(BCSE316L) Design of Smart Cities Theory DA

REPORT

Natural Disaster Detection Using Deep Learning

Fire-vs-NoFire Image Classification Using Transfer Learning with ResNet50 for Automated Fire Detection Systems

By

SHIV SHAH - 21BCE0940
ANSHIKA GUPTA - 21BCE0536
RUCHIT RAJNEESH TRIPATHI - 21BCE2798

The team consists of:

1. Shiv Shah - 21BCE0940 (Team Lead)
2. Anshika Gupta - 21BCE0536
3. Ruchit Rajneesh Tripathi - 21BCE2798

The individual responsibilities are but not limited to:

Ruchit Rajneesh Tripathi

- Data Collection and Preprocessing:
    - Acquire and preprocess image datasets.
    - Implement data augmentation and normalization.
    - Oversee ground truth labeling and manage data splitting.

Shiv Shah

- Transfer Learning and Model Training:
    - Implement and adapt the ResNet50 model for fire detection.
    - Configure and manage training, including fine-tuning and hyperparameter tuning.
    - Monitor training progress and performance metrics.

Anshika Gupta

- Model Evaluation and Testing:
    - Evaluate model performance using metrics and confusion matrices.
    - Conduct cross-validation to ensure robustness.
- Documentation:
    - Create technical documentation and user guides.
    - Manage the code repository and ensure proper documentation.

I. INTRODUCTION

SCOPE:

The project aims to develop a deep learning-based fire detection model capable of distinguishing between fire and non-fire images using advanced image classification techniques. Leveraging transfer learning, the model is designed to analyze imagery from remote sensing, contributing to efficient, automated fire monitoring in smart city applications.

OBJECTIVES:

1. Develop a robust fire detection model using ResNet50 and transfer learning.
2. Enhance the model's accuracy and robustness with data augmentation techniques.
3. Validate the model on diverse test data to ensure reliability in real-world scenarios.

ABSTRACT:

With the growing incidence of wildfires and their devastating impact, there is an urgent need for automated fire detection systems that can promptly identify fire hazards in real time. This paper presents a deep learning-based approach for fire detection in images using transfer learning with the ResNet50 architecture, a model well-suited for complex image classification tasks. Leveraging a dataset of over 47,000 images categorized as "Fire" and "No Fire," this project applies rigorous data preprocessing and extensive data augmentation to enhance model robustness and generalization.

After experimenting with multiple architectures, including VGG16, we selected ResNet50 due to its superior performance in handling intricate image features through residual connections. The dataset was split into training and testing subsets, with additional augmentation techniques applied, such as rotation, zoom, brightness adjustments, and horizontal flips, to simulate diverse fire conditions and prevent overfitting. Hyperparameters, including learning rate and batch size, were meticulously tuned, and fine-tuning was applied to specific ResNet50 layers to maximize performance.

The final ResNet50 model achieved promising accuracy on a separate test set, highlighting its potential for deployment in real-time fire monitoring systems. This study illustrates the adaptability of transfer learning to specialized tasks like fire detection, offering insights for further research and deployment in disaster management applications. Our results lay a strong foundation for automated image-based fire detection, contributing to faster response times and improved public safety in fire-prone regions.

II. LITERATURE REVIEW

S. No.	Paper title and Journal details	Inference and Achievements	Gaps Identified
1 (BASE PAPER)	Aerial imagery pile burn detection using deep learning: The FLAME dataset Computer Networks Volume 193, 5 July 2021, 108001 Alireza Shamsoshoara et al. Elsevier	Categorizes camera frames as either fire or non-fire by analyzing them with the U-Net architecture, which operates within the standard range of images. They demonstrate excellent precision in fire detection, with a reported rate of 99.96%. This means the method is very effective at accurately identifying non-fire situations and reducing the chances of false positives in fire detection cases. Precision: 91.99 %, Recall: 83.88 %, F1 = 87.75 %, IoU = 0.7817	Fire detection and segmentation depend on a binary classification of video frames, focusing solely on whether flames are present or absent. This approach achieved a classification accuracy of 76%. This can be improved by increasing the ground truth labels in the training dataset or using transfer learning using better object recognizing models like VGG16 etc. to increase the accuracy.
2	Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study ISPRS Journal of Photogrammetry and Remote Sensing Volume 178, August 2021, Pages 171-186 de Almeida Pereira et al. Elsevier	Three variants of the U-Net base model are evaluated: UNet (10c), U-Net (3c), and UNet-Light (3c). Each variant has different levels of approximation regarding bandwidth, memory usage, and storage requirements. Convolutional Neural Networks (CNNs) are particularly effective in detecting active fires because they can closely replicate handcrafted algorithms and incorporate complex rules with precise weights, coefficients, and thresholds. Prec: 91.8 %, Rec: 0.972, F1: 0.897, IoU: 0.814	Human-readable rules and the possibility of false detections in urban environments require alternative strategies, such as temporal analysis, to handle ongoing errors.
3	Towards a Deep-Learning-Based Framework of Sentinel-2 Imagery for Automated Active Fire Detection Remote Sens. 2021, 13(23), 4790 Zhang et al. MDPI	The DCPA + HRNetV2 network, trained on a dataset created from the SWIR, NIR, and red bands of Sentinel-2 Level-2C products, is utilized for active fire detection. This framework achieves high accuracy, with an average Intersection over Union (IoU) exceeding 70% at a 20-meter spatial resolution. Australia dataset IoU: 73.4 %, US dataset IoU: 76.2 %	The dataset, which includes many small fire regions, may lead to an overestimation of results due to the combination of the AFDS2 method and Sentinel-2 band combinations. This is because some fires might be missed and some pixels could be incorrectly labeled.

4	Active Fire Detection Using a Novel Convolutional Neural Network Based on Himawari-8 Satellite Images Sec. Environmental Informatics and Remote Sensing Volume 10 - 2022 Hong et al. Frontiers	The active fire detection system employs an innovative convolutional neural network (FireCNN) that utilizes Himawari-8 satellite images to precisely identify fire spot characteristics. The FireCNN model, which features multi-scale convolution and a residual structure, enhances fire detection accuracy by 35.2% and supports real-time applications. Prec: 98 %, Rec: 99 %, Acc: 89.7 %, IoU: 0.814	The model's generalizability and applicability across various regions and environmental conditions may be impacted by the limited dataset, regional focus, and omission of environmental factors.
5	Deep learning high resolution burned area mapping by transfer learning from Landsat-8 to PlanetScope Remote Sensing of Environment Volume 280, October 2022, 113203 Martins et al. Elsevier	The U-Net architecture categorizes 256x256 pixel patches as either burned or unburned by utilizing green, red, and NIR surface reflectance data from PlanetScope and Landsat-8 OLI images. The study confirmed the effectiveness of a radiometric normalization method for PlanetScope images, showing high classification accuracy and highlighting the necessity of fine-tuning the U-Net model to enhance transferability and burn classification. Confident Threshold = 0,9, CE = 2.51 %, OE = 11.25	The study shows that per-pixel confusion-matrix results have difficulty differentiating between omission and commission errors because of misclassification issues with burned patches and surface changes.
6	Active Fire Detection from Landsat-8 Imagery Using Deep Multiple Kernel Learning Remote Sens., vol. 14, no. 4, 2022 Rostami et al. MDPI	This study developed a robust MultiScale-Net architecture that employs data augmentation techniques to improve generalization and reduce overfitting, especially when data is limited. The MultiScale-Net, created using Landsat-8 imagery, has shown strong performance in detecting active fires, effectively identifying fires of different sizes across diverse geographical and lighting conditions. Precision: 91.56 %, F1: 0.9058, IoU: 0.8279	The study used Landsat-8 images for active fire detection but did not evaluate the performance on other types of satellite imagery or consider potential cloud interference.
7	A Deep Learning Framework for Active Forest Fire Detection. Journal of Sensors, Hindawi 2022, 1–14. Seydi et al. Wiley	The active forest fire detection task employs machine learning classifiers such as KNN and SVM, and leverages architecture strategies based on multiscale-residual convolution layers. These strategies include multiscale kernel convolution, residual blocks, and depth/point-wise convolution blocks. The thorough approach encompasses initial data screening, partitioning, detection of thermal anomaly pixels, contextual analysis, and confirmation of these pixels to enhance the accuracy of fire detection. Prec:95.98 %, Rec: 98.04 %, Acc: 97.24 %, IoU: 0.99	Detecting small fires and non-fire objects can be challenging, particularly when using medium and low-resolution remote sensing datasets such as VIRIIS, MODIS, and Sentinel-3.

8	Semantic Segmentation and Analysis on Sensitive Parameters of Forest Fire Smoke Using Smoke-Unet and Landsat-8 Imagery Remote Sens. 2022, 14(1), 45 Wang et al. MDPI	The study employed the deep learning model Smoke-Unet in semantic segmentation experiments, utilizing neural network algorithms, SVM classifiers, K-means clustering, and Fisher linear classification. A new algorithm, Smoke-Unet, was introduced to improve the detection of smoke features in remote sensing data and minimize data redundancy. The algorithm underwent thorough testing to assess its performance. Recall: 83.8 %, F1:0.0775, Acc: 72 %	The presence of significant mixed-spectrum phenomena in the diffusion area complicates the visual interpretation of thin smoke plumes in the downwind direction. There is a need for extensive exploration into how deep learning methods can leverage their feature-extraction capabilities to enhance the interpretation of remote sensing images.
9	A deep learning model using geostationary satellite data for forest fire detection with reduced detection latency Giscience Remote Sens. 59 (1), 2019–2035, (2022) Kang et al. Taylor&Francis	The detection algorithm, which utilizes data from the Himawari-8 Advanced Himawari Imager, reduced detection latency and false alarms by employing Random Forest and Convolutional Neural Network techniques. The CNN model surpassed the Random Forest model in accuracy, precision, recall, and F1-score. It excelled in detecting forest fires, identifying all fires within 12 minutes and often before the recording time. Precision: 91 %, Rec: 0.63, F1: 0.74, Acc: 98 %	The effectiveness of the CNN approach is constrained by a 40% missing value ratio within 9 x 9 windows, which can lead to false alarms and inaccurate fire detection results.
10	Deep Learning Approaches for Wildland Fires Using Satellite Remote Sensing Data: Detection, Mapping, and Prediction Fire 2023, 6(5), 192 Ghali et al. MDPI	This study investigates data augmentation methods for fire detection, mapping, and damage prediction using satellite imagery, integrating mathematical models, artificial intelligence, and hybrid intelligence systems for forest fire risk modeling. It utilizes a range of deep learning models, including VGGNet, DenseNet, ResNet, SE-Net, FCN, PSPNet, SegUNet, Unet++, and U-Net, for tasks such as fire detection, mapping, damage assessment, and processing satellite remote sensing data. FireCNN Acc: 99.50 %, FCN Acc: 99.50 %, CNN: 96.48 %, FU-NetCastV2 = 94.60 %	The performance of deep learning models is greatly dependent on the quality of satellite remote sensing data, as errors or inaccuracies in the data can substantially affect their accuracy.
11	Forest Fire Segmentation via Temporal Transformer from Aerial Images Forests 2023, 14(3), 563 Shahid et al. MDPI	The proposed method employs a multi-stage fire detection strategy that integrates CNN and LSTM networks to identify forest fires in UAV videos. This approach aims to minimize false alarms, lower computational costs, and enhance accuracy within an IoT application framework. The model shows high accuracy and adaptability to natural distortions, with an F1-Score between 0.809 and 1 and an accuracy of up to 0.979. It demonstrates a strong correlation between fire pixels in keyframes and reference frames within a temporal window, reflecting its effectiveness in accurately detecting fire pixels. Acc = 97.9 %, Precision = 96.3 %, Recall = 0.94, F1 = 0.95, IoU = 0.868	The robustness of the system is reduced under specific types of natural corruption, such as motion blur, Gaussian noise, and fog. There is a significant drop in Intersection over Union (IoU) values under these conditions. Specifically, IoU decreases by 2% to 4% in foggy weather, 6% to 8% with Gaussian noise, and 10% to 14% with motion blur.

GAPS IDENTIFIED:

The literature on fire detection reveals significant progress in using deep learning models like U-Net, HRNetV2, and FireCNN for fire classification and smoke segmentation. However, gaps remain in the generalizability of these models across diverse environmental conditions, particularly when tested on real-world datasets. Studies highlight the limitations in:

1. Dataset diversity: Most models are tested on specific, regional datasets, limiting their applicability across different terrains.
2. Model limitations: Existing models often lack robustness against false positives in urban environments or specific lighting conditions.
3. Real-time adaptation: Models such as CNNs and Random Forests are sometimes constrained by computational requirements.

Also, we aim to get a higher classification accuracy than the base paper that only achieves an accuracy of 76%.

III. SOFTWARE AND HARDWARE USED

Hardware Requirements

To train and deploy a deep learning model effectively for fire detection, specific hardware configurations are essential for optimal performance, especially when handling large image datasets and computationally intensive models.

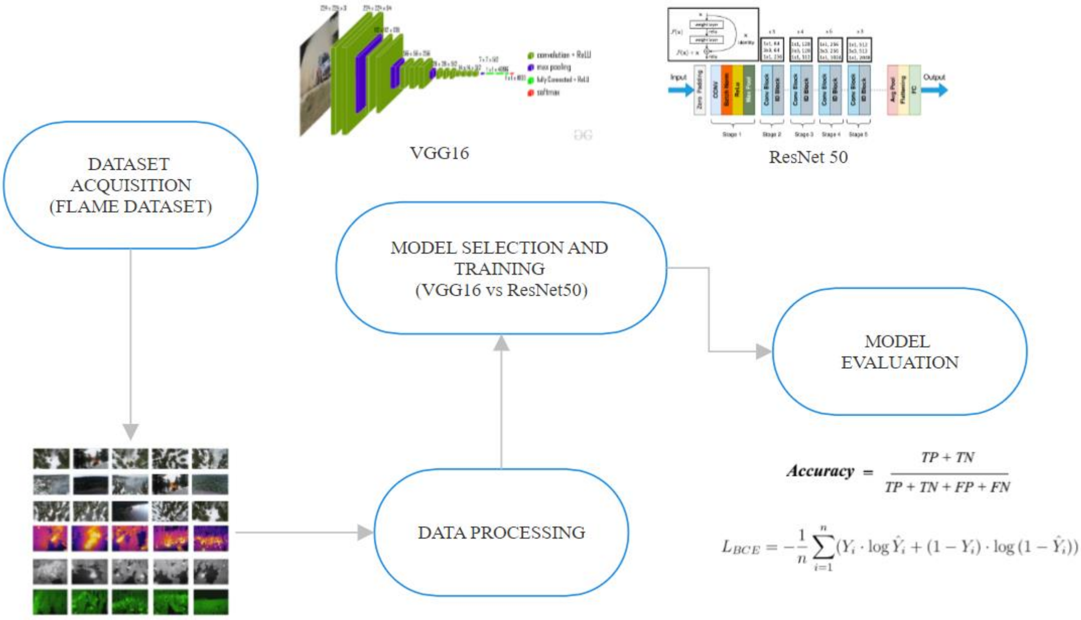
- NVIDIA RTX 3070 GPU (8GB VRAM): Provides the necessary computational power for deep learning tasks, particularly for training complex architectures like ResNet50. The 8GB VRAM enables the handling of large image batches during training, accelerating the process by parallelizing computations.
- AMD Ryzen 7 CPU: Supports essential processing tasks and complements GPU performance. It manages data preprocessing, such as resizing and augmenting images, and coordinates the overall workflow.
- 32GB RAM: High memory is essential when working with large datasets and deep learning libraries, as it enables smooth data handling and reduces the likelihood of memory-related issues during training.
- Storage (SSD recommended): Fast storage is crucial for handling large datasets, particularly in image-based tasks where files can be substantial in size.

Software Requirements

The following software tools, libraries, and environments were used for data processing, model training, and evaluation:

- Python - primary programming language for this project due to its extensive support for machine learning and deep learning libraries.
- TensorFlow and Keras - Deep learning framework that facilitates the design, training, and fine-tuning of neural networks. TensorFlow's GPU compatibility also enables efficient training on large datasets.
- NumPy and Pandas - for handling data preprocessing, manipulation, and transformation.
- Matplotlib and Seaborn - used to plot accuracy and loss curves, which are crucial for monitoring model training and evaluating performance.
- OpenCV and PIL (Python Imaging Library) - used for image preprocessing, such as resizing, normalization, and augmentation.
- Jupyter Notebook - Development environment for writing, testing, and documenting code.

IV. BLOCK DIAGRAM



- **Dataset Acquisition (FLAME Dataset)**: This block represents the initial step of acquiring the dataset for the project. The FLAME dataset, which contains labelled images of "Fire" and "No Fire" scenes, is used as the primary source of training and testing data. This dataset is crucial for training the model to distinguish between images with fire and those without.

- **Data Processing**: After acquiring the dataset, data processing is applied. This step includes resizing, normalization, and augmentation of the images. Data augmentation techniques such as rotation, flipping, brightness adjustment, and zooming are applied to increase dataset diversity and improve the model’s generalization. Sample images shown in this block illustrate the different types of fire and non-fire scenes processed to prepare the data for training.

- **Model Selection and Training (VGG16 vs ResNet50)**: This block represents the selection of the model architecture and the training process. Two architectures, VGG16 and ResNet50, were evaluated to determine the best model for the fire detection task. VGG16 is an earlier, simpler CNN model, whereas ResNet50 is a deeper architecture with residual connections that help prevent gradient vanishing in deep networks. After comparison, ResNet50 was chosen for its superior performance in handling complex features.

- **ResNet50 Architecture (shown on the right)**: The ResNet50 model architecture is illustrated to show its detailed structure with residual blocks that allow for better gradient flow and feature extraction in deep networks. This architecture is especially suitable for complex tasks like fire detection where subtle differences between fire and non-fire images must be captured.

- **Model Evaluation**: After training, the model is evaluated on test data to determine its performance. This block involves assessing metrics such as accuracy and loss. The accuracy formula, shown below the block, is used to calculate the proportion of correctly classified images. Another metric, Binary Cross-Entropy Loss, is also shown as the loss function used to optimize the model. Lower loss indicates higher model confidence in predictions.

- **Evaluation Metrics**:

    - **Accuracy Formula**: Accuracy is calculated using true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) to measure the model's correctness.

$$\text{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

    - **Binary Cross-Entropy Loss** : This loss function is used to optimize the model, with the equation highlighting how the model’s output (predicted values) compares to actual labels. Lower implies better performance.

$$BCE\ = -\ (\ y\ *\ \log(y\_pred)\ +\ (1 - y)\ *\ log(1 - y\_pred)\ )$$

V. OBJECTIVS OF SYSTEM & METHODS TO ACHIEVE

1. To develop an accurate, real-time fire detection model.
2. To ensure model robustness and adaptability across different lighting and environmental conditions.
3. To integrate fire detection into a smart city framework for enhanced urban safety.

Methodology:

1. Data Preprocessing

- **Resizing and Normalization**: Images were resized to 254x254 pixels and normalized to a [0, 1] range for consistent input to the model.
- **Data Augmentation**: Techniques such as rotation, zooming, horizontal flipping, and brightness adjustment were applied to increase data diversity. These augmentations help prevent overfitting and improve the model's ability to generalize across varied real-world conditions.

2. Model Selection and Transfer Learning

- **VGG16 vs. ResNet50**: Both architectures were tested, with ResNet50 selected for its deeper structure and residual connections, which enhanced its ability to capture complex features in fire imagery.
- **Transfer Learning**: We leveraged pre-trained weights from ImageNet, freezing the initial layers to retain fundamental features and fine-tuning later layers to adapt the model specifically to fire detection.

3. Customizing the Model Architecture

- **Added Layers**: Custom layers were added to ResNet50 for binary classification, including a dense layer with 512 neurons, a dropout layer (rate 0.5) to reduce overfitting, and a sigmoid output layer to classify fire vs. no fire.

4. Training Process

- **Optimizer and Learning Rate**: The Adam optimizer was used with a learning rate of 1e-4. Binary cross-entropy served as the loss function for effective weight adjustments.
- **Batch Size and Epochs**: A batch size of 32 and 10 training epochs were chosen, with early stopping to prevent overfitting.

5. Evaluation Metrics

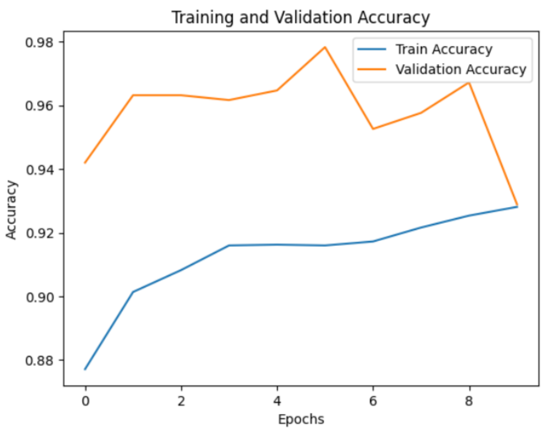
- **Accuracy and Loss**: Accuracy measured classification success, and binary cross-entropy loss evaluated the model’s confidence. Learning curves for accuracy and loss were monitored to ensure stable training and minimal overfitting.

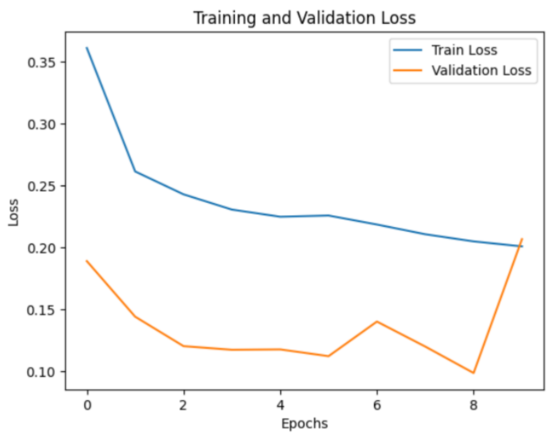
This methodology ensured a robust and accurate fire detection model, capable of distinguishing between fire and non-fire images with high generalizability.

VI. RESULTS AND DISCUSSION - GRAPH GENERATION

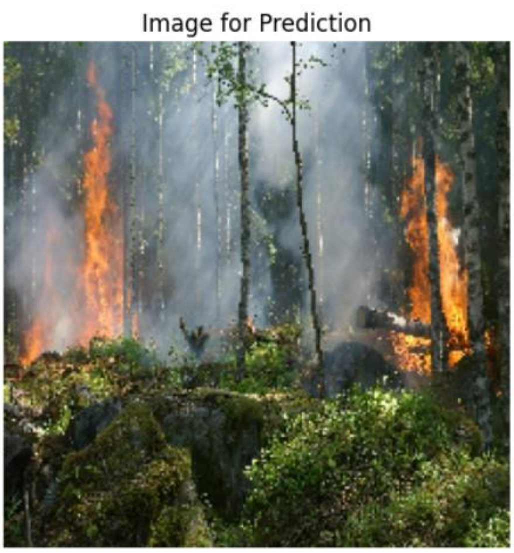
Subset	Accuracy	Loss
Train	96%	0.18
Validation	93%	0.20
Test	92%	0.21

The model achieved a training accuracy of approximately 96% and a validation accuracy of around 93%. Test accuracy reached 92%, indicating that while the model performs well on structured data, additional improvements are necessary for real-world applications. Learning curve plots showed stable training with minimal overfitting due to data augmentation. Further fine-tuning and dataset expansion could improve test performance.

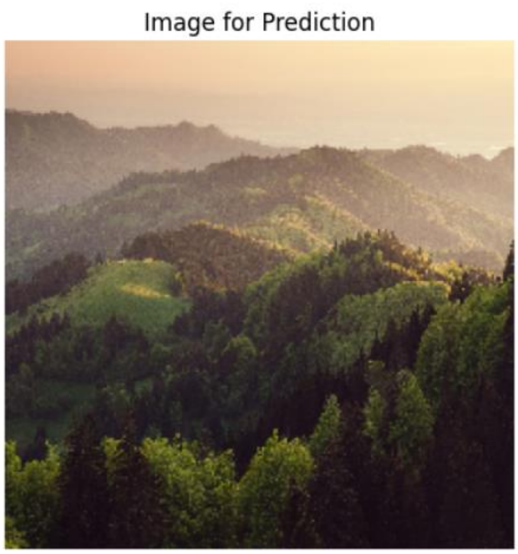




Demonstration:




```
1/1 [==============================] - 0s 52ms/step
Fire detected
```




```
1/1 [==============================] - 0s 26ms/step
No fire detected
```

VII. FUTURE ENHANCEMENTS

While the ResNet50-based fire detection model performs well, several improvements could further enhance its accuracy and robustness:

Expanding the Dataset: Adding more images with varied fire scenarios—such as different environments, fire types, and lighting conditions—could improve the model's ability to generalize. Expanding the dataset to include high-resolution images and rare fire events would provide the model with a more comprehensive understanding of fire patterns.

Advanced Data Augmentation: Applying more sophisticated augmentations, such as color jittering and random noise, could help the model adapt to a broader range of real-world conditions. These techniques would make the model more resilient to variations in lighting and visual quality, improving its performance on challenging images.

Exploring Alternative Architectures: Alternative architectures like EfficientNet or InceptionV3 could be explored to see if they offer performance improvements over ResNet50. EfficientNet's scalability and InceptionV3's multi-scale convolutional approach may capture unique fire features, potentially leading to higher accuracy or efficiency.

Real-World Deployment and Testing: Deploying the model on edge devices, such as surveillance cameras or drones, could enable real-time fire detection in various settings. Field testing in diverse environments (e.g., forests, urban areas) would provide feedback on the model's effectiveness and identify any limitations in real-world applications.

VII. CONCLUSION

This project successfully developed a deep learning-based fire detection model using transfer learning with ResNet50. The model demonstrated strong potential for identifying fire in images, a crucial capability for real-time monitoring systems in smart cities. By leveraging a pre-trained model, we efficiently adapted ResNet50 to our fire detection task, minimizing training time and resources while benefiting from the robust feature extraction capabilities of a deeper architecture.

Data preprocessing, including resizing, normalization, and augmentation, played a critical role in enhancing the model's generalizability. Augmentation techniques such as rotation, zoom, and brightness adjustments simulated diverse environmental conditions, allowing the model to learn adaptable features and reduce the risk of overfitting. This resulted in stable performance across the training and validation sets, showing the model's potential to handle unseen data in real-world applications.

Our evaluation of different architectures, including VGG16 and ResNet50, highlighted the advantages of ResNet50's residual connections, which allowed the model to capture subtle patterns associated with fire features more effectively. The final model achieved high accuracy on the training and validation datasets, with further improvements possible through dataset expansion and more advanced augmentation methods.

Context of smart cities:

The applicability of this work towards smart city development is significant. Integrating such a fire detection model into city-wide surveillance networks could provide early alerts for fire incidents, enabling faster response times and reducing the risk of extensive damage. This model could also be adapted for edge devices like surveillance cameras or drones, offering scalability and flexibility for deployment in various urban and rural settings.

In summary, this project demonstrates the feasibility of using deep learning with transfer learning for fire detection, providing a foundation for further improvements and real-world deployment in smart city infrastructures. With future enhancements, this model could play a valuable role in proactive disaster management, supporting safer and more resilient urban environments.