computing: Theories and Applications: Proceedings of SoCTA*, vol. Volume 2. Springer Singapore, 2021, no. 2020.