Vision AI: **Image Recognition in 5 Days**

Name: K. M. C. Chamikara Bandara

Bootcamp: DevTown || Build An AI That Sees

Date: 2025-08-13

# Table of Contents

1. Abstract

2. Introduction

3. Objectives

4. Dataset Overview

5. Methodology

5.1 Day 1 – Dataset Preprocessing & Visualization

5.2 Day 2 – Baseline CNN Model (MNIST)

5.3 Day 3 – Data Augmentation & Model Evaluation (CIFAR-10)

5.4 Day 4 – Transfer Learning (Cats vs. Dogs)

5.5 Day 5 – Model Deployment & Demo

6. Results and Analysis

7. Conclusion

8. Future Work

9. References

# 1. Abstract

In this project, it demonstrates the development of an image recognition system using deep learning techniques. We began with simple convolutional neural networks (CNNs) in first day and progressed to transfer learning with MobileNetV2. Then go with multiple datasets including MNIST, CIFAR-10, and Cats vs. Dogs from Kaggle. The final deliverable is a deployable image recognition tool capable of real-time predictions, accompanied by complete documentation and visual performance metrics.

# 2. Introduction

Artificial Intelligence (AI) has revolutionized computer vision, enabling machines to perform tasks that require human-like visual understanding. In this project, we designed image recognition within five days as a bootcamp activity host by DevTown, transitioning from basic model implementation to advanced transfer learning. Using Python and TensorFlow/ Keras

# 3. Key Objectives

- Basic Knowledge about deep learning and how to apply it.

-Understand Data Preprocessing

- Understand image preprocessing and augmentation techniques.

- Build and train CNN from scratch.

-Understand about Dropout and why use that technique.

- Apply model evaluation metrics to assess performance.

- Implement transfer learning with MobileNetV2.

-Learn how to fine-tune a model.

- Deploy a trained model for real-time prediction.

# 4. Dataset Overview

Here is the data set we used:

1. [MNIST:](https://www.kaggle.com/datasets/hojjatk/mnist-dataset) 70,000 grayscale images of handwritten digits (28×28).

2. CIFAR-10: 60,000 color images in 10 classes (32×32).

3. [Cats vs. Dogs](https://www.kaggle.com/datasets/tongpython/cat-and-dog): 25,000 labeled images (varied dimensions), resized to 224×224.

# 5. Methodology

5.1 Day 1 – Dataset Preprocessing & Visualization

- Loaded MNIST dataset.

-Normalized pixel values to [0,1], reshaped to (28,28,1).

- Visualized sample images using Matplotlib.

5.2 Day 2 – Baseline CNN Model (MNIST)

- Built a Sequential CNN with:

-Conv2D → MaxPooling → Conv2D → MaxPooling → Flatten → Dense layers.

- Used adam optimizer and sparse\_categorical\_crossentropy loss.

- Trained for 5 epochs with 80/20 train-validation split.

5.3 Day 3 – Data Augmentation & Model Evaluation (CIFAR-10)

- Applied ImageDataGenerator with rotation, shift, and flip augmentation.

- Trained a deeper CNN on CIFAR-10 (input size (32,32,3)).

- Evaluated performance using confusion matrix and classification report.

5.4 Day 4 – Transfer Learning (Cats vs. Dogs)

- Downloaded dataset from Kaggle, split into training/validation sets.

- Used MobileNetV2 (pretrained on ImageNet) with frozen base layers, followed by

Fine-tune

- Evaluated with ROC curve and AUC score.

5.5 Day 5 – Model Deployment

- Implemented image upload and prediction pipeline for cats/dogs classification.

- Displayed predictions along with visual comparison of dataset accuracies.

Additional Development

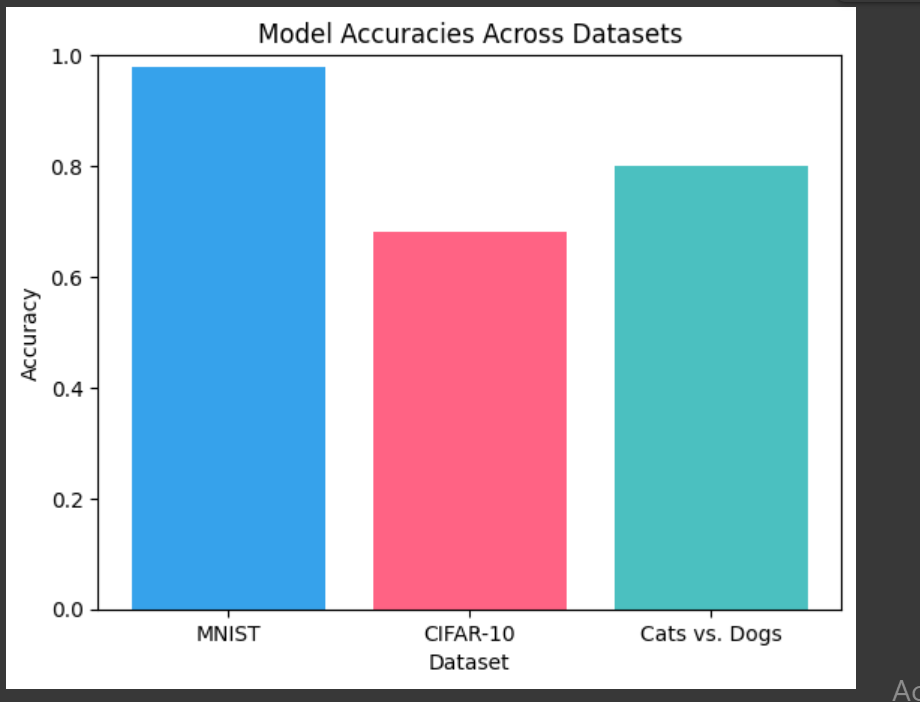
- Build a simple frontend to upload and check the results using gardio.

# 6. Results and Analysis

- MNIST Baseline CNN: ~98% accuracy.

- CIFAR-10 CNN: ~68% accuracy.

- Cats vs. Dogs MobileNetV2: ~78% accuracy after fine-tuning.



# 7. Conclusion

In the end we were able to build a successful model that can recognize images in real time. The bootcamp was so engaging that I got a clear vision about image recognition by building this. Special Thanks for DevTown team to conduct this amazing bootcamp.

# 8. Future Work

- Use larger and more diverse datasets.

- Deploy models to web or mobile applications.

# 9. References

- Kaggle Datasets: <https://www.kaggle.com/>