Implementation of a simple Content-Based Image Retrieval system.

Carlos Morote García

David Lorenzo Alfaro

Universidad Politécnica de Madrid
ETSIINF

March, 2022

Contents

1 Motivation **2**

2 State of the art **2**

2.1 Colour-histogram techniques 2

2.2 Colour search 4

2.3 Keypoint descriptors 4

3 Implementation **4**

3.1 Data preprocessing 5

3.2 Colour-based retrieval 5

3.3 Histogram-based retrieval 6

3.4 SIFT 7

References **8**

A Additional results. **10**

1 Motivation

Content-based image retrieval (CBIR) aims at finding similar images from large scale datasets against query images. The degree of commonality between images is often computed accounting for (i) a set of representative features of each sample, and (ii) a set of similarity measures, e.g., distance functions, that quantify the extent to which each feature in each pair of samples matches. Visual cues such as colour, texture and shape are among the most prominent image feature descriptors, and exploiting the appropriate ones for a given problem is key to the success of the retrieval process.

In this work, we review various techniques that enable users retrieve images, provided a set of meaningful search criteria. We state such criteria to be meaningful in the sense that they are easy and fast to formulate, i.e., effort and time required for the user to specify search conditions are measurably lesser than manually inspecting the dataset; and in such manner that they help to best describe and discriminate target images.

Specifically, data under consideration, published in (Kolman, 2021), is a collection of images of 143 different Pokemons, fictional creatures characterized by their inter-class heterogeneous colours and shapes (see Figure 1). Regardless of the very problem domain, what we hereby aim at reviewing is a set of simple techniques that may be transversely applicable to other paradigms of interest where those descriptors are of vital relevance. Namely, we explore the use of different histogram comparison methods, unsupervised classification techniques to yield the most relevant colours of an image, and algorithms for Keypoint description such as SIFT (Lowe, 1999).

2 State of the art

2.1 Colour-histogram techniques

One of the most common approaches to compare the colour content between images consists of comparing their colour histogram. First introduced in (Swain & Ballard, 1991), this idea builds on top of the computational representation scheme typically used to encode the pixel values of an image, each corresponding to a visible colour. Colours can be represented as a combination of some set of base components (e.g., in RGB colour space, the red, green and blue components). Based on this rationale, per base component colour histograms can be computed and subsequently compared by means of a distance function (e.g., Euclidean distance) to identify relative proportions of pixels within specific values. Accordingly, the semantic-wise interpretation of this technique is that similar images contain similar proportion of certain colours (Jain & Vailaya, 1996).

From the computational standpoint, this feature descriptor is rather inexpensive and suitable for real-time CBIR systems, which combine this technique along with other texture and shape descriptors to find similar images for a query image, e.g., SIMPLIcity (Wang, Li, & Wiederhold, 2001), CIRES (Iqbal & Aggarwal, 2002) or QBIC (Flickner et al., 1995).

An outline of a basic color-histogram approach to CBIR, further described in (Jain & Vailaya, 1996) is the following.

Figure 1: Individuals in the dataset are heterogeneous in colour and shape.

1. Read RGB pixel values of the query and database images.
2. Compute n-bin normalized histograms for each base component of each image.
3. Compute the pairwise Euclidean distance between the query histograms and those of each image in the database.
4. Sort the database images in descending order of Euclidean distance scores to the query image and return as result.

It is worth noting that, other than Euclidean distance, there is plethora of metrics to measure commonality between histograms. Indeed, because a normalized histogram constitute a probability density function, PDF, matching two histograms is analogous to comparing how two distributions match with each other. Sung-Hyuk Cha provides an exhaustive survey on similarity measures between PDFs (Cha, 2007), suitable for many pattern recognition problems such as classification, clustering, and retrieval problems.

In spite of its simplicity, a shortcoming of this basic colour histogram comparison algorithm lies in its inability to consider information about the spatial distribution of the colours along images, i.e., for two images to be actually similar, not only they need to feature common colours, but also to have them spatially arranged similarly. Further research have shown how to effectively account for such information by, for instance, utilizing advanced vector quantization strategies (Jeong, Won, & Gray, 2003).

That notwithstanding, the use of colour quantization techniques (i.e., those that deal with grouping colours into more general, representative bins) can lead to perceptually similar colours being quantized in to different bins. Utilizing colour spaces and similarity measures that effectively allows for perceptually meaningful comparison of colours (e.g., CIELUV or CIELAB) (Wyszecki & Stiles, 1982) can alleviate this problem. There are several proposals in the literature which attempt to overcome this, along with other renowned problems attributed to the classical approaches to compare histograms. One is described in (Lu & Phillips, 1998), which introduces the idea of Perceptually Weighted Histograms (PWHs).

Essentially, to induce a PHW, instead of dividing each colour channel by a constant quatization step when obtaining histogram, they compute the ten most similar representative colours (encoded in CIELUV) for each pixel, and assign each of them a weight, which is inversely proportional to the colour distances. Euclidean distance (L_2 distance) is used to compute distance between an arbitrary pair of colours i and j defined by their respective L, u, and v base components, i.e., $d_{i,j} = \sqrt{(L_1 - L_2)^2 + (u_1 - u_2)^2 + (v_1 - v_2)^2}$, and the weight assigned for a pixel to the k^{th} bin is $w_k = \frac{1/d_k}{(1/d_1)+\ldots+(1/d_m)}$, where m is the total number of bins.

2.2 Colour search

Yet another approach of colour retrieval consists in querying for colour values that will be searched for in images from a database. This entails capturing salient colours of each image in the database, e.g., via clustering or back-projection of binary colour sets (Smith & Chang, 1996). This type of image retrieval is, to date, available in Google's image search engine. Albeit conceptually simple, the problem of colour perception becomes even more challenging. As aforementioned, the use of colour spaces with improved perceptual uniformity, such as CIELAB, CIELUV or Munsell is of help to overcome this shortcoming.

2.3 Keypoint descriptors

Keypoint descriptors are aimed at (i) extracting distinctive invariant features from images, (ii) performing reliable matching between different views of an object or scene, and (iii) providing invariance w.r.t scale changes, rotations, affine distortions, viewpoints, noise, illumination (Ramalingam, 2017). Some popular algorithms are SIFT (Lowe, 1999), FAST (Rosten & Drummond, 2006), BRIEF (Calonder, Lepetit, Strecha, & Fua, 2010), or ORB (Rublee, Rabaud, Konolige, & Bradski, 2011), which combines the FAST and BRIEF algorithms.

In this work, we propose the use of the SIFT algorithm (Scale Invariant Feature Transform) which, as stated by its author, is a method for image feature generation which transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling and rotation, and partially invariant to illumination changes and affine or 3D projection. The algorithm can be divided in four main stages:

1. Scale-invariant feature detection. Key locations are defined as maxima and minima of the result of difference of Gaussian kernels applied in space scale.
2. Feature matching and indexing. To index and store the identified SIFT keys, Lowe uses a modification of the k-d tree, the best-bin-first search method (Beis & Lowe, 1997), which allows for identifying nearest neighbours in an efficient fashion. Nearest neighbours are those minimizing the Euclidean distance from a target descriptor vector.
3. Cluster reliable model hypotheses using the Hough transform to search for keys that agree upon a particular model pose.
4. Verify identified clusters by a least-squares solution for the affine projection parameters relating the model to the image. When at least 3 keys agree on the model parameters with low residual, there is strong evidence for the presence of the object.

3 Implementation

In this section we discuss the main highlights of the implementation of a basic content-based information retrieval system. The implementation is in *python*, relying mainly on the open-source library for computer vision, *openCV*[1]. We also utilize the *numPy*[2] library to operate on images via high-level operations robustly optimized for multi-dimensional arrays. Also,

[1] `https://opencv.org/`
[2] `https://numpy.org/`

the *pickle*[3] library comes handy to create and retrieve portable persistent representations of static information useful for the retrieval process, enhancing efficiency of the system and allowing it to scale transparent to the user.

Besides, special emphasis has been placed in producing modular, maintainable and well documented[4] code that could be trivially utilized on other datasets, or extended to include other image descriptors.

3.1 Data preprocessing

The original dataset is composed of 17 thousand images of Pokemons with little preprocessing and heterogenous spatial extents. In order to reduce space requirements and to achieve a noticeable speedup in the retrieval process we downsample images, constraining them to have a fixed width of 255 pixels and a proportional reduction in height to preserve the original aspect ratio. Besides, the original dataset features between fifty and two hundred images for each creature. We sample sixteen random of them from each directory, totalizing 2288 images.

3.2 Colour-based retrieval

As aforementioned, this type of retrieval involves users to input a colour, aiming at searching for images where that colour is relevant. Our approach here consists of first computing the dominant colours of each image in the database via k-means[5]. In all our experiments, we use the five most dominant colours, albeit larger values can lead to more precise results (finer representation of the image). Then, quality of matching between the query colour and each image in calculated accounting for its most relevant colours, where the similarity score can be computed either (i) selecting the colour among the most relevant that minimizes the Euclidean distance to that of the query (Equation 1), or by (ii) computing the sum of Euclidean distances between the query colour and each dominant colour weighted by its normalized frequency of appearance (i.e., its relative relevance) (Equation 2). Notice that in order to compare colours in a perceptually meaningful fashion, we encode them usign the CIELAB colour space.

$$d_{q,I} = \min_{c \in I} \sqrt{(L_q - L_c)^2 + (a_q - a_c)^2 + (b_q - b_c)^2} \tag{1}$$

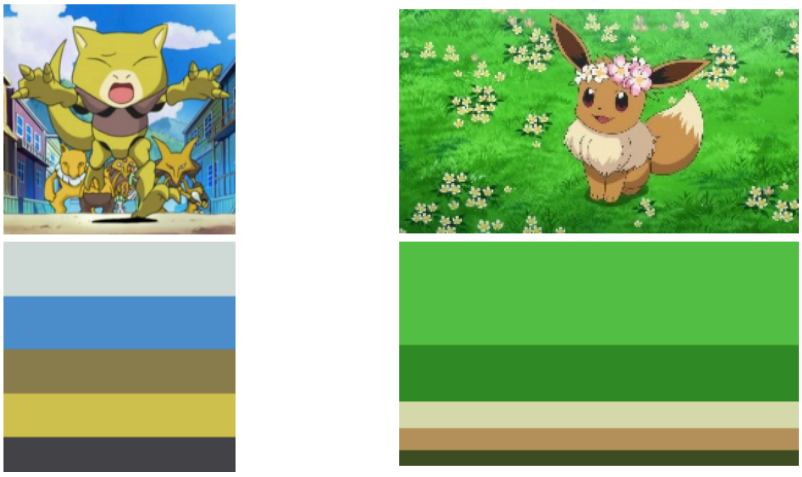
$$d_{q,I} = \sum_{c \in I} \sqrt{(L_q - L_c)^2 + (a_q - a_c)^2 + (b_q - b_c)^2} * f(x, I) \tag{2}$$

Where I is an image, c is a relevant colour in image I, q is the query colour, L, a and b are the base components of each colour, encoded in the CIEL*a*b colour space; and $f(x, I)$ is a function that computes the normalized frequency of appearance of colour x in image I.

[3] `https://docs.python.org/3/library/pickle.html`

[4] Documentation for the code is available at `https://irei-cbir.readthedocs.io/en/latest/`

[5] Because images are high-dimensional in nature, this is a very costly process, being necessary to store (e.g., vi serialization) these computations in order for this approach to be suitable in real-time applications, a very much desirable feature in information retrieval systems.



(a) (b)

Figure 2: Examples of the top five most dominant colours of two images. Per colour relevance is encoded in the height of the colour grid. This naive approach can lead to background colours having more relevance than the target objects (b).

In general, we find both score schemes to work rather decently. Empirically, the frequency weighting similarity score favours retrieving images where the colours close to that being queried is, in absolute terms, more present, whereas the minimum Euclidean distance criterion retrieves images featuring a dominant colour closest to the query colour (see Figures 5 and 6).

3.3 Histogram-based retrieval

As heretofore mentioned, histograms are a powerful, inexpensive image descriptor which has been shown to be a standalone decent technique to retrieve meaningful images, given another image query - or more specifically, another histogram(s). In this work, we first explore the use of a straightforward histogram comparison method, consisting in comparing per channel histograms of images in RGB space, utilizing the high-level directives in OpenCV to that end. Thus, in this regard we solely develop wrapper methods integrating IR specific tasks and scoring methods.

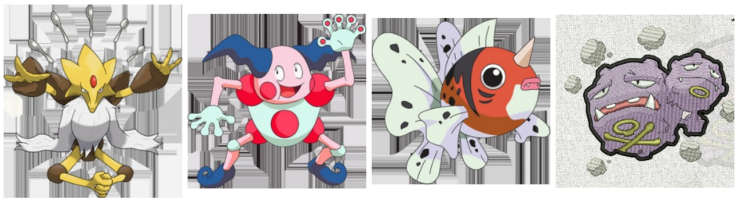
A possibly smarter approach consists in computing and comparing histograms of regions in images, to subsequently aggregate the similarity scores obtained in each comparison to obtain the degree of commonality between a pair of images. This technique allows to compare colour distributions in a more meaningful fashion, because it accounts for the spatial distribution of the colours along the image. However, the number of regions to be compared needs to be manually tuned; plus they introduce some overhead that can be of vital importance in real-time applications. In our experiments, we segment images into equally-sized regions of images, in grids of 5x5. The size of the grid was empirically tuned, striking a balance between the overhead in computational complexity and the increase in performance.

In general, we find that grid-wise histogram comparison outperforms the classical comparison approach. Nevertheless, because all resultant regions in an image are assigned the same weight, we often found that some images topping the ranking have a similar background

to that in the query, whilst the Pokemon in the image (the object of ulterior interest) may be very different, in terms of colour, to the one in the query (see Figure 3).

(a) Query image

(b) Standard histogram descriptor.



(c) Smart grid histogram descriptor.

Figure 3: Top 4 results using histogram-based retrieval for query (a), using the Bhattacharyya distance, along with standard histogram descriptor (b) and the (smart) 5x5 grid histogram descriptor (c).

Furthermore, the are four different histogram comparison metrics available in *OpenCV*[6]. Namely, the correlation, chi-square, intersection and Bhattacharyya distance scores. Upon perfoming several queries with them, we find Bhattacharyya distance, along with the intersection method to perform consistently well. The correlation comparison method delivers decent performance too. Results from Chi-square method, however, are not semantically meaningful and are clearly suboptimal (see Figures 7, 8).

3.4 SIFT

With respect to SIFT, it is worth noting that we have found cases in which keypoints are sensible to variances in rotations or translations. For example, in Figure 4b, the Pokemon depicted is the very same in both images, but one of them is mirrored. In this case, keypoint matching does not seem to work properly. That notwithstanding, when one of the images is mirrored to match the alignment of the other, keypoint matching performs decently (Figure 4a).

[6] `https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html`

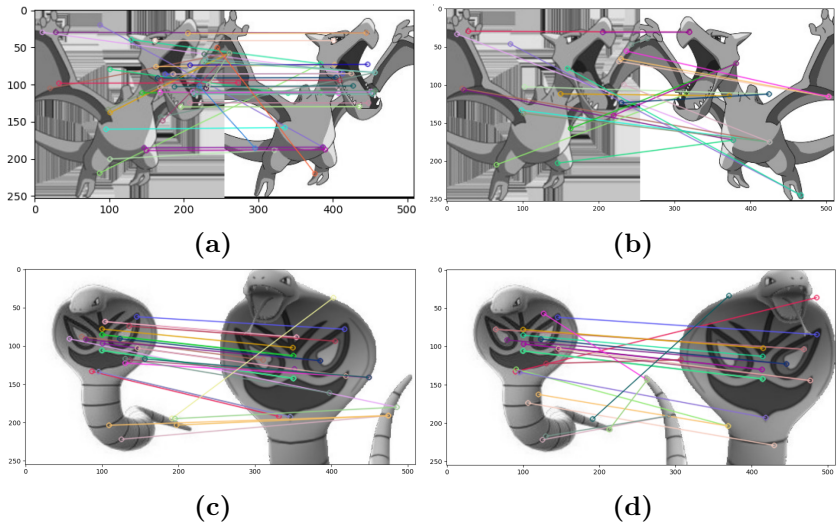


Figure 4: SIFT matches between several images.

References

Beis, J., & Lowe, D. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of ieee computer society conference on computer vision and pattern recognition* (p. 1000-1006). doi: 10.1109/CVPR.1997.609451

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6314 LNCS*, 778-792. Retrieved from `https://link.springer.com/chapter/10.1007/978-3-642-15561-1_56` doi: 10.1007/978-3-642-15561-1_56

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, *4*, 300–307.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., … Yanker, P. (1995). Query by image and video content: the qbic system. *Computer*, *28*(9), 23-32. doi: 10.1109/2.410146

Iqbal, Q., & Aggarwal, J. (2002). Cires: a system for content-based retrieval in digital image libraries. In *7th international conference on control, automation, robotics and vision, 2002. icarcv 2002.* (Vol. 1, p. 205-210 vol.1). doi: 10.1109/ICARCV.2002.1234821

Jain, A. K., & Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, *29*(8), 1233-1244. Retrieved from `https://www.sciencedirect.com/science/article/pii/0031320395001603` doi: https://doi.org/10.1016/0031-3203(95)00160-3

Jeong, S., Won, C., & Gray, R. (2003, 11). Image retrieval using color histograms generated by gauss mixture vector quantization. *Computer Vision and Image Understanding*, *94*. doi: 10.1016/j.cviu.2003.10.015

Kolman, M. (2021, jun). *First Generation Pokemon Images.* Retrieved 2022-03-24, from `https://www.kaggle.com/datasets/mikoajkolman/pokemon-images`

-first-generation17000-files

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision*, *2*, 1150–1157. doi: 10.1109/ICCV.1999.790410

Lu, G., & Phillips, J. (1998). Using perceptually weighted histograms for colour-based image retrieval. In *Icsp '98. 1998 fourth international conference on signal processing (cat. no.98th8344)* (Vol. 2, p. 1150-1153 vol.2). doi: 10.1109/ICOSP.1998.770820

Ramalingam, S. (2017). *Keypoints and descriptors.* Retrieved from https://www.cs.utah.edu/~srikumar/cv_spring2017_files/Keypoints&Descriptors.pdf

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3951 LNCS*, 430-443. Retrieved from https://link.springer.com/chapter/10.1007/11744023_34 doi: 10.1007/11744023_34

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. *Proceedings of the IEEE International Conference on Computer Vision*, 2564-2571. doi: 10.1109/ICCV.2011.6126544

Smith, J. R., & Chang, S.-F. (1996). Tools and techniques for color image retrieval. In *Electronic imaging.*

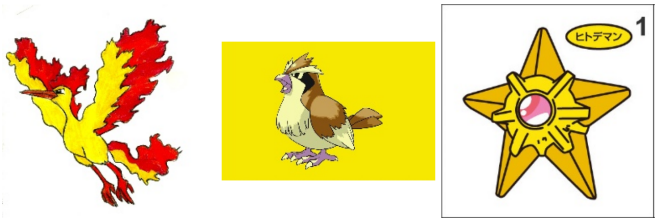
Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision 1991 7:1*, *7*(1), 11–32. Retrieved from https://link.springer.com/article/10.1007/BF00130487 doi: 10.1007/BF00130487

Wang, J., Li, J., & Wiederhold, G. (2001). Simplicity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(9), 947-963. doi: 10.1109/34.955109

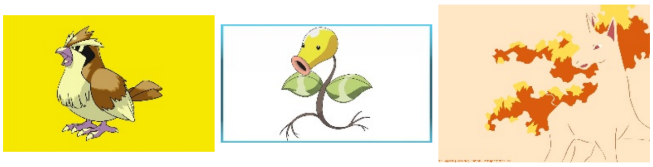
Wyszecki, G., & Stiles, W. (1982). Color science: Concepts and methods, quantitative data and formulae, 2nd edition..

A Additional results.

(a) Query colour

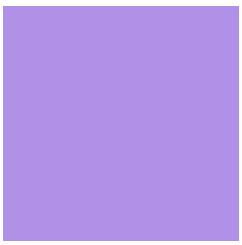


(b) Top 3 results **without** frequency weighting scheme.



(c) Top 3 results **with** frequency weighting scheme.

Figure 5: Top 3 results using colour-based retrieval when querying for yellow images (a), using the minimum distance score scheme (b) and the frequency weighting score scheme (c).



(a) Query colour

(b) Top 3 results **without** frequency weighting scheme.

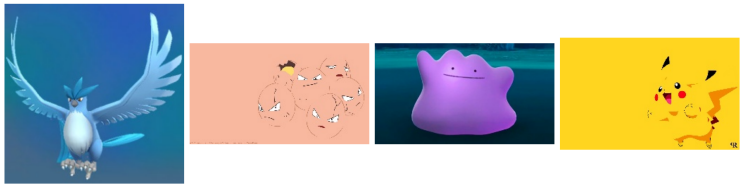
(c) Top 3 results **with** frequency weighting scheme.

Figure 6: Top 3 results using colour-based retrieval when querying for purple images (a), using the minimum distance score scheme (b) and the frequency weighting score scheme (c).

(a) Query image

(b) Bhattacharyya distance

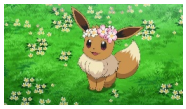
(c) Correlation



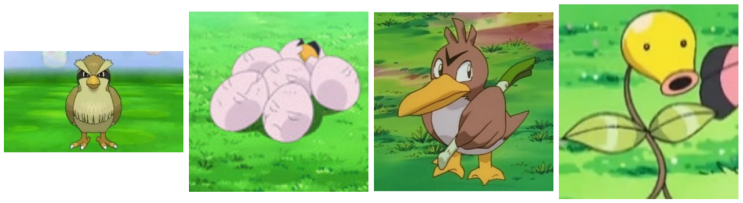
(d) Chi-square

(e) Intersection

Figure 7: Top 4 results for query (a), using different similarity metrics for histogram comparison.

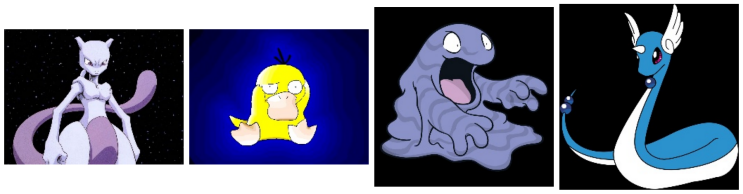


(a) Query image



(b) Bhattacharyya distance

(c) Correlation



(d) Chi-square

(e) Intersection

Figure 8: Top 4 results for query (a), using different similarity metrics for histogram comparison.