Forest Fire Detection using Deep Learning

AICTE Student ID: STU65637bb9deec51701018553

Learning Objectives

- Understand how **Convolutional Neural Networks (CNNs)** work for image classification.
- Learn how to **load and preprocess image datasets**.
- Build and train a deep learning model using **Keras and TensorFlow**.
- Evaluate model performance using accuracy and loss graphs.
- Deploy the model to **predict fire presence in new images**.

Source : www.freepik.com/

Tools and Technology used

- **Python:** Programming language
- **TensorFlow / Keras:** Deep Learning framework
- **Matplotlib:** For data visualization
- **NumPy & OS:** For data handling and manipulation
- **Kaggle Hub:** To download and manage the dataset
- **Google Colab / Kaggle Notebook:** Training environment
- **GPU Acceleration:** For faster model training
- **Data set link:** https://www.kaggle.com/datasets/elmadafri/the-wildfire-dataset
- **Git hub repository link:** https://github.com/UdayKiran-05/Forest_Fire_detection.git

Methodology

Step-by-step Process:

- **Data Loading & Visualization:** Explore images from both classes
- **Data Preprocessing:** Normalize pixel values using ImageDataGenerator
- **CNN Model Building:**
  - 3 Convolution layers
  - 3 Max Pooling layers
  - Flatten + Dense layers with Dropout
- **Model Compilation:** Using adam optimizer and binary_crossentropy
- **Model Training:** For 12 epochs with training and validation data
- **Performance Evaluation:** Accuracy and loss graphs
- **Prediction Function:** To test on new unseen images

Problem Statement:

- **Forest fires are one of the most destructive natural disasters, causing significant ecological, economic, and social damage**. Detecting these fires early is critical to reducing their impact. However, manual monitoring methods are often slow, resource-intensive, and not scalable over large forest areas. In this project, we address this issue by building a **deep learning-based solution** using **Convolutional Neural Networks (CNNs)** to automatically detect the presence of fire in images. Leveraging **Python and TensorFlow**, we used the **"Wildfire Dataset"** from Kaggle, which contains labeled images categorized as either **'fire' or 'nofire'**. The images are preprocessed using image generators with rescaling, then passed through a CNN model designed with multiple convolution and pooling layers followed by dense layers for classification. The model is trained, validated, and evaluated using appropriate performance metrics, and its accuracy is visualized using training and validation curves. **Finally, the trained model is used to make predictions on new, unseen images, offering a practical and scalable approach to assist in real-time forest fire detection and environmental safety.**

Solution:

- ✓ **Deep Learning Approach**
  Used a Convolutional Neural Network (CNN) to classify forest images into fire and no fire categories.

- ✓ **Dataset Used**
  Utilized the Wildfire Dataset from Kaggle, with separate training, validation, and test folders.

- ✓ **Preprocessing**
  Images resized to 150x150 pixels and normalized using ImageDataGenerator for better model performance.

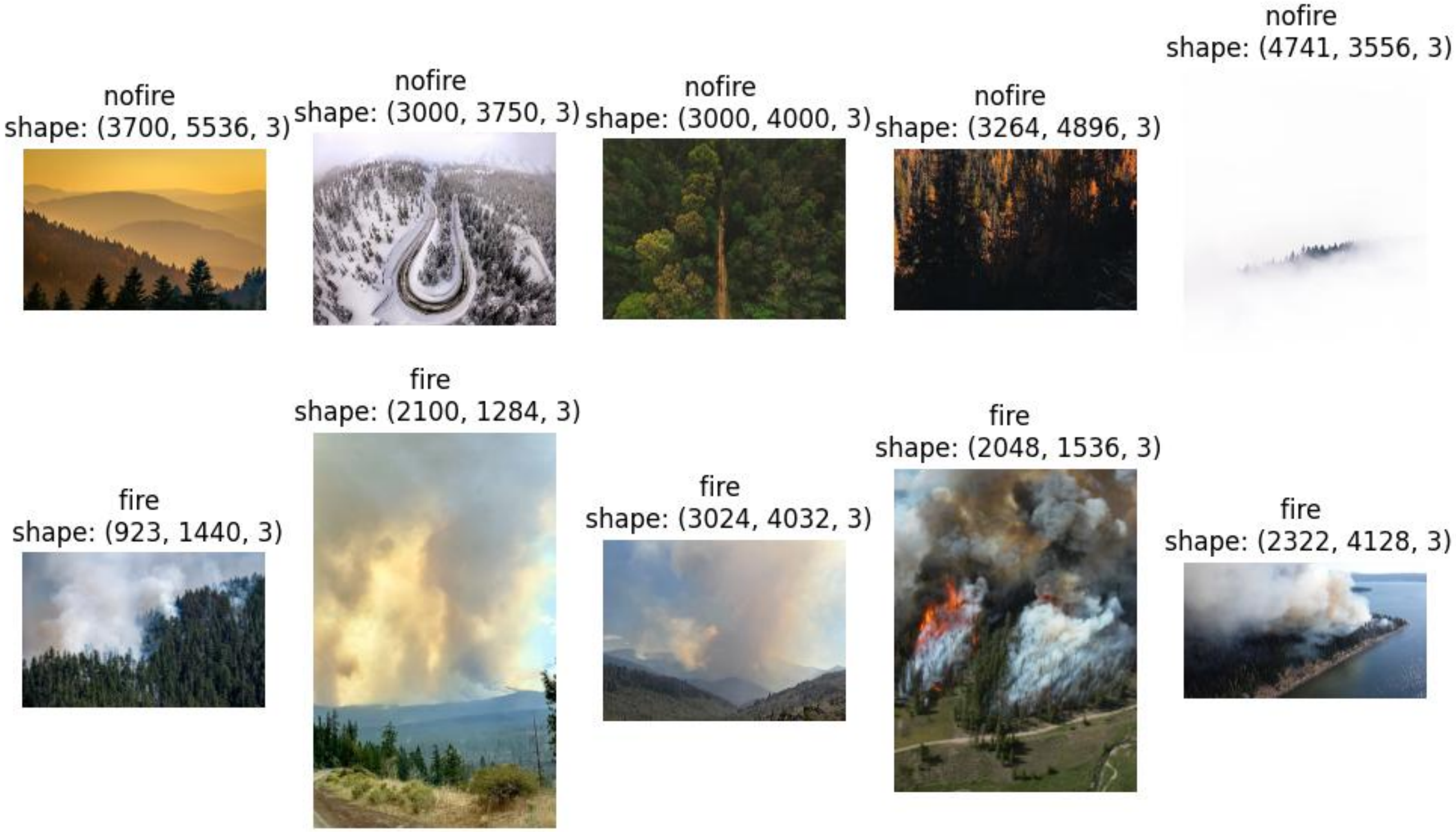
- ✓ **Model Architecture**
  Built a CNN with multiple Conv2D, MaxPooling, Dense, and Dropout layers using Keras Sequential API.

- ✓ **Model Training**
  Trained the model for 12 epochs with real-time accuracy and loss monitoring on training and validation data.

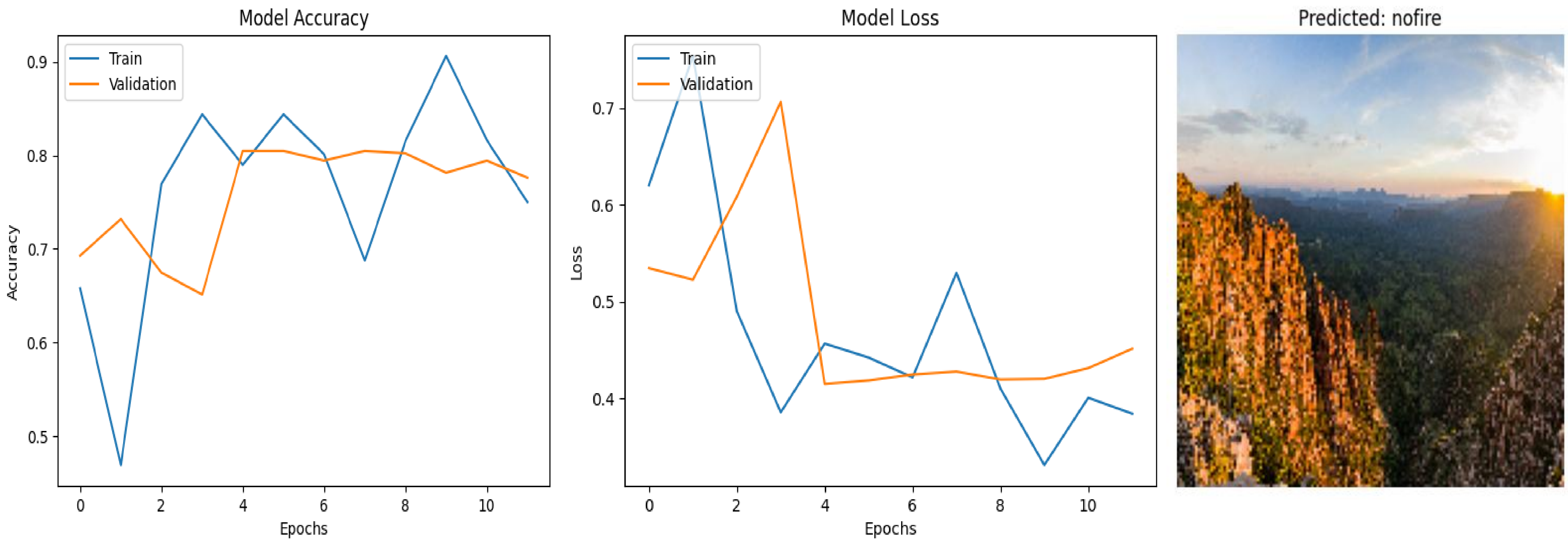
- ✓ **Prediction and Evaluation**
  Developed a custom function to predict fire in new images and evaluated the model using test data accuracy.

- ✓ **Scalable Application**
  The trained model can be integrated into real-time systems for early forest fire detection and alerts.

Screenshot of Output:



Screenshot of Output:



Conclusion:

From this We conclude that:

- ✓ Developed and trained a CNN in TensorFlow/Keras that accurately distinguishes fire from non-fire images.
- ✓ Employed **ImageDataGenerator** for effective image normalization and managed training, validation, and test splits.
- ✓ Monitored accuracy and loss curves over 12 epochs to confirm consistent model convergence and reliability.
- ✓ Implemented a **predict_fire()** function for real-time inference, ready for integration into live monitoring systems.
- ✓ Delivered a scalable, automated early-warning solution—with paths for future enhancement via larger datasets, transfer learning, and edge-device deployment.

Future Scope:

- ➢ Improve accuracy by adding more training data
- ➢ Integrate model with real-time video streams or drone footage
- ➢ Deploy the model on edge devices (e.g., Raspberry Pi with camera)
- ➢ Add multiclass classification (e.g., smoke, small fire, large fire)
- ➢ Explore transfer learning using pre-trained models like ResNet or MobileNet