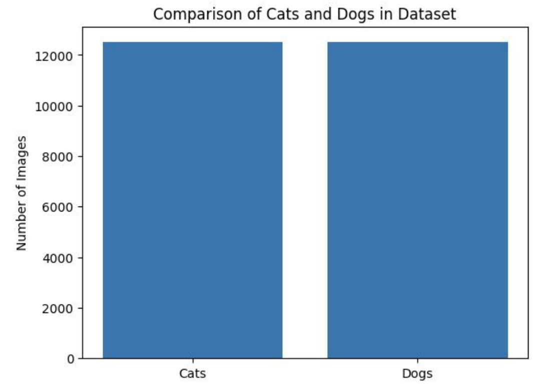
# Image Classification: Cats and Dogs

Authors:
- Rahiq Raees
- Mulin Chen
- Basim Hilal
- Jingchao Zhu

# Competition Overview and Data Exploration

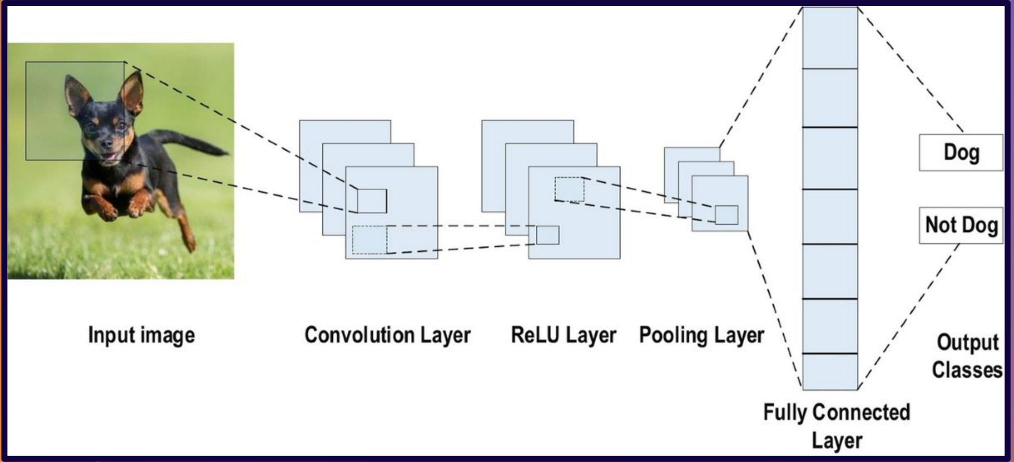
Directory

```
train/
    dogs/
        dog001.jpg
        dog002.jpg
        ...
    cats/
        cat001.jpg
        cat002.jpg
        ...
test/
    000001.jpg
    000002.jpg
    ...
```

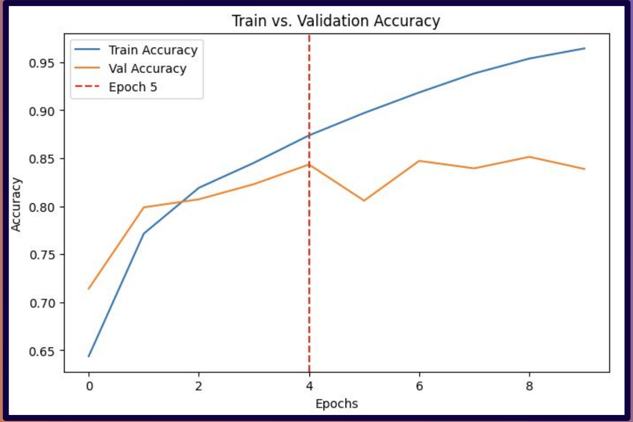

```
The number of cats in our training set are 12500
The number of dogs in our training set are 12500
The dataset is perfectly balanced!
```

# Data Processing and Methodology

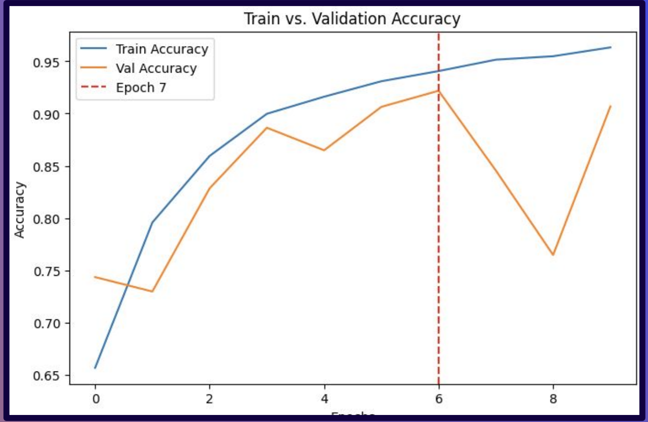



# CNN Results

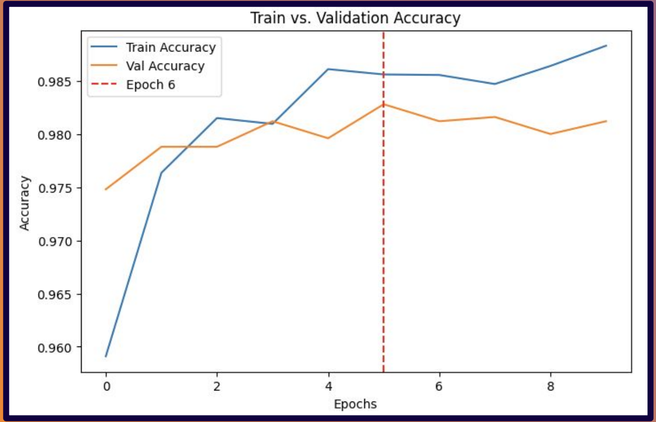
Simple CNN



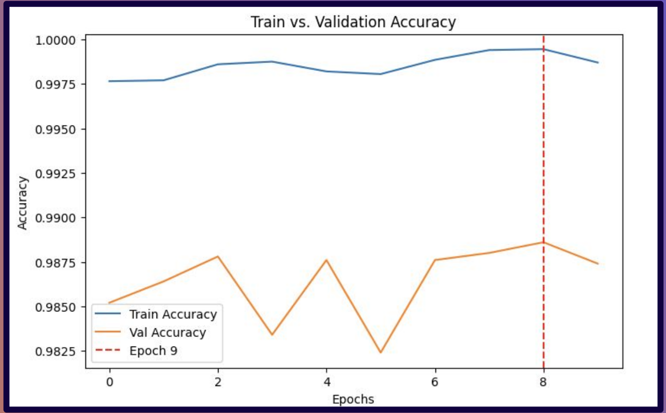

Deep CNN




VGG16




VGG16 Tuned



# Hyperparameter tuning

## Simple CNN

| Epoch | Train Acc | Val Acc | Notes |
|---|---|---|---|
| 10 | 78.6% | 79.6% | Strong learning |
| 20 | 84.9% | **85.3%** | good |

## Simple CNN + random search

| Epoch | Train Acc | Val Acc | Val Loss | Notes |
|---|---|---|---|---|
| 1 | 96.8% | 77.1% | 0.7963 | Overconfident model |
| 2 | 98.4% | 77.9% | 0.8770 | Val loss worsens, LR reduced |
| 3 | 99.4% | 78.3% | 1.1089 | Val loss explodes |

Overfiting: RandomSearch likely chose too many filters, too large dense layer, low dropout, or high learning rate → perfect train accuracy, poor validation.

No learning in tuning strategy: RandomSearch blindly picked 3 random configs — too shallow to find optimal settings.

Bad tuning sample: tuned on just 2,000 images — that subset didn't reflect the full training set → overfit to tuning data, failed to generalize.

## Simple CNN + Bayesian

| Epoch | Train Acc | Val Acc | Val Loss | Notes |
|---|---|---|---|---|
| 1 | 96.8% | 77.1% | 0.7963 | Overconfident model |
| 2 | 98.4% | 77.9% | 0.8770 | Val loss worsens, LR reduced |
| 3 | 99.4% | 78.3% | 1.1089 | Val loss explodes |

Overfitting via overpowered configs: Bayesian search likely favored too many filters, oversized dense layers, low dropout, or aggressive learning rates → led to perfect training accuracy but poor validation.

Shallow search: Only ran 3–5 trials — far too few for Bayesian Optimization to explore the hyperparameter space effectively → stuck in local minima, no real tuning.

Poor tuning subset: Used just 2,000 images for tuning — subset didn't represent the full training distribution → model overfit to the sample, failed to generalize.

# Findings and Challenges

Model comparison:
Deep CNN outperformed Simple CNN across all key metrics — accuracy, loss, and generalization.
Transfer learning (e.g., VGG16) demonstrated competitive performance with reduced training time and lower overfitting risk.

Evaluation metrics:
Binary cross-entropy worked well for probabilistic outputs. Accuracy, validation loss, and early stopping were used to track and compare model quality.

Conclusions are Well-Supported:
The chosen Deep CNN architecture is justified by strong empirical results.Hyperparameter tuning methods must shift from manual to automated (Bayesian/Random search) to scale experiments efficiently.

# Thanks for Listening!