Digital Image Processing

Topic:

Color Descriptors

Zoran Kostovski – 221522

Description

This project focuses on developing and evaluating algorithms for comparing image similarity using color descriptors.

The primary goal is to determine how similar two images are based on their color distribution.

The dataset used in this project consists of flower images, chosen for their rich diversity in color and texture, which provide an excellent testing ground for color-based analysis. And with a variety of flowers with similar colors but different shapes it will be a nice test to see if the algorithms will find flowers of the same species similar.

The project implements three models: color moments, color correlograms, and color histograms, each of which captures distinct aspects of the color distribution in an image.

The models are evaluated individually and in combination to find which models are better at perceptual similarity rather than computational.

The results from these models have potential applications in areas such as plant species identification and color-based image retrieval systems. These algorithms can also be used in image search engines.

This work aims to demonstrate the power of color-based descriptors in efficiently and accurately comparing visual content.

Color Descriptors

Color descriptors are a fundamental tool in image processing, used to capture and represent the distribution of colors within an image.

Unlike raw pixel values, which provide a pixel-by-pixel representation of color, descriptors summarize color information in a compact, interpretable form, making them highly efficient for tasks like image retrieval and similarity measurement.

Their ability to distill an image’s color characteristics into meaningful numerical features is one of their most powerful aspects.

One of the most notable advantages of color descriptors is their robustness. They are less sensitive to changes in resolution, small image distortions, or cropping, as the overall color distribution often remains consistent.

Color descriptors are also computationally efficient, especially when compared to more complex image representation methods such as deep learning-based features, and can be applied across various domains, including content-based image retrieval, object recognition, and scene classification.

However, color descriptors are not without limitations. They are highly dependent on lighting conditions, which can significantly alter the perceived color distribution of an image.

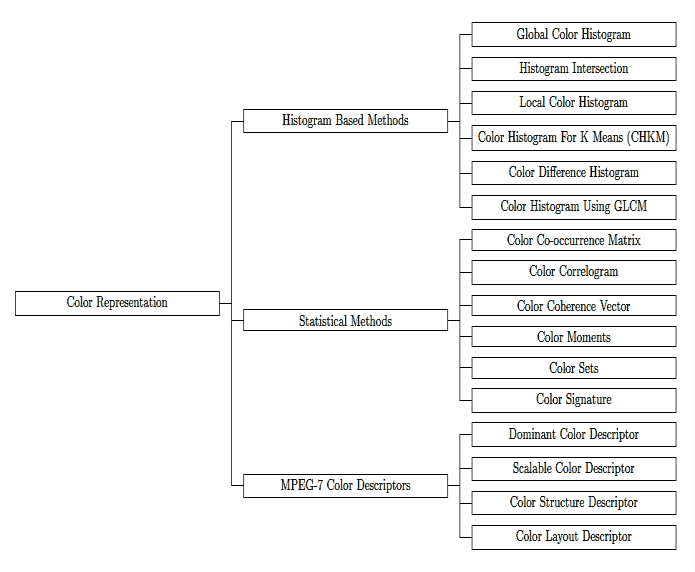
For instance, changes in ambient light or shadows can lead to discrepancies in descriptor values, potentially affecting similarity results. Also, color descriptors often overlook spatial information, meaning that two images with similar color distributions but different arrangements of those colors may be incorrectly classified as similar.

While some methods, such as the color correlogram, partially address this by incorporating spatial relationships, most descriptors still struggle with complex scenes where color arrangement is critical.

The outcome of this drawback is that sometimes images can look nothing alike to the human eyes but if they have a similar histogram for example, they will appear to be similar to the algorithm.

To minimize the drawback, three types of color descriptors are used to analyze and compare images: color moments, color correlograms, and color histograms. Each method captures different aspects of the image’s color distribution.

The combination of these descriptors allows for a more comprehensive evaluation of image similarity where the drawbacks of each model are negated by the strengths of the other algorithms



Img. 1: Types of Color Descriptors

Models for Generating Color Descriptors and Finding Similarity

The models for generating color descriptors in this project extract and summarize color information from images in three distinct ways.

The color moments model captures statistical properties like the average color, how much colors vary, and their distribution shape, providing a compact summary of the image’s color profile.

The color correlogram analyzes how often pairs of colors appear together at specific distances, adding a spatial element that helps in understanding how colors are distributed across the image.

The color histogram counts the frequency of each color within an image, creating a simple yet effective representation of the overall color distribution.

Each model captures a unique aspect of an image’s color properties which yields interesting similarity results.

Next we are gonna take an in-depth view of each model separately.

Model 1

The color moments model uses statistical measures to summarize the distribution of colors in an image.

This approach assumes that the essential information about an image's color content can be captured by three moments: mean, variance, and skewness.

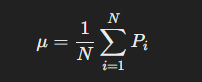
These moments are computed for each color channel (e.g., Red, Green, Blue in RGB or Hue, Saturation, Value in HSV), creating a concise descriptor that can be used to compare images.

Using the numpy package we already have functions ready for calculating the mean and variance and we just need to calculate the skewness.

**Mean**

The mean represents the average intensity of a color channel, capturing the overall color in the image.

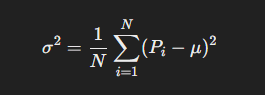
This value gives an indication of the dominant color within the channel.



**Variance**

Variance measures the spread of the intensity values around the mean, providing insight into how much the colors vary in the image.

A low variance means the colors are uniform, while a high variance indicates significant diversity.



**Skewness**

Skewness measures the asymmetry of the color intensity distribution, describing whether the values are skewed toward lighter or darker colors.

A positive skewness value indicates a bias toward brighter pixels, while a negative skewness indicates darker pixels.

By calculating these moments for each channel, the model generates a feature vector that represents the image's color distribution in a compact and descriptive way. Images with similar color moments tend to have similar color distributions, making this approach effective for comparing visual content.

This simplicity and efficiency make it a powerful technique, especially in scenarios where computational resources are limited.

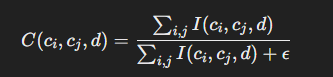
Model 2

The **color correlogram** is a descriptor that captures the spatial relationship between colors in an image.

Unlike simpler descriptors that only consider the frequency of colors, the correlogram measures how often a pair of colors ci ​ and cj​ appears at a specific spatial distance from one another. By incorporating both color and spatial information, the correlogram becomes a more robust method for comparing images with complex color distributions.

The key idea behind the correlogram is to compute a matrix where each entry corresponds to the probability of finding a pair of colors ci and cj ​ separated by a given distance d. This process involves two main steps:

1. **Quantizing Colors**:  
   To make the computation feasible, the continuous range of colors is reduced to a fixed number of bins (e.g., 64 or 128 bins). Each pixel in the image is assigned to the nearest color bin, which simplifies the computation.
2. **Computing the Correlation**:  
   For each pixel, the algorithm examines its neighbors within a specified distance d and counts how often color pairs ci ​ and cj ​ occur. The correlogram entry for a pair (ci, cj) at distance d is calculated as:



where *I*(ci, cj,d) is an indicator function that equals 1 if the pixel pair (ci, cj) appears at distance d, and 0 otherwise. The term ϵ is a small constant added to avoid division by zero.

The result is a matrix where each row represents a color, each column represents a distance, and each entry represents the likelihood of a specific color pair occurring at that distance.

In its simpler form, called the **autocorrelogram**, the model considers only pairs where ci ​= cj, making the computation more efficient.

The color correlogram captures both the color distribution and spatial relationships, offering a more detailed representation of an image compared to methods like histograms. However, it is computationally expensive and requires careful tuning of parameters such as the number of color bins and the distance range.

Model 3

The **Color Histogram** is one of the most widely used and simplest methods for color representation in image processing. A color histogram is a graphical representation of the distribution of colors in an image.

Each axis in the histogram represents the frequency of a particular color or a range of colors in the image. The color histogram is independent of the image's spatial arrangement, meaning it only counts how often each color appears in the image but ignores their positions.

For an image, a histogram is constructed by dividing the entire color space into bins (e.g., 256 bins per channel in the case of RGB images).

The image is then analyzed by counting the number of pixels that fall into each bin. This count represents the frequency of that color in the image. The more colors there are in a specific bin, the higher the value for that bin in the histogram.

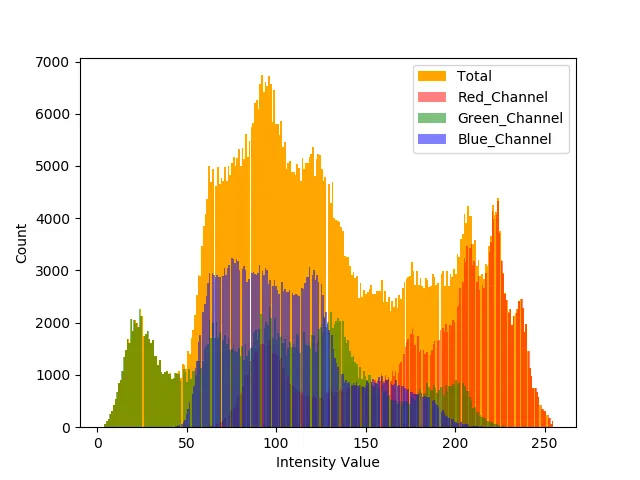
The process typically follows these steps:

1. **Color Space Representation**:  
   The color space is divided into discrete bins for each color channel. Each pixel in the image is assigned to a bin corresponding to its color value. For example, in an RGB image, the pixel’s Red component might be mapped to the bin corresponding to its intensity value.
2. **Histogram Calculation**:  
   Once all the pixels are assigned to their respective bins, the histogram is formed by counting the number of pixels in each bin. The resulting histogram for each channel (Red, Green, Blue) is stored as a vector.
3. **Comparison**:

A common method for comparing histograms is the **Bhattacharyya distance**, which measures the similarity between two histograms. This distance computes the overlap between the two histograms by comparing the values in corresponding bins.

The **advantages** of using histograms include their simplicity and the fact that they are fast to compute. However, one **drawback** is that histograms do not capture spatial information, meaning they do not account for the arrangement of colors in an image.

This makes them less effective in images where the spatial relationship of colors is important (such as textures or patterns). Despite this limitation, histograms are often effective in simpler scenarios where the color distribution alone is sufficient for comparison.



Testing Models

In order to get results that are more accurate or rather more extreme, I created a program that runs all three algorithms on every possible image pair.

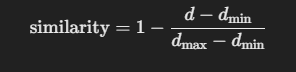
There are roughly 20000 image pairs and the program finds the top five most similar pictures for each of the algorithms.

By testing each algorithm with a number of pictures we can see whether the results are depicting computational similarity or perceptual.

Our goal is to tweak the code so that the computational similarity is as close to the perceptual as possible.

Each of the algorithms yield a coefficient of similarity unique to them, in order to get a score that is perceptible for people it needs to be converted.

To scale the output between 0 and 1 for the color moments algorithm, we applied min-max normalization to the Euclidean distance:

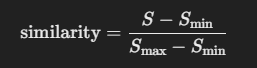


d is the Euclidean distance between the color moments of the two images.

dmin and dmax ​ are the minimum and maximum distances observed across the entire dataset of images.

The color correlogram model, on the other hand, computes a similarity score directly (typically ranging from 0 to 1) based on how often certain color pairs appear at specific distances in an image. The similarity measure produced by the correlogram tends to be higher for more similar images.

The formula we used here for normalization is:



 S is the raw similarity score for the correlogram.

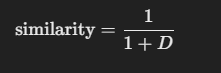
 Smin and Smax​ are the minimum and maximum correlogram similarity scores observed in the dataset.

For correlograms, a higher similarity score (closer to 1) already represents more similar images. So, there's no need for a transformation like 1− (like in the previous example) because the correlogram's similarity values already fit the desired behavior

In the case of histograms, the similarity between two images is computed using the Bhattacharyya distance, which measures the overlap between two color histograms.

The output of this distance is typically between 0 and infinity, with lower values indicating higher similarity.

To convert this distance into a scale between 0 and 1, we used inverse scaling:



where:

D is the Bhattacharyya distance between the histograms of the two images.

This formula ensures that the similarity score is within the range [0, 1].

We will review the most similar pairs for each algorithm.

**Table 1: Color Moments - Top Pairs**

|  |  |  |
| --- | --- | --- |
| Rank | Image Pair | Similarity Score (0-1] |
| 1 | |  | | --- | | 0015.png, 0180.png | | 0.9932 |
| 2 | 0112.png, 0123.png | |  | | --- | | 0.9851 | |
| 3 | 0011.png, 0111.png | 0.9826 |
| 4 | 0180.png, 0192.png | 0.9825 |
| 5 | 0007.png, 0176.png | 0.9821 |

**Table 2: Color Correlogram - Top Pairs**

|  |  |  |
| --- | --- | --- |
| Rank | Image Pair | Similarity Score (0-1] |
| 1 | |  |  | | --- | --- | | |  | | --- | | 0035.png, 0083.png | | | 0.9927 |
| 2 | 0066.png, 0115.png | |  | | --- | | 0.9924 | |
| 3 | 0022.png, 0134.png | 0.9923 |
| 4 | 0132.png, 0145.png | 0.9899 |
| 5 | 0113.png, 0145.png | 0.9799 |

**Table 3: Histogram - Top Pairs**

|  |  |  |
| --- | --- | --- |
| Rank | Image Pair | Similarity Score (0-1] |
| 1 | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | 0054.png, 0073.png |  | | | 0.9574 |
| 2 | 0082.png, 0090.png | |  | | --- | | 0.9521 | |
| 3 | 0082.png, 0208.png | 0.9426 |
| 4 | 0041.png, 0114.png | 0.9404 |
| 5 | 0007.png, 0022.png | 0.9403 |

The most similar pair for the color moments algorithm is image 0015.png and 0180.png which are displayed below.

Each of the algorithms as you will see in the below guessed the correct type of flower. However for this one the angle is completely wrong and the lighting is not the same, even though the computational similarity is 0.9932 out of 1.

The most similar pair for the color correlograms algorithm is image '0035.png', '0083.png'which are displayed below.

Same as the color moments picture the flower is the same but the angle is vastly different.

The most similar pair for the histograms algorithm is image '0054.png', '0073.png'which are displayed below.

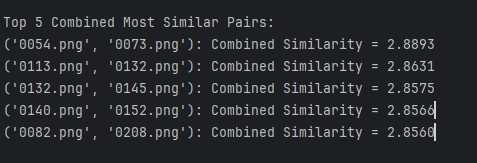
This algorithm actually managed to find the most similar picture when you combine the score of all the algorithms without giving weight to one of the either algorithms.

When you take the scores of all the algorithms and summarize their results so that now 3 is the maximum score, the winner of the histogram is also the winner of the combined results.

Even though this is the most similar picture it doesn’t appear as a top 5 similar picture in the other algorithms as you can find in the table above.

But only one other result from the histogram model’s top five list is included in the combined top five, meaning it is not defining for the combined list.

Combined Model Results



To get the combined score we just used the sum of the scores and increased the range from (0, 1] to (0,3].

All of the models have the same weight and the results are still good regarding perceptual similarity.

It has to be noted that tweaking with the weight of each algorithm will affect the perceptual similarity and an ideal weight distribution could be found.

But if we are looking for ideal results going of only on color descriptors is not the way to go, there is much more to picture similarity than just the color of the picture.

('0054.png', '0073.png')



As we talked before this pair is the histogram model’s highest score, and it is computationally and perceptually almost identical. On the first look it is hard to find a difference.

('0113.png', '0132.png')



('0132.png', '0145.png')



'0140.png', '0152.png'

'0082.png', '0208.png'

When we look at the images in the top five we can tell that high comptutational similarity leads to high perceptual similarity when using pictures that have a defining color characteristic. But the pairs are not perfect.

For example in pairs four and five a person would pair 0082.png with 0208.png instead of the current arrangement, because of the bee in the middle of the flower and overall position of the flower.

That is one reason why when comparing similarity of images one needs more than color descriptors to develop a program that finds similar pictures. Because although all the color moments, histograms and other descriptors might match, the pictures might not be so similar.

When we look at the second and third pair and their position in the table for the individual algorithms we can see that only one of the algorithms consider pair two a top five match, while pair three is nowhere to be found. This means that although one algorithm might find a pair to be really similar it might not be such high scoring in other algorithms.

But to appear in the top five combined the pairs must be in the top 1% of pictures in all the algorithms.

On the other hand we can have two pictures that are perceptually almost identical but the color descriptors might not be so similar. A picture can have its colors slightly edited and the change might be hard to perceive for humans but it will make a lot of difference in the algorithm.

Conclusion

This project demonstrated the effectiveness of using color descriptors to measure image similarity, exploring three models: color moments, color correlograms, and histograms.

Each model offers a unique approach to capturing color distribution, with the color moments focusing on statistical properties, the color correlogram incorporating spatial relationships, and the histogram providing a straightforward color frequency analysis.

By evaluating the models individually and in combination, we observed that combining different models improved the overall accuracy of image similarity, especially for images with distinct color characteristics.

While these methods proved useful, they also have limitations, such as sensitivity to lighting conditions and a lack of spatial information in some cases.

This suggests that color descriptors alone may not be sufficient for achieving perfect perceptual similarity, and further refinement or the inclusion of additional features is necessary for more complex image analysis tasks.

Used Literature

G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. IEEE

Workshop on Applications of Computer Vision,

Jing Huang, S. R. Kumar, M. Mitra, Wei-Jing Zhu and R. Zabih, "Image indexing using color correlograms," Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, 1997,

Noah Keen, Color Moments, 0341091, February 10, 2005

Michael Freeman (2005). The Digital SLR Handbook. Ilex. ISBN 1-904705-36-7.