Notes and Analysis of Image Compression

An individual report for GM2 Technology for the Poorest Billion
by Roshen Sidhu rs2248 2023

Contained in these notes are a description of the files within the Image Compression folder of this repository and an analysis of how image compression with the JPEG algorithm affects a trained convolutional neural network's accuracy in classifying papilledema from images of the fundus. It is kept in mind that for use in developing countries, image acquisition and transmission methods used must be balanced with cost. The Q=80_99.27 model has been trained and obtained an accuracy of above 97% for images with a JPEG compression quality of 20 and above. This opens the discussion of the compression levels that can be used, how image resolution may affect a model's classification accuracy, and features that may be necessary for classification.

Contents

1. Repository Contents

1.1 Data Folder

The data folder contains all the training data used to train the models. The folder is divided into the different compression levels performed by the "Image Compression" file using the JPEG algorithm on the original dataset. The original folder contains all the original images from the dataset found here: https://osf.io/2w5ce/ [1]. All the folders are then divided into the classes of Normal, Pseudopapilledema, and Papilledema where the images are contained.

1.2 Images to Evaluate Folder

This folder contains images that will be used by the "Use Model to Evaluate Images" file. Place images in this folder and change the file path in the "Use Model to Evaluate Images" file to obtain the prediction of a model.

1.3 Information Folder

1.3.1 Model Testing Information Folder

This folder contains the confusion matrices from all the models when they were tested on the test sets from all image quality levels. The plots are named (Model) on (Dataset), where the models are named (Training Dataset)_(Accuracy on the test set from the same dataset).

1.3.2 Model Training Information Folder

This folder contains the training and validation loss and accuracy plots of the models trained using the "Model Training" and "Model Training Script" files.

1.3.3 Compression Size

This file contains the average size of the images of different quality levels in bytes.

1.3.4 Test_paths

This file contains the relative path of images in the test set. This was kept constant for all models by using a seed when creating the training, testing, and validation sets during model training. This file is used by the "Testing Model Accuracy" file.

1.3.5 Test_results

This folder contains the accuracy results for the different models when tested on the different qualities of images.

1.4 Models Folder

This folder contains all the models that have been trained by the "Model Training" or "Model Training Script" files. The models are named according to the dataset they were trained on (D) and the accuracy rating when tested on the test set from the same dataset (T), D_T.

1.5 Image Compression

This file compresses the original images in the "Data\Original" folder using the JPEG algorithm for different quality levels. It also contains a module to obtain the average file size for each quality level. It saves the average file size as an Excel file "Compression Size" under the "Information" folder.

1.6 Model Training

This file contains the code that can be used to train an EfficientNetB3 model on a chosen dataset for a specified number of epochs, learning rate, and dropout rate. It saves the training model according to the dataset they were trained on (D) and the accuracy rating when tested on the test set from the same dataset (T), D_T in the "Models" folder. The training and validation loss and accuracy plots for the model are saved under the "Information\Model Training Information" folder. This file is modified from https://www.kaggle.com/code/gpiosenka/papilledema-f1-score-98. A key change is the removal of trimming the test set. This is a form of undersampling utilized to counter class imbalances. The ratio of 5:2:2 of Normal to Pseudopapilledema to Papilledema images does not constitute a significant class imbalance (such as 1:100), leading to greater accuracy achieved by the model due to more training images being used when trimming is removed.

1.7 Model Training Script

This file is a modified version of the "Model Training" file that loops through the different quality levels within the "Data" folder and trains a model on it for 20 epochs.

1.8 Testing Model Accuracy

This file tests all the models in the "Models" folder with the test set images from all the different quality levels in the "Data" folder. The path of the test set images is obtained from the "Test_paths" file in the "Information" folder. The test set is kept constant by using a seed when creating the training, testing, and validation set splits. As such, the test images have not previously been seen by the models. The confusion matrices of the models when tested on all the images are saved in the "Information\Model Testing Information" folder. The accuracy of the models for each quality level is saved in the Excel file "Test_results" under the "Information" folder. The results are then analyzed in another module in the file.

1.9 Use Model to Evaluate Images

This file reads an image in the "Images to Evaluate" folder and uses a selected model from the "Models" folder to obtain the prediction of the model for the given image.

2. Training a Model

A Model accuracy of 99.27% has been obtained when trained on the Original dataset for 20 epochs with the following model type, parameters, and image augmentation techniques.

2.1 Model Selection

Convolutional Neural Networks (CNNs) can be trained to perform classification using images of the back of the eye. Generally, as the complexity of the model increases, which can be quantified as the number of parameters, the accuracy of the model increases [2]. In the context of a cheap and handheld device, the computational power and hardware required to run and store the model need to be considered. The EfficientNetB3 model was selected as a medium-sized model in the EfficientNet family.

2.2 Model Parameters

The EfficientNetB3 model used for training was based on a model that was pre-trained on the ImageNet dataset and its base layers were set to trainable. The dropout rate was set to 0.4, and the learning rate is relatively low at 0.001 to prevent over-training.

2.3 Image Augmentation

Image Augmentation was applied to the training set when the model was trained. This results in horizontal flips, height and width shifts, zoom, and rotation being applied randomly to the training images when a training batch was produced. Image Augmentation allows for more variation in the training data to allow the model to be more generalizable from a limited dataset.

3. Image Quality

Potential bottlenecks in obtaining the images include having a low image resolution by using a lower resolution camera to obtain the images in the interest of decreasing the cost of the device, and compression that may be required to transmit the images so that they can be analyzed or stored offsite.

3.1 The JPEG Compression Algorithm

The JPEG compression algorithm is a form of lossy compression where information from the original image is lost during the compression process. This is done by the quantisation of high-frequency components within the image resulting in a loss of finer details within the image[3]. The JPEG compression algorithm can be utilized to compress images leading to a significant size reduction and simulating lower-resolution images. Analyzing the performance of a trained CNN model on the compressed images can lead to information relating to acceptable image resolution and compression levels for a specified model accuracy.

3.2 Image Size and Transmission

In some developing countries, SMS may be the only reliable means of transmitting information. SMS utilizes the GSM 03.38 encoding scheme that can send messages of up to 160 characters. The sizes of the images significantly decrease as the quality level of the images decreases. It was stated initially that the project team was considering transmitting the images as hexadecimal. Even at the highest compression level, 20 messages are still required to transmit an image. Having such a large number of messages needed to transmit the images may not be desirable due to the increased cost associated. One option is to perform a base change to utilize the entire GSM 03.38 encoding scheme. This involves a change to base 128 that halves the number of messages needed to transmit an image as compared to hexadecimal. Another is to consider if MMS or other methods of transmission can be utilised instead of SMS. MMS can allow for images up to the size of MegaBytes to be sent without hassle. Images in the original dataset that were about 20KB.

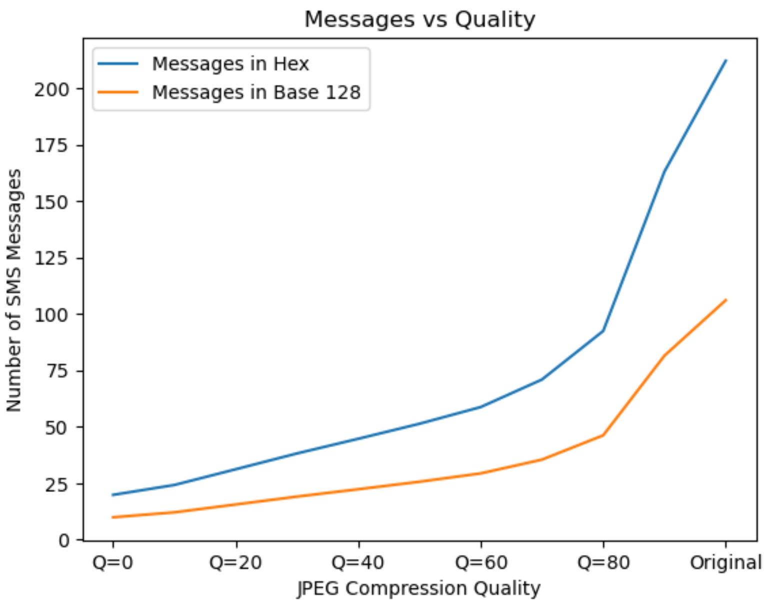


Figure 3.2.1: SMS Messages needed to transmit Images

4. Models Trained on Compressed Images

4.1 Training and Testing Methods

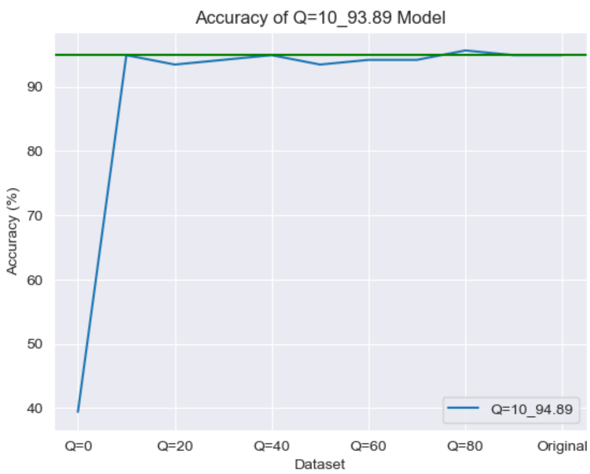
To train a model using a particular image compression level, the ”Model Training Script” file was used. The models were trained using the same parameters (described in 2.Training a Model), with only the compression levels of the images used for training and validation being different. A seed was placed into the Train_Test_Split function to keep the test images the same for all the models. This ensured that the test images were new to all the models when testing for accuracy was performed. The models were trained for 20 epochs.

4.2 Results

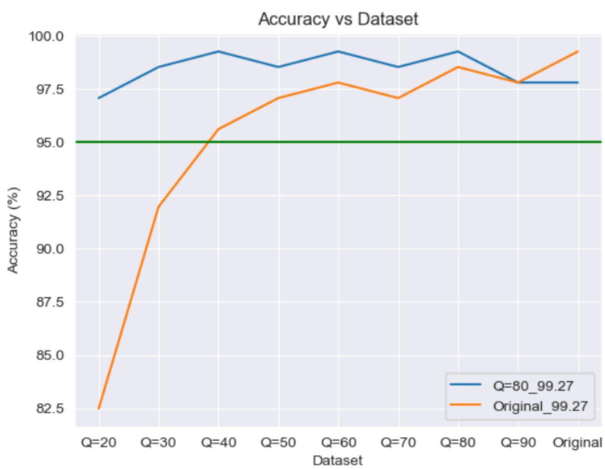
Figure 4.2.2 shows all the models that were trained and the accuracy they achieved when tested on images with different JPEG compression quality levels. Interestingly, the model trained on the images with a JPEG compression quality of 0 was able to obtain an accuracy of 91.24% but this did not generalize well to the other images. None of the models were able to obtain an accuracy of above 95% for the images with a JPEG compression quality of less than 20. Across the images with a quality of 20 and above, the model trained on the quality level of 80 (Q=80_99.27) had the highest accuracy of all the models except for the model trained on the original images for the original images as seen in Figure 4.2.1b. The minimum number of messages to obtain accuracy percentages between 97.08 and 99.27 are shown in Table 4.2.1.

Q=80_99.27 Model Accuracy (%)	Number of Messages to Transmit an Image in Base 128
97.08	16
98.45	20
99.27	23

Table 4.2.1: Minimum Number of Messages to Achieve a Given Accuracy



(a) Model Trained on Images with JPEG Compression Quality = 10



(b) Model Trained on Images with JPEG Compression Quality = 80

Figure 4.2.1: Highlighting Selected Models

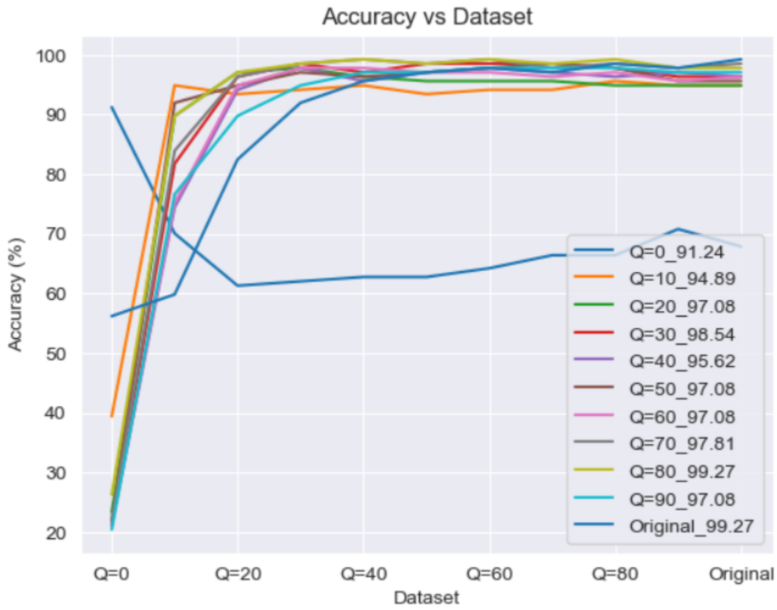


Figure 4.2.2: All Models Trained on Images with Different JPEG Compression Quality

4.3 Implications

Achieving an accuracy of over 95% for the images between the JPEG compression qualities above 20 as well as how the accuracy of the model trained on images with a JPEG compression quality of 10 fluctuates around 95% (as seen in Figure 4.2.1a) indicate that the frequency components preserved in this range of compression might be necessary for classifying papilledema. As such it might be possible to have significant compression of the images or to have the images be taken with a low-resolution camera to be used for classification by a trained model. Investigations can be done to identify features that are preserved in this range of JPEG compression.

4.4 Limitations

The results obtained regarding the accuracy of the models when trained on the compressed images may not generalize to other model types that may be selected due to cost or computational power constraints. It should also be noted that using JPEG compression alone to represent lower-resolution images may be insufficient leading to overconfidence in the model's ability to perform. To better simulate lower resolution images, simulating lens distortion, blur, noise, and colour distortion can be applied to the images simultaneously with compression. Verification of the camera resolution needed for the models to be accurate may only be accomplished when testing with those cameras is performed.

Another limitation is that the models were only trained for 20 epochs. This was due to limits on time and computational power. Most of the models had a decreasing validation loss which would indicate that the models were not yet overfitting to the training data and can be trained further to improve their accuracy.

Lastly, the image-capturing process used ought to closely resemble the images within the dataset in order for the model to be effective. To create a more robust model, more datasets can be utilized for training and testing.

Bibliography

[1] Ungsoo Kim. *Machine learning for Pseudopapilledema.* Aug. 2018. DOI: 10.17605/OSF.IO/2W5CE. URL: osf.io/2w5ce.

[2] Mingxing Tan. *EfficientNet: Improving accuracy and efficiency through AutoML and model scaling.* May 2019. URL: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-%20and.html#:~:text=Accuracy%5C%20Comparison.,than%5C%20the%5C%20best%5C%20existing%5C%20CNN.

[3] Technote. *How does the JPEG compression work?* Sept. 2011. URL: https://www.image-engineering.de/library/technotes/745-how-does-the-%20jpeg-compression-work#:~:text=The%5C%20JPEG%5C%20compression%5C%20is%5C%20a,setting%5C%20(or%5C%20chose%5C%20presets)..