A System to Detect Forest Fires Early by Analyzing Satellite Images of Forested Areas

Pratyush Ranjan
Lovely Professional University
Phagwara, Punjab, India
pratyushranjan87@gmail.com

Sujal Gupta
Lovely Professional University
Phagwara, Punjab, India
sujalg2806@gmail.com

Abstract— **Forest fires are among the most devastating natural disasters, posing a severe threat to biodiversity, climate stability, and human livelihood. Their rapid and unpredictable spread makes early detection a critical component in effective disaster response and mitigation. Traditional methods such as satellite imaging, sensor-based systems, and human surveillance are often expensive, time-consuming, and limited in real-time applicability. In this research, we present an efficient and accurate forest fire detection system leveraging deep learning techniques. Specifically, we employ Transfer Learning using a pretrained ResNet18 model in the PyTorch framework to classify images as either 'fire' or 'no fire'. The model is trained on a publicly available dataset featuring annotated fire and no-fire images. We also introduce a custom image preprocessing pipeline that enhances feature extraction by converting images to the HSV color space and applying Gaussian blurring for noise reduction. This preprocessing significantly improves model performance by accentuating color patterns typical in fire imagery. The system achieves high classification accuracy on the validation set, demonstrating its ability to distinguish between complex visual patterns of fire and non-fire scenes. Our solution provides a scalable, cost-effective, and real-time capable approach to forest fire detection, with potential for integration into drone surveillance and automated monitoring systems.**

Keywords— ***Forest Fire Detection, Deep Learning, Transfer Learning, ResNet18, PyTorch, Image Classification, Computer Vision***

I. INTRODUCTION

Forest fires are a critical environmental issue with far-reaching consequences for ecosystems, human settlements, and climate change. These fires can lead to the destruction of biodiversity, release of greenhouse gases, degradation of air quality, and substantial economic losses. They are often triggered by both natural causes such as lightning and anthropogenic activities like illegal logging, unattended campfires, or arson. Once ignited, fires can spread rapidly, particularly in dry conditions, making timely detection vital to prevent widespread damage.

Conventional fire detection systems include satellite imagery analysis, sensor-based networks, and human surveillance using watchtowers or patrolling. While effective to a degree, these methods are associated with drawbacks such as high costs, low temporal resolution, dependence on weather conditions, and susceptibility to human error. Moreover, remote sensing techniques can introduce significant time delays in detection and response.

The rapid advancements in artificial intelligence, particularly deep learning and computer vision, have opened new possibilities for real-time fire detection. Deep learning models, especially Convolutional Neural Networks (CNNs), have proven highly successful in image classification tasks by learning hierarchical features from raw image data. Transfer Learning—a technique that adapts a pretrained network to a new task—allows us to use state-of-the-art models like ResNet18 trained on large datasets like ImageNet and fine-tune them for specific applications with limited data.

This study explores the application of Transfer Learning using ResNet18 within the PyTorch framework to develop a robust forest fire detection system. The model is trained to classify input images into two categories: 'fire' and 'no fire'. The training process is enhanced through custom image preprocessing that includes resizing, conversion to HSV color space, and Gaussian blurring, which improves the model's sensitivity to fire-specific visual features. Through systematic evaluation using performance metrics such as precision, recall, F1-score, and confusion matrix, the system's effectiveness is validated. Our research proposes a scalable and deployable solution that can serve as a vital tool in early forest fire detection and disaster management.

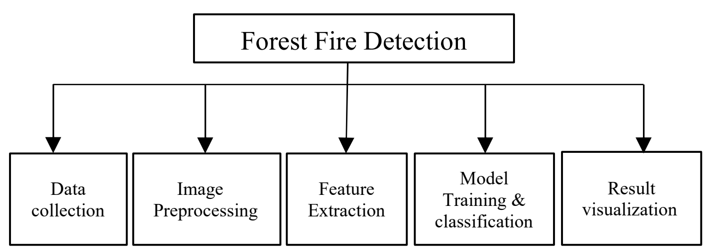

Fig 1. The process involves in Forest Fire Detection.

II. RELATED WORK

Various studies have explored fire detection using conventional image processing techniques such as flame and smoke detection through thresholding and motion detection. These methods often rely on handcrafted features like color, shape, and motion patterns to identify fire, which makes them susceptible to inaccuracies under complex backgrounds, lighting conditions, and varying environmental factors.

Recent advancements in machine learning have greatly enhanced the accuracy and efficiency of fire detection systems. Several works have leveraged CNNs for automatic feature extraction from images. For instance, CNNs have been used to detect fire in surveillance videos, and models like YOLO and Faster R-CNN have been employed for real-time object detection tasks including fire detection. Although these methods show promising results, they often demand considerable computing power and extensive labeled datasets.

Transfer Learning has emerged as a promising solution to overcome data limitations and computational overhead. Research using pretrained models such as VGG16, InceptionV3, and ResNet50 has shown high performance in image classification and object detection tasks in various domains, including medical imaging, traffic monitoring, and agriculture. By fine-tuning these models on fire-specific datasets, researchers have achieved improved accuracy and faster convergence.

In the context of forest fire detection, studies have demonstrated the effectiveness of ResNet variants due to their ability to retain spatial features across multiple layers through residual connections. This research builds on prior work by utilizing ResNet18, which offers a balance between performance and computational efficiency, and enhancing it with domain-specific preprocessing for improved fire detection accuracy.

III. METHODOLOGY

The proposed methodology is a multi-phase approach designed to develop a high-performance forest fire detection system leveraging deep learning. The overall workflow integrates data acquisition, advanced preprocessing, deep neural network customization, model training, and rigorous evaluation. The implementation utilizes the PyTorch framework, known for its modularity and flexibility in developing and training deep learning models.

The key idea is to enhance the model's ability to detect fire-like visual patterns by applying custom preprocessing techniques that emphasize color and shape characteristics typical of fire. Instead of relying on raw RGB data, we convert images into the HSV color space, where the hue and saturation channels help isolate fire hues more effectively. Additionally, Gaussian blurring is applied to reduce background noise and strengthen the representation of flames and smoke.

Once preprocessing is complete, we use Transfer Learning with ResNet18, a pretrained convolutional neural network known for its strong feature extraction capabilities. The final layer of ResNet18 is modified to perform binary classification ('fire' vs. 'no fire'). The model is trained on a labeled dataset using an appropriate loss function and optimizer, with continuous monitoring of training performance metrics like loss, accuracy, and confusion matrix.

A. Data Collection

The dataset used in this study is publicly available on Kaggle and comprises two distinct classes:

Fire: Images depicting forest fires, flames, or fire-afflicted regions.
No Fire: Images of normal forest landscapes with no visible fire or smoke.

The dataset was unzipped and categorized into a Training directory containing two subfolders (fire and nofire). To evaluate performance, 80% of the dataset was used for training while the remaining 20% was set aside for validation, leveraging PyTorch's random_split function. This helps in assessing the model's generalization ability on unseen data.

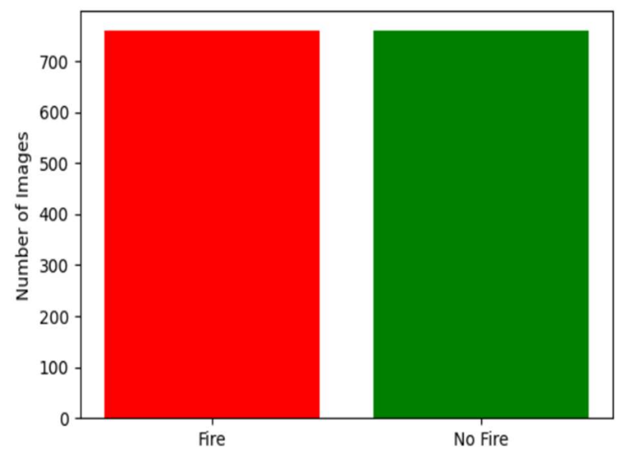

Fig 2. Class Distribution b/w Fire and No Fire.

B. Data Pre-processing

Image preprocessing plays a critical role in enhancing the model's learning. A custom transformation pipeline is implemented using the following steps:

Resizing: All input images are resized to 224x224 pixels, aligning with ResNet18's input requirements.

Color Space Conversion: Images are transformed from the RGB (Red-Green-Blue) color space to HSV (Hue-Saturation-Value). HSV enhances color separation, making fire-related hues (reds, oranges) more distinguishable from background vegetation.

Gaussian Blurring: A (3x3) Gaussian filter is applied to the HSV-converted image. This helps in reducing high-frequency noise, smoothing irrelevant edges, and enhancing dominant fire features such as color blobs or flame contours.

Normalization: Using the ImageNet dataset's mean and standard deviation values, images are normalized to bring pixel intensities within a standard range. This ensures faster convergence during training.

Conversion to Tensors: Finally, images are converted to PyTorch tensors, making them compatible with the deep learning model input format.

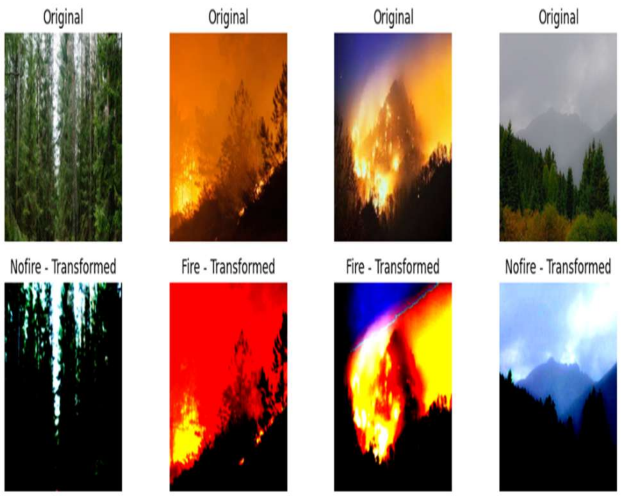

Fig 3. Original vs Transformed (Mixed Classes)

C. *Model Architecture*

We employ Transfer Learning using the ResNet18 architecture, a deep convolutional neural network that was originally trained on the ImageNet dataset containing over a million images across 1,000 categories. Transfer Learning allows us to utilize the pretrained layers of ResNet18 that have already learned to identify basic image features like edges, textures, and shapes. These learned features are then fine-tuned to detect fire in forest scenes.

The architecture of ResNet18 includes:

The architecture begins with a convolutional layer that captures local patterns, followed by a max pooling layer which helps downsample the feature maps.

Four groups of residual blocks that contain skip connections. These connections help avoid issues like vanishing gradients during training by allowing gradients to flow directly across layers.

Toward the end of the network, a global average pooling layer is used to condense each feature map into a single value, effectively creating a compact feature vector.

A final fully connected (FC) layer, which we modify to output two classes ('fire' and 'no fire') instead of the original 1,000 classes.

This structure makes ResNet18 lightweight yet powerful, ideal for classification tasks with limited training data.

D. *Training Configuration.*

The model is trained using the following configuration:

Loss Function: Cross Entropy Loss, ideal for multi-class classification.

Optimizer: Adam optimizer with a learning rate of 0.0001, providing efficient gradient-based learning.

Batch Size: 32 images per batch, balancing memory usage and update frequency.

Epochs: 4 iterations over the entire training set.

Each batch undergoes forward propagation, loss computation, backpropagation, and weight update. The training loop logs the average loss per epoch to visualize convergence.

IV. Evaluation Metrics

After training, the model's performance is assessed using a range of evaluation metrics:

Classification Report: Provides detailed metrics such as precision, recall, and F1-score for each class, giving insight into false positives and false negatives.

	precision	recall	f1-score	support
fire	0.99	0.99	0.99	159
nofire	0.99	0.99	0.99	145
accuracy			0.99	304
macro avg	0.99	0.99	0.99	304
weighted avg	0.99	0.99	0.99	304

Fig 4. Classification Report.

Prediction Visuals: A set of validation images are displayed alongside the model's predicted and actual class labels, helping validate qualitative performance.

Confusion Matrix: A normalized matrix is plotted to visualize the distribution of true versus predicted labels. This aids in detecting any model bias. It helps you understand how well your classification model is performing by comparing the predicted labels with the actual labels.

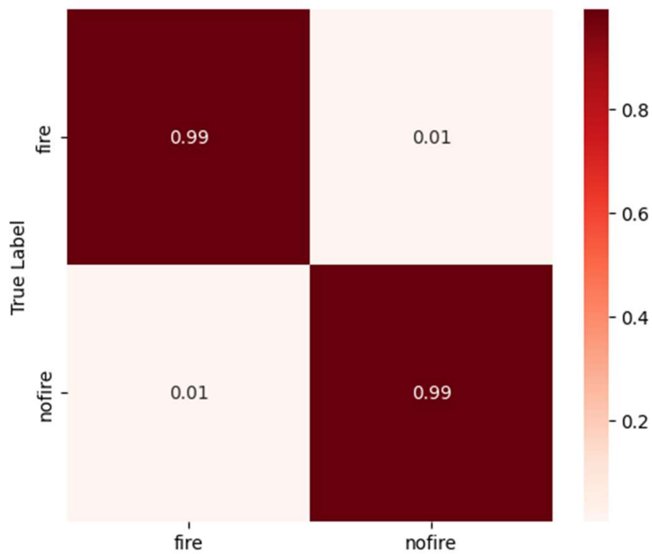

Fig 5. Normalized Confusion Matrix.

Accuracy Curve: The training loss is plotted against epochs to ensure proper model convergence and detect overfitting or underfitting.

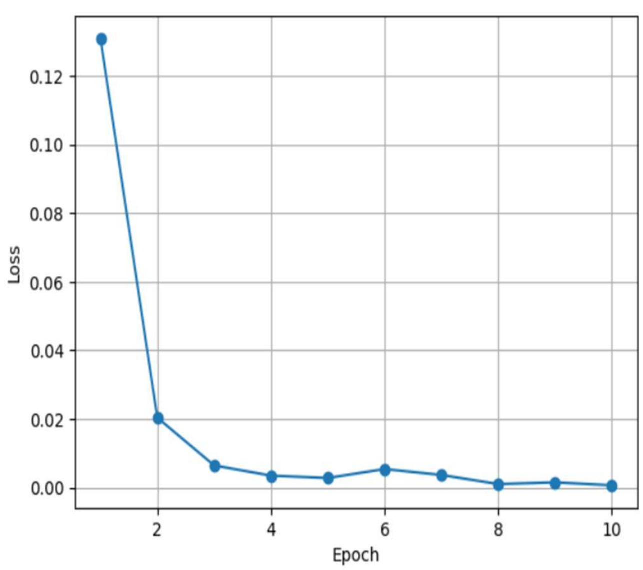

Fig 6. Training Loss Over Epochs.

V. Results and Discussion

The proposed forest fire detection system using ResNet18 and Transfer Learning was trained and evaluated on a carefully preprocessed dataset. The model demonstrated consistent learning behavior across the training epochs, as evidenced by the steadily declining training loss. This indicates effective learning without signs of overfitting, despite the relatively small dataset size.

The classification report on the validation dataset showed strong performance in both classes—'fire' and 'no fire'. High precision implies that the model rarely misclassified 'no fire' images as 'fire', which is critical for avoiding false alarms in real-world applications. Similarly, high recall values ensure that actual fire incidents are correctly detected, reducing the chances of missing crucial events. The F1-score, which balances both precision and recall, further confirmed the model's overall reliability.

The confusion matrix, both raw and normalized, revealed that the majority of images were correctly classified, with only a small number of misclassifications. Most of these errors occurred in visually ambiguous cases such as misty backgrounds, sunset glows, or dense orange foliage, which may resemble fire-like characteristics. Despite these challenges, the system maintained robust accuracy.

Visual inspection of predictions further reinforced model performance. Sample predictions from the validation set—overlaid with their true and predicted labels—demonstrated the model's strong generalization capabilities. The model was not only accurate on clear and distinct fire images but also effective in borderline or visually complex scenes.

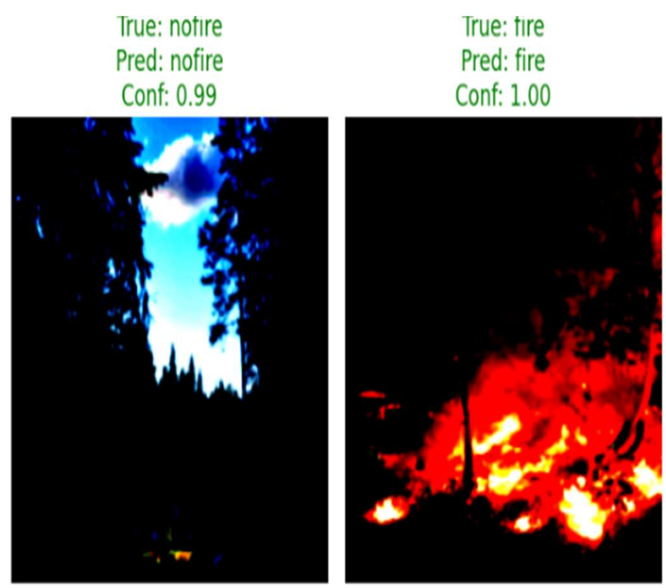

Fig 7. Prediction on few images.

A significant contributor to the model's success was the custom preprocessing pipeline. The transformation of images to the HSV color space enhanced the model's sensitivity to the warm color tones typically associated with flames. The Gaussian blur further helped reduce high-frequency noise, allowing the model to focus on critical regional patterns instead of irrelevant texture details. This preprocessing step was particularly useful in dealing with environmental noise such as smoke, foliage, and lighting variation.

Overall, the proposed system showed promising results, confirming that lightweight deep learning models like ResNet18, when combined with targeted preprocessing and Transfer Learning, can be effectively used for real-time forest fire detection. These results pave the way for deploying this system in real-world scenarios such as drone-based monitoring or automated surveillance networks.

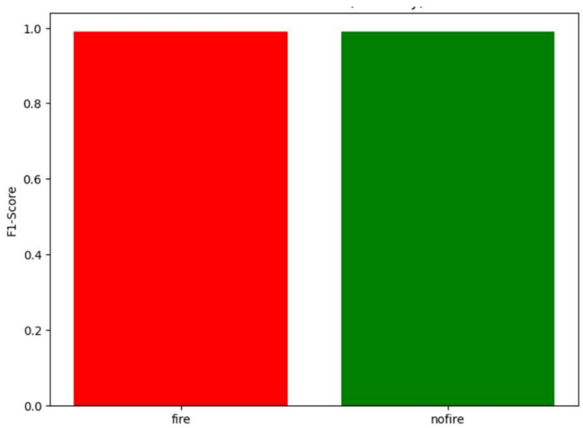

Fig 8. Class-wise F1-Score (Accuracy).

VI. Conclusion

This study demonstrates the effectiveness of Transfer Learning using ResNet18 for forest fire detection from images. The proposed approach is efficient, accurate, and suitable for real-time applications in surveillance systems. By combining deep learning with custom preprocessing, the system is capable of identifying fire instances even in challenging environmental conditions. The integration of color-space transformations and blurring techniques adds robustness to the detection mechanism.

To visually represent the approach, the flowchart below illustrates the step-by-step methodology employed in our research:

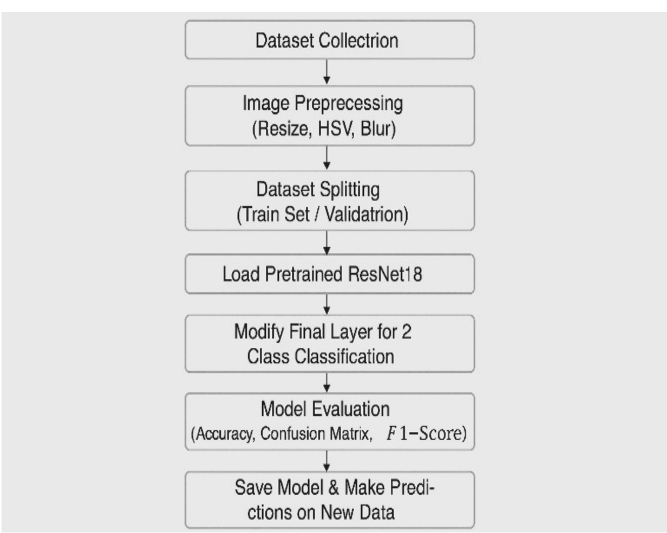

Fig 9. Forest Fire Detection using Transfer Learning.

Future Work includes:

Expansion of dataset with more diverse scenes, including various weather conditions, lighting scenarios, and geographical regions, to improve model generalization.

Real-time integration with aerial surveillance platforms such as drones or UAVs for on-the-fly detection and alerting systems.

Deployment on low-power edge devices like Raspberry Pi or NVIDIA Jetson for remote forest monitoring.

Comparative analysis using other state-of-the-art architectures such as EfficientNet, MobileNet, or Vision Transformers to explore trade-offs between performance, speed, and resource utilization.

Addition of fire segmentation capabilities to not just detect but localize the fire regions within the image.

VII. References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.Ng, A. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the 21st International Conference on Machine Learning (ICML).

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Adv. Neural Inf. Process. Syst., vol. 25, 2012.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] S. Muhammad, M. Khan, and S. Rho, "Fire Detection Using Deep Convolutional Neural Networks with Spatial and Channel Attention," *Sensors*, vol. 21, no. 4, p. 1345, 2021.

[5] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in Proc. Int. Conf. Learn. Represent., 2021.

[6] A. Deng, M. Liu, and Z. Wang, "A Forest Fire Detection System Based on Ensemble Transfer Learning," IEEE Access, vol. 8, pp. 93441–93453, 2020.

[7] J. Chen et al., "Real-Time Fire Detection Using Improved YOLO Models with Attention Mechanism," Fire Technol., vol. 59, pp. 109–129, 2023.

[8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA: MIT Press, 2016.

[9] PyTorch Documentation: https://pytorch.org/docs/stable/index.html

[10] OpenCV Documentation: https://docs.opencv.org

[11] Scikit-learn Documentation: https://scikit-learn.org/stable/documentation.html