Image compression and ML Report

Jose Miguel Alfaro Castillo

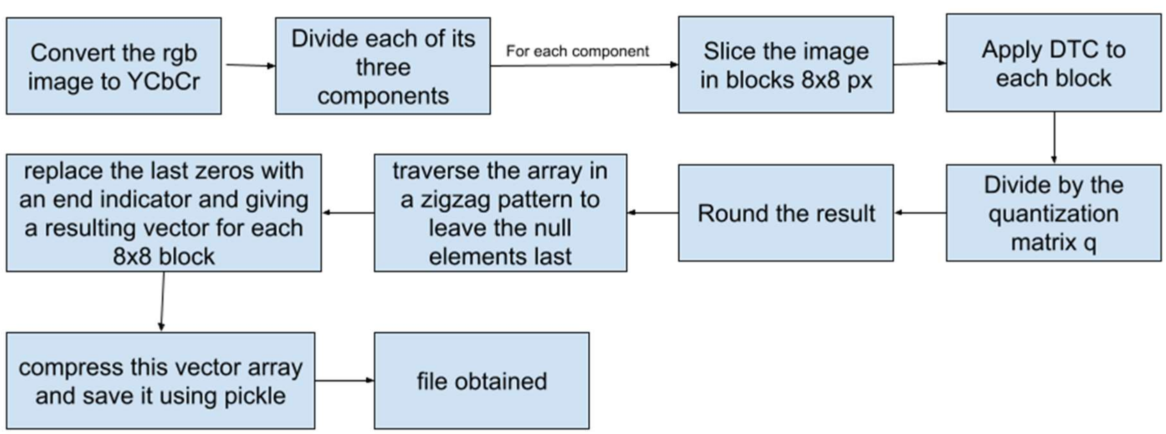
Introduction:

This TP was divided into three main parts, which were: finding and selecting an image compression algorithm using DHT; training different machine learning algorithms to classify handwritten numbers from 0 to 9; repeating the process after preprocessing the images with HOG (Histogram of Gradient Feature); and then comparing these two results to find which one performs better. Finally, the last part discusses the process of generating a mosaic of thumbnails, starting with a large image. Three possible processes will be considered, varying the characteristics taken by each block of the image and also the way of calculating the distance between each block of the image and the available thumbnails.

Methodology:

1. Image compression with DHT:

The first step was to define the process for compressing an image using the DHT algorithm. This process is illustrated in the following image:



The decompression process is the reverse of this, using the same q that was used previously.

To provide a basis for comparison, this algorithm was initially performed with a default q that is used to compress JPEG images.

Esta es la matriz de cuantización de jpeg:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

The reason why the values increase as rows and columns increase is because this matrix seeks to generate as many zeros as possible when the DHT coefficient matrix is divided.

In order to find a q that achieves better results than what we will call q_JPEG, a random experiment is proposed where a q matrix is first defined as follows:

$Q(i,j)=a+b(i+j)$

During 200 iterations, a will take a random value between 3 and 100, and b will take a random value between 1 and 40. With each of these values, the compressed image called q_a{a}_b{b} is generated, we calculate its size, decompress it, obtain the RGB image by performing the opposite process to that shown for compression, and this image is comparable with the JPEG-compressed image and with the source image. In order to find out if, using this method, there are some values of a and b that achieve a better compression and quality ratio, taking as a reference the size and the difference between the original image and the compressed image measured by the PSNR indicator.

2. Image classification:

To carry out this point, 70,000 labeled images were obtained from mnist_784. The aim was to train several machine learning models and find the most optimal one for classifying this database.

The first step is to normalize the data so that it is between 0 and 1, ensuring that no classification algorithm has problems with its procedure. To train this data, three of the methods seen in class were used: KNN, decision tree, and SVM. The aim is to compare their performance.

Each of these machine learning models is trained and evaluated using K-fold cross-validation divided into 5, and the results of each are weighted to compare them.

As a second part of image processing, the use of HOG (Histogram of Gradient Feature) is proposed. Initially, in order to understand how it works, it was applied to one of the available PPM images, and both the image weight histogram and an image illustrating the HOG for each of the blocks are shown.

Preprocessing with HOG is then applied to all data from mnist_784, and model training is repeated using these HOG features rather than all the image data.

The performance of each model is compared after using HOG to determine whether there is an improvement in model accuracy.

Finally, we are asked what we can expect to happen if the process is carried out with images compressed with JPEG QF=75. The most likely outcome is that the images will be classified less accurately when they are pixelated because too much information is cropped out. To be sure of this, the test will be carried out with the data and the training process will be repeated to find the answer.

3. Large Mosaic:

The idea is to generate a mosaic of small images, using a large image as a reference, so that the small images can be identified without losing sight of the overall picture.

The first step is to select a large image and choose a size to replace the image with blocks, keeping in mind the goal of conveying the general idea of the large image without losing sight of the smaller ones. Therefore, a block size of 32x32 was selected, which is the current size of the small images. The large image is resized so that all its pixels can be replaced more easily.

The next step is to select the characteristics that will describe the small images. Three different measurements will be taken: initially the rgb average, secondly the HSV histogram, and finally a method that weights these.

The next step is to select the distance measurement that will be used to find the thumbnail closest to a given block of the image. Two distances will be taken into account: Euclidean and cityblock.

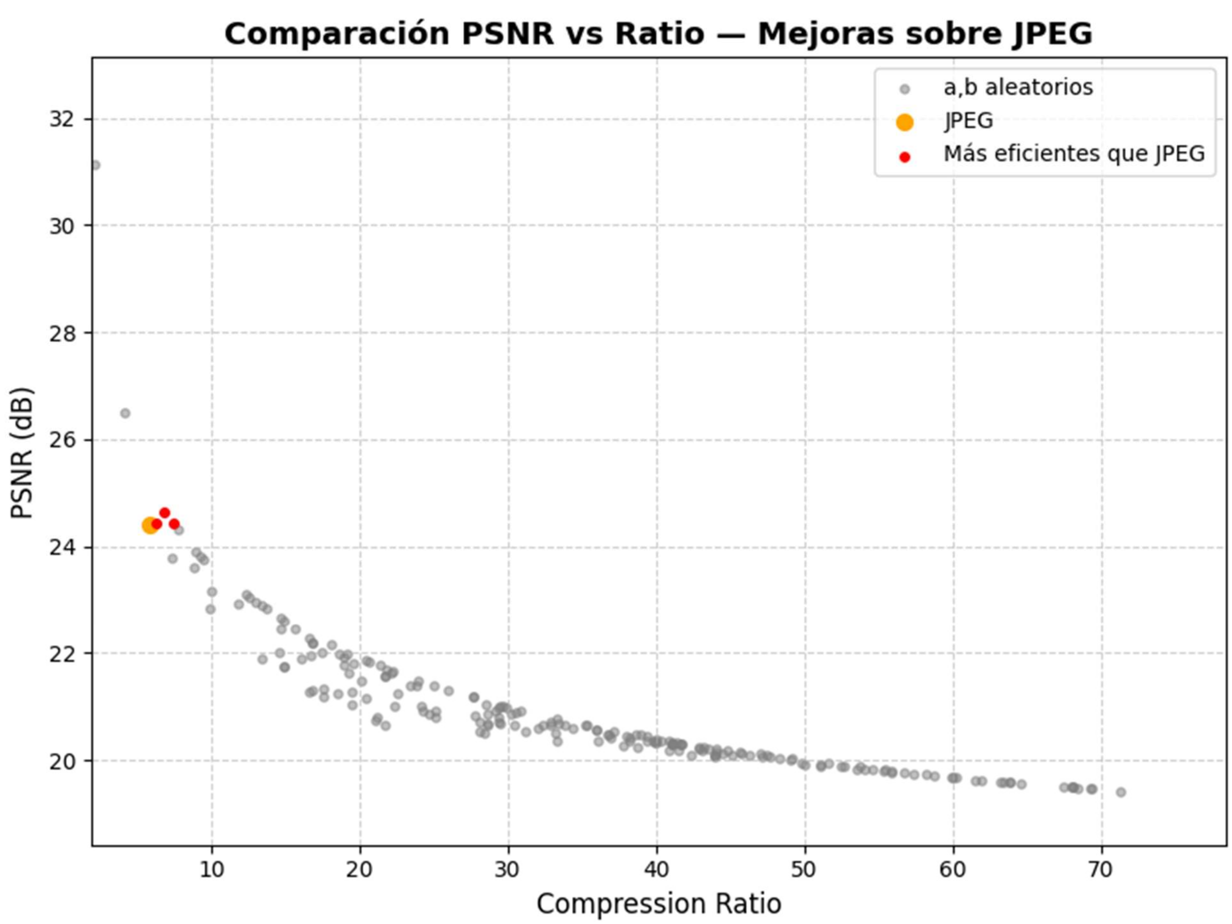
Finally, the three results obtained will be compared to draw conclusions about how the characteristics taken and the distances can influence performance.

Analysis of results:

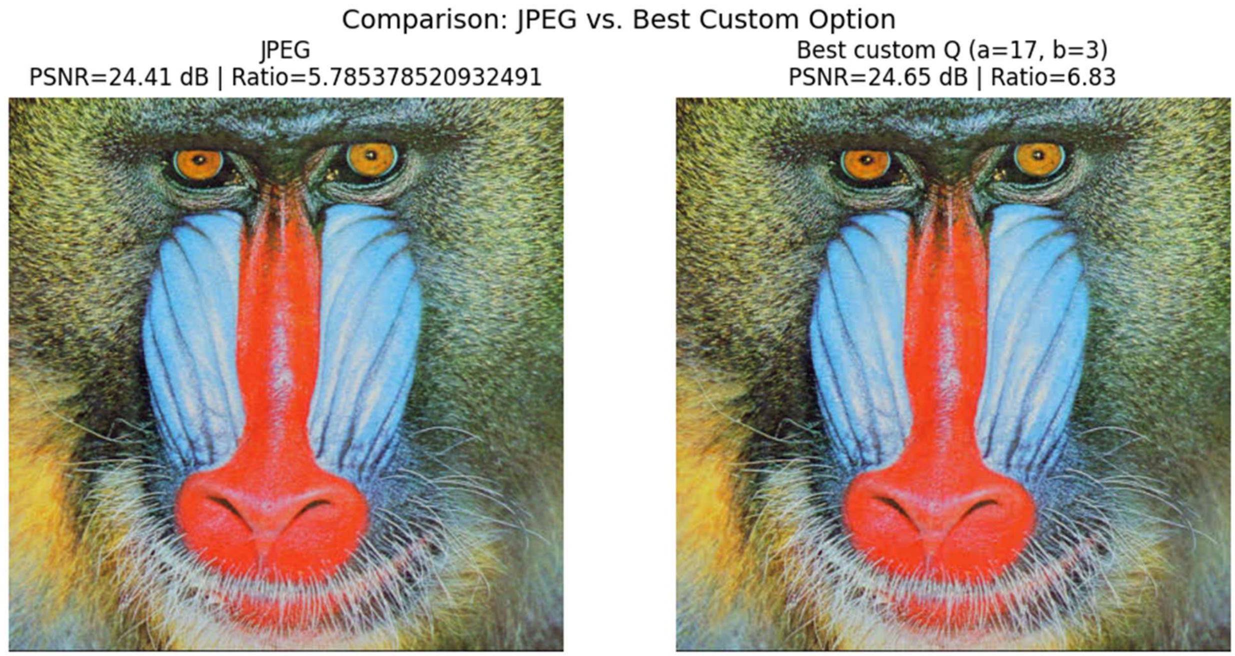
1. Image compression with DHT:

To consider any of the results more efficient than JPEG, they must achieve a higher compression rate than JPEG, while the image quality measured with PSNR must be equal to or better than JPEG, or the image quality must be better than that of JPEG with equal or greater compression.

Taking this condition into account, 200 random tests were performed, yielding the following results:



As can be illustrated, three images with different values of a and b were found that achieve better performance than JPEG. This is based on the PSNR indicator, which is a mathematical indicator. The following comparative image serves as a visual aid.



At least at first glance, we can say that there are no significant differences, but the image on the right manages to compress the initial image, making it lighter and obtaining a better PSNR.

2. Image classification

Model	**Average Accuracy**	**Standard Deviation**
KNN (k=3)	0.971217	0.001026
Decision Tree (Gini)	0.870100	0.001932
SVM (RBF)	0.983133	0.000602

The three models tested allowed MNIST images to be classified directly without extracting features.

The RBF kernel SVM achieved the best average accuracy (97%), followed by KNN (96%), while the decision tree (87%) gave lower results but was faster to train.

HOG would then be used, which looks like this for a PPM image

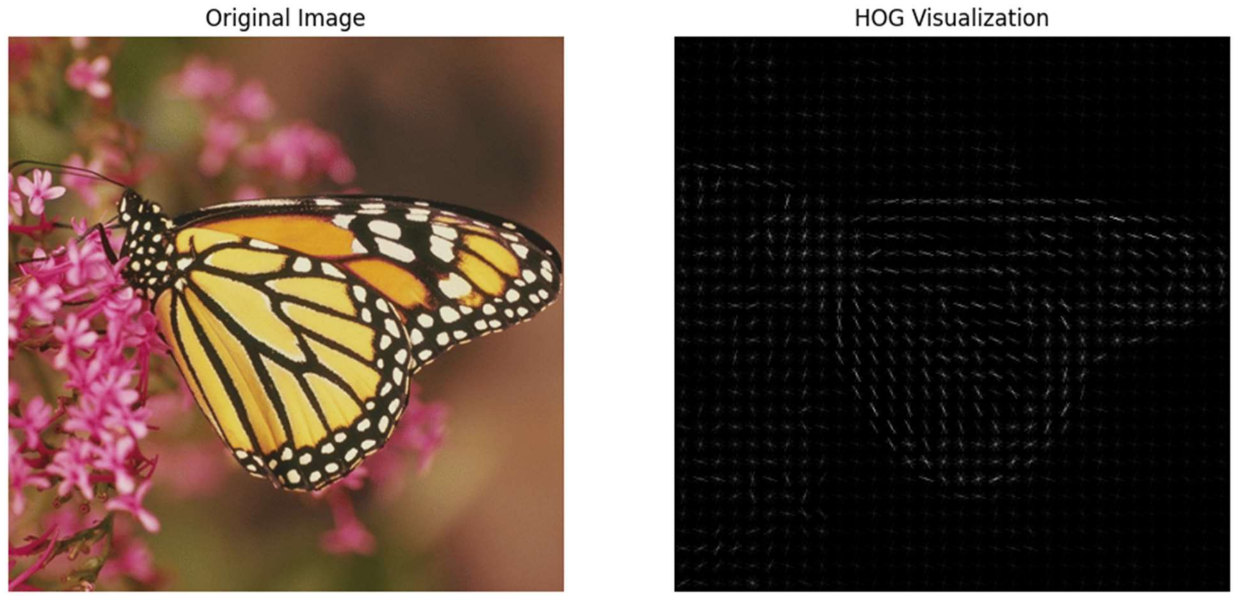


Table Results of training model with HOG

Model	Average Accuracy	Standard Deviation
KNN (k=3)	0.973717	0.001026
Decision Tree (Gini)	0.868717	0.001932
SVM (RBF)	0.989383	0.000602

Base results table without preprocessing vs. preprocessing with HOG

Model	Average Accuracy	Average Accuracy	Standard Deviation
KNN (k=3)	0.971217	0.973717	0.001026
Decision Tree (Gini)	0.870100	0.868717	0.001932
SVM (RBF)	0.983133	0.989383	0.000602

1. Superior Performance of the SVM (RBF)

The SVM (Support Vector Machine) model with RBF kernel consistently shows the best accuracy (0.983133 without HOG and 0.989383 with HOG) and the lowest standard deviation (0.000602).

RBF (Radial Basis Function) kernel: By using the RBF kernel, the SVM is able to map the data to a higher-dimensional space where the classes can be linearly (or nearly linearly) separable. This allows it to handle the nonlinear and complex relationships inherent in image features.

Low Standard Deviation: The very low standard deviation indicates that the SVM's performance is extremely stable and consistent across different partitions of the training data (e.g., in cross-validation). This confirms that the model is not overfitted and has a high capacity for generalization.

2. The Impact of HOG

The Histogram of Oriented Gradients (HOG) is a feature descriptor designed to capture the shape and structure of an object by encoding the distribution of intensity gradients (i.e., edges and contours) in an image.

Increased Accuracy: HOG converts raw image pixels into a more informative and compact feature vector.

Specific Improvement in SVM and KNN: For SVM, providing it with well-defined and less “noisy” features such as those from HOG allows it to find an even better separating hyperplane.

3. Mosaic of images:

The initial long image is as follows:

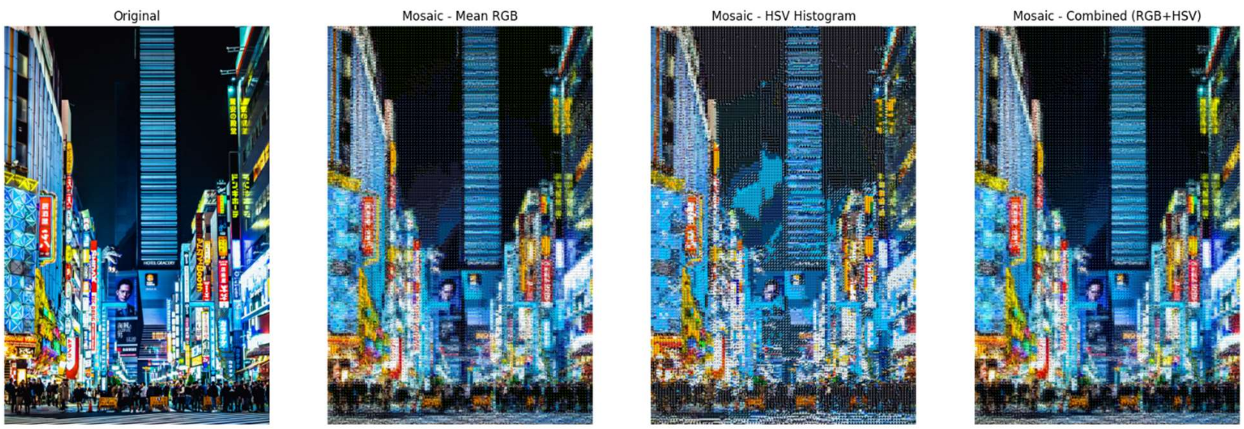
Based on this image and using thumbnails from cifar10, three methods were used to convert it into a mosaic. The efficiency results for each method are as follows:

Method	PSNR	SSIM	ΔE (avg)
Mosaic - Mean RGB	11.531	0.117	19.010
Mosaic - HSV Histogram	10.022	0.145	22.124
Mosaic - Combined (RGB+HSV+Adjust)	11.622	0.131	18.411

Best PSNR: Mosaic - Combined (RGB+HSV+Adjust)

Best SSIM: Mosaic - HSV Histogram

Best Color Match (lowest ΔE): Mosaic - Combined (RGB+HSV+Adjust)



At first glance, the option obtained by HSV seems to be slightly less accurate, but according to the SSIM indicator, it is the best option.

Why is this?

The HSV method may have chosen thumbnails with luminance or texture patterns more similar to the original block, albeit with incorrect colors.

Visually, our brain penalizes color error much more than SSIM, which only sees that the “shapes” or “reliefs” are similar.

That is why it looks worse to the eye, but mathematically “more structurally similar.”

Finally, according to the ΔE indicator, the combined method achieved the greatest similarity with the base image. It seeks to compare color fidelity and, in this way, is an index that can be correctly linked to it looking more similar.

Conclusions:

1. **Image compression with DHT**

- The compression method implemented using DHT proved effective in compressing the image without visible loss of quality.
- Random experiments with different parameters of a and b allowed us to identify configurations that exceed the JPEG standard in terms of PSNR, while maintaining a similar or even better compression ratio. It is important to note that this was one of the possible ways to find a Q matrix. For future experiments, I would recommend testing with a random test for many more ranges of values than those allowed by an equation of the form $Q(i,j)=a+b(i+j)$.
- This shows that the classic JPEG-type quantization structure is not necessarily optimal for all images, and that adaptive adjustments of the coefficients can improve visual fidelity without increasing file size.

2. **Image classification (MNIST)**

- Among the models tested (KNN, Decision Tree, and SVM), SVM with RBF kernel offered the best performance, achieving an average accuracy of 98.9% and the lowest standard deviation.
- The application of the HOG descriptor slightly improved the results, demonstrating that the extraction of structural features provides more relevant information than raw pixel values.
- In general, it can be said that HOG tends to improve performance when analyzing images by providing structural features that contribute important information to the model.

3. **Image mosaic**

- Of the three approaches tested, the Combined method (RGB + HSV + Adjust) offered the best overall compromise between color fidelity (lower ΔE), structure (SSIM), and color ratio (PSNR).
- In this case, HIV achieved a high SSIM, but visually it does not achieve as good a similarity as the other two methods, possibly because it focuses too much on local structure.