Analysis of JPEG Image Compression Effect on Convolutional Neural Network-Based Cat and Dog Classification

Yueming Qu, Qiong Jia, Euee S. Jang
Hanyang University

Abstract

The process of deep learning usually needs to deal with massive data which has greatly limited the development of deep learning technologies today. Convolutional Neural Network (CNN) structure is often used to solve image classification problems. However, a large number of images may be required in order to train an image in CNN, which is a heavy burden for existing computer systems to handle. If the image data can be compressed under the premise that the computer hardware system remains unchanged, it is possible to train more datasets in deep learning. However, image compression usually adopts the form of lossy compression, which will lose part of the image information. If the lost information is key information, it may affect learning performance. In this paper, we will analyze the effect of image compression on deep learning performance on CNN-based cat and dog classification. Through the experiment results, we conclude that the compression of images does not have a significant impact on the accuracy of deep learning.

1. Introduction

The Cat vs Dog image classification problem has been around for a long time now. Cats and dogs, as the two most close and common animal species in our human life, have been accompanied by people from thousands of years ago, and have continuously spawned thousands of breeds. Currently, there are 73 cat breeds recognized by TICA (The International Cat Association)[1], 369 dog breeds recognized by FCI (Fédération Cynologique Internationale)[2], and many breeds have not been registered and certified. Different breeds of cats and dogs have great differences in appearance, and the appearance characteristics of cats and dogs have many things in common, which makes the problem of the classification between cats and dogs need a large amount of data to support machine learning to achieve a more ideal effect[3].

With the continuous development and improvement of CNN (Convolutional Neural Network) technology, it will be more widely used in image classification and related applications in our daily life. Then, a key challenge of how we can more conveniently and efficiently process large amounts of training data within a limited storage device also emerges. Therefore, we consider using the image compression processing technology to solve this problem. For such massive image data processing, we usually use lossy compression, which will lead to the degradation of image quality, because it discarded part of the information contained[4]. Depending on the compression ratio, the higher compression ratio will result in more loss of the image information. In this paper, we compress the dataset images using different JPEG compression ratios. Then, we want to show the effect of compressed data according to each JPEG compression ratio on training and test results in deep learning.

This paper will be divided into five parts. Except for section 1, the remaining sections are introduced as follows. In section 2, we will review the work that has been done in relation to image classification using deep learning. The method used and proposed in conducting the experiment will be introduced in section 3. Section 4 will discuss the experimental results. Finally, the experimental results are analyzed and conclusions are summarized in section 5.

2. Related Work

2.1. Classification using deep learning

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) from 2010 to 2014 showed that tremendous improvements and advances have been made in large-scale object recognition[5]. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held in 2010, a large and deep CNN trained to classify 1.2 million high-resolution images into 1000 different classes achieved 37.5% in top-1 test errors and 17.0% in top-5 test errors[6]. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held in 2012, AlexNet took 1st place with 15.3% of top-5 test errors[7]. AlexNet has proven the superiority of CNN, the GPU implementation of CNN structure and application of dropout have become common since AlexNet. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held in 2014, GoogleNet took the top-5 with an error rate of 6.7%. GoogleNet utilizes a CNN structure that improves the depth and width of the network by combining multi-scale ideas[5]. Afterwards, STL-10, an image recognition dataset, was used as a benchmark in developing unsupervised learning[8].

3. Proposed Method

3.1. Image Encoding and Decoding using JPEG

Before proceeding with the experiment, the dataset preparation process for the experiment is as follows. First, we downloaded a cat and dog dataset with PNG format. Then we used FFmpeg and libjpeg-turbo to compress the cat and dog image dataset in PNG format into JPEG[9][10]. Libjpeg-turbo is an unofficial reference software as a JPEG codec for fast encoding and decoding using SIMD instructions. FFmpeg is a computer program that records and converts digital audio and video streams into various formats. JPEG compression cannot be done directly from PNG format using libjpeg-turbo. We convert the dataset with PNG format to bmp using ffmpeg before proceeding with JPEG compression. Then, we compressed the dataset in BMP file format into 9 types of quality by using JPEG quantization parameters (QP) at intervals of 10 from QP 10 to QP 90. Fig. 1 is one of the cat datasets compressed by JPEG into 9 types at intervals of 10 from QP 10 to QP 90.

Figure 1. Cat images compressed in JPEG at intervals of 10 from QP 10 to QP 90

3.2. Deep Learning in Cat vs Dog Image Classification

This process mainly includes three procedures: data preprocessing, model training, and test. First, we perform information extraction on all images and resize all images to 64x64 and store their labels as cat or dog. The dataset is then divided into the training set and the test set. Among them, there are 800 images of cats and dogs in the training set, and 200 images of cats and dogs in the test set.

After data preprocessing, we train the training set with a CNN model using the AlexNet architecture in the model training procedure. When each epoch training is completed, we perform 1 epoch test accordingly. Datasets with different compression ratios will be trained and tested for 50 epochs and output a set of results after training and testing every 2 epochs.

4. Experiment Result

4.1. Data Set Preparation

We compressed the cat and dog dataset in PNG format into JPEG in a total of 9 types at intervals of 10 from QP 10 to QP 90. Table 1 shows the sum of the total file sizes in the dataset before and after compression for each JPEG QP and the compression ratio.

Table 1 Compressed size of the dataset .

JPEG QP	Total PNG file size (byte)	Total JPEG file size (byte)	Compression ratio (%)
10	2,649,598,416	59,195,569	2.23
20	2,649,598,416	84,197,464	3.17
30	2,649,598,416	106,147,272	4.00
40	2,649,598,416	129,821,186	4.89
50	2,649,598,416	147,465,452	5.56
60	2,649,598,416	169,000,505	6.37
70	2,649,598,416	202,841,641	7.65
80	2,649,598,416	251,931,403	9.50
90	2,649,598,416	343,886,647	12.97

4.2. Training Environment

The experimental environment conducted in this paper is as follows. It was done with a hardware configuration of Intel Core i7-7700K@4.2GHz, DDR4 32GB, NVIDIA GeForce GTX-750 Ti and 8GB RAM. The software configurations consisted of Python 3.7, Tensorflow 2.3.0, Keras 1.0.8 and CUDA 11.3. The operating platforms are Anaconda 2022.05 and Jupyter Notebook 1.0.0.

4.3 Training Results

In the experiments, each group of JPEG QPs were trained for 50 epochs and output 1 set of results every 2 epochs, for a total of 25 sets of results. Table 2 shows the average losses and the average accuracy of all training for each QP. And all the average accuracy of each QP is about 90 percent.

Table 2 Training average result per QP.

JPEG QP	Loss	Accuracy
10	0.106463	90.14%
20	0.111249	90.18%
30	0.107193	90.94%
40	0.109958	90.46%
50	0.115895	89.94%
60	0.108013	90.35%
70	0.104689	89.44%
80	0.108742	90.73%
90	0.108221	90.35%

And Fig. 2 shows the training accuracy per epoch on QP range 10~90.

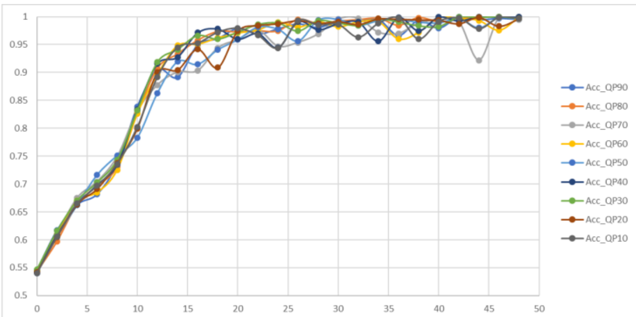

Figure 2. Training accuracy per epoch

4.4 Test Results

Table 3 shows the mean, maximum and minimum values for a total of 25 test result sets collected every 2 epochs for 50 epochs at each QP.

Table 3 Compressed size of the dataset .

JPEG QP	Avg. Accuracy	Max Accuracy	Min Accuracy
10	71.34%	76.25%	54.75%
20	70.71%	76.25%	54.50%

30	72.09%	76.75%	55.00%
40	71.11%	76.25%	55.00%
50	71.60%	76.50%	54.50%
60	72.49%	77.50%	55.00%
70	71.55%	76.50%	54.50%
80	72.25%	77.00%	54.50%
90	70.76%	75.00%	55.25%

And Fig. 2 shows the training accuracy per epoch on QP range 10~90.

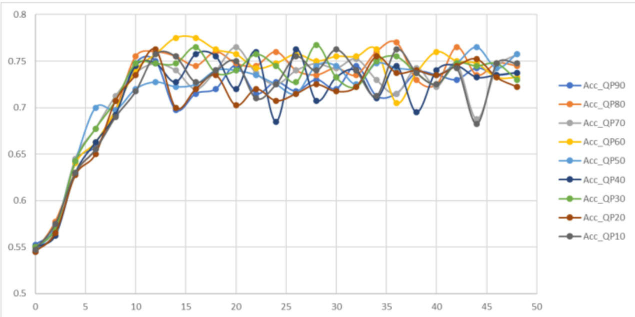

Figure 3. Test accuracy per epoch

5. Conclusion

In the training results, the smallest average loss is 0.104689 at QP 70, and the largest is 0.115895 at QP 50. The largest of the accuracy averages is 90.94 percent at QP 30, and the smallest is 89.44 percent at QP 70. In the test results, the highest accuracy average is 72.49 percent at QP 60, and the smallest is 70.71 percent at QP 20. The largest accuracy value is 77.5 percent at QP 60, and the smallest is 54.5 percent at QP 20, 50, 70, and 80. The accuracy per epoch shows an upward trend around the first 20 epochs and fluctuates slightly after that. However, it can be seen from Fig. 2 and 3 that the accuracy of images with different compression rates does not have much difference and regularity, and the test accuracy is generally lower than the training accuracy. There was no phenomenon in which the accuracy was lowered or the accuracy was increased only in a specific QP. Therefore, we conclude that the compression of JPEG images does not have a significant impact on the accuracy of its deep learning.

From the additional analysis, it can be seen that overfitting occurred in this experiment. However, we did not deliberately use a technique to solve overfitting, such as increasing the number of data through transformation, because we tried to analyze the effect of the image according to JPEG compression on the test result. Assuming that the overfitting problem has been solved, further studies using more data are needed to confirm the effect of JPEG on the deep learning results.

Reference

[1]. The International Cat Association, [Online]. https://tica.org/fi/.

[2]. Fédération Cynologique Internationale, [Online]. http://www.fci.be/en/.

[3]. Jajodia T, Garg P. Image classification-cat and dog images[J]. Image, 2019, 6(23): 570-572.

[4]. Gandor T, Nalepa J. First Gradually, Then Suddenly: Understanding the Impact of Image Compression on Object Detection Using Deep Learning[J]. Sensors, 2022, 22(3): 1104.

[5]. Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.

[6]. Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

[7. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).

[8]. Coates, Adam, Andrew Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning." Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011.

[9]. FFmpeg, Accessed: May 30, 2022. [Online]. Available: http://ffmpeg.org.

[10]. Libjpeg-turbo, Accessed: May 30, 2022. [Online]. Available: https://libjpeg-turbo.or