Fire Detection using YOLOv8: Model Selection and Future Enhancements

Alexandre Quach

September 26, 2024

Introduction

This document presents the approach used to select the top-performing model for forest fire detection using a historical dataset of fire images. The primary models considered were YOLOv5, Faster R-CNN, and YOLOv8. Due to the nature of the task and the computational constraints on Google Colab, YOLOv8 was selected as the optimal model.

Model Selection

Why YOLOv8?

YOLOv8 was selected over YOLOv5 and Faster R-CNN based on the following considerations:

- **Performance and Speed**: YOLOv8 offers significant improvements in both speed and accuracy over YOLOv5. It is specifically optimized for real-time object detection, which is critical for the early detection of fires in forested areas.

- **Efficiency in Colab**: Given the limited computational resources of Google Colab, YOLOv8's efficiency made it the best candidate. Faster R-CNN, while potentially more accurate, was computationally too heavy for the available environment.
- **Task Requirements**: Fire detection requires both speed and precision. YOLOv8's ability to quickly identify and localize fires, even when obscured or in low-light environments, aligned well with the dataset and task.

Model Performance

Initial tests using YOLOv8 showed the following:

- High precision in detecting fire locations, with accurate bounding boxes around the fire zones.
- Occasional false negatives in cases where the fire was obscured by smoke or background noise. These issues could be addressed with further data augmentation or fine-tuning.
- The model's performance in terms of speed (frames per second) was highly suitable for real-time fire detection, which is a critical requirement for this project.

Below are four images that demonstrate the model's performance in detecting fire and smoke.

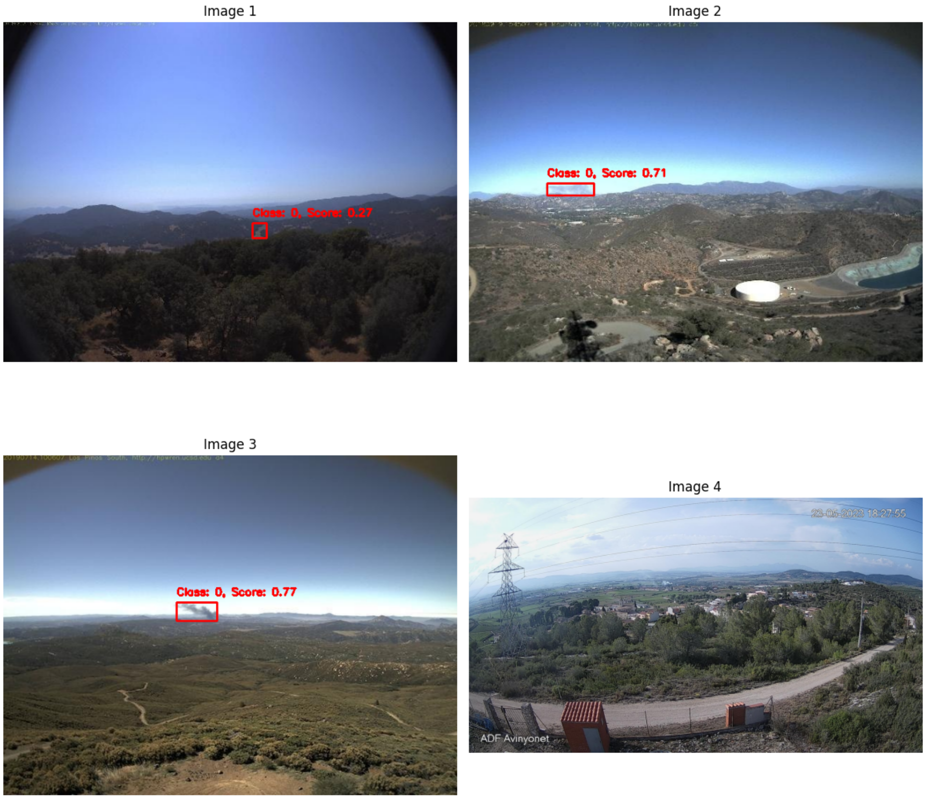

Figure 1: YOLOv8 Fire and Smoke Detection: Model Predictions with Different Scores

- **Image 1**: The model assigned a score of 0.27 for classifying smoke in this image. However, this is a poor detection as the model encircles a region where there is no smoke visible. This indicates a false positive result.

- **Image 2**: In this image, the model assigned a score of 0.71 for detecting smoke (Class 0). The prediction seems correct, as there is a visible presence of smoke in the outlined region, which means the model performed well here.

- **Image 3**: Here, the model predicted smoke with a score of 0.77. The detected area indeed contains smoke, so the model correctly identified smoke in this scenario with a high confidence score.

- **Image 4**: This image shows a landscape without any smoke or fire. The model correctly did not predict any smoke or fire, thus demonstrating a correct negative prediction.

One Month Plan

Given more time and resources, the following enhancements could significantly improve model performance:

- **MLFlow Integration**: **MLFlow** would be implemented to track experiments and hyperparameter tuning systematically. This tool would allow comparison between different models (YOLOv5, YOLOv8, Faster R-CNN) and help in identifying the most effective configuration.
- **Increased Epochs**: Due to the time limitations and Colab's restricted resources, the model was trained with fewer epochs than desired. Extending the training time would likely improve the model's accuracy, particularly in detecting smaller or more obscured fires.
- **Enhanced Data Augmentation**: More sophisticated augmentation techniques, such as random cropping, brightness/contrast adjustments, and adding noise, would help the model become more robust across varying lighting conditions, smoke presence, and fire sizes.

Conclusion

In conclusion, YOLOv8 was chosen for its balance of speed, accuracy, and efficiency given the constraints of the Colab environment. With additional time and resources, improvements such as MLFlow integration and extended training would further enhance model performance, making it more suitable for real-world fire detection tasks.