Image quantization as a dimensionality reduction procedure in color and texture feature extraction

Moacir Ponti, Tiago S. Nazaré, Gabriela S. Thumé

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, SP — Brazil — 13566-590

Abstract

The image-based visual recognition pipeline includes a step that converts color images into images with a single channel, obtaining a color-quantized image that can be processed by feature extraction methods. In this paper we explore this step in order to produce compact features that can be used in retrieval and classification systems. We show that different quantization methods produce very different results in terms of accuracy. While compared with more complex methods, this procedure allows the feature extraction in order to achieve a significant dimensionality reduction, while preserving or improving system accuracy. The results indicate that quantization simplify images before feature extraction and dimensionality reduction, producing more compact vectors and reducing system complexity.

Keywords: Image quantization, LPP, compact features extraction
1. Introduction

Image recognition problems, including classification and retrieval, often present the extraction of image features as the first step in the system’s pipeline. The feature extraction step is crucial since the features are going to represent images and, therefore, will be used as input for machine learning or content-based retrieval methods. Most feature extraction methods operate on a single channel 8-bit image, equivalent to a grayscale image. Even color feature extraction methods are often applied in 8-bit images, instead of using the original color spaces [1, 2]. It is important to note, however, that these gray levels are actually RGB colors mapped to a lower range of values. For this reason we will refer to color as an 8-bit integer that is mapped from RGB to an 0-255 interval.

In the extraction step, one needs to choose from a wide variety of methods that will work well in some situations, while fail in others [3]. For instance, color feature extractors often used in the literature are the color coherence vectors (CCV) [4], the border/interior classification (BIC) [5] and the color correlograms [6]; while the texture extraction methods often used are the Haralick descriptors [7] and the local binary patterns (LBP) [8]. Each one can produce very different results, depending on the application.

After extracting features it is convenient to obtain a smaller feature space since most learning and retrieval methods have computational complexity proportional to the number of examples (in our context, images) and the number of dimensions. A high dimensional space can also hamper the classification accuracy due to the curse of dimensionality. The problem of dimensionality reduction, using feature selection and/or space transforma-
tion, is often addressed using methods such as Principal Component Analysis (PCA) [9] and Wrappers, which can also be combined with more sophisticated methods [10, 11, 12], and used in different applications [13, 14]. In the literature we also find manifold learning methods, such as the Isometric Mapping (ISOMAP) and Locality Preserving Projection (LPP) [15], which often construct adjacency graphs in order to capture the nonlinear information of original data [2]. Another possible approach is to select subsets of descriptors that best suit each problem [16, 17], but even descriptor selection methods are still application dependent and an open research problem. Thus, although these methods overcome issues of simpler and linear methods, their running time is higher.

A study comparing dimensionality reduction methods for image retrieval showed that linear methods are faster and, on average, slightly decrease the accuracy of the original set of features [2]. The PCA method was found to be the best linear method, while LPP was highlighted as the method with the best precision/complexity ratio, specially for nonlinear spaces. On the other hand, the same study pointed out that manifold methods (such as the LPP) can be unstable and need parameter tuning, but maintain the accuracy of the original feature set.

Instead of focusing on complex methods, in order to transform the feature space or select a subspace, we propose to reduce the complexity of the problem at the beginning of the process of image recognition, by using a quantization procedure before the feature extraction (see Figure 1 for a diagram of the proposed method). Although quantization is commonly part of the pipeline of image classification and retrieval, it is uncommon to find
studies that describe the quantization method and its parameters. When
neglecting the quantization step, one can lose an important opportunity of
reducing the dimensionality and/or the running time (depending on the fea-
tures and methods involved).

Our hypothesis is that using a reduced number of colors with
a proper quantization method will significantly reduce the dimen-
sionality, while improving or maintaining the accuracy of a classification
system. More specifically, we claim that it is possible to encode the color
information using less than 8 bits to extract several features with reduced
dimensionality. This procedure is simple and has potential to improve speed
and reduce memory consumption, depending on the methods involved, on the
following steps: feature extraction, feature selection and recognition. Also,
we investigate several quantization methods, showing that some methods can
improve even the final accuracy. Finally, some intuition about the effect of
quantization in others feature extraction scenarios is discussed.

1.1. Related work and Contribution

Only a few image-based visual recognition studies report the method
used to obtain the intensity image before feature extraction. However, it is
known that the choice of the color-to-grayscale method can have a significant
impact on image recognition [1]. Kanan and Cottrell [1] showed that color-to-
grayscale methods yield different results, producing images that can hinder
even robust methods, such as SURF (Speeded Up Robust Features) and
SIFT (Scale Invariable Feature Transformation). Reducing the image color
complexity is also explored in [18, 19, 20], by using a restoration or spatial
filtering to improve image segmentation results. In addition, according to [21]
Figure 1: Pipeline of methods used in this study. In summary, it investigates the quantization step in order to reduce both the complexity of the system and the dimensionality of the resulting feature vector $\mathbf{x}$. We also evaluated the impact of LPP for dimensionality reduction.

Reducing the number of colors in the feature vector does not severely impact the accuracy of image classification. While Kanan and Cotrell [1] evaluates conversion methods (except for the most significant bit method), Ponti and Escobar [21] extended the most significant bit method without a deep investigation on the quantization parameter for different conversion methods. This gap motivated our investigation.

We investigate different quantization methods using the guidelines of [1], that pointed the methods Luminance’, Intensity’ and Gleam$^1$ to be the best among standard conversion methods. We also included in the experiments a bitwise quantization, that was originally proposed in [23] for producing 6-bit images, and then extended in [21] to allow the user to select the number of

---

$^1$Luminance’ and Intensity’ are gamma-corrected versions of the original Luminance and Intensity methods
The main contribution of this paper is to demonstrate that it is possible to obtain compact and effective feature vectors, by extracting features from images with reduced pixel depth (i.e. palette of intensity levels) at a low computational cost. In addition, since quantization is one of the first steps in the image recognition pipeline, we show how the feature extraction and dimensionality reduction are affected by different quantization methods. Therefore, compact descriptors can be obtained by extracting features from simpler images (in the sense of intensity level possibilities), speeding-up all further processes in classification and retrieval pipeline.

Our findings provide an addition – or an alternative – to feature selection, by using image quantization methods. Due to limiting the number of colors in the original image, the amount of possible features to be extracted is reduced, specially color ones. Texture feature extraction is also facilitated, since it often computes patterns and uses memory proportional to the number of gray levels/colors [7]. Consequently, when using color descriptors there will be a significant reduction of the feature vector size, while using texture descriptors can reduce the running time.

2. Technical background

2.1. Image feature extractors

Previous studies showed that there is no clear guideline to choose a single extraction method [3]. Therefore one option is to perform dimensionality reduction after many image features are extracted, in order to obtain a more suitable set of features. To study the impact of using different quantization
parameters and several features, we choose four color descriptors and a texture descriptor based on co-occurrence matrices \cite{7}. These descriptors were chosen using the results of a previous work by Penatti et al. \cite{3}.

All images, originally in the RGB color space, were converted to a single channel image with $C$ intensity levels, as described in Section \ref{sec:method}. Since the levels are values mapped from color, we will also refer them as colors, even though their visual representation are gray levels. After preprocessing the images, the following methods were used to extract features:

*Global Color Histogram (GCH) \cite{27}*: This descriptor calculates a global histogram $h[.]$, where each value $h[j]$ represents the frequency of some level $j$ in the image. Consequently, it generates a compact representation of the color information, with a $C$-dimensional feature vector.

*Color Coherence Vectors (CCV) \cite{4}*: Is a method designed to capture information about how colors are organized in connected regions. It classifies each region as coherent or incoherent based on the area, i.e., whether or not it is part of a large similarly-colored region. Then, it produces two coherence vectors, one with coherent and the second with incoherent pixels of each color. These vectors are computed by counting the area of regions above and below a threshold $T$, depending on the image resolution. The feature vector is originated by concatenating the two coherence vectors, resulting in $2C$ dimensions.

*Border/Interior Classification (BIC) \cite{5}*: This extractor also tries to capture at the same time color and structure information. To achieve this, it generates a representation of the image color distribution by computing two
histograms: one for the pixels classified as border and another one for those classified as interior. A pixel is classified as border if at least one of its neighbors has a different quantized color, and classified as interior otherwise. The final vector is compounded by concatenating the two histograms and has $2C$ dimensions.

Auto-correlogram of colors (ACC) \[6\]: In this method the spatial auto-correlation of color levels is computed. It computes the occurrence of a same color within a set of $k$ distance values $D$, given some arbitrary distance function. The output for some fixed distance value is a single vector. However it is recommended to use a set of distances $D = \{1, 3, 5, 7\}$. Therefore, the feature vector is a concatenation of the auto-correlograms computed for each distance. It produces a feature vector with $k \cdot C$ dimensions.

Haralick-6 \[7\]: It calculates a co-occurrence matrix with size $C \times C$ using a fixed relationship between a pair of pixels (one is considered the reference pixel and the other one the neighbor pixel). Then, it extracts 6 Haralick features from this matrix: Entropy, Homogeneity, Contrast, Correlation, Maximum Probability and Uniformity.

2.2. Dimensionality Reduction by LPP

Although the most widely used method for unsupervised dimensionality reduction is the Principal Component Analysis (PCA) \[9\], the use of methods that keep the manifold of the original data usually produces better projections in terms of class separability. For instance, the Locality Preserving Projection (LPP) \[15\] has achieved the best compromise between computational
complexity and dimensionality reduction, while maintaining the recognition accuracy [2].

Let $N$ be the total number of features, $D$ the dimension of each eigenvector and $X \in \mathbb{R}^{D \times N}$ the data. The dimensionality reduction can be defined by a projection matrix $W = w_i \in \mathbb{R}^{D \times d}$, where $d < D$ is the number of output features. The low-dimensional features are found by:

$$Y = W^T X$$

LPP is a linear approximation of nonlinear Laplacian Eigenmaps algorithm. It overcomes the disadvantages of linear methods such as PCA [25], while is still able to obtain low-dimensional projected points. The LPP algorithm follows three main steps:

1. Construct an adjacency graph and compute the $k$-nearest neighbors of each point in the graph — the $k$ is a parameter of the algorithm;
2. Find the weights, i.e., compute $W_{i,j} = 1$ if the vertices $i$ and $j$ are connected by an edge;
3. Compute the eigenvectors and eigenvalues by:

$$XLX^T a = \lambda XD_{diag} X^T a,$$

where $D_{diag}$ is a diagonal matrix with $D_{i,j}$:

$$D_{i,j} = \sum_j W_{j,i} L_p = D_{diag} - W,$$

in which $L_p$ is the positive semi-definite Laplacian matrix. According to the condition $a^T XD_{diag} X^T a = 1$, the minimization function can be
reduced to:
\[
\arg\min_a a^T XL_p X^T a
\] (4)

The eigenvectors corresponding to the \(d\) smallest non-zero eigenvalues are then used as output features.

3. Image quantization

In order to extract color or texture features from images, it is often necessary to reduce the number of colors, obtaining a grayscale version of the images. Most studies do not describe the method used to compute it, but it was shown that this step can have a significant impact on the classification results [1].

One alternative to color-to-grayscale methods is the use of the three original RGB channels to compute the features. However, by using this approach, the resulting dimensionality is tripled. In this paper we explore color-to-grayscale methods in order to produce simpler images and to obtain smaller feature vectors than using conventional methods.

3.1. Color-to-grayscale methods

Among the well known methods, we choose the three best reported by [1]: Gleam and the gamma corrected versions of Intensity and Luminance. The gamma operation used is the standard gamma correction function \(\Gamma(z) = z' = z^{1/2.2}\). More sophisticated methods to obtain color quantizations could be applied, such as clustering [26], however, when dealing with big data applications, this kind of method will increase the running time of the system.
For this reason, we choose to use only faster methods, that are already part of the visual recognition task.

Perhaps the simplest way to perform the conversion from a 24-bit image, often a RGB image, to an 8-bit image with a single channel, is to compute the Intensity, i.e., the mean of the RGB channels:

\[ Q_{\text{Intensity}} = \frac{(R + G + B)}{3}. \]  

(5)

After that we can obtain the gamma corrected version with \( Q_{\text{Intensity'}} = \Gamma(Q_{\text{Intensity}}) \). Alternatively, by correcting the channels first and then performing the same linear combination, we obtain Gleam:

\[ Q_{\text{Gleam}} = \frac{(R' + G' + B')}{3}, \]  

(6)

where \( R' \), \( G' \) and \( B' \) are the gamma corrected channels. This method yields very different results when compared to Intensity because the \( \log(\Gamma(z)) \) function is known to be concave in \( z > 0 \) [27]. This can be used to obtain the following inequality:

\[ Q_{\text{Intensity}} \leq Q_{\text{Gleam}} \leq \Gamma(Q_{\text{Intensity}}). \]  

(7)

**Luminance**, the third method, is designed to capture the human brightness perception, which is most sensitive to green and least sensitive to the blue spectrum [28]. It computes a weighted combination of the RGB channels:

\[ Q_{\text{Luminance}} = 0.299R + 0.587G + 0.114B. \]  

(8)

These coefficients are defined by the standard ITU-R Recommendation BT.601, originally issued in 1982 [29]. For this reason, this is the standard RGB to
grayscale method used in many image processing software and libraries, e.g. GIMP and OpenCV.

Considering that the weighted sum presented does not match the logarithmic nature of human vision, a subsequent gamma correction can be performed: \( Q_{\text{Luminance'}} = \Gamma(Q_{\text{Luminance}}) \).

All transformations are independently applied to each pixel, producing a gray level image \( Q \) with 256 colors by merging the color channels on each pixel. A second step is required to obtain an image with less colors, which can be done by merging similar colors and consequently producing less bins in the image histogram.

3.2. Quantization based on the most significant bits

Instead of computing a linear combination of color channels, the most significant bits (MSB) quantization tries to emphasize the chromatic differences by ordering the actual color bits in a single channel. It computes a quantized image in a single step, with \( C = 2^p \) colors. This algorithm was proposed in \[23, 21\] and it is explored here in the context of dimensionality reduction.

Initially, the algorithm computes the amount of bits from each color channel that will contribute to the final image. The order of preference is \( G, R \) and then \( B \). This order is based on the range of stimulus of the cone cells\(^2\) for visible light wavelengths \[30, 31\]. The same order of preference can be seen in the weights of \textit{Luminance} method.

\(^2\)there are three types of cone cells located on the human eye, sensible to ranges of wavelengths that are dominated by the colors green, red and blue.
The amount of bits used from channels $G$, $R$ and $B$ are, respectively: $N_g$, $N_r$ and $N_b$. The bits are extracted from the bit code of the original channels. Considering a little-endian system and the channels to be represented as 8 bits numbered from 0 (least significant) to 7 (most significant), the bits mask for each channel can be defined as:

$$Gm = \sum_{i=(8-N_g)}^{7} G_i \cdot 2^i,$$

$$Rm = \sum_{i=(8-N_r)}^{7} R_i \cdot 2^{i-N_g},$$

$$Bm = \sum_{i=(8-N_b)}^{7} B_i \cdot 2^{i-(N_g+N_r)},$$

where $G_i$, $R_i$, $B_i$ are the $i^{th}$ bit code of the channels $G$, $R$ and $B$, respectively. The new image, with $2^p$ colors, is computed by the composition of the extracted bits mask for each channel using Equation 12:

$$Q_{MSB} = Rm + Gm + Bm.$$  

This method is illustrated in Figure 2 for $2^6 = 64$ colors.

![Figure 2: An example of MSB quantization with $2^6 = 64$ colors, using the 2 most significant bits from each channel in a little-endian system. Adapted from [21].](image-url)
3.3. Behavior of quantization methods

Each method behaves differently for a given RGB image. For instance, the Intensity maps all permutations of the same values in RGB to a same color. Therefore, it produces a plane that cuts the RGB cube as depicted in Figure 3. The Gleam result is similar, but spanning a curved surface, due to the nature of the gamma function. The same effect is achieved using Intensity'. In all cases, the result is the mapping of very different chromatic features to similar intensity values. The Luminance method tries to overcome that by weighting the linear combination of the channels.

![Figure 3: Plane computed by Intensity method when one of the color channels has value of 255](image)

An example of generated images using the discussed quantization methods can be seen in Figure 4. In this case, it is possible to notice that both Luminance and MSB methods can better discriminate different colors. In addition, the MSB color map yielded a greater number of unique colors. The
gradient bar below the pencils image demonstrates how the quantization methods behave regarding color variations.

Table 1 presents some numerical examples, with the output of each method. In this case, the inputs are tuples of values \((R, G, B)\). Note that the gamma correction must be computed in a 0-1 interval of real values, that is mapped afterwards to the 0-255 interval.

<table>
<thead>
<tr>
<th>Method / RGB</th>
<th>(255, 50, 50)</th>
<th>(50, 255, 50)</th>
<th>(50, 50, 255)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gleam</td>
<td>165</td>
<td>165</td>
<td>165</td>
</tr>
<tr>
<td>Intensity'</td>
<td>180</td>
<td>180</td>
<td>25</td>
</tr>
<tr>
<td>Luminance'</td>
<td>175</td>
<td>212</td>
<td>144</td>
</tr>
<tr>
<td>MSB</td>
<td>60</td>
<td>228</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 1: Outputs of the four different quantization methods. Note that MSB produced values that better discriminate the colors, while the other methods in general produced similar outputs for different RGB inputs.

In Figure 5 we exhibit an example of color reduction using the MSB method for a pair of images from the Caltech-101 dataset. One can notice the color preservation, specially between 256 and 64 colors. With 32 colors, the images are still satisfying, however there is some information loss.

3.4. Quantization as a dimensionality reduction method

The basic pipeline of image recognition comprises: acquisition, pre-processing, feature extraction and feature selection or transformation. The final feature subset is then used in retrieval or learning algorithms. As reported earlier, the most common setup is to use an 8-bit grayscale image with 256
Figure 4: Original image (a) and its single-channel versions produced with (b) Gleam: 184 unique colors, (c) Intensity’: 184 unique colors, (d) Luminance’: 184 unique colors, (e) MSB: 232 unique colors.
colors as a basis for feature extraction. By applying a quantization in the preprocessing step we expect to reduce the dimensionality of the feature vector in the beginning of the system, with potential to benefit all other steps. It also reduces the running time and memory complexity of the Haralick extractor, because it is based on a $C \times C$ co-occurrence matrix.

Our hypothesis is that using a reduced number of colors will significantly reduce the dimensionality, while improving or maintaining the accuracy of a classification system. We expect that it will be even better for the MSB quantization method because of the analysis carried out in Section 3.3. In the next section, a set of experiments is described to test our hypothesis.

4. Experimental setup

4.1. Reproducibility

In order to comply with reproducible research, there is a release of a repository\(^3\) that corresponds to the last version of the code and extracted

\(^3\)Repository release DOI: [http://dx.doi.org/10.5281/zenodo.15932](http://dx.doi.org/10.5281/zenodo.15932)
features upon publication. Also, all datasets described in this paper are available.

4.2. Image datasets

Three image datasets were used in the experiments:

Corel-1000[^4] contains ten balanced categories of natural images, including some well defined classes and confusion among some specific classes.

Caltech101-600[^5] has photos and drawings with different poses. This dataset is difficult to classify using only global descriptors. We used a subset with 6 balanced classes: airplanes, bonsai, chandelier, hawksbill, motorbikes and watch.

Produce[^32] (also referred as Tropical Fruits and Vegetables dataset): consists of images with a constant background but changes in illumination, number of objects and scale, including partial occlusions of objects.

The Produce and Caltech101 datasets were modified to balance the available classes by removing images from the majority ones. The reason for this is to avoid unbalanced problems, since it could be a possible confusing cause in the analysis of the results, which are focused on dimensionality reduction.

The datasets were chosen so that it would be possible to have a very well

[^5]: available at [http://wang.ist.psu.edu/docs/related/](http://wang.ist.psu.edu/docs/related/)
 behaved data (Produce dataset), another one with confusion between specific classes (e.g. classes Africa/Elephants and Mountain/Beach from Corel dataset), and finally a dataset with a higher intraclass variance specially in terms of color features (Caltech101) and class overlapping. Examples of each dataset are shown in Figure 6 and their characteristics are summarized in Table 2. In this figure, two images from each Caltech101 dataset class are displayed to show the high intraclass variability.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Samples</th>
<th>#Classes</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corel-1000</td>
<td>1000</td>
<td>10</td>
<td>384 × 256</td>
</tr>
<tr>
<td>Caltech101-600</td>
<td>600</td>
<td>6</td>
<td>around 420 × 380</td>
</tr>
<tr>
<td>Produce</td>
<td>1500</td>
<td>15</td>
<td>1024 × 768</td>
</tr>
</tbody>
</table>

4.3. Description of experiments

In Figure 7 we present our flow diagram, including details of the pipeline, and the methods used in the experiments. Note that our contribution is to show the effects of the quantization step and how it can be used to reduce the dimensionality of the feature space.

The experiments begin by quantizing the images in 256, 128, 64, 32 and 16 colors and extracting the features. After that, we perform the following two types of experiments:

1. Experiments using each individual descriptor, followed by classification without feature selection;
Figure 6: Examples of each class of the datasets used in the experiments.
2. Experiments using the full feature vector created by concatenating all descriptors, followed by classification with and without feature selection.

The main topics under investigation are the following:

- **The effects of the quantization over the descriptors performances**: how classification accuracy is changed when using different quantization methods and reducing the color/level parameters;

- **The effects of different quantization methods in reducing dimensionality** for:
  - individual descriptors;
  - concatenation of descriptors.
• The combination of quantization and LPP methods in order to reduce the dimensionality.

One important scenario is when a substantial number of features is available and it is required to obtain a small feature vector from it. In our experiments we analyze if the concatenation produces better results than single descriptors and how much the feature space can be compressed while keeping a similar accuracy.

4.4. Parameters and settings

The parameters and settings used in the experiments for each method are described below:

• Feature extractors settings: We extracted features using the following configurations:

  ACC: using a set of 4 distances $D = \{1, 3, 5, 7\}$ using the $L^\infty$-norm distance (i.e. chessboard distance);

  BIC: utilizing a 4-neighborhood pixel;

  CCV: adopting a threshold value of 25 for coherent/incoherent classification of regions;

  Haralick-6: employing a neighborhood relationship of $\Delta(x, y) = (1, 0)$, i.e. the neighbor pixel for computing the co-occurrence matrix is defined to be the pixel to the right.

All parameters above were chosen by following the original papers recommendations. Note that, after feature extraction all descriptors were individually normalized using $L1$ norm.
• **Dimensionality reduction settings:** The LPP projection was carried out using $k = 128, 64, 32$ and $16$ dimensions, with a fixed value of $k = 10$ neighbors. This parameter was determined empirically and does not influence severely the accuracies.

• **Classification and evaluation settings:** For the classification task, we used *Support Vector Machines* (SVM), since it has learning guarantees and was shown to have many benefits concerning dimensionality. The parameters for it were found using a grid search inside the training set.

The experiments were performed with a **stratified 10-fold cross validation setting**. Because all datasets are balanced and the sampling for the cross validation is stratified, the **accuracy** is a suitable measure to evaluate the classification performance [33].

5. Results and Discussion

5.1. Individual descriptors experiments

The mean accuracy for the first set of experiments is shown in Figure 8. It presents a general perspective of the results. For each dataset and descriptor we displayed six accuracies, corresponding to the quantization using $256, 128, 64, 32, 16$ and $8$ colors (levels).

It is evident that the method used to obtain the single channel image has significant impact on classification accuracy. In this way, the choice of quantization methods can improve or hamper the discriminative capacity of color and texture descriptors. This result confirms previous findings [1].
Figure 8: Results for Corel (a), Produce (b) and Caltech (c) datasets, with all quantization methods. For each descriptor it is shown the accuracy while using 256, 128, 64, 32, 16 and 8 colors, from left to right.
However, the cited paper did not explore the same descriptors neither include the MSB method in the analysis. Our results show the Gleam method does not work well for color or texture extractors, while in the previous work it was referred among the best methods. Also, the MSB method show the best accuracy mean in general. For the texture descriptor, however, MSB is not ideal, but the Intensity and Luminance methods seems to be the best choices. Another important observation is that the reduction from 256 to a lower number of levels often maintains the accuracies and sometimes can even slightly increase them, specially for 128 and 64 quantization levels.

Since the results showed in Figure 8 are only an overall view, we performed a deeper analysis using the following pairs of methods: BIC descriptor with MSB quantization; and Haralick descriptor with Luminance quantization. Also, considering that 16 and 8 colors degraded the results, we used color levels of 256, 128, 64 and 32 for the remaining experiments.

The ANOVA statistical test was performed to compare accuracies obtained by methods presented in Figures 9 and 10. In order to find which method had significant difference we used the Tukey’s Honest Significant Difference test. A significance level of $\alpha = 0.01$ was used in both tests. To observe the results in more detail, a boxplot for 256, 128, 64 and 32 levels using BIC descriptor is shown for the MSB method in Figure 9 and for Luminance and Haralick descriptor pair in Figure 10. The boxplots shaded in gray correspond to data with significant average difference when compared to the accuracy of the 256 level baseline, obtaining $p < 0.01$.

By using color features and lower quantization levels provided by the MSB method, we obtained significantly better results when compared with
the 256 levels for the Corel (128, 64 and 32 levels) and Caltech (64 levels) datasets, according to the statistical test (see Figure 9). It is important to note the results were not significantly worse, except for 32 levels in Produce dataset. Therefore, we believe a 64 level value represents a good choice for the quantization parameter. Otherwise, texture features were better extracted using a Luminance image, for which lower quantization levels can keep or degrade the results depending on the dataset (see Figure 10).

![Figure 9: Accuracies for MSB quantization considering 256, 128, 64 and 32 color images with BIC descriptor for the Corel (a), Produce (b) and Caltech (c) datasets. Boxplots in gray correspond to significances when compared to the 256 level base accuracy with $p < 0.01$.](image)

Another important comparison is between the dimensionality reduction obtained using the quantization and the LPP method. We also show this comparison using the MSB and BIC methods, which represent the best pair of methods. In this case the comparison is pair-wise. For each dataset, we used as input the image quantized by MSB with 256 colors. A BIC descriptor is extracted and used as input for the LPP method in order to produce reduced versions of the vectors with 256, 128 and 64 dimensions. The accuracies ob-
Figure 10: Accuracies of Luminance quantization in 256, 128, 64 and 32 color images with Haralick descriptor for the Corel (a), Produce (b) and Caltech (c) datasets. Boxplots in gray correspond to significances when compared to the 256 level base accuracy with \( p < 0.01 \).

Obtained with the LPP vectors are then compared with the accuracies obtained by the vectors reduced only using the quantization parameter. Since the comparison must be carried out in pairs, the Student’s \( t \)-test were performed to compare the accuracies obtained by the methods showed in Figures 11.

By analyzing the data and its variances, the tests were calculated under the assumption of independent two-sample with unequal variances, and a significance level of 0.01. The quantization method obtained lower accuracies when compared to the LPP method in three experiments: 256 dimensions in Corel dataset; 256 and 64 dimensions in Produce dataset. For the Caltech dataset, the quantization method for dimension reduction performed better for 256 and 128 dimensions. The remaining experiments did not present statistical difference. Despite the accuracy loss in some scenarios, it is interesting to note that, by using a correct quantization parameter, it is possible to keep or even improve the accuracies after dimensionality reduction (for instance, the

27
improvement obtained in Caltech dataset as can be seen in Figure 11 (c).

Figure 11: Comparison of accuracies for MSB quantization and LPP methods for dimensionality reduction using BIC descriptor: Corel (a), Produce (b) and Caltech (c) datasets. The comparison is performed in pairs of methods (LPP vs MSB) with the same dimensionality. Boxplots in gray corresponds to significances when compared to the base accuracy, with $p < 0.01$.

However, the problem dimensionality using only one descriptor can be considered low. Also, it is common to extract many descriptors for a given problem, since there is often no clear guideline to choose a single method for feature extraction. For this reason, the next section shows experiments using feature concatenation.

5.2. Full feature space experiments

The results of the previous experiment guided this second one, where we concatenated all descriptors to produce a larger feature vector. The use of 128 and 64 colors was shown to be the best settlement between feature vector size and accuracy. Here, the idea is to test if a concatenation of all descriptors can improve the results, and also to compare the use of 256 and
the reduction to 128 and 64 colors, in order to verify if the quantization can provide an alternative to a regular dimensionality reduction procedure (the LPP in this case).

By changing the number of colors we can reduce the original dimensionality $D$. The numbers of features with respect to the quantization parameters are: 256 colors – 2310 features; 128 colors – 1160 features; 64 colors – 582 features; 32 colors – 294 features; 16 colors – 150 features.

First, we used the regular 256 color setting, starting with $D = 2310$ and then applied LPP with $d = 1160, 582, 294$ and 150, producing feature vectors with the same size as obtained by quantization. In Figure 12 we show the results for each dataset when using LPP. When analyzing the quantization methods, the MSB method was, again, better than the competing methods.

Figure 12: Results for the quantization methods Gleam, Intensity, Luminance and MSB for the full feature space, starting with $D = 2310$ and then reducing it with LPP to $d = 1160, 582, 294$ and 150.

By using all descriptors we obtained an improvement in accuracy, when compared to the best individual descriptor. Here we compare the results for the LPP method with the quantization settings MSB 128 and MSB 64.
Figure 13 presents a boxplot for the datasets, comparing the accuracies of the original feature space with LPP and MSB quantization methods used to reduce dimensionality. The ANOVA test was performed to compare the accuracies obtained by the feature space $D = 2310$ with the reduced spaces, followed by a Tukey’s Honest Significant Difference post hoc test. Both tests used $\alpha = 0.01$ as significance level. The only results that did not significantly change the original accuracies were the MSB 1160 and MSB 582 in Corel dataset, and with MSB 1160 in both Produce and Caltech datasets, according to the statistical test for $p < 0.01$. More importantly, the results were not significantly worse, except for 32 levels in Produce dataset. Therefore, we believe that 64 levels represent a good choice for the quantization parameter.

Figure 13: Results for Corel (a), Produce (b) and Caltech (c) datasets, comparing the use of LPP projection and MSB quantization to reduce dimensionality.

The results clearly indicate that quantization can be used as a dimensionality reduction method for visual data, specially for 128 and 64 colors. In addition to the dimensionality reduction obtained by the quantization, we applied LPP over the vector obtained by the MSB using 256 colors (baseline), i.e. a vector with $d = 2310$, and 64 colors, i.e. a vector with $d = 582$,
in order to see if a further compression is possible. Figure 14 shows that the use of \( d = 100 \) is still able to maintain the accuracies for all datasets using 64 colors. It is interesting to note that LPP projections were better, in general, using a MSB 64-color quantized version of the images than the original 256 version. We believe the reason behind this result is that quantization removes confusing information. It simplifies the images in a way that the remaining colors – after quantization – better describes a given category.

![Figure 14: Results for LPP projection over the feature space produced by MSB method using 256 \((d = 2310)\) and 64 colors \((d = 582)\).](image)

### 5.3. Complexity analysis in a scenario of dimensionality reduction

The vector containing all concatenated descriptors has \( 9C + 6 \) dimensions, where \( C \) is the number of colors in the input image. In addition, the overall running time function for the extraction of all features is \( f(N) = 42N + 6C^2 \), where \( N \) is the number of pixels.
In order to compute the reduced vector the LPP requires for each instance in the dataset $D^2 + kD + d^2$ operations, where $D$ is the size of the original feature vector ($D = 9C + 6$), $d$ is the size of the output vectors and $k$ is the number of neighbors used in the algorithm.

It is possible to perceive the potential of quantization in helping reduce dimensionality by the following example: using 256 colors, the feature vector will have $D = 2310$ features. Considering 100 instances to be processed, this demands around 231.6 million computer instructions for extracting features and perform a LPP-based dimensionality reduction (with $k = 10$ and $d = 50$). By using 64 colors, the dimensionality is reduced to $d = 582$ (74.8% of reduction) and the number of instructions is reduced to 58.7 million (74.6% of reduction). Thus, independently of using a feature selection method, by choosing a proper quantization method and settings it is possible to reduce the dimensionality and speed up the procedures that precede the recognition.

6. Conclusions

This research gives an important contribution regarding the use of image quantization to dimensionality reduction of visual data. Considering scenarios with a large number of images, it could produce a significant impact. We stated that an original feature vector of $D$ dimensions could be reduced to $d \approx D/4$ using only quantization parameter changing, while producing good results. Another possibility is to use this method as a first reduction step and then use the LPP transformation in order to obtain only 100 features that can better represent the data, achieving similar or higher accuracies. In this paper we also investigate the use of MSB quantization method, which
produced superior results for color feature extraction. We believe that improved variations of this method have potential to be even more effective on generating compact feature vectors.

It is important to note that image quantization does not require training, and it is already a task in the recognition pipeline. For this reason, it will not increase the computational cost of the system, while simplifying the subsequent steps. It reduces the dimensionality of feature vector for color descriptors and the running time for computing the texture descriptors. Another important observation is that quantization is used specially for visual data, so it is not a general method for dimensionality reduction.

Image quantization can possibly produce effects on other feature extraction techniques that can be explored in future investigations. By using simpler (quantized) images, the descriptors based on orientation (HoG, SIFT), bag of visual words and Fisher vectors, would probably be more sparse. In deep learning, one can investigate if the use of simpler images would help or hamper the feature learning.

Future studies can also explore different color models and variations of the MSB method in order to produce better color maps for recognition purposes, as well as investigate the spaces generated by such methods in order to understand better the class-discriminative capability.

Acknowledgment

The authors would like to thank CNPq and FAPESP (grants #11/22749-8 and #11/16411-4).
References


[8] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, Pat-


[22] M. Ponti, C. Picon, Color description of low resolution images using fast bitwise quantization and border-interior classification, in: Acous-
tics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, IEEE, Brisbane, Australia, 2015.


