Study of the Impact of Standard Image Compression Techniques on Performance of Image Classification with a Convolutional Neural Network

Mathieu Dejean-Servières, Karol Desnos, Kamel Abdelouahab, Wassim Hamidouche, Luce Morin, Maxime Pelcat

Study of the Impact of Standard Image Compression Techniques on Performance of Image Classification with a Convolutional Neural Network

Internship Report, IETR-Vaader 2017

Mathieu Dejean-Servières[1], Karol Desnos[1], Kamel Abdelouahab[2], Wassim Hamidouche[1], Luce Morin[1], and Maxime Pelcat[1,2]

[1]IETR, INSA Rennes, CNRS UMR 6164, UBL, Rennes, France
[2]Institut Pascal, Clermont Ferrand, France

December 20, 2017
Version v1.0

Abstract

In this study, we have measured the impact of image compression on the classification performance of Convolutional Neural Networks (CNNs). By using a pre-trained CNN to classify compressed images, we have shown that on average, an image can be compressed by a factor 7, 16, 40 for a JPEG, JPEG200 and an HEVC encoder, respectively, while still maintaining a correct classification by the CNN. This study also showed that pre-trained AlexNet CNN was making use of JPEG artifacts learned during the training phase to perform classification.

To further study the impact of compression on CNN-based classification, a large set of encoding parameters was explored: color-space, resolution, Quantization Parameter (QP). Main conclusions of this study are that color is essential for classification with AlexNet CNN, and that classification is resilient to image downscaling.

Finally, we have studied the correlation between classification performance of a CNN and image quality measured with two objective metrics, namely the Peak Signal to Noise Ratio (PSNR) and the Structural SIMilarity (SSIM). We have found that the SSIM metrics was more appropriate to measure the degradation of an image with regards the CNN performance.

1 Introduction

Recent victory of the AlphaGo program over a professional human Go player has thrown light on the tremendous capabilities of machine learning techniques. In the past few years, machine learning techniques based on artificial neural network have been used to address many technological challenges of our society, like autonomous driving vehicle [12], text recognition [9], or medical diagnosis [10]. Convolutional Neural Networks (CNNs), studied in this report, are a kind of artificial neural networks inspired by the human visual cortex. The main purpose of CNNs is to recognize and classify shapes (objects, characters, animals, ...) in processed 2D images.

In parallel, the era of Internet-of-Things (IoT) has accelerated the trend towards a world where a network of distributed objects and smart sensors communicate to provide new kinds of services. In particular, combining IoT objects with machine learning techniques paves the way towards futuristic applications like smart cities, home automation, and e-health.

Because of the distributed and low-power nature of IoT sensors, processing of acquired data is likely to be offloaded on a remote server where more computational power is available. Hence, acquired data should be compressed before transmission in order to save bandwidth and power. Smart cameras will thus rely on standard image and video compression algorithm (e.g. Joint Photographic Experts Group (JPEG), High Efficiency Video Coding (HEVC), ...) to transmit images to be processed by CNNs. One of the basic principles of these compression algorithms is to compress images by introducing an acceptable loss of image quality in order to transmit less information.

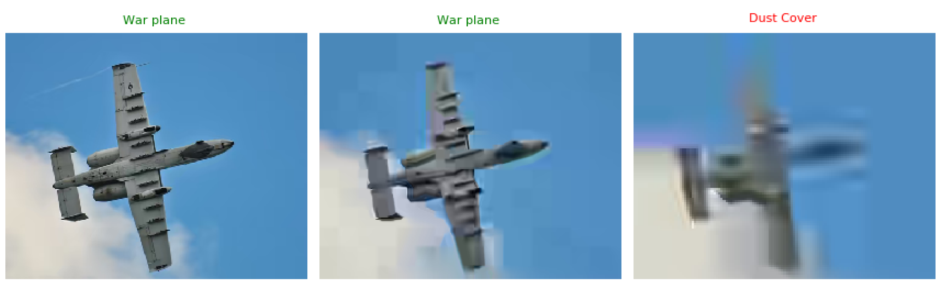


Figure 1: Classification results of an image with different compression ratio.

The objective of this report is to study the impact of quality degradation induced by compression of images on the performance of their classification with a CNN. Figure 1 illustrates this problem with the example of an image compressed with three different parameters, leading to different loss of image quality. As can be seen in this example, classification of the image as a *war plane* is successful for the first two qualities, but fails for the most degraded image.

The steps followed in this study are:

1. Creation of a database of degraded images by applying standard image/compression algorithm to a database of images.
2. Perform classification of the degraded images with pre-trained State-of-the-Art CNNs.
3. Study the influence of compression algorithms on classification performance and model the impact of compression parameters on the efficiency of the classification.

The main objective of this work is to identify which parameters can be used to compress as much as possible images, without losing the efficiency of the classification with a pre-trained CNN.

The report is organized as follows. Section 2 introduces the working principle of CNNs and image compression techniques. Section 3 explains our approach to build the database of compressed images and Section 4 analyses of the classification of the compressed images with a CNN. Finally, Section 5 concludes this reports.

2 Background

2.1 Convolutional Neural Network (CNN)

Deep learning [8] is a machine learning technique whose principle is to teach a task to computers by feeding the algorithm with many input examples and associated "correct" outputs. To be concise, the Artificial Intelligence (AI) will adapt its behavior depending on the data it is fed with, and will thus "learn by itself". In particular, it is important to note that no human intervention is needed to develop and adjust his algorithm. Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

Deep learning computational model is inspired by how the biological neurons work. As illustrated in Figure 2, a neural network is a directed graph where vertices represent the neurons and edges represents channels for transmitting signals between neurons. A neuron usually implements a simple function applied to its input signals to produce an output. Neural networks are usually organized into layers of neurons, where all neurons belonging to layer n only receive signals from neurons of layer $n-1$, and produces signals processed by neurons of layer $n+1$. The first and the last layers of a neural network are called the *input* and *output* layers, respectively. Layers between the input and output layers are called the *hidden* layers of the network. The term "deep" usually refers to the important number of hidden layers in a deep learning neural network.

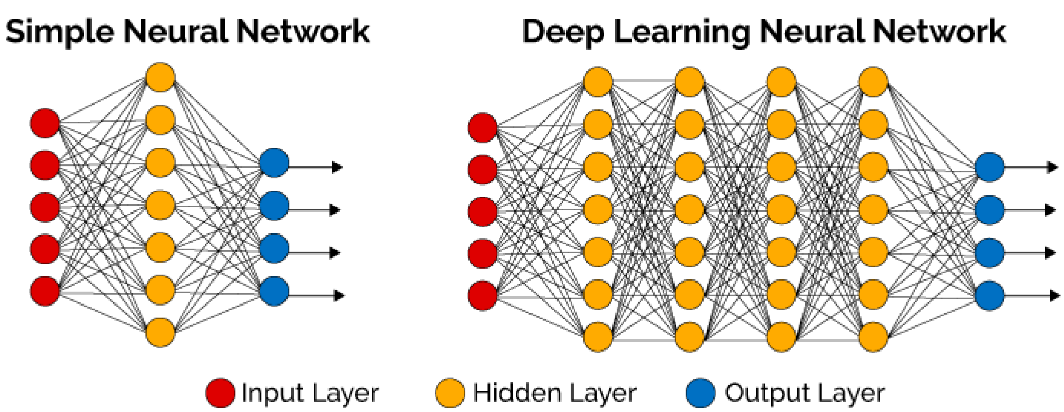


Figure 2: Neural networks, which are organized in layers consisting of a set of interconnected nodes. Networks can have tens or hundreds of hidden layers.

An image classification algorithm is an algorithm whose purpose is to automatically identify the content of an image among a predefined set of image classes. For example, in

Figure 1, the first two images were identified as belonging to the class *war plane*, and the third image was associated to the *dust cover* class.

A Convolutional Neural Network (CNN) [7] is a specialized neural network model inspired by the human visual cortex, specially fitted for image classification. Intuitively, a CNNs alternates convolutional and pooling layers, each processing and producing 2D images. In convolutional layers, neurons produce 2D images by convolving learned 2D features, with input 2D images. In pooling layers, 2D images produced by multiple neurons from the previous layer are mixed, usually using a non-linear function, and the resulting 2D image may be downsampled. Connections between layers can be sparse, typically near input layers, or fully connected, typically near output layers. The convoluted features are not pre-trained, they are learned while the network trains on a collection of images. This automated feature extraction makes CNNs highly accurate for computer vision tasks such as image classification [6, 8].

As shown in [15] some invisible perturbations, for human observers, can prevent the image from being correctly classified by a CNN. A special technique, presented in [11], was specifically developed to produce the smallest perturbations to fool such network. Hence, the question of the resilience of CNNs to artifacts introduced during the compression of the classified image is not trivial. Indeed, although image compression artifact aim at preserving the quality of images for human subjects, they may still introduce imperceptible artifacts that would undermine the efficiency of CNNs.

2.2 Image compression

2.2.1 Used Encoders

In this report, the impact and performances of three different encoders are studied: JPEG, JPEG2000 and Better Portable Graphics (BPG).

JPEG

JPEG [16] is one of the most commonly used algorithm for lossy image compression. In this study, JPEG will therefore be the reference to compare the performance of encoders. JPEG can typically achieve a compression ratio (see Section 2.2.2) of 10 with smalls perceptible loss in image quality [3]. The algorithm is particularly efficient on images with smooth variations of color, like photographies. JPEG uses a lossy form of compression based on a decomposition on the encoded image into 8x8 pixels blocks, and the application of a Discrete Consine Transform (DCT) to each block of pixels. The DCT operation converts each field of the image from the spatial domain into the frequency domain. JPEG then compress information by quantizing high frequencies coefficients. This loss of information is acceptable, as the human psycho-visual system discards high-frequency information like sharp transitions in intensity. The main drawback of JPEG compression algorithm is the "tiling" effects that appears at high compression ratio.

JPEG2000

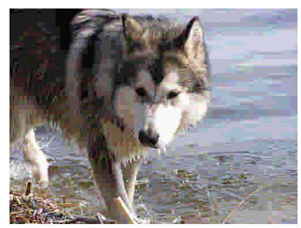
JPEG2000 [14] intends to overcome several of the shortcomings of JPEG such as better compression ratios, compression scalability, and resolution accuracy. JPEG2000 is an evolution of JPEG, where the main differences are the substitution of the DCT with a wavelet-based method with better scalability characteristic, and the adoption of a more sophisticated entropic coding algorithm. Compared to the previous JPEG standard, JPEG2000 delivers a typical compression gain in the range of 20%, depending on the image characteristics.

BPG

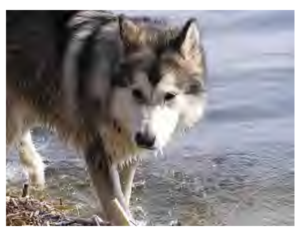
BPG [2] is a lossy and lossless picture compression format based on the Main Still Picture profile (HEVC-INTRA) of the video compression standard HEVC, also known as H.265. It supports grayscale, YCbCr, RGB, YCgCo color spaces with an optional alpha channel. HEVC specifies 33 directional modes for intra-prediction compared with the 8 directional modes for intra-prediction specified by former standard. BPG outperforms JPEG both in terms of compression quality and compression ratio and also avoids the "tilling" effects. BPG seems to be an excellent format for the IoT [1].

Figure 3 illustrates the compression artifacts introduced when encoding an image with each of the studied image encoding algorithm.

(a) Original image, 147.8 Ko



(b) JPEG image, quality 5, 12.4 Ko



(c) JPEG2000 image, quality (100), 5.6 Ko

(d) BPG image, quality 47, 3.2 Ko

Figure 3: Identical image compressed with three different encoders.

2.2.2 Metrics

In this work, objective indicators of the characteristics of compressed images are needed in order to study their impact on image classification.

Compression ratio

Image compression is the process of reducing the amount of data required to represent a given image. The compression ratio C_R measures the compression efficiency of an algorithm for a given image.

$$C_R = \frac{n_1}{n_2} \tag{1}$$

where n_1,n_2 is the number of information carrying units (bits) respectively for the original image and the compressed image.

Quality metrics

Quality metrics are objectives indicators of the degradation of images induced by the different compression algorithms.

As the amount of data used in our study is huge, a subjective visual quality test passed by humans is excluded. Two objective numeric metrics will be used to evaluate the quality loss between the original and the compressed image: the PSNR and the SSIM.

The Peak Signal to Noise Ratio (PSNR) is the most commonly used metric to compare the quality of a compressed with its uncompressed counterpart. The **PSNR** is defined through the Mean Squarred Error (MSE) as follow:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2 \tag{2}$$

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{3}$$

where:

- Y_i represents the pixels of the original image.
- $\hat{Y_i}$ represents the pixels of the compressed image.
- n represents the number of pixels of the image.
- MAX_I represents the maximum pixel value in the original image.

Although the PSNR is the most used quality metric for assessing the efficiency of image compression algorithms, it fails at reproducing the quality evaluation of human beings. For this reason, the Structural SIMilarity (SSIM) metric was introduced in [17], in an attempt to reproduce the subjective the image quality evaluation of the human vision system. The **SSIM** is defined as follow for comparing a compressed image x to the original image y:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4}$$

where:

- μ_x the average of x
- μ_y the average of y
- σ_x^2 the variance of x
- σ_y^2 the variance of y
- σ_{xy} the covariance of x and y;
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator
- L the dynamic range of the pixel-values (typically this is $2^{bits\ per\ pixel} - 1$)
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

3 Contribution

The dataset studied in this work has been classified using Caffe [5] with the pre-trained network `bvlc_alexnet`, as described in [6].

3.1 Compressed Images Dataset

The performance of the selected CNN was measured on images from the validation set used by ILSVRC12 [13]. The validation set ensures that the images have never been seen during the training.

To generate the dataset used for this study, images from the validation set have been compressed with various compression algorithms, and with various encoding parameters (see Section 3.2). The original database containing over 50000 images, it is too big to be analyzed exhaustively in this work. For this reason, a subset of the original database consisting of 55 images, detailed in Appendix 1, was used for this study.

When applying the pre-trained CNN to the selected subset, a top-1 accuracy of 56,3% and a top5 accuracy of 81,8% are achieved, which is representative of the accuracy of the whole dataset (respectfully 57,2% and 80,3% with bvlc_alexnet from the Caffe codebase) [4].

The creation of the dataset of compressed images is fully automated with Python scripts. The user simply gives as inputs the desired encoding parameters and the source images. Because image compression is time consuming, a particular attention was given to avoid recreating several times the dataset of compressed images. For each configuration, corresponding to the compression of a given image with given encoding parameters, the scripts searches if such data is already in the database. If yes, the result is extracted and added to the data to visualize. If not, the data is created and added to the database. Such data can be used for another viewing later with no additional cost computing.

3.2 Encoding Parameters

For each image of the input dataset, 432 variations were created by compressing the image with the following parameters and encoders.

JPEG

The JPEG encoder supplied by `LibJPEG`[1] is used in this study. The parameters used to compress images from the dataset with the JPEG encoder are listed in Table 1.

The scale parameter of the JPEG encoder controls the resolution of the compressed image. Downsampling the original image by a scale of 1/2 means that both the height and the width of the image are divided by 2 (i.e. the number of pixels is divided by 4).

Images encoded with JPEG were compressed either in the default RGB color mode, which actually compresses the image within the YCbCr color-space, or converted to grayscale images before compression.

In the JPEG encoder, the quality parameter, expressed as a percentage, is used to control the quantization process. Hence, this quality parameter is used to control the tradeoff between quality and size of the compressed image. The best image quality is reached with a Quantization Parameter (QP) of 100 and the most compressed image a QP set to 1.

[1] `http://libjpeg.sourceforge.net/`

Scale	Color	Quality
1, 1/2, 1/4, 1/8	RGB, Gray	1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 18, 20, 25, 30, 40, 50

Table 1: Encoding parameters for generation of the JPEG images

JPEG2000

The JPEG2000 encoder supplied by OpenJPEG [2] was used in this study. The parameters used with the JPEG2000 compiler are listed in Table 2.

All the images compressed with JPEG2000 use the default Code Block Size of 64x64, a precinct size of 215x215 and the Irreversible Color Transform (IRC) activated, using the Cohen–Daubechies–Feauveau 9/7 wavelet transform. This choice was made because it was found empirically that changing these parameters was causing a decrease of the quality and compression ratio of the compressed image.

In addition to the image scale and color parameters, already used in the JPEG encoder, JPEG2000 enables selecting the number of quality layers and their associated compression level. For example, "[200,40,20]" means that the compressed image has three quality layers, with a compression ratio of 200 for the first layer, 40 for the second and 20 for the third. The more layers with a small number of compression ratio there are, the better the quality of the compressed picture is.

Scale	Color	Quality Layers
1, 1/2, 1/4, 1/8	RGB, Gray	[200,40,20],[200,40],[200],[100,20,10],[100,20],[100] [50,40,20],[50,40],[50],[25,20,10],[25,20],[25] [15,10,1],[15,10],[15],[10,5,1],[10,5],[10]

Table 2: Encoding parameters for generation of the JPEG2000 images

BPG

The BPG encoder supplied by F. Bellard[3] was used in this study. Parameter used for compressing images with the BPG encoder are listed in Table **??**

The BPG encoder offer a compression level parameter that makes it possible to limit the amount of computing resources and the time used to encode the image. All the images were compressed with a compression level value set to 1, which means that no restriction was given.

Quality of images compressed with the BPG encoder can be controlled using a QP. The best image quality is reached with a QP set to 1, and the most compressed image with a QP set 51.

Scale size	Color Mode	Quantization Parameter (QP)
1, 1/2, 1/4, 1/8	RGB, Gray	30, 35, 36, 37, 38, 39, 40, 41, 42 ,43, 44, 45, 46, 47 48, 49, 50, 51

Table 3: Encoding parameters for generation of the BPG images

[2] `http://www.openjpeg.org/`

[3] `https://bellard.org/bpg/`

4 Results

4.1 Encoders Performance

Each of the following subsections, one for each encoder, presents statistics on the classification performance and on the compression ratio for selected compression parameters.

When classifying an image with a CNN, the CNN produces a list of probabilities. Each probability denotes the likelihood that the classified image belongs to a class learned during the training phase. The expected result of the classification process is that the highest probability corresponds to the true class of the image. For this reason, classes are ranked in order of decreasing probability to identify the most probable classes of a classified image.

The classification performance is measured as the *rank difference* in presented results. The rank difference is obtained by computing the difference between the rank of the true class of an image in the classification results of the original and the compressed image. A negative rank difference means that the classification of the image was deteriorated due to the compression. For each set of encoding parameter, the mean, the median, and the standard deviation of the rank difference were computed on the whole dataset.

4.1.1 JPEG

Table 4 show the results obtained with the JPEG encoder for different encoding parameters. The given parameter sets were selected to highlight how changes in compression resulting in similar compression ratio may have very different classification results. In particular sets of parameters were selected such that:

- For each scale the quality parameter which provides a good compression ratio (not necessary the best one) and either a minimal impact on rank (median and mean near 0 when possible).
- Result where a parameter has a direct influence on the classification, like the Gray color space (see further info in Section 4.3).

Compression Parameters			**Rank Difference**			**Compression Ratio**		
Scale	Color	Quality	Mean	Median	Std Dev	Mean	Median	Std Dev
1/1	RGB	30	0.65	0.00	5.73	2.65	2.89	1.13
1/2	RGB	40	0.73	0.0	19.52	7.70	7.91	3.55
1/2	Gray	50	-15.84	0.0	52.22	22.96	23.53	10.87
1/4	RGB	50	-4.22	0.0	35.49	22.24	22.92	11.05
1/8	RGB	50	-34.06	-2.0	99.36	68.95	68.6	36.13

Table 4: Classification and compression statistics for selected compression parameters with the JPEG encoder

4.1.2 JPEG2000

Table 5 presents classification and compression results with the JPEG2000 encoder, for different parameter sets.

Compression Parameters			**Rank Difference**			**Compression Ratio**		
Scale	Color	Quality	Mean	Median	Std Dev	Mean	Median	Std Dev
1/1	RGB	200,40	-0.98	0.0	11.29	8.11	8.32	4.26
1/1	RGB	50	-1.18	0.0	11.30	510.14	10.36	5.33
1/2	RGB	100,20,10	-1.54	0.0	21.75	8.11	8.28	4.27
1/2	RGB	100,20	-1.07	0.0	26.68	16.19	16.53	8.51
1/2	RGB	200,40,20	-1.38	0.0	27.09	16.24	16.61	8.54
1/4	RGB	15,10,1	-3.84	0.0	34.40	7.52	8.45	3.26
1/4	Gray	15,10,1	-21.87	-1.0	75.59	15.57	16.53	7.19
1/8	RGB	10,5,1	-15.60	-1.0	51.31	25.62	27.16	11.52

Table 5: Classification and compression statistics for selected compression parameters with the JPEG2000 encoder

4.1.3 BPG

Table 6 presents classification and compression results with the BPG encoder, for different parameter sets.

Compression Parameters			**Rank Difference**			**Compression Ratio**		
Scale	Color	QP	Mean	Median	Std Dev	Mean	Median	Std Dev
1/1	RGB	35	0.01	0.0	7.37	8.53	8.23	4.91
1/1	RGB	39	-0.25	0.0	15.16	14.68	13.83	10.02
1/1	RGB	41	-1.61	0.0	16.38	19.63	17.74	14.01
1/2	RGB	30	-0.12	0.0	19.94	15.99	15.58	8.87
1/2	RGB	35	-0.05	0.0	26.67	28.02	26.41	17.13
1/2	RGB	38	-1.07	0.0	24.66	40.46	38.82	23.79
1/2	RGB	40	-2.98	0.0	26.75	52.32	51.19	30.17
1/4	RGB	35	-7.25	0.0	46.42	89.012	86.87	46.91
1/4	RGB	40	-13.94	0.0	75.08	162.11	157.39	87.35
1/8	RGB	30	-27.50	-1.0	83.07	162.81	158.97	83.66

Table 6: Classification and compression statistics for selected compression parameters with the BPG encoder

The BPG performance on highly compression ratio is directly correlated to its ability to preserve the structure of the image as measured with the SSIM metric. Indeed, Figure 4 shows that, when increasing the compression ratio, the SSIM value decreases slower with the BPG encoder than with any other encoder. For clarity, all plots presented in Figure 4 were obtained with a single image of the dataset. Similar plots were observed for other images in the dataset.

4.2 Influence of Scaling on Classification

Figure 5, plots the PSNR as a function of the rank obtained for the true class of compressed images, for different image scales. To avoid overcrowding the the plot, results are only given for two images of the "schooner" class. Blue, orange, pink and green dots corresponds to

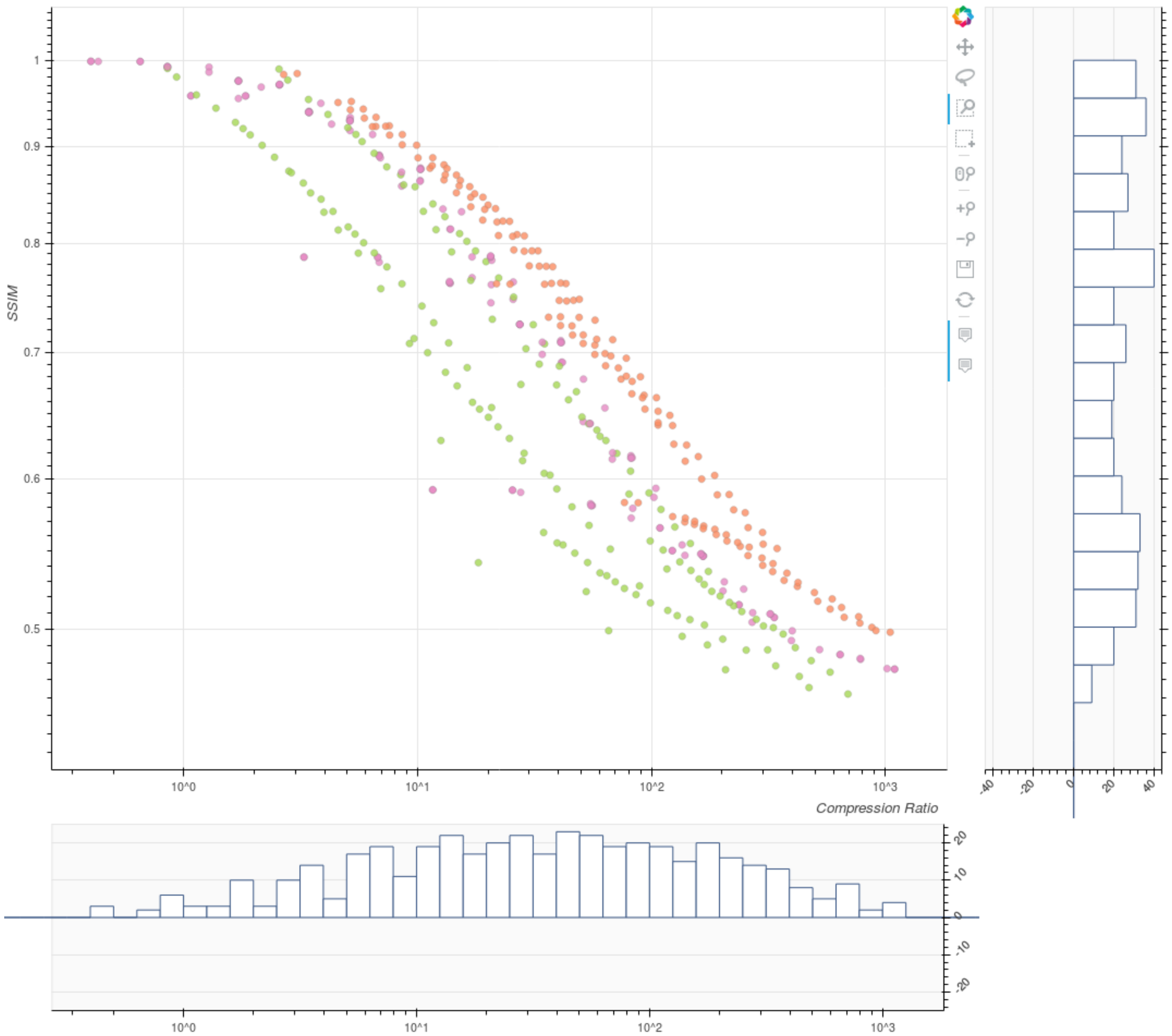


Figure 4: BPG encoding, in orange, preserve more SSIM through compression ratio and thus disturb less the classification than JPEG in green or JPEG2000 in violet.

a scaling of 1, 1/2, 1/4, and 1/8, respectively. The three encoders where used with the parameters described in the tables from Section 3.2.

As can be seen in Figure 5, the blue dots, which represent unscaled images, are mostly aligned at rank 0. This observation shows that even when images are highly compressed by the encoders, with a PSNRs down to 35dB, the images maintain a classification rank between 0 and 5. As revealed by the distribution histogram below the plot, most dots in the plot are stacked at the first classification rank, making it impossible to discern them.

The orange dots, which represent a 1/2 scaling, with light compression are near the rank 0, but if the compression becomes too high, the classification is more likely to fail. Pink dots, representing 1/4 scaling, are less stacked and more stretched horizontally. Green dots, representing 1/8 scaling, are stretched all along, that means that even with low compression, the resolution is too low, here 92x48 and 27x38, to be well recognized by the CNN.

Combining scaling followed by compression is thus possible, but requires a minimum resolution.

4.3 Influence of color on classification

Statistics on the classification performance of compressed images in RGB and GrayScale color modes are given in Table 7. These results were obtained on the complete dataset generated with the parameters presented in Section 3.

	RGB	**GRAY**
Mean Rank	107.3	258.1
Median Rank	0	156
Std Dev	190.0	239.6

Table 7: Classification performance depending on the color mode used

These results show a huge drop of classification performance between RGB and Gray color mode, for equal compression parameters. This shows that the RGB color mode is so absolutely necessary for the used pre-trained AlexNet CNN to get acceptable results of classification.

4.4 SSIM Threshold

Figure 6 plots the SSIM of compressed images as a function of their rank difference compared to the original image. Results are presented for all images belonging to the "Granny Smith" class, and were generated with all three studied encoders.

As can be observed in Figure 6, for each image, there is a threshold for the SSIM metrics from where the image is not longer correctly classified. This threshold is specific to each image and is invariant of the image scale or the used encoder.

These results show that the SSIM evaluation of the image structure is correlated to how CNN performs its classification. The same study conducted with the PSNR evaluations, but no clear correlation or threshold was found, as illustrated in Figure 5.

As can be seen in Figure 6 where only images from the "Granny Smith" are studied, no correlation seems to exist between the SSIM threshold value and the class of the image. Moreover, it is interesting to note that this threshold is not marked for images where the original image is already badly classified by the CNN.

4.5 JPEG over-fitting

In some cases, as illustrated in Figure 7, compression can improve classification rank of an image, leading to a positive rank difference. An example of this phenomenon can be observed with image 1358 from the dataset, where the non-compressed source is badly classified (rank 221). In this example, the improvement is particularly marked for the JPEG encoder, represented with green dots.

In Figure 7, the ordinate axis represents the compression ratio. The higher the compression ratio is, the more the image is compressed, and the more compression artifacts appear. The abscissa axis represents the rank difference with the non-compressed original image. A negative value on this axis denotes a degradation of the classification rank after compression, and a positive value denotes an improvement.

When the compression ratio is moderate, between 10 and 200, the JPEG encoder leads to the best classification results, reaching a rank difference up to 200. The observed improvement of the classification is at odds with previously observed correlation between image quality and classification performance. Indeed, as shown in Figure 4, the JPEG encoder is the encoder inducing the largest degradation of image quality, for identical compression ratio. Since, as shown in previous section, image quality is correlated with classification performance, the improvement of classification with the JPEG encoder is not logical. Our explanation for this JPEG "over-fitting" phenomenon is that, because the used CNN was trained on a database encoded with the JPEG encoder, its training has led the CNN to rely on some the compression artifacts introduced by the JPEG encoder.

This phenomenon was observed in all images in the dataset that were originally badly classified with a rank greater than 50.

Nevertheless, at higher compression ratios, the structure of the images are too strongly altered by the JPEG encoder and the BPG encoder logically produces the best results.

5 Conclusion

5.1 General summary

In this work, we have created a database of compressed images and studied the loss of classification efficiency by a CNN, due to compression artifacts.

Experiments show that on average, images can be compressed by a factor 7, 16, and 40 with the JPEG, the JPEG2000, and the BPG encoder, respectively, with a loss of 1 on classification rank. Thus the BPG encoder outperforms classic compression method in this task.

Experiments also show that classification performance of the CNN was correlated with the quality of the images, as evaluated by the SSIM metric. Interestingly, each image can be associated to an SSIM threshold below which its classification rank drops.

Finally, this study shows that because its training set consists of image encoded with JPEG encoder, the used CNN partially relies on JPEG artifacts to better classify images.

5.2 Future Work

The future work will focus first on extending the database with more images, for example by using additional encoders, such as the Portable Network Graphics (PNG) encoder, or by studying new image databases.

A further step could be to train the CNN specifically for compressed data before evaluating its efficiency.

Another direction for future work is to use different encoders for the images in the training set in order to build a more robust CNN, by decreasing the JPEG "over-fitting" phenomenon.

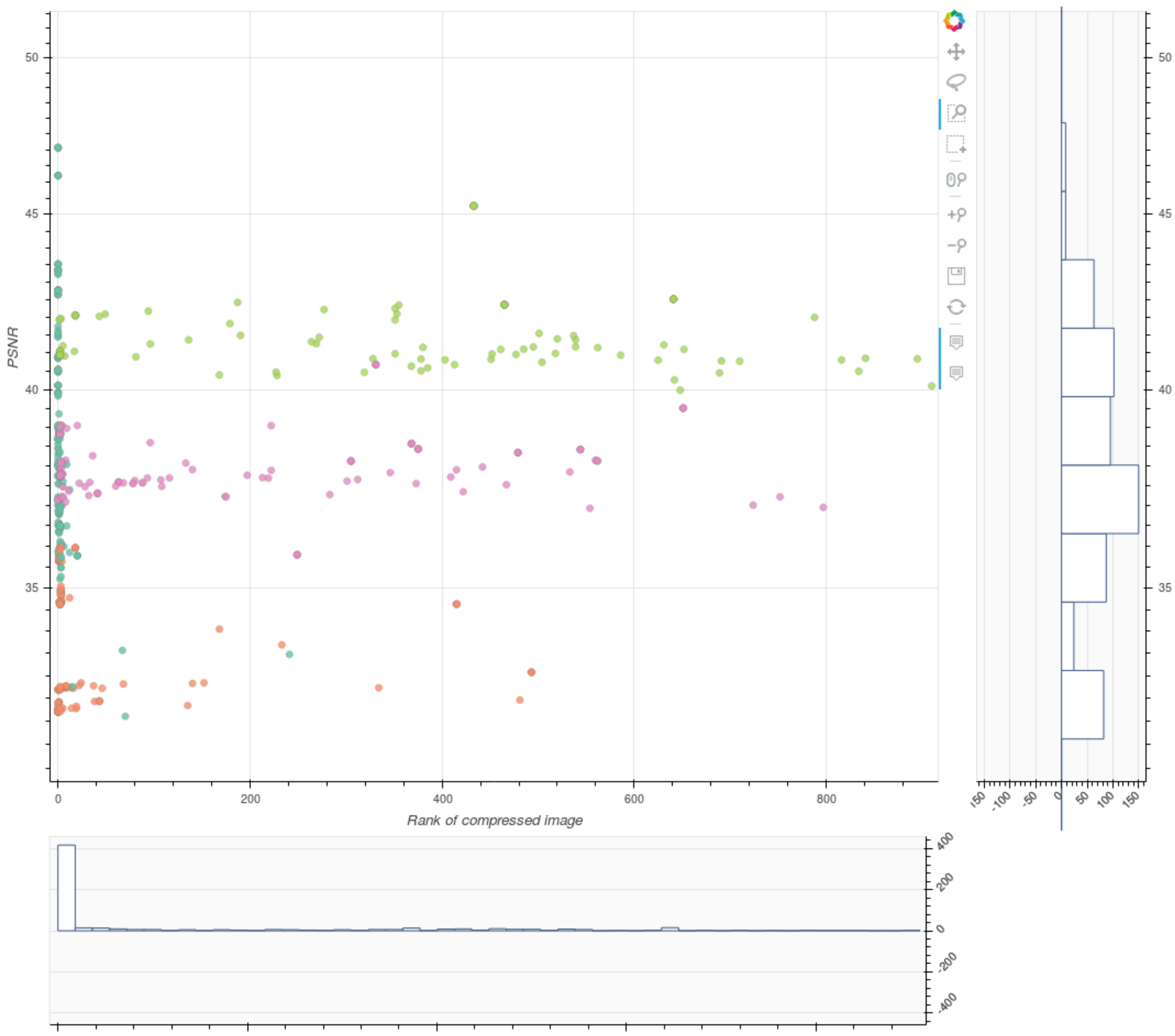


Figure 5: Influence of scaling of two images of class ”schooner” on classification rank. Blue is no scaling, orange scaling 1/2, pink scaling 1/4, green scaling 1/8

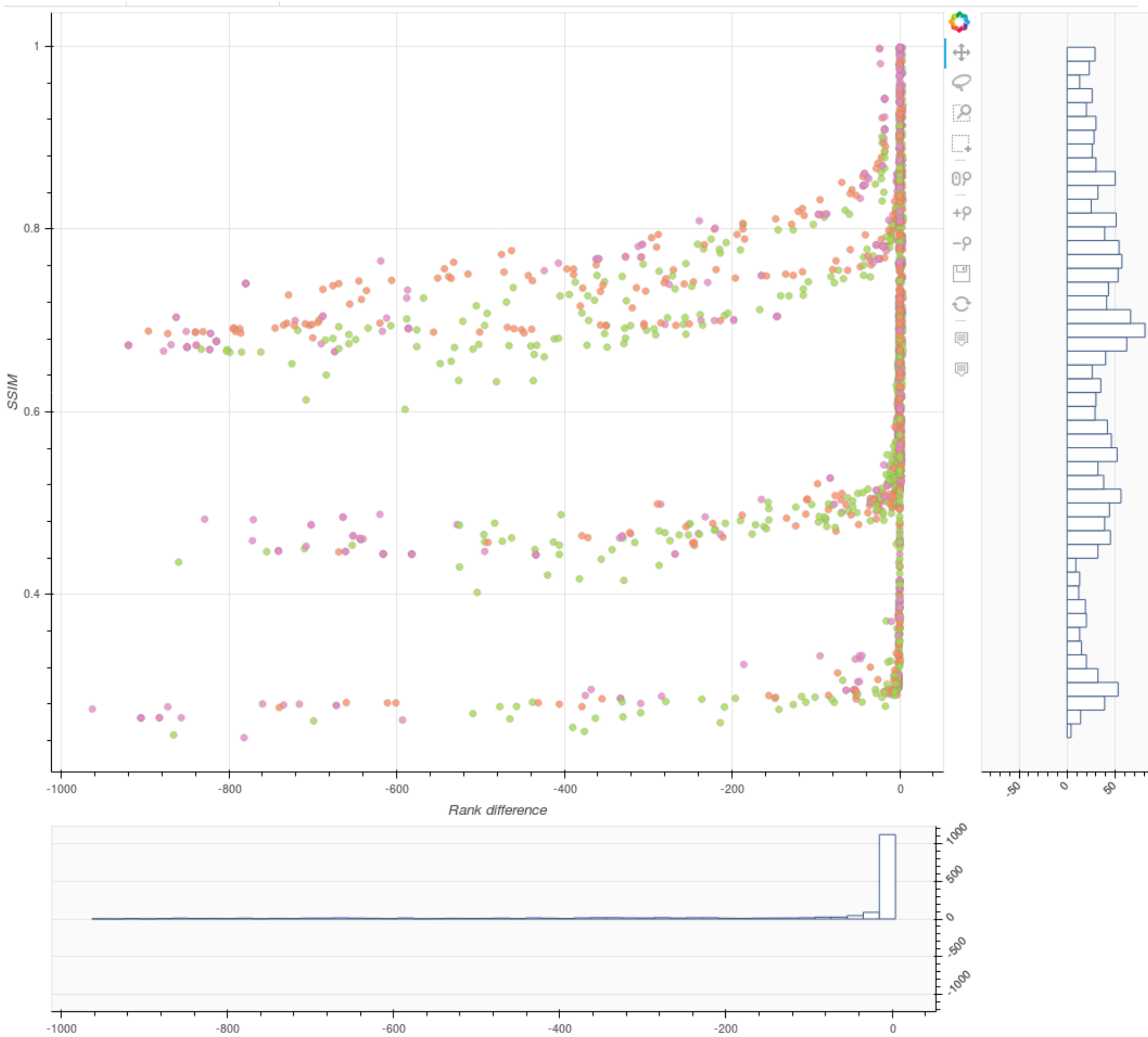


Figure 6: Images belonging to the class “Granny Smith”. Green is JPEG, Orange is BPG, Pink is JPEG2000

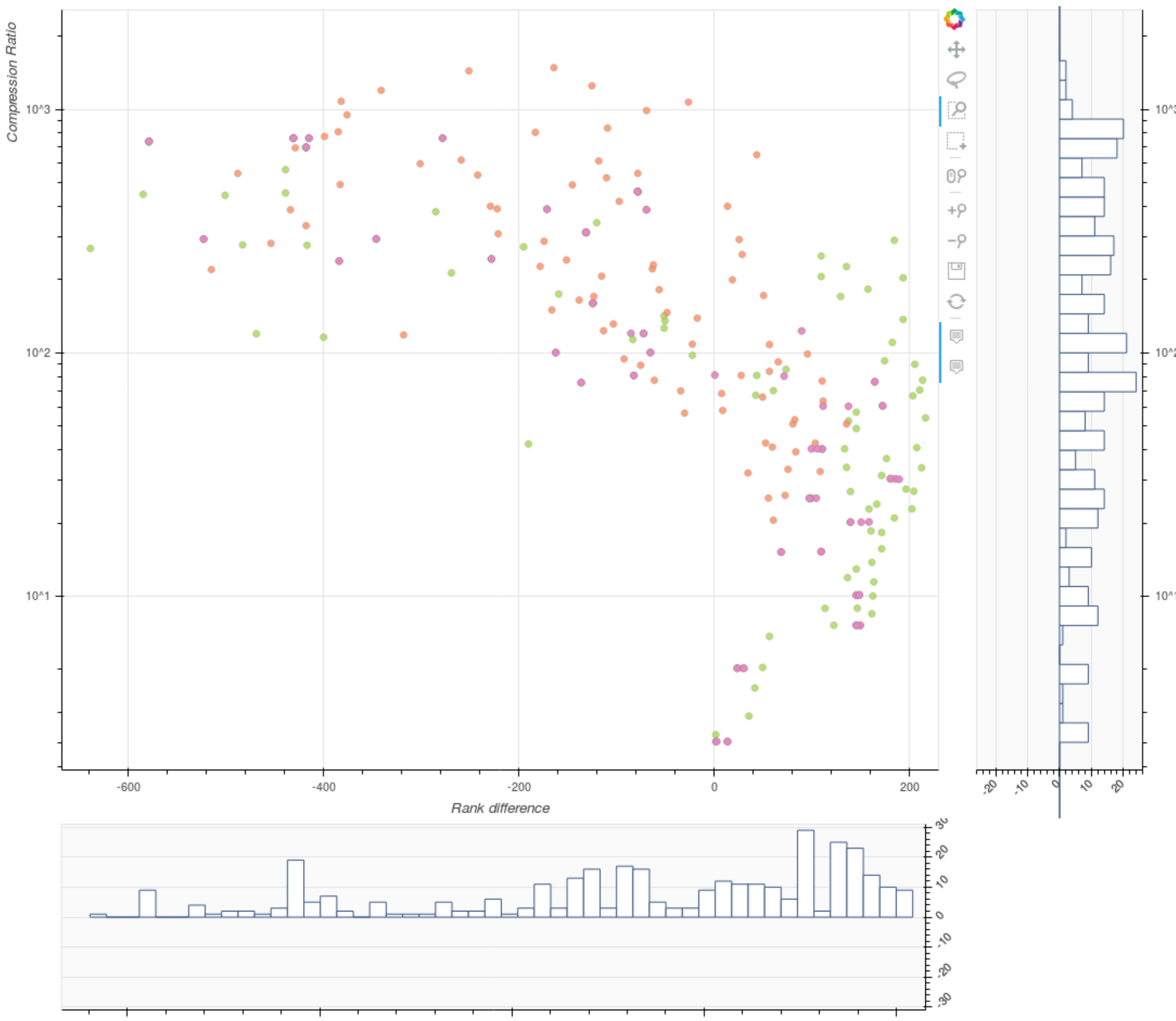


Figure 7: Overfitting on image 1358, JPEG compression in green has the best positive shift over JPEG2000 in violet and BPG in orange

Bibliography

[1] U. Albalawi, S. P. Mohanty, and E. Kougianos, "Sbpg: A secure better portable graphics compression architecture for high speed trusted image communication in the iot," in *Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), 2016 17th International Conference on*. IEEE, 2016, pp. 1–5.

[2] F. Bellard, *The BPG Image Format*, accessed the 2017-09-30.

[3] R. F. Haines and S. L. Chuang, "The effects of video compression on acceptability of images for monitoring life sciences experiments," 1992.

[4] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[7] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, June 2004, pp. II–97–104 Vol.2.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[10] S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural networks*, vol. 8, no. 7, pp. 1201–1214, 1995.

[11] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *CoRR*, vol. abs/1511.04599, 2015. [Online]. Available: http://arxiv.org/abs/1511.04599

[12] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[14] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg 2000 still image compression standard," *IEEE Signal processing magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

[16] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

Appendix 1: Selected Dataset

Truth Label	File name	Truth label rank	File size (Ko)
n07742313 Granny Smith	00000023	0	47,5
n07742313 Granny Smith	00043125	0	34,0
n07742313 Granny Smith	00046567	0	71,6
n07742313 Granny Smith	00046718	0	33,1
n07742313 Granny Smith	00049370	0	17,8
n04147183 schooner	00000094	0	116,7
n04147183 schooner	00049923	0	11,3
n02981792 catamaran	00001063	1	132,7
n02981792 catamaran	00001171	1	84,9
n02981792 catamaran	00004218	4	201,4
n02981792 catamaran	00043109	0	145,7
n04483307 trimaran	00005196	1	309,1
n04483307 trimaran	00005581	0	98,5
n03041632 cleaver, meat cleaver, chopper	00003462	109	98,1
n03041632 cleaver, meat cleaver, chopper	00048820	23	143,9
n03041632 cleaver, meat cleaver, chopper	00008526	45	141,0
n03041632 cleaver, meat cleaver, chopper	00004379	55	62,0
n07802026 hay	00042652	14	214,9
n07802026 hay	00043120	0	214,5
n01871265 tusker	00000067	0	279,2
n01871265 tusker	00002625	1	213,6
n01871265 tusker	00002998	0	120,3
n01871265 tusker	00001358	218	142,4
n01871265 tusker	00000754	0	208,3
n02504013 Indian elephant, Elephas maximus	00000597	1	584,7
n02504458 African elephant, Loxodonta africana	00033235	1	36,8
n02504458 African elephant, Loxodonta africana	00048781	1	31,7
n02089973 English foxhound	00000028	2	201,7
n02107574 Greater Swiss Mountain dog	00000531	1	103,2
n02088364 beagle	00029932	0	191,1
n02111277 Newfoundland, Newfoundland dog	00043152	0	116,7
n02115641 dingo, warrigal, warragal, Canis dingo	00000785	9	122,0
n02120505 grey fox, gray fox, Urocyon cinereoargenteus	00048756	0	98,1
n02119022 red fox, Vulpes vulpes	00000517	1	143,3
n07742313 n02115913 dhole, Cuon alpinus	00001487	0	125,2
n02134418 sloth bear, Melursus ursinus, Ursus ursinus	00001745	11	9,5
n03482405 hamper	00003890	30	75,5
n04447861 toilet seat	00004573	57	175,5
n04591157 Windsor tie	00004858	53	50,8
n02992211 cello, violoncello	00005046	0	131,1
n03187595 dial telephone, dial phone	00005223	0	173,5
n01833805 hummingbird	00027986	0	189,7
n04039381 racket, racquet	00028105	0	112,8
n02423022 gazelle	00029641	0	126,5
n02110063 malamute, malemute, Alaskan malamute	00029783	0	147,8
n04552348 warplane, military plane	00030069	0	88,8
n02892201 brass, memorial tablet, plaque	00031164	0	198,0
n07753113 fig	00032827	0	170,8
n01580077 jay	00032831	0	144,6
n07714571 head cabbage	00033354	1	170,3
n03125729 cradle	00038941	1	117,2
n02086910 papillon	00040587	0	225,7
n01770393 scorpion	00048800	0	104,5
n02951358 canoe	00043133	0	130,0
n02071294 killer whale, killer, orca, grampus, sea wolf, Orcinus orca	00048861	0	156,0