Wildfire Detection using Computer Vision

Andrew Choi
University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
jxc190056@utdallas.edu

Micheal Kempner
University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
mwk190000@utdallas.edu

Chloe Pascual
University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
cmp220004@utdallas.edu

Philip Wallis
University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
ptw190000@utdallas.edu

Abstract

Wildfires are a common and costly problem that occur globally. Early detection of wildfires is a key element necessary to respond to and prevent their spread. If computer vision techniques could be properly applied to observations of areas where wildfires are known to occur, earlier detection and response may be possible. This would likely result in the reduction of damage to life and property. Team 3 approached this problem by using PyTorch and a large dataset of images to determine if a computer could be trained to correctly identify if an image contained a wildfire. Results were encouraging, but more work would be required to make this method viable. [8]

1. Introduction

Between 2010 and 2022, wildfires burned between 4.3 million and 10 million acres of land in the United States every year [12], averaging 7.2 million acres annually [3]. These fires cause billions of dollars in damages every year [10] from personal and environmental damages. Wildfires can be more easily managed or even prevented if they are detected earlier [9]. This presents an opportunity for a study of how computer vision may be applicable in a solution.

The team approached this problem by locating a dataset containing approximately 2,700 images of wildfires [8]. These included both aerial and terrestrial photographs of varying resolutions of forests and wildfire. The images were categorized into distinct classes containing images of actual fires and smoke as well as confounding images such as sunsets or fog. This set was chosen because of its overall size as well as its numerous high-resolution images and variants which allowed for superior feature detection.

Once the dataset was chosen, a method of creating and training the model had to be determined. PyTorch was an obvious choice for a library thanks to its robust documentation and widespread use. Designing and constructing a bespoke convolutional neural network was considered, but ResNet-50 [4] was ultimately chosen as the CNN.

A potential usages of the created model is implementation with live feed from a camera mounted onto watch towers across forests to detect real-time forest fires for early identification; this can also be provided in a form of an Application Programming Interface, or API, to let anyone access the model to watch their own forests across the globe.

2. Related Work

While the subject of wildfire detection and the application of computer vision to solve the problem is an interesting idea, Team 3 was not the first to explore it. Other studies had previously been conducted with promising results, encouraging Team 3 to continue exploring the subject to see what new avenues of approach could be explored. Two similar studies were used as an inspiration and guidebook for how wildfires in computer vision have been studied before. Once review of these related works was complete, Team 3 had a more robust understanding of the subject. This allowed Team 3 to conduct a study that approached the problem from a unique point of view with a new way of attempting to tackle the issue.

2.0.1 UAV Image Capturing

"Early Wildfire Detection Technologies in Practice—A Review", a research paper by Ankita Mohapatra and Timothy Trinh, discusses various wildfire techniques, their advantages, and their challenges. Various of which use image-detection similar to this project. The technology is implemented in Unmanned Aerial Vehicles (UAV) which use

YOLOv3 and YOLOv5 networks to recognize wildfires based on video footage captured by UAV. Similar deep-learning techniques are used with stationary cameras and satellite footage to detect fires as well. These methods in practice have their own advantages and disadvantages presenting a time and place to use each one, but they demonstrate there is a space for wildfire detection technology that is varied and growing. This paper's collection of modern wildfire-recognition methods shows us the rationale to pursue creating a detection model that could be more accessible and reliable enough where it can be applied to similar technologies in the future [2].

2.0.2 Tensorflow

Another resource that proved to be useful to this project was the source code of a TensorFlow wildfire-detection model by Philo Pateer-Georgei uploaded to the data science platform Kaggle. His model demonstrated ¿85% accuracy and was a reliable resource for our foray into PyTorch classification. His model consisted of four main steps: loading dependencies, getting data and preprocessing, creating a model and compiling it, and finally adding callbacks and training the model. Although his code used TensorFlow while ours used PyTorch it served as a practical guide to kickstart this project [11] .

2.0.3 Support Vector Machines with ANN

The paper "Modeling Spatiotemporal Wild Fire Data with Support Vector Machines and Artificial Neural Networks" Georgios Karapilafis et al. is another resource that acts as a combination of the topics of the previously mentioned works. It takes the real life application of wildfire detection using machine learning like the first source and provides a more technical breakdown like the second source. The artificial neural networks (ANN), referenced in this paper take into account many factors into account as neurons in the Input Layer including: average altitude, relative humidity, air temperature, wind speed, slope, vegetation density, grassland density, intervention time and type of wildfire. Different models were created for different regions of Greece to recognize fires in different environments to test a variety of ANN models. They concluded that ANNs with the use of Soft Computing algorithms is a useful combination in the world of wildfire detection. For this project, this paper depicted another wildfire detection technique with the use of machine learning models reliably, which allows more motivation to pursue the topic on a smaller scale. The mention of taking into account other factors also presents the opportunity to build upon this model to gather more data from the dataset other than solely wildfire recognition that could allow the model to be more versatile [7].

3. Method

To approach this problem, the best decision the team came up with is to use a machine learning model to categorize between fire and no fire. By using a machine learning model with computer vision, we can identify the difference between a fire and no fire elements of the forest, as well as its subcategories. With pytorch and its modules, we can quickly train a model for this exact use case. We can load a predefined model called resnet50, and use an Adam Optimizer as well as cross entropy loss function and an lr_scheduler to change the step size for each step of training the model.

3.0.1 Data

We decided to use a wildfire data set from kaggle, which includes 2700 images of forests with and without fire as well as elements that can define fire, and elements that could be seen as a fire from a computer, such as fog, sun glare, and reflections. The images will go through several transforms, which includes resizing to smaller pixels, random horizontal flip as well as a plot with a little bit of random rotation in case there are cameras mounted or the pictures taken with a slight tilt. Finally a normalization will be applied to the images to intensify the fire as well as fog and other confounding elements. The data set will be loaded using the image folder module with the transforms applied to the images and the data loader function will be used to load the data into the tensors.

3.0.2 Evaluation

To evaluate the success of the model we will look at its losses, as well as its accuracy to the testing set as well as the validation set. A goal in this project is to reach at least 80% in accuracy, the model can be useful in the real world scenarios.

4. Experiments

4.1. Data Selection

First step taken to generating a satisfiable model is using clean and well curated data. As seen by Large Language Models such as GPT-4, a crucial step involves using clean data [1]. The team decided to use a pre-categorized dataset from Kaggle called The Wildfire Dataset which includes 2700 images of multiple forests with multiple features such as fire, smoke, sun, and fog. With a large image resolution with an average size of 4057 x 3155 pixels, it provided unique feature points that allowed the model to be trained on a dynamic set of data. The data is categorized into two folders, fire and no-fire, as well as sub-categories fire_and_smoke, fire_smoke_only, nofire_fire_confounding,

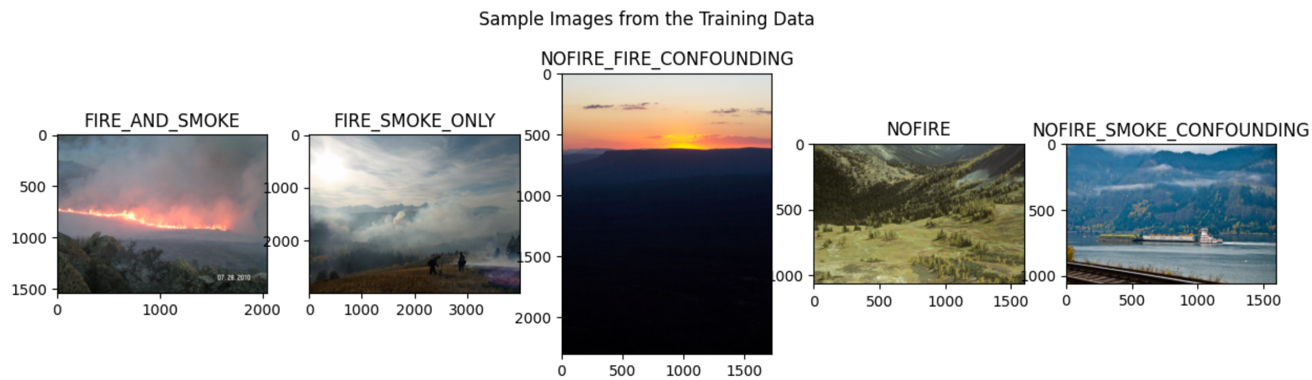

Figure 1. Sample images from the train dataset, with the pre-categorization labels. First two images on the left are part of the fire category, while the three images on the right are part of the nofire category.

nofire, and nofire_smoke_confounding. Although this depth of categorization is useful for the purposes of further categorization of the feature detection, the decision was made to only categorize the data to fire and nofire to improve model accuracy, as well as lessen the severity of the project as a whole.

4.2. Data Preparation

4.2.1 Transforms

Using PyTorch's transforms module, multiple transforms were applied to the data. First step was to resize the image to 224 x 224; resizing not only allows quicker model training, as well as testing, but it also helps in a potential use case where any camera mounted on watch towers, regardless of the image quality provided, can be used to strongly utilize the model generated. A random horizontal flip was applied to augment the data; although it may seem unnecessary, it is especially useful for forests where its terrestrial shape may be unknown. A random rotation of 10 degrees was also applied; this reduces the stress when it comes to mounting the actual live feed camera to the watchtowers, where a precise mounting mechanism won't be needed for a successful usage of the model. Finally, normalization was applied, for easier detection for the features that we want to identify, such as smokes or fire.

4.2.2 Loading the Data

The dataset had already split up data into three separate folders which included the train, test, and validation sets, with about 70% of the data in the training set. The data was loaded to tensor using Pytorch's ImageFolder and DataLoader modules. The transforms were applied to the images with the ImageFolder module, with the folder path defined. Then using the DataLoader, each set of data was loaded with a batch size of 32, and on the train data, shuffle was set to true in order to truly have a random set of data on each training step.

4.3. Model Preparation and Training

Before creating the model, we utilized CUDA to assure quick model training through the usage of a GPU, as well as set a manual seed of 42 to ensure each run will provide equal values for the results, for better reproducibility. Rather than creating a brand new model from the get-go, a choice was made to use a well-known Convolutional Neural Network called ResNet-50, to train our data. More pre-defined functions were borrowed to create an accurate model. We used an Adam optimizer, Cross Entropy Loss function, and an lr_scheduler to train the model. With 10 epochs per batch, 60 batches were trained.

4.4. Results

4.4.1 Performance Summary

The performance of our wildfire detection model demonstrated promising results in the realm of computer vision applied to environmental monitoring. With a training accuracy of 71.38% and a test accuracy of 68.29%, the model showcases its potential in accurately identifying wildfire occurrences from diverse image sets. Notably, the model adeptly distinguished between images containing fire and smoke, as well as those depicting similar but unrelated phenomena such as sun glare or fog. This distinction is crucial, as false positives in wildfire detection can lead to unnecessary resource allocation and potential delays in responding to actual fires.

The model's ability to classify images with varying features, as shown in Figure 2 through Figure 6, underlines its robustness. For instance, in Figure 2, the model correctly identified an image with both fire and smoke, showcasing its sensitivity to key wildfire indicators. Similarly, in Figure 7, the model accurately categorized an image with a

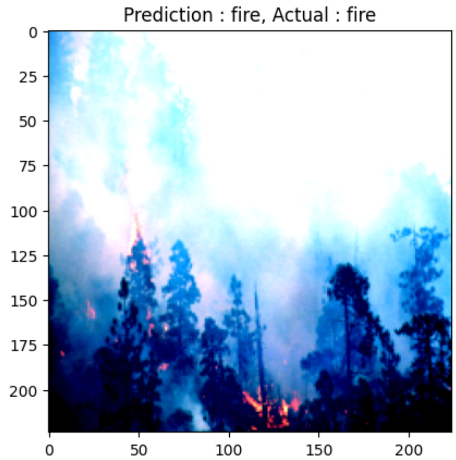

Figure 2. Prediction for an image with fire and smoke - Correct

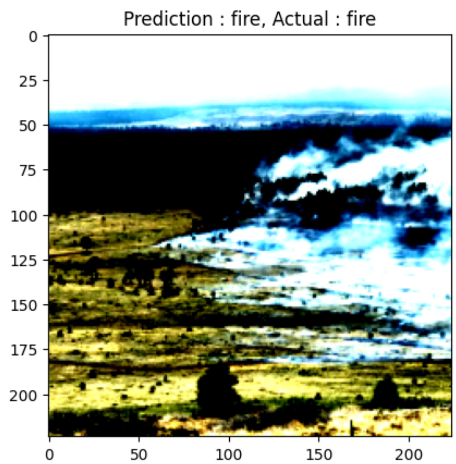

Figure 3. Prediction for an image with smoke - Correct

sunglare as a non-fire scenario, demonstrating its capability to avoid common misclassifications. These instances of accurate classification highlight the model's nuanced understanding of complex visual inputs, a critical factor in real-world applications.

Furthermore, the analysis of loss metrics, specifically a training loss of 0.81858 and a test loss of 0.64919, provides insights into the model's learning efficiency and generalization capabilities. The slight discrepancy between training and test performance indicates areas for future improvement, particularly in addressing potential overfitting issues. Nevertheless, the current model serves as a solid foundation for further refinement and application in wildfire detection systems.

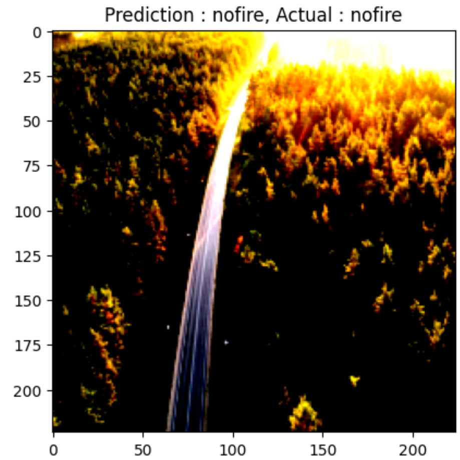

Figure 4. Prediction for an image with a sunglare - Correct

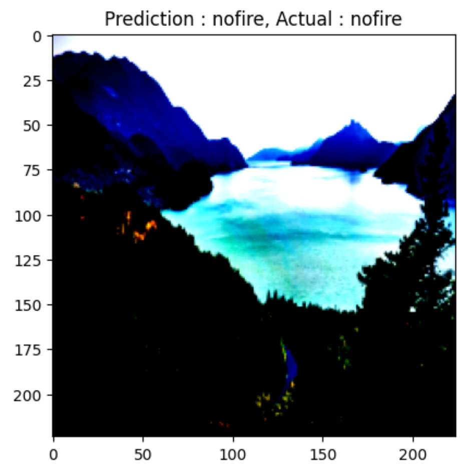

Figure 5. Prediction for an image with a regular forest - Correct

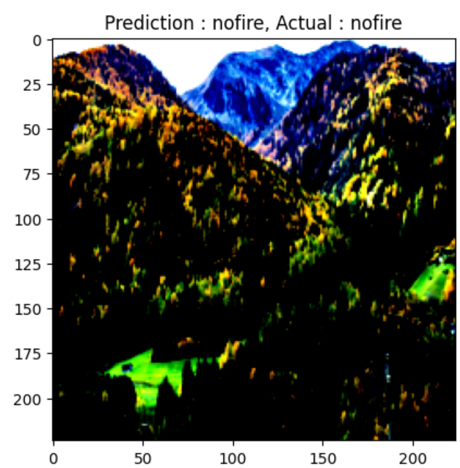

Figure 6. Prediction for image with fog - Correct

	precision	recall	f1-score	support
fire	0.63	0.51	0.57	156
nofire	0.72	0.81	0.76	246
accuracy			0.69	402
macro avg	0.68	0.66	0.66	402
weighted avg	0.69	0.69	0.69	402

Figure 7. Precision, Recall, and F-1 Score

4.4.2 Quantitative Analysis

A deeper quantitative analysis reveals that the model exhibits a commendable balance between precision and recall, with a precision of 69% and a recall of 69%. These metrics are indicative of the model's ability not only to correctly identify the presence of wildfires but also to minimize false negatives, which is crucial in emergency response scenarios. The F1-score, sitting at 69%, further confirms the model's balanced performance in both identifying true positives and avoiding false positives.

Additionally, a confusion matrix analysis showed that the model has a higher tendency to correctly classify true positives (actual wildfires) and true negatives (non-wildfire scenarios), with a relatively lower incidence of false positives and false negatives. This aspect of the model is particularly important in real-world applications, where the cost of false alarms and missed detections can be substantial.

In summary, the experimental results, backed by quantitative metrics, underscore the model's effectiveness in wildfire detection using computer vision techniques. While there is room for improvement, particularly in enhancing the model's generalizability and reducing overfitting, the current findings are encouraging. Future iterations of the model, incorporating more diverse datasets and advanced machine learning techniques, could further elevate its accuracy and reliability, making it a valuable tool in combating the ever-present threat of wildfires.

5. Conclusion

5.1. Experimental Results and Further Improvements

The model was able to achieve a training loss of: 0.81858, a training accuracy of 71.38%, a test loss of 0.64919, and finally a test accuracy of 68.29%. The slight improvement that the training results have over the test results may be due to overfitting during the training period. One of the best solutions to fixing overfitting is having a large dataset. The dataset used was quite large to begin with but adding to it using web scraping or bringing video footage could help prevent overfitting instances in the future. Also, reducing the complexity of the model can help with overfitting as well [6]. Further fine-tuning the parameters and nodes of the network can work to fight this issue. Overall, these results were not a failure, but not a stellar success either. The aforementioned improvements could be possible solutions to further increase the test accuracy and build a more dependable model.

5.2. Summary and Future Applications

With a test accuracy at 68.29%—just about a point below the industry standard for success—the wildfire detection model can be recognized as an accomplishment. Its ability to recognize fires by images of fire or smoke while being able to weed out different cases of fire-representative or unrepresentative images demonstrates its range in utility. With further improvement of the model such further experimenting with hyperparameters, using regularization, matrix analysis, and understanding the model better, could turn it into something that could be useful for application in the real-world [5]. Additional implementation for the model to be able to use live data while implementing motion detection would also bolster its practicality so that wildfires could be recognized and tracked in real time. Team 3 has also considered publishing an API for the model so that community collaboration could present new ideas to further improve on this project. Largely, there is a lot of space for the model to be built upon, but the foundation itself is already sound.

References

[1] Open AI. Improving language model behavior by training on a curated dataset, 2023. 2

[2] Timothy Trinh Ankita Mohapatra. Early wildfire detection technologies in practice—a review, 2022. 2

[3] CRSReport. Wildfire statistics, 2023. 1

[4] Datagen. Resnet-50: The basics and a quick tutorial, 2023. 1

[5] Rohith Gandhi. Improving the performance of a neural network, 2018. 5

[6] Chirag Goyal. Complete guide to prevent overfitting in neural networks (part-1), 2021. 5

[7] Iliadis L. Spartalis S. Katsavounis S. Pimenidis E. Karapilafis, G. Modeling spatiotemporal wild fire data with support vector machines and artificial neural networks, 2013. 2

[8] ISMAIL EL MADAFRI. The wildfire dataset, 2023. 1

[9] U.S. Department of Homeland Security. Technology to reduce the impacts of wildfires, 2023. 1

[10] Congressional Budget Office. Wildfires, 2022. 1

[11] Philo. 85+ accuracy, 2023. 2

[12] USAFacts Team. How many wildfires occur in the us?, 2023. 1