Object Color Estimation with Dominant Color in Automated Image Semantic Description Generation Tasks

Aghasi S. Poghosyan
Institute for Informatics and Automation Problems of NAS RA
e-mail: agasy18@gmail.com

Abstract

The automated image tagging is an important part of modern search engines. The generated image tags can be constructed from object names and their attributes, for example, colors. This work presents an object color name detection real-time algorithm. It is applicable to any automatic object detection and localization systems. The presented algorithm is fast enough to run after the existing real-time object detection system, without adding visible overhead. The algorithm uses k-means to detect the dominant color and selects the correct name for the color via Delta E (CIE 2000).

Keywords: Dominant color, Image color palette, Color name selection, Object color detection, Image semantic analysis.

1. Introduction

Automated image semantic description generation is one of the fundamental problems in computer vision. One of the ways to generate a semantic description of the given image is the object detection and localization. There are many solutions, which can find objects by localizing and naming them. In recent years, neural networks have become a leading method for high-quality object detection tasks. They have enough accuracy to be used in different areas, such as self-driving cars, robotics, automatic surveillance and also in web search engines. In web search engines, the object detectors are used for automated image tagging. These tags allow the user to find images without any annotations using a simple keyword search. The simplest way to automatically generate image tags is to detect objects in the image and use the names of objects as tags. This kind of tags do not allow the user to find images, which contain an object with a specific attribute. To solve this issue, we can add additional attributes for each object. One of the main attributes for each object is color. This kind of addition will allow users to search, for example, a “red car” or a “white cat” and find the corresponding images.
This work presents an object color name detection algorithm, which is applicable to automatic object detection systems. The presented algorithm is fast enough (28 ms on i7 CPU) to run after the existing object detection real-time system without adding visible overhead on each iteration.

2. Related Works and Methodology

The modern object detectors are based on Convolutional Neural Networks (CNN), such as Faster Region-based Convolutional Neural Network (Faster R-CNN) [1], Region-based Fully Convolutional Network (R-FCN) [2], Multibox [3], Single Shot Detector (SSD) [4] and YOLO: Real-Time Object Detection [5]: for example, networks can be used to generate image tags from the detected object names.

There are networks which, can generate a single caption [6], [7] for the given image. There are also networks which can generate the image caption and detect the objects [8], [9]. Those works are not computing color for each object. All of these works can be used as object detectors for this work.

In [10], the authors present a method, which assigns a color attribute to a recognized object. They are using an image segmentation method based on graph cuts. After segmentation, they are filtering the results by “Attribute Recommender” [10]. This approach is slightly slow for real-time systems.

This work introduces a color name extraction method, which has a low computational cost. To exact color name for each detected object, we will detect the dominant color and then select the correct name for it.

3. Dominant Color Detection

At first sight, an image dominant color detection is a simple task, and we can compute the mean color for an image. But we will be disappointed, because we will get something close to gray (see Fig. 1).

![Fig. 1: Image and its mean color.](image)

The second approach that we can apply, we can split the image as a grid, and then compute the mean value for each grid’s cell. This approach will be approximated to a downscaled image with an interpolation algorithm, which uses neighbor pixels, such as liner, bicubic or spline methods.
As we see in Fig. 2, we can’t select the central pixel color as the image dominant color of the downscaled image.

To solve this problem, we will use a clustering algorithm to extract a color palette from the image. At first step, we will downscale our image to 100x100 dimensionality via bicubic interpolation method (see Fig. 3). This step will remove rarely encountered pixels by including them in the final pixel of the downscaled image. In addition, this technique will reduce the number of pixels, which will have a huge impact on the clustering algorithm. After that, we will cluster the image pixels in sRGB (standard Red Green Blue) color space via the k-means [11] clustering method. Experimentally we found that our color palette is more meaningful when the cluster count is equal to 9. The k-means cauterization will give us cluster indices (labels, per pixel mappings to their cluster) and the cluster center values (means). To get the image color palette, we will sort by clusters size in descending order. Each palette’s color will correspond to cluster centers (see Fig. 3). The first color of the pallet will be selected as the dominant color for the image.
4. Color Name Selection

As we have a dominant color value in sRGB color space, now we need to select a name for this color. Selection will be made from a predefined color table (see Table 1). We will compare each table’s color with our computed value and find the nearest one. But we can’t do this using the Euclidean distance (1), because visually close colors are not located near in sRGB space.

\[
\text{distance} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}.
\]  

One of the best algorithms to compute the color visual difference (distance) - \( \Delta E \) (2) between a sample color \((L_2, a_2, b_2)\) and a reference color \((L_1, a_1, b_1)\) is Delta E (CIE 2000) [12]:

\[
\Delta E = \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right)\left(\frac{\Delta H'}{K_H S_H}\right)}.
\]  

In equation (2) \(\Delta L', \Delta C', \Delta H'\) and \(R_T\) values are computed by equations (3)-(4). We will take \(K_L = 1, K_C = 1, K_H = 1\) by default.

\[
\Delta L' = L_2 - L_1, \quad \Delta C' = C_2' - C_1',
\]  

\[
\Delta H' = 2\sqrt{C_1' C_2'} \sin \left(\frac{\Delta h'}{2}\right), \quad R_T = -R_C \sin(2\Delta \theta)
\]  

\(R_C\) and \(\Delta \theta\) values are computed by equations (5)-(10).

\[
\bar{H} = \begin{cases} 
\frac{h_1' + h_2' + 360°}{2}, & \text{if } |h_1' - h_2'| > 180° \\
\frac{h_1' + h_2'}{2}, & \text{otherwise.}
\end{cases}
\]  

\[
T = 1 - 0.17 \cos \left(\bar{H}' - 30°\right) + 0.24 \cos \left(2\bar{H}'\right) + 0.32 \cos \left(3\bar{H}' + 6°\right) - 0.20 \cos(4\bar{H}' - 63°)
\]  

\[
\Delta h' = \begin{cases} 
\frac{h_2' - h_1'}{2}, & \text{if } |h_2' - h_1'| \leq 180° \\
\frac{h_2' - h_1' + 360°}{2}, & \text{else if } |h_2' - h_1'| > 180° \text{and } h_2' \leq h_1' \\
\frac{h_2' - h_1' - 360°}{2}, & \text{otherwise}
\end{cases}
\]  

\[
\Delta \theta = 30 \exp \left\{-\left(\frac{\bar{H}' - 275°}{25}\right)^2\right\}, \quad \bar{L}' = \frac{L_1 + L_2}{2}, \quad \bar{C}' = \frac{C_1' + C_2'}{2}
\]  

\[
S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}}, \quad S_C = 1 + 0.045\bar{C}' \quad \text{S}_H = 1 + 0.015\bar{C}' T
\]  

\[
R_C = 2 \sqrt{\frac{\bar{C}'}{\bar{C}'^2 + 25}}
\]
$L_1, C'_1, h'_1$ are computed by equations (11)-(12). Also, the $L_2, C'_2, h'_2$ values are computed in the same way.

$$C_1 = \sqrt{a_1^2 + b_1^2}, \quad G = \frac{1}{2} \left( 1 - \frac{C'}{C' + 257} \right), \quad a'_1 = a_1(1 + G) \quad (11)$$

$$C'_1 = \sqrt{a'_1^2 + b'_1^2}, \quad h'_1 = \begin{cases} \arctan \left( \frac{b'_1}{a'_1} \right), & \text{if } \arctan \left( \frac{b'_1}{a'_1} \right) \geq 0, \\ \arctan \left( \frac{b'_1}{a'_1} \right) + 360^\circ, & \text{otherwise}. \end{cases} \quad (12)$$

We will use metric (2) to find the nearest color and get the color name from Table 1.

<table>
<thead>
<tr>
<th>Table 1: CSS color table.</th>
</tr>
</thead>
</table>

4. Conclusion

We have developed a fast object color name extraction method. To test the algorithm on end to end system, we used a system, which generates the image caption, detects and localizes the objects [9]. For each detected object, we computed the object dominate color and found the correct name for that color. Each color name was used as an attribute for that object (see Fig. 4). For detections, which were predicated as humans, we did not use the color attribute.
Fig. 4: Image caption generation, object detection, localization and color estimation, second row presents dominant color detection process.

References


Object Color Estimation with Dominant Color in Automated Image Semantic Description Generation Tasks


Աղասի Ս. Պողոսյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ

e-mail: agasy18@gmail.com

Աշխատանք

Այս աշխատանքում ներկայացված է օբյեկտների գույնի անվանման հայտնաբերման արագ ալգորիթմ, որը կիրառվում է ավտոմատացված օբյեկտները հայտնաբերող համակարգի հետ։
Определение цвета объекта с использованием доминантного цветного в задачах автоматической генерации семантического описания изображений

Агаси С. Погосян

Институт проблем информатики и автоматизации НАН РА
e-mail: agasy18@gmail.com

Аннотация

Автоматическая маркировка изображений является важной частью современных поисковых систем. Сгенерированные маркировки изображений могут быть построены из имен объектов и их атрибутов, например, цвета. В данной работе представлен быстрый алгоритм определения имени цвета объекта. Это применим к любым системам автоматического обнаружения и локализации объектов. Представленный алгоритм достаточно быстр (28 мс) для запуска с существующими системами обнаружения объектов в реальном времени, без видимых дополнительных расходов. Алгоритм использует метод k-средних для обнаружения доминирующего цвета и Delta E (CIE 2000) для выбора правильного имени цвета.

Ключевые слова: доминирующий цвет, цветовая палитра изображения, выбор названия цвета, определение цвета объекта, семантический анализ изображения.