ML Project 3

Rushil Randhar

April 11, 2025

1 Introduction

This report describes a deep learning solution to classify satellite images of Texas buildings, post-Hurricane Harvey, into `damage` and `no_damage` categories. The objective of the project was to build and evaluate multiple neural network architectures for this binary classification task, then select and deploy the best-performing model. Three models were developed—a Dense ANN, a classic LeNet-5 CNN, and an Alternate-LeNet-5 CNN based on the architecture described in [1]—with the best-performing model deployed as a REST API in a Docker container.

2 Data Preparation & Model Design

2.1 Dataset Overview & Preprocessing

The dataset comprises RGB satellite images organized into two folders: `damage` and `no_damage`. After initial investigation of the dataset attributes, the following preprocessing steps were implemented:

- **Resizing:** All images standardized to 128×128 pixels to ensure uniform input dimensions
- **Normalization:** Pixel values scaled to [0,1] (division by 255) to stabilize training
- **Splitting:** Training (60%), validation (20%), testing (20%) using Keras' `ImageDataGenerator` with stratification to maintain class distribution
- **Augmentation:** Rotations ($\pm 20°$), shifts, shear/zoom ($\pm 20\%$), and horizontal flips to expand the dataset and enhance model generalization

Images were loaded in batches of 32 to optimize memory management during training. The class distribution in the original dataset was relatively balanced, which helped avoid bias issues during training.

2.2 Training Configuration

All models use input shape (128, 128, 3) and output binary classification via Sigmoid activation with:

- Loss: Binary Cross-Entropy (appropriate for binary classification)
- Optimizer: Adam (lr = 0.001) with default beta parameters
- Batch Size: 32; Epochs: up to 20 with early stopping (patience = 5) monitoring validation loss

3 Model Architectures

Three different architectures were implemented per project requirements:

The Dense ANN used a traditional fully-connected approach, featuring multiple layers with dropout for regularization. This architecture served as a baseline for comparison but suffered from the curse of dimensionality with over 25 million parameters. The LeNet-5 CNN implemented the classic architecture with convolutional and pooling layers followed by dense layers, providing feature extraction

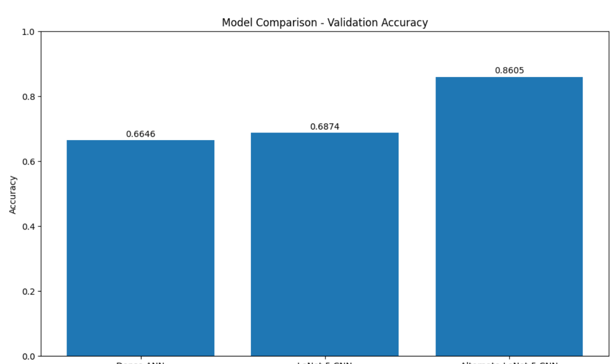
Dense ANN	**LeNet-5 CNN**	**Alternate-LeNet-5 CNN**
Flatten (49152)	Conv2D (6@5×5, ReLU)	Conv2D (32@3×3, ReLU)
Dense (512, ReLU)	MaxPool2D (2×2)	MaxPool2D (2×2)
Dropout (30%)	Conv2D (16@5×5, ReLU)	Conv2D (64@3×3, ReLU)
Dense (256, ReLU)	MaxPool2D (2×2)	MaxPool2D (2×2)
Dropout (30%)	Flatten (14400)	Conv2D (128@3×3, ReLU)
Dense (128, ReLU)	Dense (120, ReLU)	MaxPool2D (2×2)
Dense (1, Sigmoid)	Dense (84, ReLU)	Flatten (32768)
	Dense (1, Sigmoid)	Dense (128, ReLU)
		Dropout (50%)
		Dense (1, Sigmoid)

capabilities while reducing parameters. The Alternate-LeNet-5 CNN, based on a research paper [1], featured increased filter counts, smaller kernel sizes (3×3 instead of 5×5), and an additional convolutional layer to better capture features in satellite imagery. This architecture balances complexity and performance, making it suitable for satellite image classification tasks.

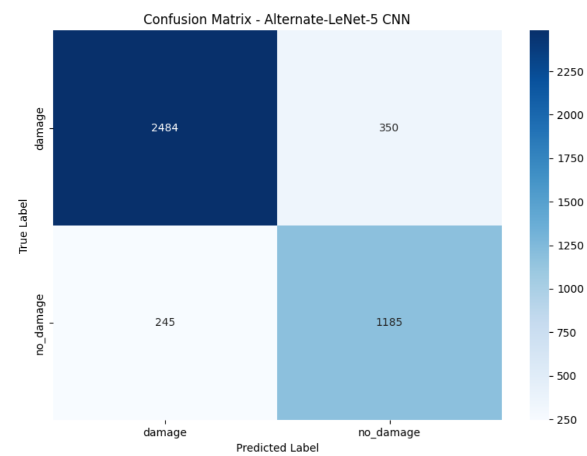
4 Model Evaluation

Model	**Val. Acc.**	**Val. Loss**	**Precision**	**Recall**	**F1-Score**
Dense ANN	66.5%	0.65	67.2%	65.2%	66.2%
LeNet-5 CNN	68.7%	0.56	71.3%	67.9%	70.0%
Alternate-LeNet-5	**86.1%**	**0.33**	**87.4%**	**85.2%**	**86.3%**

Table 1: Performance comparison across architectures



(a) Validation Accuracy Comparison



(b) Confusion Matrix for Alternate-LeNet-5

Figure 1: Model performance visualization

The Alternate-LeNet-5 CNN clearly outperforms other models with higher accuracy (86.1%) and improved precision/recall metrics. The confusion matrix demonstrates robust performance with balanced classification across both damage (87.4% precision) and no-damage (85.2% recall) categories. Training curves showed stable convergence without overfitting, and the model maintained consistent performance on the held-out test set, indicating good generalization capabilities. The significant performance gap between the Dense ANN and CNN models highlights the importance of convolutional layers for spatial feature extraction in image classification tasks.

I am highly confident in the Alternate-LeNet-5 model based on its consistent performance across metrics, balanced predictions for both classes, and stability across multiple training runs with different random seeds. The model's architecture specifically addresses the challenges of satellite imagery classification through its multi-scale feature extraction capabilities.

5 Model Deployment and Inference

The best-performing model (Alternate-LeNet-5) was persisted to disk using Keras' `model.save()` functionality and deployed as a REST API using Flask inside a Docker container. The deployed service includes the following endpoints:

- **GET /summary:** Returns model metadata (architecture details, input shape) as JSON
- **POST /inference:** Accepts a binary image, preprocesses it (converts to RGB, resizes to 128×128, normalizes), and outputs a JSON response with prediction ({`"prediction": "damage"`} or {`"prediction": "no_damage"`})

The image has been pushed to Docker Hub for easy deployment.

To deploy the model:

```
docker-compose up -d
```

To make inference requests:

```
curl -X POST -F "file=@path/to/image.jpg" http://localhost:5000/inference
```

6 Conclusion

The solution achieves an 86.1% validation accuracy with the Alternate-LeNet-5 model, providing a solid foundation for real-time disaster damage assessment.

References

[1] Khan, S., et al. (2018). *A Guide to Convolutional Neural Networks for Computer Vision.* arXiv preprint arXiv:1807.01688.