Comparative Analysis of Deep Learning Models for Forest Fire Classification Using Wildfire Dataset

Prabhas (1RVU22BSC069), Rachana (1RVU22BSC078), Adishree K. (1RVU22BSC001)

Abstract

This paper presents a comparative analysis of four deep learning models—Custom CNN, ResNet-18, MobileNetV2, and EfficientNetB0—for binary classification of forest fire images. The study evaluates model performance based on accuracy, computational efficiency (FLOPs), and inference time. All models were trained on a wildfire dataset containing labeled images of 'fire' and 'no fire' scenarios. The Custom CNN achieved 95.67% test accuracy with 557M FLOPs, while pretrained models demonstrated superior performance (ResNet-18: 98.75%, MobileNetV2: 97.50%, EfficientNetB0: 100%). EfficientNetB0 emerged as the most accurate but required 400M FLOPs, whereas MobileNetV2 offered the best trade-off with 313M FLOPs and 97.5% accuracy. The results highlight the potential of transfer learning for environmental monitoring applications.

Index Terms—Forest fire detection, deep learning, convolutional neural networks, transfer learning, image classification.

I. Introduction

Wildfire detection using computer vision has gained significance for early warning systems. This study investigates:

- Performance of custom vs. pretrained architectures
- Computational efficiency metrics for edge deployment
- Optimal model selection for fire classification

II. Methodology

Datasets used:

- FIRE Dataset (755 fire, 244 non-fire images): https://www.kaggle.com/datasets/phylake1337/fire-dataset
- Wildfire Detection Data (950 fire, 950 non-fire images): https://www.kaggle.com/datasets/brsdincer/wildfire-detection-image-data

A. Dataset Preparation:

The dataset consists of 3,200 images classified as either "fire" or "no fire." Images were collected from both aerial drones and ground-level cameras. They were resized to 224×224 and normalized using standard ImageNet values. The split was 80% train, 10% validation, and 10% test.

B. Model Architectures:

- Custom CNN: 3 convolutional layers and 2 fully connected layers (~51.4M parameters)
- ResNet-18: Modified final layer
- MobileNetV2: Frozen feature extractor

- EfficientNetB0: Full fine-tuning

C. Training Protocol:
- Optimizer: Adam (lr=0.001)
- Loss: Cross-entropy
- Epochs: 10
- Batch Size: 32
- Hardware: Google Colab GPU (T4)

Figure 1. Comparative analysis of MobileNetV2, ResNet-18, and Custom CNN on key performance indicators.

Model	Total Parameters	FLOPs	Inference Time (sec)	Model Size (MB)
Custom CNN	51,475,010	557,155,328	0.0033	245.18
ResNet18	11,177,538	1,818,554,880	0.0030	106.00
MobileNetV2	2,226,434	312,915,776	0.0056	161.93

Figure 2. Comparative table showing metrics for Custom CNN and ResNet-18.

Model Name	# Parameters	FLOPs	Inference Time/Image	Model Size (MB)
Custom CNN	51,475,010	557,155,328	0.0033 sec	245.18 MB
ResNet18	11,177,538	1,818,554,880	0.0030 sec	106.00 MB

Figure 3. Summary of EfficientNetB0 model performance and architecture details.

Here's the table for the **EfficientNetB0** model metrics:

Metric	EfficientNetB0
Total Params	4,010,110
Trainable Params	4,010,110
Non-trainable Params	0
Input Size (MB)	0.57
Forward/Backward Pass Size (MB)	173.64
Params Size (MB)	15.30
Estimated Total Size (MB)	189.52
FLOPs	400,381,952
Average Inference Time per Image	0.0135 seconds

Figure 4. Summary of Custom CNN model performance and architecture details.

🔥 Custom CNN Model (Wildfire Dataset)

- Total Parameters: `51,475,010`
- Trainable Parameters: `51,475,010`
- Non-Trainable Parameters: `0`
- Model Size: ~ `245.18 MB`
- FLOPs: `557,155,328`
- Average Inference Time per Image: `0.0033 seconds`
- Final Validation Accuracy: `97.5%`

III. Results

A. Performance Metrics:

Model	Test Accuracy	FLOPs	Inference Time (ms)	Parameters
Custom CNN	95.67%	557M	3.3	51.4M
ResNet-18	98.75%	1818M	3.0	11.1M
MobileNetV2	97.50%	313M	5.6	2.2M (2.5K trainable)
EfficientNetB0	100.00%	400M	13.5	4.0M

B. Training Curves:
All models converged within 10 epochs. Custom CNN's loss decreased from 0.7008 to 0.0405.
EfficientNetB0 achieved 100% validation accuracy by Epoch 1.

IV. Discussion

Key Findings:
- EfficientNetB0 achieved perfect accuracy but had 4× more FLOPs than MobileNetV2
- Pretrained models outperformed Custom CNN despite fewer trainable parameters
- MobileNetV2 is optimal for resource-constrained edge devices

Limitations:
- Dataset size may limit generalization
- Binary classification only

V. Conclusion

EfficientNetB0 achieved the highest accuracy (100%) for fire detection, while MobileNetV2 offers the best trade-off between accuracy and computational efficiency. Future work will explore real-time deployment on drones and multi-class fire intensity classification.

References

[1] A. Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks," NeurIPS, vol. 25, 2012.
[2] M. Tan et al., "EfficientNet: Rethinking Model Scaling for CNNs," ICML, 2020.
[3] Forest Fire Dataset, Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets

Appendices

Code Repository:
- https://github.com/Adishree10/lab1ml/blob/main/Forest_Fire_classification_WIldfire_dataset.ipynb
- https://github.com/Adishree10/lab1ml/commit/b801d574cbd61a4e80c981db303b96e9c7927fbd