CS 203: Software Tools & Techniques for AI
IIT Gandhinagar
Sem-II - 2024-25

LAB 05

Total marks: 10
Submission deadline:
Submission guidelines:

1. Code should be added to a GitHub repository, and the repository details should be shared.
2. Late submissions will be penalized 20% per day.
3. Google form submission link:

Note: By submitting this assignment solution you confirm to follow the IITGN's honor code. We shall strictly penalize the submissions containing plagiarized text/code.

Model Architecture

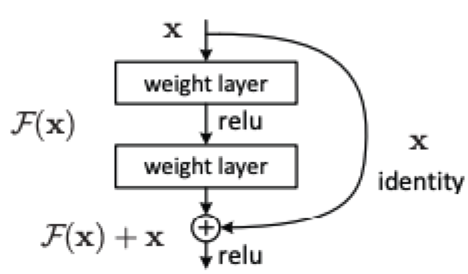
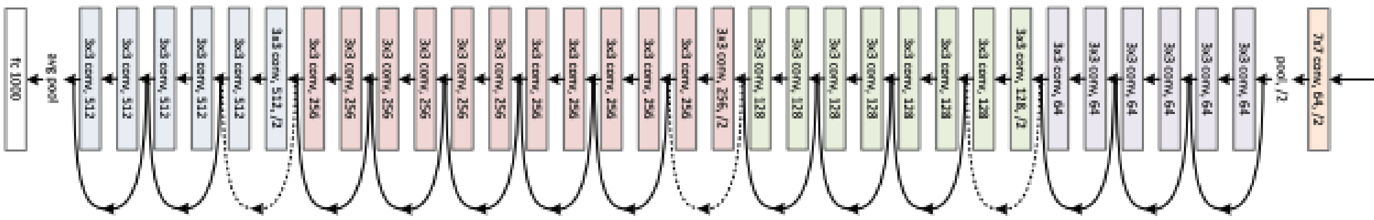


Figure 2. Residual learning: a building block.

Source: Deep Residual Learning for Image Recognition, Microsoft Research



Resnet 34 Network (50 is a larger version of this)

ResNet50 is a deep convolutional neural network with **50 layers**, designed for image recognition. It uses **residual connections (skip connections)** to prevent vanishing gradients and enable training of very deep networks. Pre-trained on **ImageNet**, it achieves high accuracy on various vision tasks. It’s widely used in **transfer learning** for object detection, classification, and feature extraction.

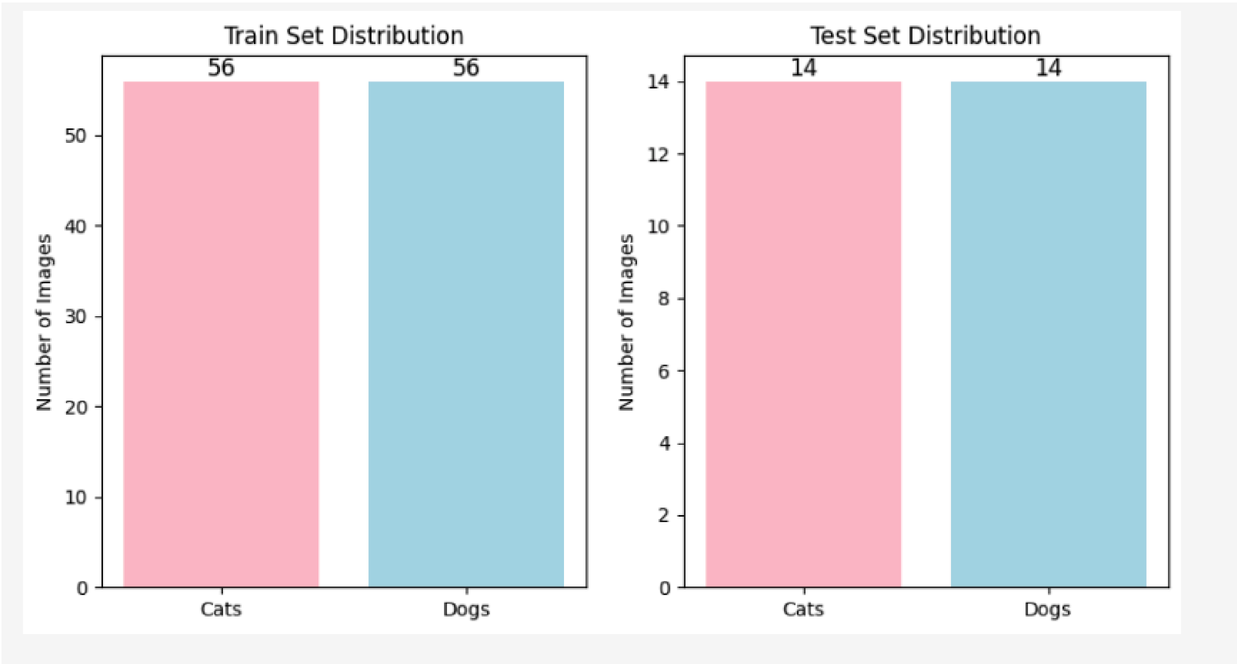
Additionally, our setup uses

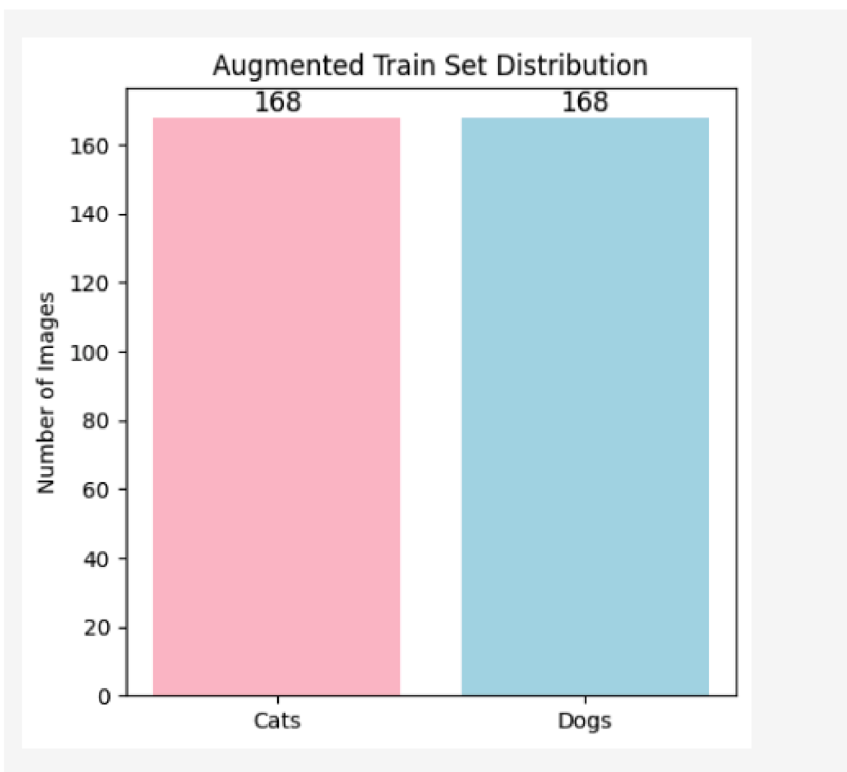
50 layers, including convolutional, batch normalization, and ReLU activation layers.

Bottleneck residual blocks, which improve computational efficiency while maintaining performance.

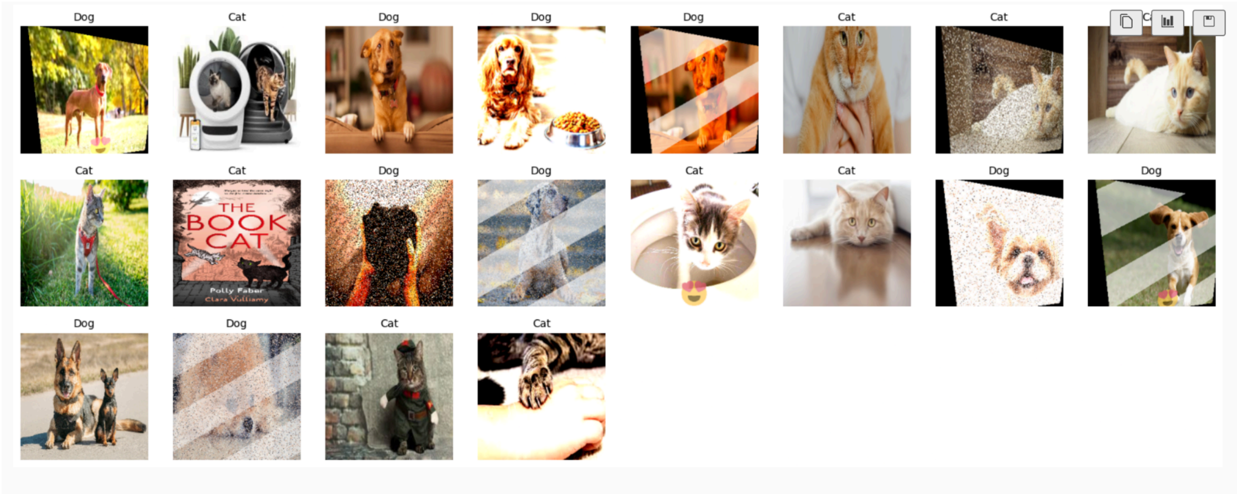
Global average pooling followed by a fully connected (FC) layer for classification.

Graphs


Train Set Distribution
56
56
Number of Images
0
10
20
30
40
50
Cats
Dogs
Test Set Distribution
14
14
Number of Images
0
2
4
6
8
10
12
14
Cats
Dogs

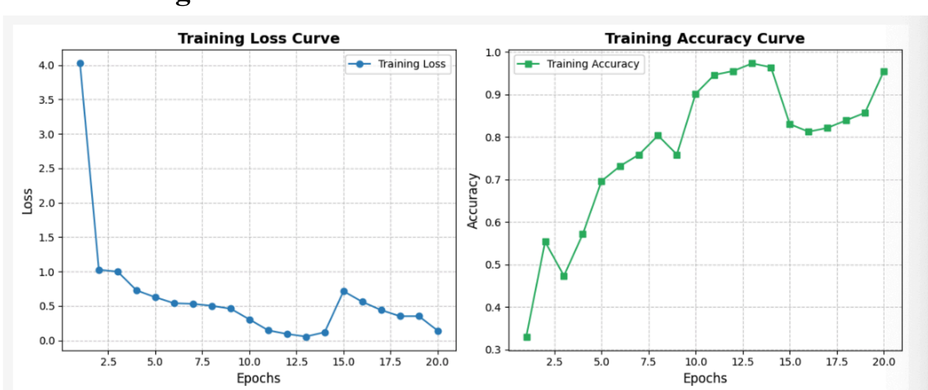

Augmented Train Set Distribution
168
168
Number of Images
0
20
40
60
80
100
120
140
160
Cats
Dogs

Augmented images (few)



Augmented train set training

Normal train set training



Scores

```
ResNet50 (Normal Training) Results:
  Accuracy:  0.6786
  Precision: 0.6923
  Recall:    0.6429
  F1 Score:  0.6667

ResNet50 (Augmented Training) Results:
  Accuracy:  0.6429
  Precision: 0.6429
  Recall:    0.6429
  F1 Score:  0.6429
```

Interpretation

Traintime

- We observe that during training, the accuracy for the augmented dataset does not rise as quickly as the normal dataset. This might seem counterintuitive, but it happens because augmentation adds variations, making the task harder initially. However, this increased diversity helps the model generalize better over time.

Testtime

- For normal training, the accuracy is **67.86%**, meaning the model has learned something beyond random guessing (which would be 50% for binary classification). However, augmentation seems to slightly hurt accuracy, bringing it down to **64.29%**. This suggests that while augmentation helps with generalization, it might have introduced some noise that made learning harder in this specific case.
- Interestingly, **all metrics (accuracy, precision, recall, F1-score) are identical for the augmented model**, indicating a balanced behavior across classes. This could mean the model has become more conservative in its predictions, rather than being overly confident in one class.

Extra

- **Recall drops** (from **64.29% → 64.29%**, meaning it remains the same), indicating that augmentation did not make the model more cautious about missing positive samples.
- **F1 Score remains unchanged**, meaning there's no drastic shift in the trade-off between precision and recall.
- **Precision slightly drops** (from **69.23% → 64.29%**), meaning the model has a bit more false positives.

Overall, normal training performed slightly better, but augmentation may still be useful in different scenarios or with further tuning.