Dogs vs Cats Image Classification

Austin H Kim

*Abstract*— Image classification is an active area of research where people are trying to improve and develop neural networks and other models that can accurately classify images based on their features. Convolutional Neural Networks and Deep Learning have been the preferred method for image classifications and this paper will be using Very Deep Convolutional Networks to classify various photos of dogs and cats.

# Introduction

Advancements in image classifications have been gaining traction among researchers, programmers, and the public. The topic of image classifications is not limited to the confines of classifying still images but has developed and branched out into more specific and advanced subjects such as computer vision. Machine learning and deep learning methods have evolved over time and facial recognition technology has drastically improved. The development of deep learning methods allowed for this exponential growth and enabled us to improve security, identification, and biometrics.

The most prominent method of improving and researching image classifications have been deep learning. Over the past decade, deep learning was trending among researcher, students, and hobbyists which in turn encouraged the rapid growth in the field.

The dataset, which contained 2500 photos of cats and dogs each, was taken from a competition on Kaggle that was held in 2013. The dataset was a subset of the Animal Species Image Recognition for Restricting Access (ASIRRA) dataset which was mainly used as a turing test for online purposes.

After the end of the competition in 2013, people continued to take advantage of the reliable amount of data that was available. People more commonly used convolutional neural networks and other deep learning methods. There have been attempts at using less advanced methods such as the k nearest neighbor method

# Methods.

## Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) differ from Artificial Neural Networks (ANN) where each node receives multiple units from the previous layer and incorporates their weights into the algorithm. This is advantageous to image classification since CNNs allow each node to take multiple pixels as inputs. This allows the model to analyze the relationship between neighboring pixels which also helps the model differentiate the figure they are trying to classify with other figures in images. This method of accepting multiple inputs is referred to as the convolutional inside the neural network model. Although inaccurate, it would help intuitively to view these convolutions as a method to output an average of the inputted pixels. This allows the neural network to process the images more easily without sacrificing an unnecessary number of features. Each layer is passed through a filter, which is a matrix that is used to pass the input image through. The filter transforms the original image and outputs a feature matrix through equation where f is the input image, h is the filter, and m and n are the dimensions of our feature matrix.

With each convolutional layer, the model can understand the low-level features of the images and the addition of extra layers allow the model to gradually understand the high-level features of the images.

The Convolutional Neural Network (CNN) used in this experiment is based off [1] Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG), which involves small 3x3 convolutional layers that converges into a pooling layer. The filters used are at the smallest possible size that would still be able to identify the entire surrounding of a single pixel.

Diagram

Description automatically generated

Fig 1. The Architecture of the VGG-16 model

The architecture of the model goes through several convolution layers and max pooling, but goes through a single softmax layer at the end which is represented by the function

The softmax function configures the output of the model to be between 0 and 1 so that they could be interpreted as probabilities. It is typically used in classification models where the classifications are exclusive. The summation ensures that the output vectors sum to 1 so it can be used as a probability in the model. The exponent ensures that vectors with values that are probabilistically invalid are converted to a positive value between 0 and 1 that is scaled depending on the original value.

# Experiment

## Baseline VGG-1 Model

I created a baseline CNN model that is based off the principle of the VGG-16 model where it retained the 3x3 filters along with the max pooling layer. Since the images were all different in dimensions, they were reshaped to be 100 x 100 pixels each. I initially created the CNN with only a single block at 20 epochs and got an average accuracy of 66.284%.

I added a second and a third block to the CNN model and was able to increase the accuracy percentage to 70.134% and 74.167%. With the addition of extra blocks, the model was able to achieve a higher accuracy on correctly classifying the cat and dog images.

## Overfitting and Accuracy

From the initial testing, the VGG-1 model was showing signs of significantly overfitting which continued to carry over to the VGG-3 model.

Chart, line chart

Description automatically generated

Fig 2. Relationship between Epoch and Accuracy in the VGG-1 model

The blue corresponds to the training model and the red corresponds to the testing model, which displayed overfitting and a positive correlation to the number of epochs. To overcome this issue, I introduced dropout regularization to each layer. The dropout regularization helps the problem with overfitting by thinning out the weights in our model by probabilistically omitting a percentage of the inputs. In my model, I set the dropout percentage at 20% for the first two blocks and at 50% for the third block.

Another method I implemented into the model was data augmentation. Unlike dropout regularization, this method is more self-explanatory since the method modifies some of the input images in the training data. These modifications include small shifts, rotations, and flips to the original dataset. The augmentations should not be used on our testing data since we want to test our trained model on the unmodified images. Depending on the set percentage, these modifications will be set randomly across the data and in this model the augmentation was set at 10%.

The previous results demonstrated a continual increase in accuracy as the number of epochs increased. The VGG-1, VGG-2, and the VGG-3 models were all trained through 20 epochs, but I have increased the number to 50 epochs for the final model. Each epoch instructs the model to go back and forth through each photo, which in turn increases the accuracy and decreases the loss overall.

Alongside the improvements that came from adding blocks to the VGG-1 model, the addition of Dropout Regularization, Data Augmentation, and a greater epoch value increased the accuracy to 80.485%

Chart, line chart

Description automatically generated

Fig 3. Relationships between Epochs and Loss as well as Epoch and Accuracy in the VGG-3 model.

Compared to the VGG-1 model, the issue regarding overfitting was significantly tamer. The blue corresponds to the training accuracy and loss while the red corresponds to the testing accuracy and loss. This also confirms that the increase in epochs had a positive correlation to the accuracy of the model.

# Results

Before resulting in the final model, I have saved several models and tested them out with random photos of cats and dogs that were both in and out of the testing dataset. The stochastic nature of the neural networks did not always accurately reflect in the final model. One model had an overall accuracy of 74.127% but was more consistent with guessing photos of dogs correctly compared to classifying photos of cats. The final model was overall more consistent and did not show as much over fitting as the previous models.

My final model resulted in an accuracy of 80.485 and was saved. Another addition program was created to utilize the model to predict any

# Future Work

The goal of this project was for the model to be able to predict whether a photo had a cat or a dog in it. With further development, the model can be able to predict more subtle details between classifications. This includes differentiating between different breeds of dogs, cats, and other animals. This model only had to classify between two distinct figures, but with enough data and more blocks into the same model, it should be able to differentiate among multiple different classifications.

# Conclusion

My results showed that having only three convolutional layers can still yield relatively accurate results. Although it pales in comparison to the VGG-16 model, it still showed signs of having learned the data. Considering that there were only two classifications and selecting the correct classification at random would be 50%, it still shows evidence of the potential behind deep convolutional neural networks.

##### References

1. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition” , 2014, in press.
2. A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks.
3. https://www.kaggle.com/c/dogs-vs-cats/data (December 18, 2020)
4. D. Trigueros and L. Meng, “Face recognition: from traditional to deep learning methods”, 2018, in press
5. Adrian Rosebrock, https://www.pyimagesearch.com/2016/08/08/k-nn-classifier-for-image-classification/