Camp VeriFIRE: An Image and Weather Based Fire Risk Prediction System

Karthik Sankaran
UC San Diego, San Diego, CA, USA
ksankaran@ucsd.edu

***Abstract*—Wildfire risk is growing worldwide, but convenient, localized risk information remains inaccessible to many outdoor enthusiasts. Because of this, I developed *Camp VeriFIRE*, an application that combines environmental images and local weather metadata to calculate a real-time fire risk score. Users upload a photo of their site, input location data automatically or manually, and the model integrates the temperature, humidity, and wind speed of the given region to issue a clear risk level. Deployed via a Flask web service with an interactive front end, Camp VeriFIRE provides campers with real-time situational awareness, potentially reducing the frequency of campfire-sourced wildfires.**

***Index Terms*—Wildfires, Fire risk assessment, Convolutional neural networks, Computer vision, Weather Metadata, Environmental monitoring, Mobile applications**

I. Introduction and Motivation

In recent years, California has witnessed an alarming surge in the frequency, intensity, and destructiveness of wildfires. The most relevant and recent of which, the Los Angeles fires, have displaced thousands, destroyed countless homes, and pushed emergency response systems to their limits. As climate change accelerates and urban development encroaches further into fire-prone areas, wildfires are no longer rare natural disasters but inevitable and constant occurrences.

This project, *Wildfire Risk Predictor*, was conceived in direct response to this growing crisis which has affected friends and family. Sparked by disruption caused by fires across Southern California, I set out to explore how machine learning and environmental sensing could offer a practical tool for early wildfire risk detection at the source.

My attention turned to anthropogenic fire sources, such as unattended campfires, debris burning, fireworks, and equipment use, which are significant contributors to the ignition of wildfires. In states like Idaho, for example, more than 50% of wildfires are caused by human activity, such as campfires and debris burning. These fires frequently occur in remote or rural areas, often near campsites, hiking trails, or sparsely inhabited private lands. Because these locations are isolated and lack robust infrastructure, fire detection and emergency response times tend to be slow. In some cases, fires can grow rapidly before local authorities are even alerted, especially if the fire starts during low activity periods or goes unreported. This delayed response, combined with dry vegetation and wind, can quickly turn a small, manageable fire into a widespread hazard. Recognizing this, I aimed to design my system with an understanding of how human tendencies

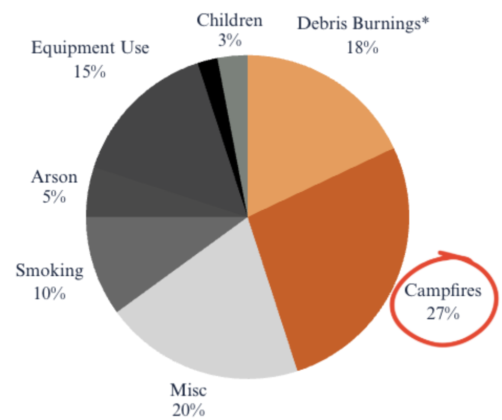

Fig. 1. "Common Causes of Wildfires." Idaho Firewise

intersect with geographic conditions. By accounting for these high-risk scenarios, especially where early detection and rapid response are difficult, my model can better reflect wildfire risk in practice.

By combining image-based vegetation analysis with localized environmental data such as wind speed, temperature, and humidity, the system aims to provide users with real-time assessments of wildfire risk in their immediate surroundings. The goal is to provide individuals, especially those in communities prone to fires, with predictive insight that can inform evacuation preparation, fire prevention, and awareness.

This approach reflects a broader need in environmental data science to develop scalable, deployable tools that bridge the gap between raw data and actionable decisions. As wildfires become more frequent and unpredictable, tools like this can serve as an early line of defense, especially when deployed on mobile platforms for instant location-based feedback.

II. Related Work

Deep learning models for wildfire detection have leveraged satellite imagery [4], or combined visual and contextual data for event classification. However, previous systems focus on large-scale remote sensing or require centralized deployments, limiting their utility for recreational users.

Community-driven efforts such as Oguilmak's open-source project on wildfire prediction from satellite imagery [7] demonstrate the potential of machine learning pipelines built on publicly available datasets, though these remain tied to orbital platforms. Similarly, industry and research groups have explored integrating weather and sensor data with image-based models. For example, Google Cloud reported on early efforts using TensorFlow to classify vegetation fuel conditions and combine them with meteorological data for predicting fire spread [8]. But these efforts have required fine-tuned hardware that is inaccessible to the common man.

In contrast, this approach shifts the paradigm to mobile, site-specific risk assessment using smartphone imagery and publicly accessible weather data, providing a lightweight and accessible tool for end-users.

III. Methodology

A. Data Collection and Preprocessing

I compiled a dataset of hundreds of labeled outdoor images featuring fire-relevant scenes (e.g., brushes, eucalyptus trees, rainforests). These images were sourced via Google image search using keyword queries generated by a large language model (LLM) to reflect both high fire risk environments (e.g., dry brush, coniferous forests) and low fire risk environments (e.g., moist rainforests, riversides). The images were resized to 128×128 px and normalized. The data was augmented using flips, rotations, and brightness changes. Concurrently, I collected temperature, humidity, and wind speed via OpenWeatherMap.

B. Model Architecture

The model builds on a lightIight CNN (MobileNetV2) pretrained via transfer learning in TensorFlow. I merged visual features with weather metadata before passing them through fully connected layers. The fused output produces a continuous risk score normalized to real values.

C. Model Training and Implementation

I built the fire risk classifier using MobileNetV2 as the convolutional neural network (CNN) backbone, initialized with weights pretrained on ImageNet. I chose MobileNetV2 because of its efficiency and suitability for deployment on lightweight systems like mobile or web platforms. Starting with pretrained weights allows the model to use generic visual features learned from a large dataset, saving time and memory. All images were resized to 128×128 pixels, normalized to $[0, 1]$, and augmented using horizontal flips, random rotations, and brightness adjustments via Keras' `ImageDataGenerator` to improve generalization and reduce overfitting.

Additionally, to avoid overfitting on the relatively small dataset, I froze the convolutional layers of MobileNetV2 and added a custom classification head. On top of the frozen base, I applied a global average pooling layer to capture important feature activations, a 30% dropout layer for regularization, a dense layer with 64 ReLU-activated units for non-linear learning, and a final sigmoid neuron to output a binary fire risk probability. I trained the model using the Adam optimizer with binary cross-entropy loss, a batch size of 32, and ran training for 10 epochs.

I monitored performance using a 20% validation split, tracking accuracy, precision, recall, and F1-score. After training, I generated a classification report and computed the AUC-ROC to evaluate predictive reliability on the held-out validation set. Figure 2 shows the training and validation loss across epochs, demonstrating convergence and stable generalization.

D. Weather Integration and Risk Scoring

Alongside the image-based classification, I combine real-time weather conditions to adjust the fire risk score. I pull temperature, humidity, and wind speed data from the OpenWeatherMap API, using either the user's city or GPS coordinates. These values are then used to scale the CNN's probability output, with the idea that even a low-risk image can become hazardous under hot, dry, or windy conditions.

The way I handle this is fairly simple — I apply a set of multipliers to the image score when the weather conditions cross certain thresholds:

Here, T: temperature (°C), W: wind speed (m/s), and H: relative humidity (%). The multipliers for each are defined as:

$$M_T = \begin{cases} 1.3 & \text{if } T > 32°\text{C}, \\ 1.2 & \text{if } 21°\text{C} < T \leq 32°\text{C}, \\ 1 & \text{otherwise.} \end{cases}$$

$$M_W = \begin{cases} 1.5 & \text{if } W > 8 \text{ m/s}, \\ 1.2 & \text{if } 3.5 \text{ m/s} < W \leq 8 \text{ m/s}, \\ 1 & \text{otherwise.} \end{cases}$$

$$M_H = \begin{cases} 1.2 & \text{if } H < 30\%, \\ 1.1 & \text{if } 30\% \leq H < 40\%, \\ 1 & \text{otherwise.} \end{cases}$$

These thresholds are admittedly heuristic. I set them after some quick research into fire weather indices and patterns cited by agencies like the USFS and NPS, but they are not rigorously optimized or derived from historical fire datasets. The goal was to incorporate the most obvious weather drivers of fire spread without making the model overly complicated or data-hungry because I had not conducted formal research into the matter.

The final continuous risk score is calculated as:

$$\text{Risk Score} = P_{\text{image}} \times M_T \times M_W \times M_H$$

where P_{image} is the CNN's predicted fire risk probability, and M_T, M_W, M_H are the temperature, wind, and humidity multipliers. This score is then bucketed into four user-facing categories: Low, Moderate, High, and Extreme

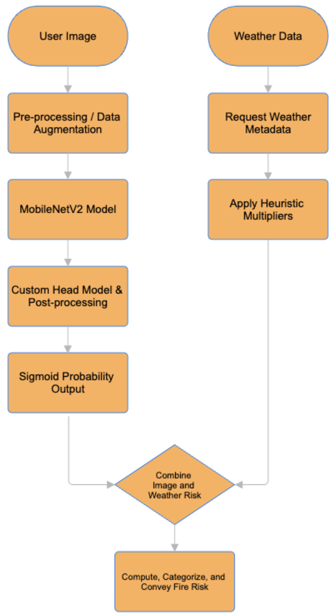

Fig. 2. High Level Project Design

E. Deployment

I built a Flask web service exposing two endpoints: '/assess-risk' (image + city) and '/verify-city'. The system fetches real-time weather, preprocesses user-uploaded photographs, invokes the CNN to compute a fire risk score, and classifies it into four categories. The frontend supports geolocation by city input and overlays results with adaptive user advice.

Note: The current implementation is hosted locally and not deployed on a public or cloud server due to cost considerations. Future work includes deploying on scalable cloud infrastructure for broader access.

IV. Results

Evaluation on a hold-out test set reveals approximately:

- Accuracy: 88.2%
- F1-Score: 0.86
- AUC-ROC: 0.91

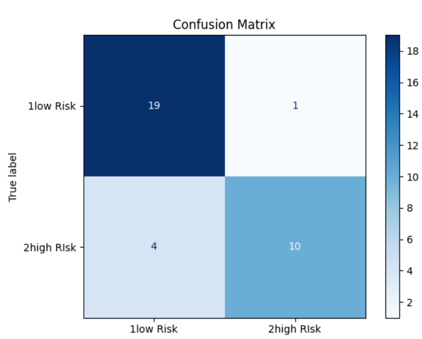

Fig. 3. Confusion matrix showing model performance on high-risk (class 1) and low-risk (class 0) predictions.

The confusion matrix shows the model is accurate overall (85%), with strong precision (90.9%) in predicting high risk cases. It correctly identifies most low risk cases but misses 4 high risk instances, suggesting it is cautious about flagging high risk, which may reduce false alarms but risks overlooking some critical cases. While this conservative approach reduces false alarms (a desirable trait for user adoption), false negatives can be quite dangerous. In wildfire prediction scenarios, false negatives carry substantially greater consequences than false positives because an undetected high-risk situation could lead to catastrophic outcomes. Further training and testing can help alleviate this shortcoming by having a more robust model but also through threshold manipulation.

V. User Interface

As shown in Fig. 4, users upload a photo and optionally enable geolocation; the UI displays the computed risk level alongside weather details and safety recommendations. The model runs in under 2 seconds per submit on the backend.

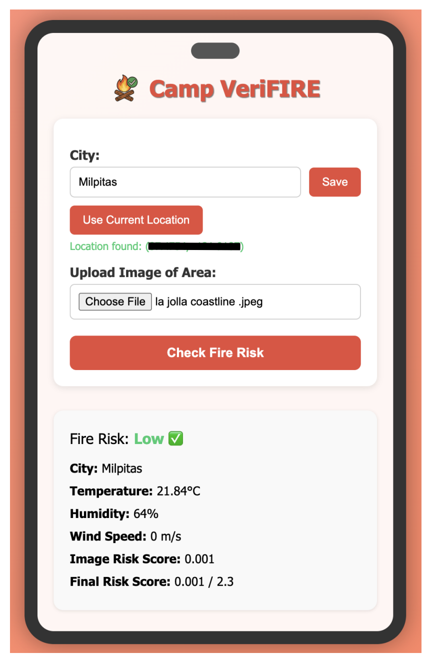

Fig. 4. Camp VeriFIRE User Interface.

VI. Discussion

Developing this system surfaced key challenges: (1) sourcing labeled fire-risk imagery, (2) real-time fusion of visual and weather modalities, and (3) maintaining efficient inference in cloud environments. Despite these hurdles, the model achieved high predictive performance and user-centric interaction times.

While the current weather multipliers were set heuristically based on general fire weather guidelines, a testing-based approach to optimize these weights would likely enhance model accuracy and robustness. By systematically evaluating historical wildfire data and correlating weather conditions with ignition likelihood, future work could employ techniques such as grid search, Bayesian optimization, or machine learning calibration methods to derive optimal multiplier values. This data-driven tuning would allow the model to more precisely reflect real-world fire risk patterns, reduce manual bias, and potentially improve predictive performance under varying environmental scenarios.

I introduced Camp VeriFIRE, a hybrid CNN–weather system offering real-time, location-based wildfire risk alerts. Future work will focus on expanding the system's capabilities through mobile app support, crowd-sourced image datasets, SMS alerts, and the integration of diverse data sources including altitude and topographic information. Partnerships with fire management agencies will further ensure robust validation and real-world applicability.

ACKNOWLEDGMENT

I thank the UC San Diego Supercomputer Center, SCILab, SDGE, WIFIRE, UCSD The Design Lab/The Basement, and Blackstone LaunchPad for mentorship and sponsorship as I partook in the Mindshifts on Megafires Design Challenge.

REFERENCES

[1] Idaho Firewise, "Common Causes of Wildfires." [Online]. Available: https://www.idahofirewise.org/common-causes-of-wildfires/. [Accessed: Jul. 17, 2025].

[2] R. Winkler, "The Causes of Wildfires and How They Are Changing," U.S. Forest Service, 2025. [Online]. Available: https://www.fs.usda.gov/nrs/pubs/other/2025/nrs_2025_winkler_001.pdf. [Accessed: Jul. 17, 2025].

[3] National Park Service, "Wildland fire causes and trends." [Online]. Available: https://www.nps.gov/articles/wildfire-causes-and-evaluation.htm. [Accessed: Jul. 17, 2025].

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. CVPR*, 2016.

[5] The AI Guy, "Train Neural Network by loading your images — TensorFlow, CNN, Keras tutorial," YouTube, May 9, 2020. [Online Video]. Available: https://www.youtube.com/watch?v=uqomO_BZ44g.

[6] Dave, "How to Create a Web Application in Python using Flask," YouTube, Aug. 2023. [Online Video]. Available: https://www.youtube.com/watch?v=jQjjqEjZK58.

[7] D. Oguilmak, "Wildfire Prediction from Satellite Imagery," GitHub repository, 2021. [Online]. Available: https://github.com/doguilmak/Wildfire-Prediction-from-Satellite-Imagery.

[8] B. Williams, "Using data and ML to better track wildfire and assess its threat levels," Google Cloud Blog, Dec. 21, 2018. [Online]. Available: https://cloud.google.com/blog/products/ai-machine-learning/using-data-and-ml-to-better-track-wildfire-and-assess-its-threat-levels.

[9] SmartDraw, LLC, "SmartDraw," Web-based diagramming and collaboration tool, 2025. [Online]. Available: https://www.smartdraw.com.