CONVOLUTION REPORT

Yashwanth Reddy Sadala

811292491

The objective of this project is to construct a convolutional neural network that, by recognizing the distinctive characteristics of each animal, can accurately and successfully identify photos of cats and dogs. A Kaggle dataset comprising 12,500 tests and 25,000 training images is used in this experiment. The same volume of dogs and cats.

We will look at the connection between training samples and whether you should use a pretrained convnet or train your model from scratch.

Problem to be defined:

The goal of the Cats-vs-Dogs dataset is to identify whether an image is of a dog or a cat.

Data:

The Cats-vs-Dogs dataset, which has a compressed size of 543MB, has 25,000 images that are split equally between dogs and cats. After downloading and unzipping the dataset, I created a new one.

Preprocessing of the Data:

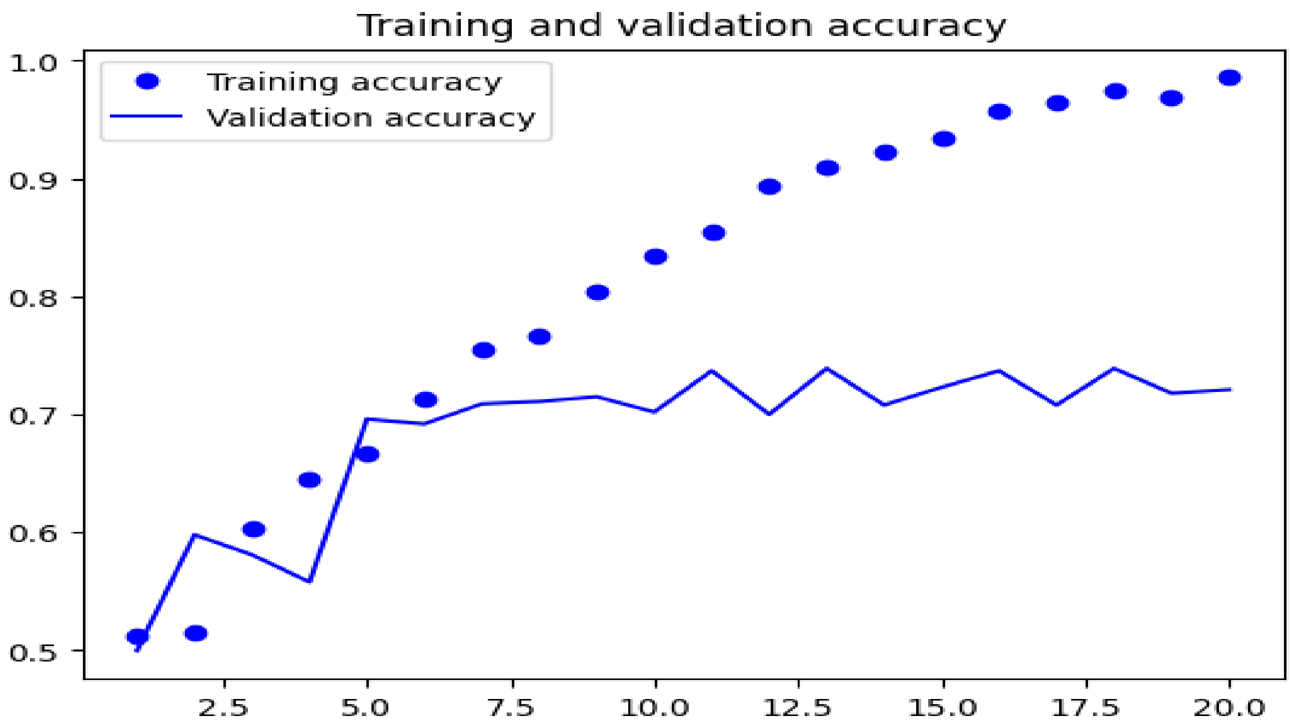
- Review the image files.
- Create RBG pixel grids from the JPEG data.
- From grids, create floating point tensors.

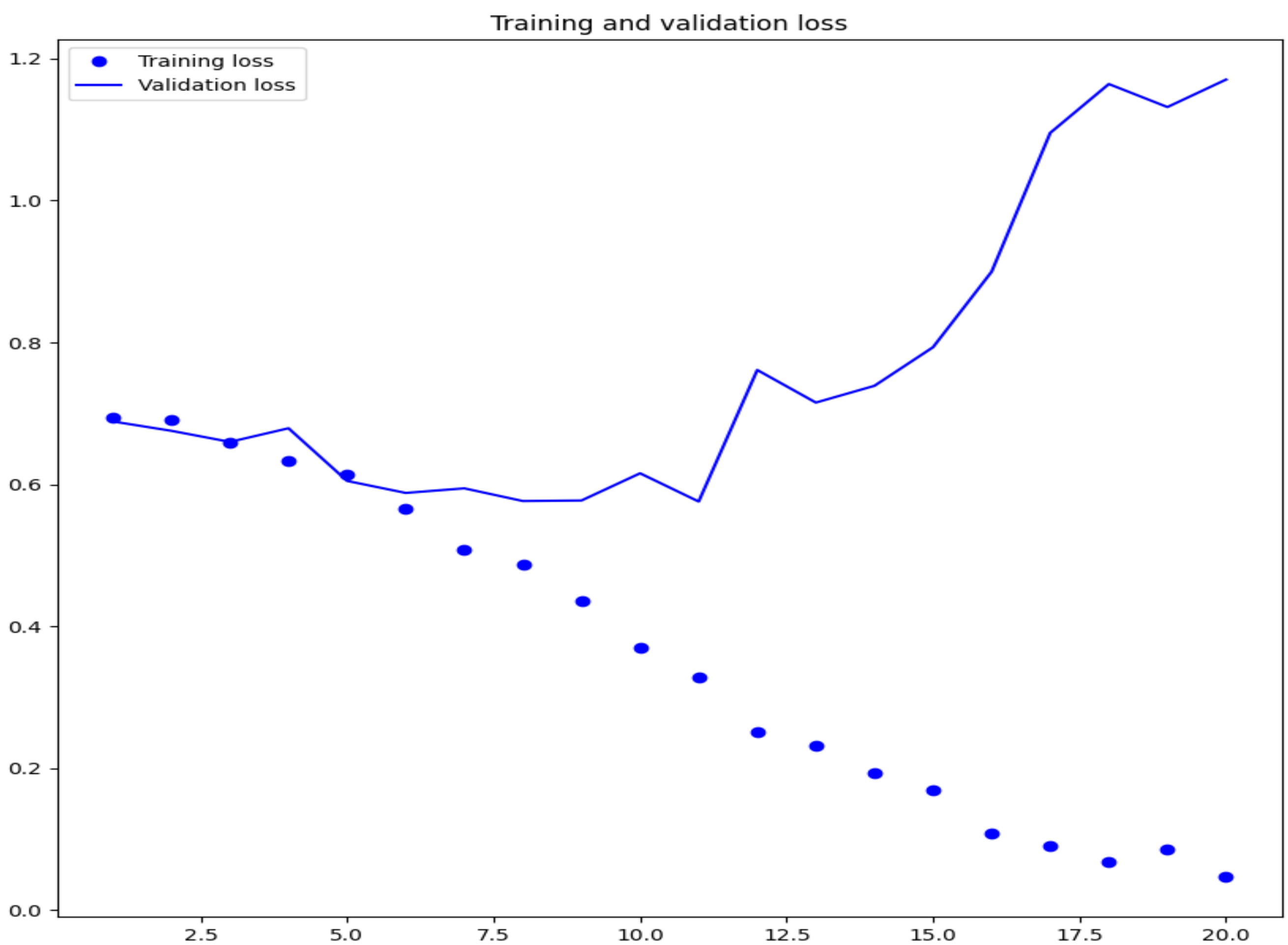
Rescaling the 0–255-pixel values to the [0, 1] interval is necessary because neural networks function best with small input values.

Q1. Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?

The data transformation is to be converted using the data flattening technique. 20 epochs helped us determine that the test accuracy was 73.4% and the validation accuracy was 73.40%.

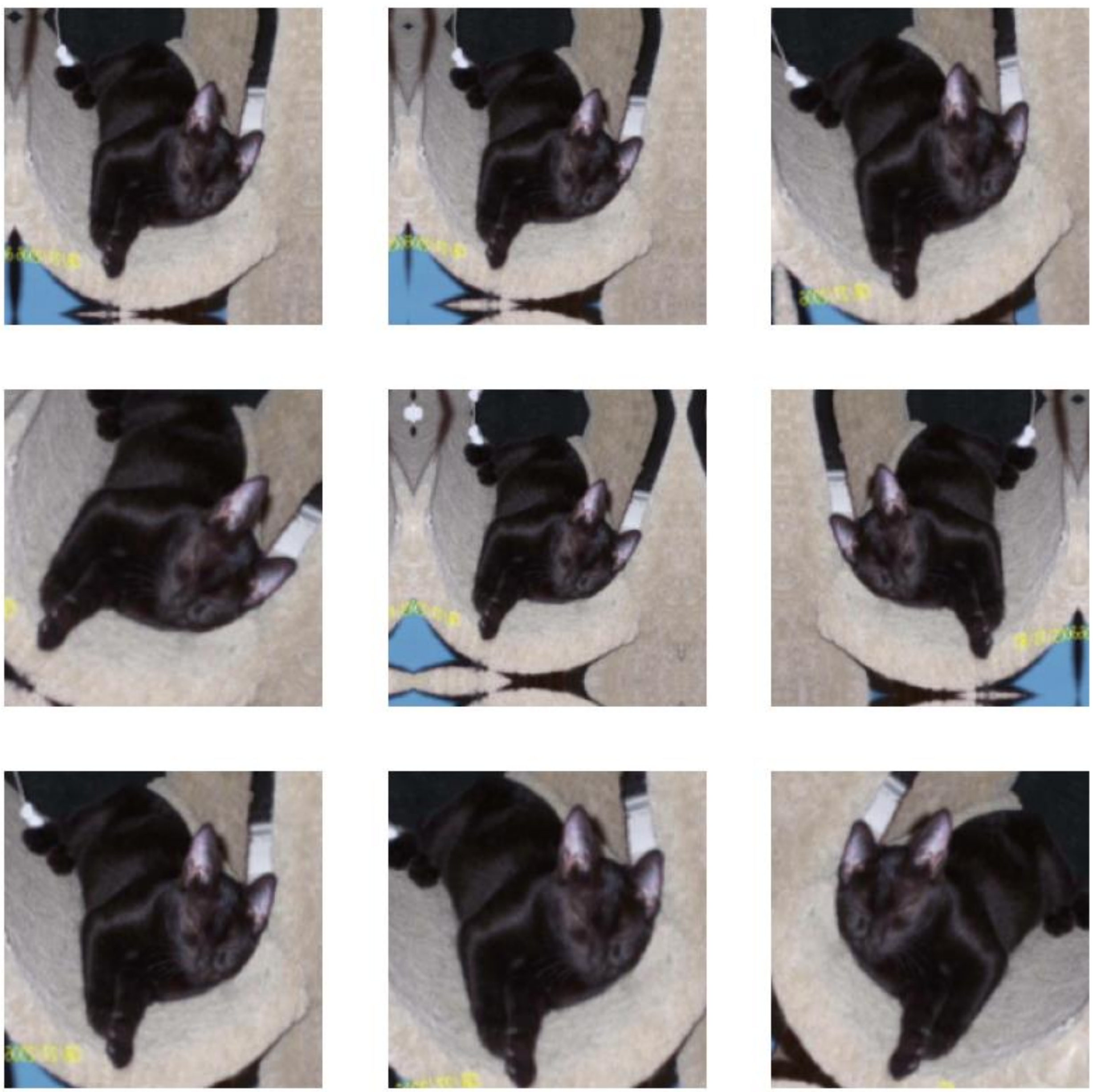
Test accuracy with no data augmentation is about 72.5% while the Training accuracy is about 97.9%.


Training and validation accuracy
Training accuracy
Validation accuracy
1.0
0.9
0.8
0.7
0.6
0.5
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Training and validation loss
Training loss
Validation loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Q2. Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?

This method can be used to increase the model's accuracy. Data augmentation is one method that enables reliable results to be obtained even with small datasets. To create new data, it comprises modifying the provided training samples at random. By ensuring that the model sees a wide variety of images during training, this technique enhances the model's ability to successfully generalize.



Test Accuracy: 89.9% Better results than the previous (Question 1) are evident from the validation accuracy of 89.8%.
The following factors have led to an improvement in the model's performance: By adding 500 1000–1500 training samples, we were able to nearly 10% improve test and validation accuracy; In addition to the convolution layer, we also used data augmentation, which helped us enhance the featured extractions and improve performance.

Q3. Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get best prediction results.

Since we are aware that using an increasing amount of data will help to improve the model's performance, we are unable to determine the appropriate sample size. Test sets comprising 500 samples and 2000 training samples with validation were used in this. I've discovered that test accuracy is higher with 1500 photos as opposed to training samples of 1000 and 2000 photos. Training accuracy increases with 1000 training samples. increasing the training sample to 2000 while keeping the validation and test set of 500 samples.

- Training samples- 1000:

  Validation Accuracy: 72,10%
  Test Accuracy: 73.4%

  Data Augmentation is NO.

- Training samples- 1500:

  Validation Accuracy: 79.10%
  Test Accuracy: 89.9%

  Data Augmentation is YES.

- Training samples- 2000:

  Validation Accuracy: 68.7%
  Test Accuracy: 69.7%

  Data Augmentation is YES.

Q4. Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use all optimization techniques to get the best performance.

The main applications of trained networks are in feature extraction and fine-tuning. A pretrained network can be used as a generic model with its features applied to a range of computer vision applications if its initial dataset is large and diverse. One of the main advantages of deep learning over other machine learning techniques is its capacity to apply learned properties to a wide range of tasks. With 1.4 million annotated images and 1,000 distinct classifications, the ImageNet dataset provides an example of analyzing a large trained convolutional neural network. The collection comprises many different animal categories, such as various dog and cat breeds. This network uses the popular and straightforward VGG16 convolutional neural network architecture for ImageNet.

Pre-trained model with no augmentation. We have a 98% validation accuracy and a 99.8% train accuracy. The graphs show that, despite using dropout at a relatively high rate, we are overfitting almost instantly.

Pre-Trained model with Data Augmentation:

Validation accuracy is 97.8%.
Test accuracy: 96.4%.

Fine-tuning a pretrained model:

Validation accuracy is 98%.
Test accuracy: 96.8%.

Conclusion:

To prevent overfitting when training from scratch, the size of the dataset and augmentation methods must be carefully considered. Model performance was greatly enhanced by applying data augmentation and increasing the size of the training sample. Pretrained networks are a strong foundation, particularly when utilizing sparse data. Pretrained models can be fine-tuned to minimize overfitting and produce high accuracy. In both cases, data augmentation was found to be essential for enhancing model generalization and lowering overfitting. Depending on the size of the dataset, available computing power, and required performance, one can choose between training a pretrained network from scratch and starting from scratch. When working with limited data and computational resources, pretrained models are useful; however, training from scratch offers greater customization and control over the model architecture. Testing various dataset sizes, augmentation strategies, and model architectures is essential for achieving optimal performance in image classification tasks.