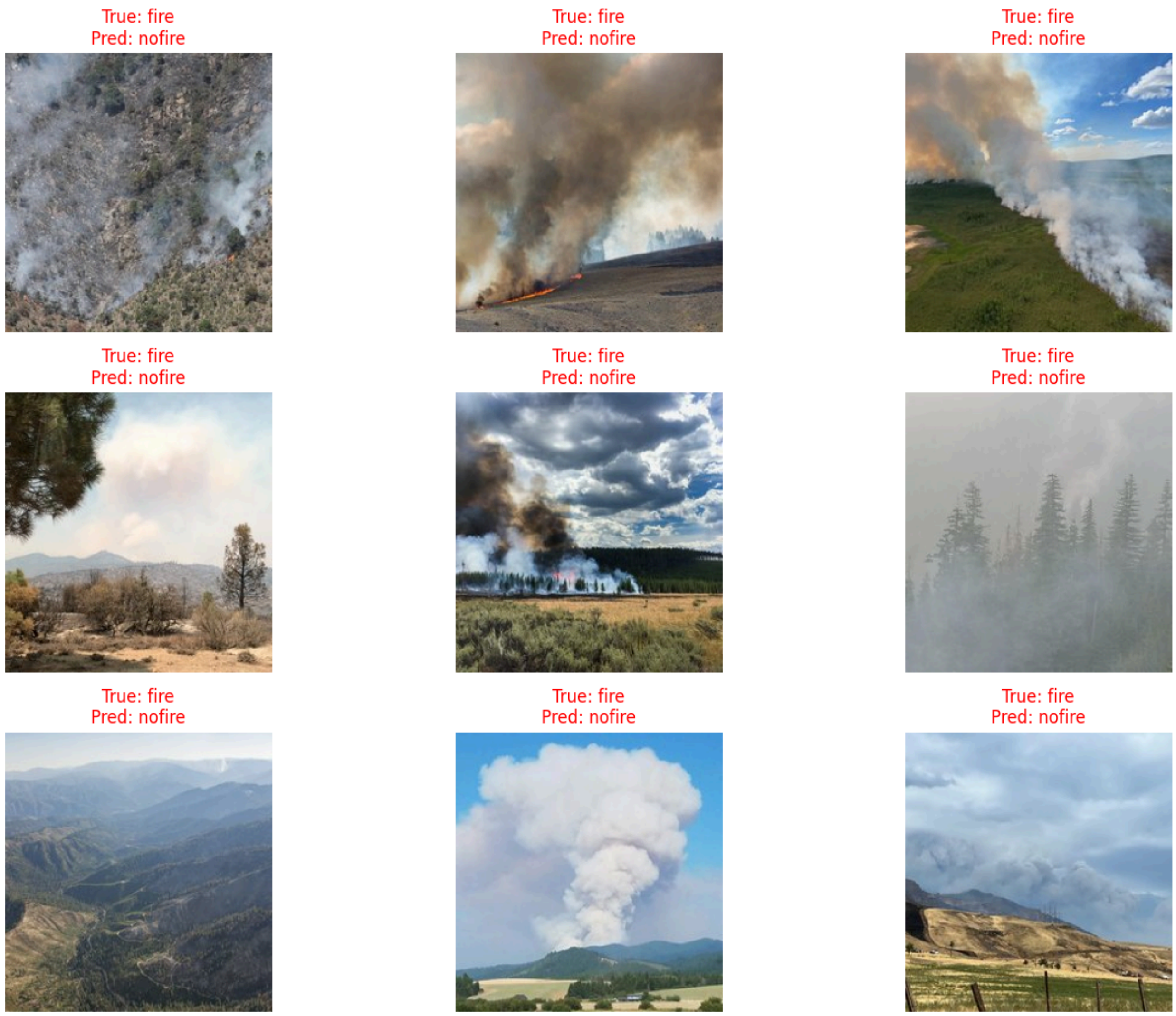
Early Wildfire Detection from Forest Images Using Transfer Learning

Jack Meghreblian, Vernon Walker, & Brody Kasprzak

April 22nd, 2026



Abstract

Wildfires are a major environmental and public safety concern, making early detection an important application of artificial intelligence. This project investigates whether deep learning can classify forest images based on whether wildfire is present. We use the Kaggle Wildfire Dataset and develop a transfer learning pipeline for binary image classification. Our modeling process followed an iterative path, beginning with an EfficientNetB0 baseline and later shifting to a ResNet50V2-based approach with data augmentation, class weighting, and threshold adjustment. The final selected model achieved a test accuracy of 88.15%, with strong class-level performance across both fire and no-fire images. At the optimized threshold of 0.3, the model achieved a precision of 0.84, recall of 0.88, and F1-score of 0.86 for the fire class, while the no-fire class achieved a precision of 0.93, recall of 0.90, and F1-score of 0.91. These results show that transfer learning can be effective for wildfire image classification, while also highlighting the importance of class-sensitive evaluation metrics rather than relying on accuracy alone.

Introduction

Wildfires cause major environmental damage, threaten communities, and create significant economic and public safety costs. In addition to destroying vegetation and wildlife habitats, wildfires can damage infrastructure, reduce air quality, disrupt local economies, and place pressure on emergency response systems. Because fire conditions can spread quickly once ignition occurs, early detection is especially important. A system that identifies wildfire presence sooner can help decision-makers respond more quickly, allocate resources more effectively, and potentially reduce the scale of damage. For these reasons, wildfire monitoring has become an important real-world application of computer vision and artificial intelligence.

This project investigates whether deep learning can detect wildfire presence in images of forest areas. More specifically, the goal is to build a binary image-classification model that predicts whether wildfire is present in a given image. We use the Kaggle Wildfire Dataset and apply a transfer learning approach to classify images into two categories: fire and no fire. This problem is well suited to supervised learning because each image is associated with a known class label, allowing the model to learn visual patterns that distinguish wildfire scenes from non-fire scenes. At the same time, the task is more challenging than a simple two-class setup may suggest. Fire-related imagery can vary greatly depending on smoke density, flame visibility, lighting, weather conditions, camera angle, and the complexity of the forest background.

Deep learning is a strong fit for this problem because convolutional neural networks can learn visual features directly from image data rather than relying on manually designed rules. Transfer learning is especially useful because it allows a model to build on features learned from large-scale image datasets and adapt them to a more specialized task. In this project, the modeling process was iterative rather than one-step. We began with an EfficientNetB0 baseline, introduced data augmentation, and then shifted to a ResNet50V2-based model with class weighting when the initial approach failed to identify fire images reliably. This progression was important because wildfire detection depends on subtle visual cues such as faint smoke, low-contrast flames, and complex forest backgrounds that may not be captured well by a simpler baseline.

The completed project moved beyond the original baseline stage and produced a final selected model with strong performance on the wildfire classification task. The fine-tuned model achieved a test accuracy of 88.15% and showed balanced class-level results across both fire and no-fire images. In addition to reporting accuracy, we evaluated the model using precision, recall, F1-score, and threshold analysis in order to better understand how well it identifies wildfire cases. This broader evaluation is important because a model that appears strong on overall accuracy may still be less useful in practice if it misses too many actual fire images.

The project therefore contributes in two ways. First, it demonstrates that transfer learning can be an effective method for binary wildfire image classification. Second, it shows why class-sensitive evaluation matters in safety-related detection problems. Rather than focusing only on whether the model predicts the correct label in aggregate, the project examines how well the model performs on the wildfire class specifically and how threshold tuning can change the balance between false negatives and false positives. This makes the project relevant not only as a technical modeling exercise, but also as a practical example of how AI systems should be evaluated in high-stakes settings.

The remainder of the paper is organized as follows. The next section reviews relevant prior research on deep learning and wildfire detection. The paper then describes the dataset, preprocessing steps, and transfer learning methodology used in the final model. After that, the experiments and results section presents the model's performance, including accuracy, class-level metrics, and threshold analysis. The paper concludes with key findings, limitations, and future directions for improving wildfire image classification.

Related Work

Deep learning has become a standard approach for image classification because convolutional neural networks can learn hierarchical visual features directly from raw image data. More recent architectures such as EfficientNet have improved model performance by scaling depth, width, and input resolution in a more balanced way, making them stronger choices for transfer learning tasks (Tan & Le, 2019). In wildfire and smoke detection, researchers have increasingly applied deep learning methods instead of relying only on traditional image-processing techniques. This shift is important because wildfire imagery can be highly variable, with differences in smoke density, flame visibility, lighting, weather, and forest background making detections more difficult. Recent review work shows that computer vision and deep learning now play a central role in modern fire detection systems because they are better able to handle these complex visual conditions than older handcrafted approaches (Özel et al., 2024).

Several published studies are relevant to our project because they apply deep learning directly to fire and smoke imagery Govil et al. (2020) developed a wildfire detection system using remote camera images and showed that deep learning could support early smoke detection in a practical monitoring setting. Their work is useful because it demonstrates that image-based models can contribute to real-world wildfire surveillance rather than only laboratory-style classification tasks. At the same time, the broader review by Özel et al. emphasizes that wildfire detection remains challenging because fire-related imagery often includes visually complex conditions such as haze, partial smoke, shifting light, and cluttered natural backgrounds, all of which can reduce model reliability (Özel et al., 2024).

Our project is most closely aligned with prior work that uses transfer learning for image-based wildfire detection. Like other fire detection studies, we rely on pretrained convolutional backbones rather than training a model entirely from scratch. However, our modeling process was iterative: we began with an EfficientNetB0 baseline and later shifted to ResNet50V2 with class weighting after observing that the initial setup overpredicted the majority no-fire class and failed to distinguish wildfire images reliably. Compared with many published systems, our project remains narrower in scope because it focuses on binary image classification rather than real-time deployment or smoke-specific monitoring pipelines (Govil et al., 2020).

Overall, the related literature supports two key ideas for this project. First, transfer learning is a reasonable baseline strategy because pretrained architectures such as EfficientNet have shown strong performance across image-based tasks and transfer settings (Tan & Le, 2019). Second, wildfire detection remains a difficult problem because environmental variability can make reliable recognition harder, which helps explain why class-level evaluation remains important in addition to overall accuracy (Özel et al., 2024).

Data

We used the Kaggle Wildfire Dataset to train and evaluate our wildfire detection model. The dataset contains labeled images for a binary classification task in which each image is categorized as either fire or no fire. Because the dataset already includes predefined training, validation, and test splits, it is well suited for supervised learning and allows the modeling process to focus on classification rather than manual labeling or custom data partitioning (El-Madafri, 2023).

In the final implementation, the dataset consisted of 1,878 training images, 398 validation images, and 405 test images. The training set included 723 fire images and 1,155 no-fire images. The validation set contained 152 fire images and 246 no-fire images, while the test set contained 154 fire images and 251 no-fire images. These counts show that the dataset is moderately imbalanced toward the no-fire class, which is important because class imbalance can affect how the model learns decision boundaries and can make overall accuracy appear stronger than class-level performance actually is. This moderate class imbalance also influenced later modeling decisions, particularly the introduction of class weighting to reduce the tendency to overpredict the no-fire class.

The images were loaded from directory-based train, validation, and test folders using TensorFlow. Each image was resized to 224 × 224 pixels so that all observations could be processed consistently by the model. Pixel values were also normalized from the original 0–255 range to a 0–1 scale, which helps improve training stability and makes the data more suitable for neural network input. These preprocessing steps ensured that the model received standardized input across all three splits.

One advantage of this dataset is that it supports a complete image-classification pipeline without requiring us to manually label images or design our own evaluation split. At the same time, the dataset remains challenging because wildfire imagery can vary substantially in flame visibility, smoke density, distance from the fire, lighting conditions, weather, and forest background. Some images may show clear visible flames, while others may contain only smoke or subtle fire-related cues. This visual variability makes it more difficult for the model to learn patterns that consistently separate fire from no-fire scenes.

The dataset also reflects an important real-world difficulty in wildfire detection: fire-related features are not always large, bright, or obvious. In some cases, smoke may be faint or partially obscured by trees, and in others the background itself may contain visually complex textures that resemble smoke or haze. For that reason, descriptive analysis of the data is important alongside model training. Even before evaluating the model, the dataset suggests that wildfire detection is not simply a matter of identifying bright flames, but rather a more nuanced image-recognition task that depends on multiple visual cues.

Methods

Our modeling approach followed an iterative development process rather than a single one-step training pipeline. The overall goal was to move from a general transfer learning baseline to a more specialized wildfire classifier that could better recognize the visual characteristics of fire and smoke. A central strategy throughout the project was transfer learning, which allowed us to build on models that had already been pretrained on the large-scale ImageNet dataset rather than training a deep convolutional network entirely from scratch. This approach was especially appropriate for our project because the wildfire dataset was much smaller and more specialized than the datasets typically required for full-scale deep learning training.

The first phase of the project used EfficientNetB0 with pretrained ImageNet weights as the initial backbone model. In this setup, the pretrained base was frozen so that its previously learned feature extractors could be reused while the custom classification layers adapted to the wildfire task. The motivation for beginning with EfficientNetB0 was that it is an efficient and widely used transfer learning architecture that performs well on many image-classification problems. However, the initial baseline plateaued at roughly 62% accuracy and showed strong majority-class behavior, effectively predicting the no-fire class for nearly all examples. This suggested that the baseline model was not learning wildfire-specific visual distinctions well enough to separate the two classes.

To address this issue, the second step introduced data augmentation through random transformations such as flipping, rotation, and zoom. The purpose of this change was to improve generalization by exposing the model to greater visual variation during training. This was important because wildfire imagery can vary substantially in camera angle, distance, smoke shape, and scene composition. By augmenting the data, we attempted to encourage the model to focus on more stable fire-related features rather than memorizing the appearances of specific training images. Although augmentation helped reduce some overfitting, it did not fully solve the earlier performance problem, and the model still remained stuck near the same accuracy range.

At that point, we reconsidered the suitability of the original backbone architecture. One likely issue was that EfficientNetB0, while strong for many object-recognition tasks, may not have been as well matched to wildfire imagery because fire and smoke often do not appear as sharply bounded objects. Instead, they are amorphous visual patterns with diffuse textures and irregular shapes. In addition, the model appeared to fall into a local minimum created by the class imbalance in the dataset, where consistently predicting the majority no-fire class produced acceptable but misleading accuracy.

To improve performance, we pivoted to a second modeling phase using ResNet50V2 as the backbone architecture and introduced class weighting during training. ResNet50V2 was selected because residual connections often support more stable optimization and stronger gradient flow in difficult image-classification tasks. Class weighting was added to address the imbalance between fire and no-fire examples by penalizing errors on the fire class more heavily. This forced the model to pay greater attention to wildfire images instead of defaulting to the majority class. Together, these changes made the model better aligned with the practical objective of wildfire detection, where missing true fire cases is generally more harmful than producing some additional false alarms.

In a later experiment, we also tested a hard-example weighting approach intended to reduce cloud-related false alarms and improve performance on visually ambiguous images. This adjustment was motivated by the fact that cloud-like textures and haze can resemble smoke and therefore create especially difficult false-positive cases. Although this experiment addressed a meaningful practical challenge in wildfire detection, it did not outperform the selected ResNet50V2 model and was therefore not used as the final reported system.

Table 1. Summary of modeling development process

Stage	Model / Change	Purpose	Outcome
Initial baseline	EfficientNetB0 with frozen pretrained layers	Establish transfer learning baseline	Stalled near 62% accuracy and overpredicted no-fire
First improvement	Data augmentation	Improve generalization across visual variation	Reduced some overfitting but did not solve class bias
Second phase	ResNet50V2	Use a backbone better suited for difficult image classification	Improved learning stability and class separation
Class balancing	Class weights	Penalize missed fire cases more heavily	Reduced majority-class bias
Final evaluation	Threshold lowered to 0.3	Improve sensitivity to wildfire cases	Increased fire recall while maintaining strong overall performance
Additional Experiment	Hard-example weighting / cloud-focused adjustment	Reduce cloud-related false alarms	Did not outperform selected model

As shown in Table 1, the project evolved through a sequence of targeted changes rather than a single modeling decision. Each stage addressed a specific weakness observed in the earlier setup, moving the model from a general transfer learning baseline toward a more specialized wildfire classifier. Although an additional cloud-focused hard-example weighting experiment was also explored, it did not outperform the selected ResNet50V2 model. This iterative process was important because it allowed us to compare results across stages and better understand which adjustments produced meaningful improvements.

After training the improved model, we evaluated performance using several complementary metrics. Accuracy was included as an overall summary measure, but it was not treated as sufficient on its own. Because this is a safety-related classification problem, we also used precision, recall, F1-score, and confusion matrix analysis to assess class-level behavior. Recall for the fire class was especially important because it captures how often the model correctly identifies wildfire cases instead of overlooking them. In the final stage of evaluation, we also adjusted the decision threshold to 0.3 in order to reduce false negatives and improve wildfire sensitivity. This threshold analysis helped show that model performance depends not only on the trained architecture, but also on how predictions are translated into final class decisions.

Empirical Applications, Experiments, and Results

The reported final results are most meaningful when interpreted in light of the project's iterative development process. The initial EfficientNetB0 baseline established a working transfer learning pipeline, but its class-level behavior showed that it struggled to identify fire images reliably and tended to overpredict the no-fire class. After introducing data augmentation, switching to ResNet50V2, applying class weights, and adjusting the decision threshold, the selected final model achieved much stronger and more balanced performance across both classes. These changes show that the project's improvements were not only numerical but also methodological, as each stage addressed a specific weakness observed in the earlier setup.

At the default evaluation stage, the selected final model achieved strong overall performance on the test set, indicating that transfer learning was an effective approach for this binary image-classification task. However, because wildfire detection is a safety-related problem, overall accuracy alone does not provide a complete picture of performance. For that reason, we also examined class-level metrics and threshold adjustment in order to better understand how the model handled missed fire cases and false alarms.

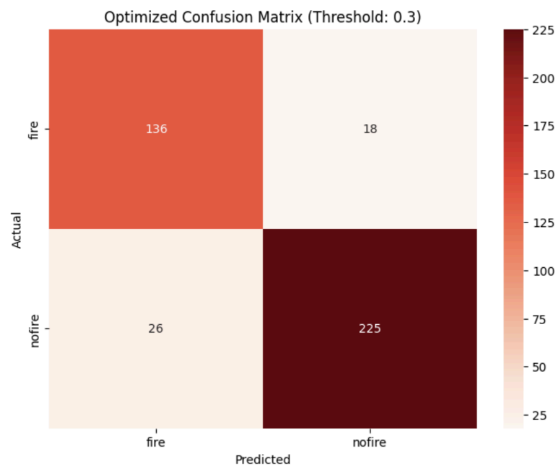


Figure 1. Confusion matrix at threshold 0.3

To further reduce false negatives, we lowered the decision threshold to 0.3 and evaluated the optimized model under this setting. Figure 1 presents the confusion matrix for the optimized model using this threshold. Under this setting, the model correctly classified 136 of the 154 fire images and 225 of the 251 no-fire images. It misclassified 18 fire images as no fire and 26 no-fire images as fire. These results show that the model was able to identify the large majority of wildfire cases while still maintaining strong classification performance on no-fire images.

Table 2. Performance at optimized threshold (0.3)

Class	Precision	Recall	F1-score	Support
Fire	0.84	0.88	0.86	154
No Fire	0.93	0.90	0.91	251

As shown in Table 2, the optimized model achieved a precision of 0.84, recall of 0.88, and F1-score of 0.86 for the fire class. For the no-fire class, it achieved a precision of 0.93, recall of 0.90, and F1-score of 0.91. Overall accuracy under the optimized threshold was 0.89. These results indicate that the model maintained strong overall performance while improving sensitivity to wildfire cases. Although additional later experiments were also explored in the notebook, including a cloud-focused hard-example weighting approach, they did not outperform the selected ResNet50V2 model reported here.

Taken together, the confusion matrix and classification metrics show that the selected final model performs well as a binary wildfire image classifier and that threshold tuning can meaningfully improve its practical usefulness. In a real wildfire detection setting, reducing missed fire cases is especially important, since false negatives may delay response to real fire events. The optimized threshold therefore provides a better balance between sensitivity and overall performance than a default decision rule would. More broadly, these findings reinforce an important lesson from the project: wildfire detection models should be evaluated with class-level metrics and confusion matrices rather than relying on accuracy alone.

Conclusion

This project investigated whether deep learning could detect wildfire presence in images of forest areas using a binary image-classification approach. Using the Kaggle Wildfire Dataset and a transfer learning pipeline, we developed a model capable of classifying images as either fire or no fire. The final selected model achieved strong performance, with a test accuracy of 88.15% and solid precision, recall, and F1-scores across both classes. These findings show that transfer learning can be an effective strategy for wildfire image classification, especially when combined with iterative refinement, fine-tuning, and careful evaluation.

One of the main lessons from the project is that performance in wildfire detection must be interpreted at the class level rather than through accuracy alone. The selected final model performed well overall, but the threshold analysis showed that changing the classification threshold can alter the balance between missed fires and false alarms. That result is especially important in wildfire detection, where missing a real fire case may be more harmful than generating a small number of extra warnings. For this reason, metrics such as recall, precision, F1-score, and confusion matrices are essential for understanding whether the model is actually useful in practice.

The project also shows the value of moving from a baseline benchmark to a more refined final system. Establishing the initial EfficientNetB0 baseline made it easier to understand the original weaknesses of the model, while the later shift to ResNet50V2 with class weighting and threshold adjustment produced substantial gains. Although an additional cloud-focused hard-example weighting experiment was also explored, it did not outperform the selected ResNet50V2 system. This progression helped show not only that the final selected model works better, but also why methodological choices such as architecture selection, class balancing, and threshold tuning matter.

From a broader perspective, the project carries both social and managerial implications. Socially, wildfire detection systems have the potential to support earlier response and reduce harm to communities, ecosystems, and infrastructure. At the same time, an unreliable model could create risk by either missing real fires or producing excessive false alarms. Managerially, this means AI-based wildfire detection should be treated as a decision-support tool rather than a fully autonomous replacement for human judgment. Emergency management settings often require balancing speed, accuracy, and resource allocation, and a model like ours is most useful when it helps human operators make faster and more informed decisions rather than acting alone.

Future Work

Although the final selected model performed well, there are still several directions for improvement. One next step would be to test alternative pretrained architectures and compare them directly with the current model. Additional data augmentation and more extensive hyperparameter tuning could also improve robustness, especially under more difficult visual conditions such as haze, weak flames, partial smoke, or cloud-like textures that may resemble smoke. An additional hard-example weighting experiment aimed at reducing cloud-related false alarms was explored in the notebook, but it did not outperform the selected ResNet50V2 model, suggesting that cloud-specific error reduction remains an important area for future work.

Another important direction would be to evaluate the model on more diverse wildfire imagery, including data collected under different environmental settings or from different camera perspectives. This would help determine whether the model generalizes beyond the current dataset. Future work could also explore smoke localization or segmentation rather than only binary classification, bringing the project closer to real-world wildfire monitoring systems.

Finally, additional threshold tuning could help adapt the model to different deployment goals. In some settings, minimizing missed fire cases may be more important than reducing false alarms, while in other settings the tradeoff may differ. Exploring these tradeoffs further would help make the model more useful in practice.

References

El-Madafri, I. (2023). The wildfire dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/elmadafri/the-wildfire-dataset

Govil, K., Welch, M. L., Ball, J. T., & Pennypacker, C. R. (2020). *Preliminary results from a wildfire detection system using deep learning on remote camera images*. *Remote Sensing, 12*(1), 166. https://doi.org/10.3390/rs12010166

Özel, B., Alam, M. S., & Khan, M. U. (2024). *Review of modern forest fire detection techniques: Innovations in image processing and deep learning*. *Information, 15*(9), 538. https://doi.org/10.3390/info15090538

Tan, M., & Le, Q. (2019). *EfficientNet: Rethinking model scaling for convolutional neural networks.* In *Proceedings of the 36th International Conference on Machine Learning* (Proceedings of Machine Learning Research, Vol. 97, pp. 6105–6114). PMLR