Documentation Details

Model File

https://drive.google.com/file/d/1hW60wTFypT5EOQlN8RULVkuur7HUIetW/view?usp=sharing

Prediction Documentation

- Prediction Script: *PredictionScript*.
- Prediction Script Details: *Prediction Code Components*.

Deployment Documentation

- Deployment Script: *DeploymentScript.py*.
- Deployment Screenshot: *Deployment Code*
- Deployment Script Explanation: *Deployment Code Components*.

Training and Validation Overview

- Training & Validation Script: TrainingScript
- Accuracy Plot: Training-set-run
- Training Script Explanation: Explanation of the Training and Validation Code Components

Forest Fire Detection Model Report

1. **Design Choices Overview**

The model is based on a Convolutional Neural Network (CNN) architecture, which consists of three convolutional layers followed by corresponding max-pooling layers. After these, a flattening layer is applied to reduce the dimensionality of the feature maps. The network then passes through two fully connected (dense) layers, with the final dense layer employing the softmax activation function for classification into three distinct categories: fire, no fire, and uncertain. To enhance the model's generalization capabilities and mitigate overfitting, data augmentation techniques were applied during training. These techniques included random rotations, shifts in width and height, shear transformations, zooming, and horizontal flipping of the input images.

2. **Performance Evaluation and Results**

The model underwent training for a total of 7 epochs with a batch size of 32. To ensure reliable evaluation, the training process included not only the primary training dataset but also a validation set, which helped monitor the model's performance on unseen data during each epoch. The model achieved an impressive training accuracy exceeding 90%, though the validation accuracy was slightly lower, suggesting a degree of overfitting. When tested on a separate test set, the model's accuracy was around 85%, indicating that, while the model performs well, there is still room for improvement in terms of generalizing to entirely new data.

3. **Future Work and Enhancements**

Looking ahead, several improvements can be explored to enhance the model's performance. One potential avenue is experimenting with deeper neural network architectures, such as ResNet or Inception, which are known for their advanced feature extraction capabilities. Additionally, further fine-tuning of data augmentation strategies—perhaps by adjusting the magnitude of the transformations—could help improve model robustness. Incorporating a larger, more diverse dataset would also be beneficial, as more varied data could help the model generalize better. Finally, performing a more comprehensive hyperparameter search using techniques like grid search or random search could lead to more optimal settings, potentially increasing the model’s accuracy and overall robustness.