Image Classification: Cats vs Dogs using Convolutional Neural Networks (CNNs)

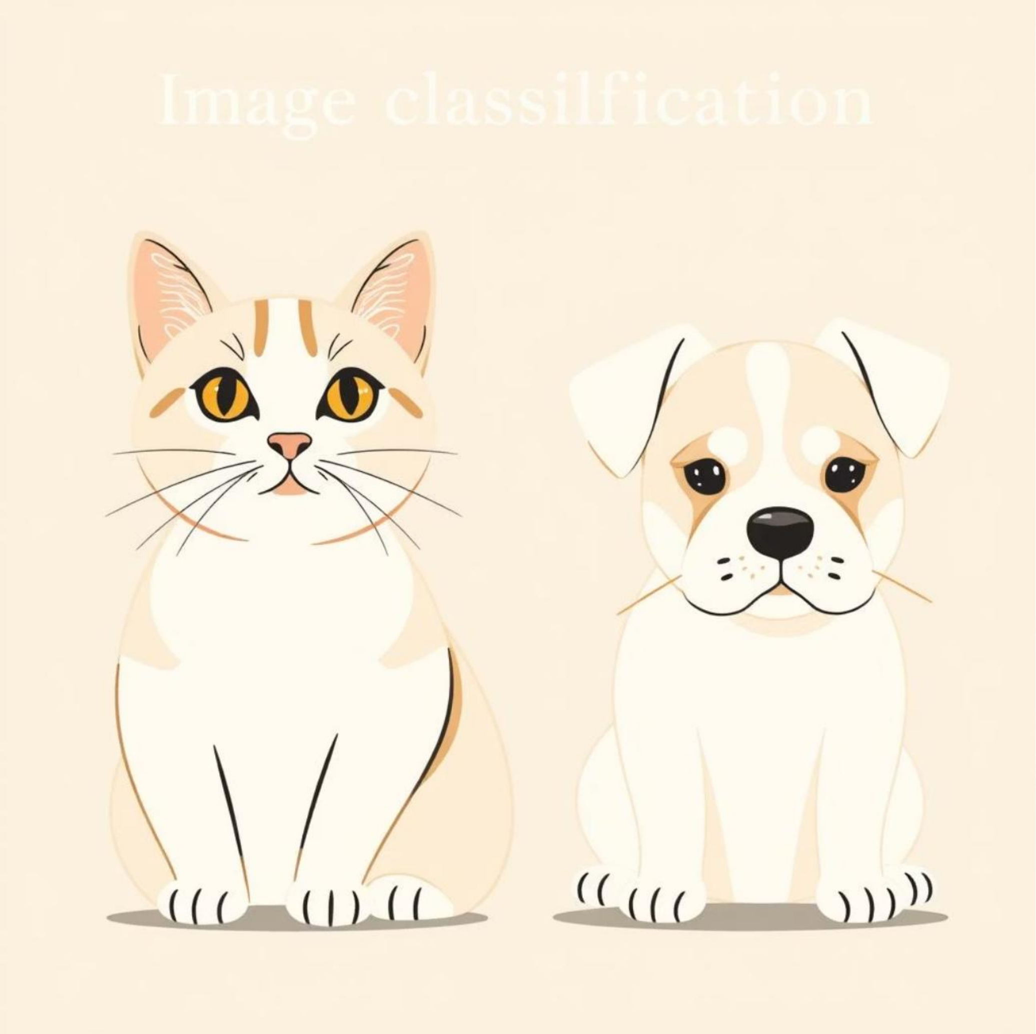
Presented by: Suyash Srivastava

Project Guide: R Dev Nath

Introduction to Image Classification

Image classification is a core task in computer vision, involving the categorization of images into predefined classes. It underpins applications from medical diagnosis to autonomous vehicles.

The "Cats vs Dogs" problem is a classic benchmark, popular for its intuitive nature and as an excellent entry point into advanced techniques like Convolutional Neural Networks (CNNs). It demonstrates how deep learning can differentiate between visually similar yet distinct categories.



Problem Statement & Objectives

Objective

To develop and evaluate a robust CNN model capable of accurately classifying images as either containing a cat or a dog.

Scope

Focus on binary classification using a publicly available dataset, exploring fundamental CNN architectures and training strategies.

Impact

Serve as a foundational project for more complex image recognition tasks and real-world applications.

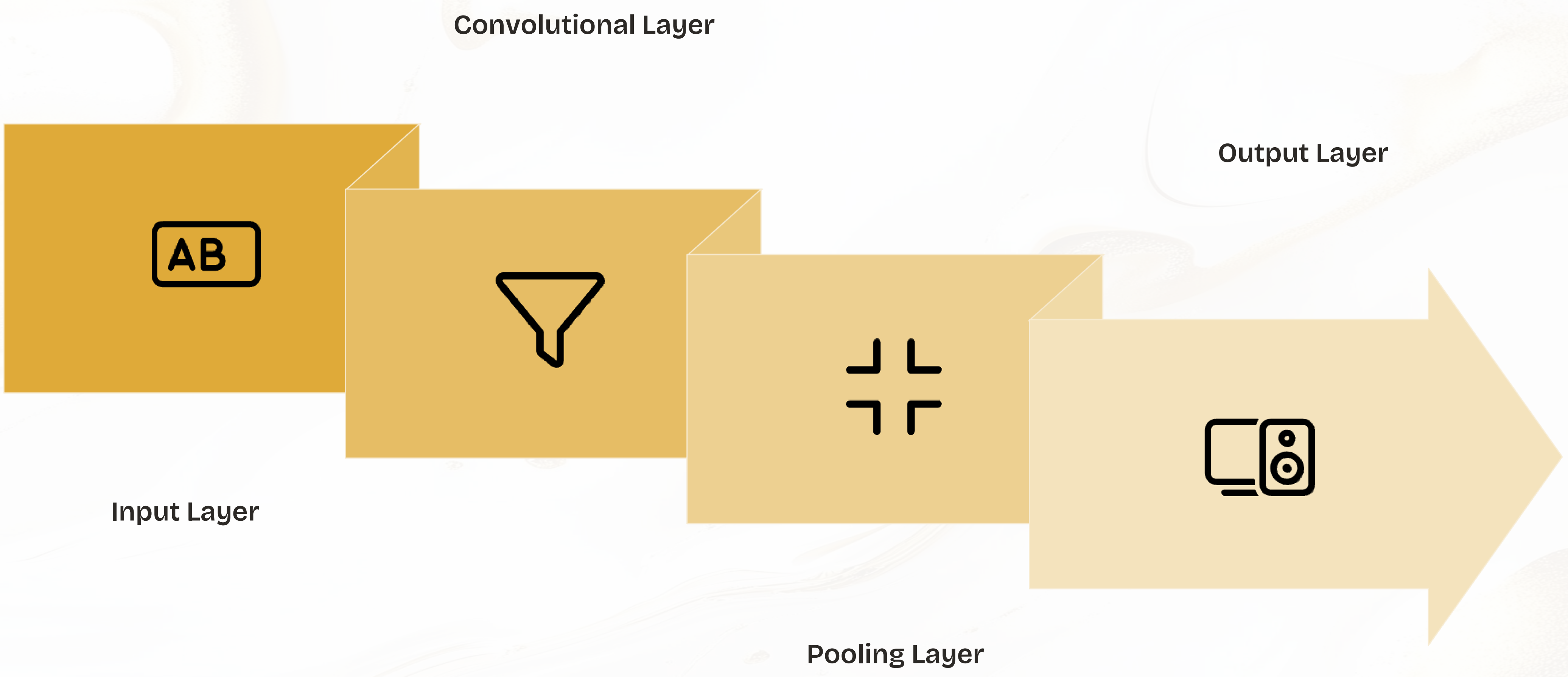
Our goal is to achieve high classification accuracy while understanding the intricacies of deep learning model development.

Dataset Overview: Kaggle Dogs vs. Cats

We utilized the Kaggle "Dogs vs. Cats" dataset, comprising 25,000 images (12,500 cats and 12,500 dogs) for training and validation. Images vary widely in resolution, lighting, and pose, introducing real-world challenges.

The dataset structure is straightforward: images are organized into 'train' and 'test' directories, with subfolders for 'cats' and 'dogs', facilitating direct loading for deep learning frameworks.

Convolutional Neural Network (CNN) Architecture



Training Process & Data Augmentation

Training Strategy

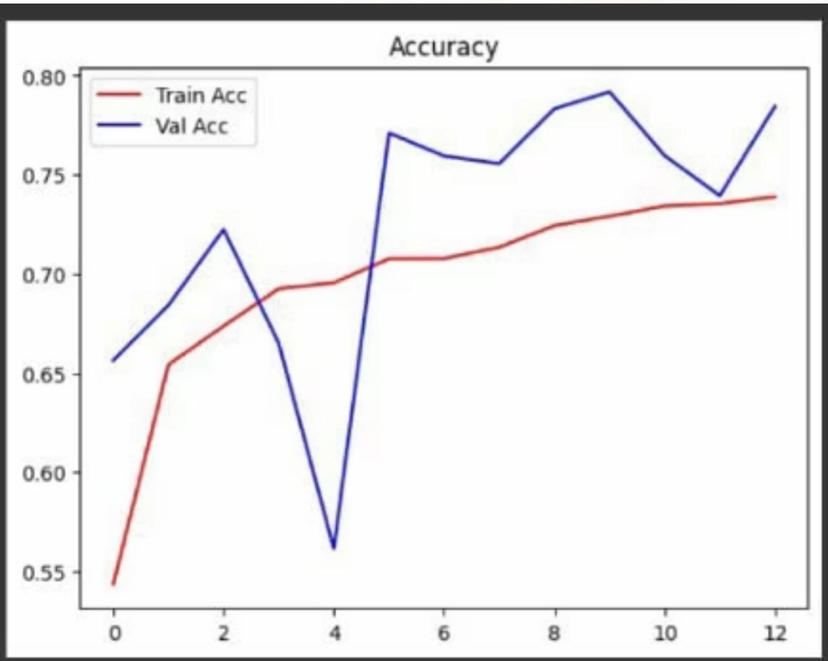
- **Hyperparameters:** 25 epochs for training.
- **Loss Function:** Binary Cross-Entropy, ideal for binary classification.
- **Optimizer:** Adam optimizer with a learning rate of 0.001, for efficient convergence.

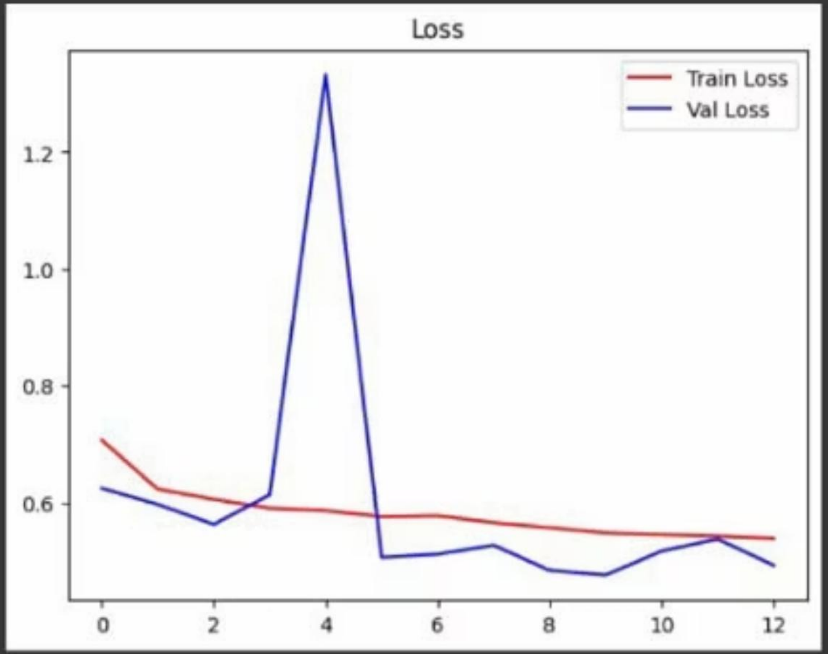
Data Augmentation

To prevent overfitting and enhance generalization, we applied real-time data augmentation:

- Random rotations and shifts
- Horizontal flips
- Zooming
- Shear transformations

Model Training Screenshots





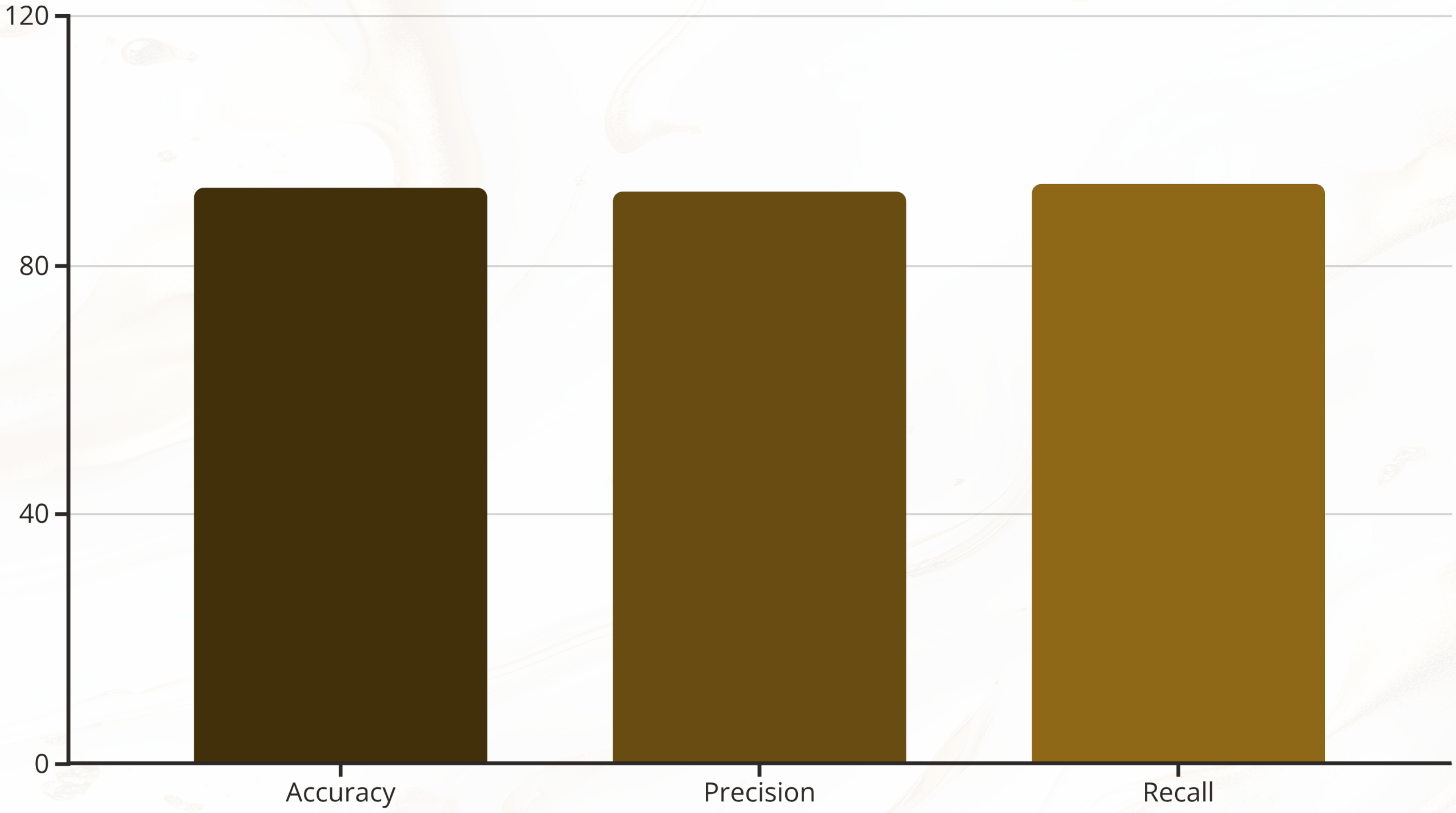
Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	896
batch_normalization (BatchNormalization)	(None, 254, 254, 32)	128
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 125, 125, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 60, 60, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten (Flatten)	(None, 115200)	0
dense (Dense)	(None, 128)	14,745,728
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

Total params: 14,848,193 (56.64 MB)
Trainable params: 14,847,745 (56.64 MB)
Non-trainable params: 448 (1.75 KB)

Evaluation Metrics & Results

Our model achieved strong performance on the test set:



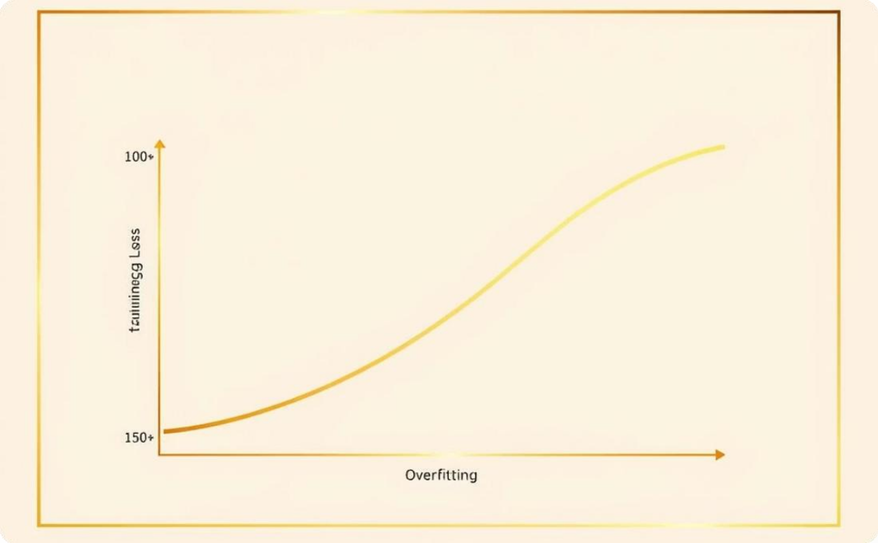
Result Screenshots

Challenges & Solutions

Challenge: Data Quality

Variations in image quality, resolution, and inconsistent labeling within the dataset.

Solution: Implemented robust data preprocessing and augmentation pipelines to normalize inputs and increase data diversity.



Challenge: Overfitting

The model performed exceptionally well on training data but poorly on unseen validation data.

Solution: Applied dropout layers, L2 regularization, and extensive data augmentation to improve generalization.

Challenge: Computational Resources

Training deep CNNs required significant computational power and time.

Solution: Utilized cloud GPU instances and optimized batch processing for more efficient training cycles.

Conclusion & Future Work

> Our project successfully demonstrated the effectiveness of CNNs for complex image classification tasks, achieving high accuracy in distinguishing cats from dogs. This serves as a solid foundation for more advanced computer vision applications.

1. **Model Improvement**

   Explore transfer learning with pre-trained models (e.g., VGG, ResNet) for even higher accuracy.

2. **Broader Applications**

   Extend the model to classify a wider range of animal species or other object categories.

3. **Real-time Deployment**

   Optimize the model for deployment on edge devices for real-time classification in practical scenarios.

Thank You!

Please feel free to ask any questions