JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Color detection using pandas an Opencv

G. Sangeetha Lakshmi, R. Hari Priya, S. Sree Mahalakshmi

M.S (It)., M.Phil., Assistant Professor & Head In Computer Science And Applications D.K.M College For Women (Autonomous), Vellore

M. SC., PG Student in Computer Science, D.K.M College for women (Autonomous), Vellore

M. SC., PG Student in Computer Science, D.K.M College for women (Autonomous), Vellore

ABSTRACT

Color detection is the process of detecting name of the color. Here this is easy task for human to detect the color and choose one. But computer cannot detect the color easily. This is tough task for computer to detect the color easily. So that's why we choose this project. Many of the project and research papers are written on this problem. But we use different techniques for this project. Pandas and openCV libraries used in python languages. Open Source Computer Vision Library. Open CV was designed for computational efficiency and with a robust specialise in real-time platform that gives video and audio encoding infrastructure.

1. INTRODUCTION

Before going into the speculations of the project it is important to know the definition of color detection. It is simply the process of identifying the name of any color. It is obvious that humans performs this action naturally and do not put any effort in doing so. While it is not the case for computers. Human eyes and brain work in coordination in order to translate light into color. Light receptors that are present in eyes transmit the signal to the brain which in turn recognizes the color. There is no exaggeration in saying that humans have mapped certain lights with their color names since childhood. The same strategy is useful in detecting color names in this project. Three different colors Red,Green and Blue are being tracked by utilising the fundamentals of computer vision. After successful compilation when we execute the code a window redirects the image displayed on it whose path is given as an argument.

2. LITERATURE SURVEY

Title: Active contour-based visual tracking by integrating colors, shapes, and motions

Author: Weiming Hu [1], Xue Zhou

Year: 2013

Description: In this paper, we present a framework for active contour-based visual tracking using level sets. The main components of our framework include contour-based tracking initialization, color-based contour evolution, adaptive shape-based contour evolution for non-periodic motions, dynamic shape-based contour evolution for periodic motions, and the handling of abrupt motions. For the initialization of contour-based tracking, we develop an optical flow-based algorithm for automatically initializing contours at the first frame. For the color-based contour evolution, Markov random field theory is used to measure correlations between values of neighboring pixels for posterior probability estimation. For adaptive shape-based contour evolution, the global shape information and the local color information are combined to hierarchically evolve the contour, and a flexible shape updating model is constructed. For the dynamic shape-based contour evolution, a shape mode transition matrix is learnt to characterize the temporal correlations of object shapes. For the handling of abrupt motions, particle swarm optimization is adopted to capture the global motion which is applied to the contour in the current frame to produce an initial contour in the next frame

Title: Evaluating Color Descriptors for Object and Scene Recognition
Author: Koen van de Sande, Theo Gevers

Year: 2009

Description: Image category recognition is important to access visual information on the level of objects and scene types. So far, intensity-based descriptors have been widely used for feature extraction at salient points. To increase illumination invariance and discriminative power, color descriptors have been proposed. Because many different descriptors exist, a structured overview is required of color invariant descriptors in the context of image category recognition. Therefore, this paper studies the invariance properties and the distinctiveness of color descriptors (software to compute the color descriptors from this paper is available from http://www.colordescriptors.com) in a structured way. The analytical invariance properties of color descriptors are explored, using a taxonomy based on invariance properties with respect to photometric transformations, and tested experimentally using a data set with known illumination conditions. In addition, the distinctiveness of color descriptors is assessed experimentally using two benchmarks, one from the image domain and one from the video domain. From the theoretical and experimental results, it can be derived that invariance to light intensity changes and light color changes affects category recognition. The results further reveal that, for light intensity shifts, the usefulness of invariance is category-specific. Overall, when choosing a single descriptor and no prior knowledge about the data set and object and scene categories is available, the OpponentSIFT is recommended. Furthermore, a combined set of color descriptors outperforms intensity-based SIFT and improves category recognition by 8 percent on the PASCAL VOC 2007 and by 7 percent on the Mediamill Challenge.

Title: Spatially Varying Color Distributions for Interactive Multi-Label Segmentation

Author: Claudia Nieuwenhuis and Daniel Cremers

Year: 2012

Description: We propose a method for interactive multi-label segmentation which explicitly takes into account the spatial variation of color distributions. To this end, we estimate a joint distribution over color and spatial location using a generalized Parzen density estimator applied to each user scribble. In this way we obtain a likelihood for observing certain color values at a spatial coordinate. This likelihood is then incorporated in a Bayesian MAP estimation approach to multi-region segmentation which in turn is optimized using recently developed convex relaxation techniques. These guarantee global optimality for the two-region case (foreground/background) and solutions of bounded optimality for the multi-region case. We show results on the GrabCut benchmark, the recently published Graz benchmark and on the Berkeley segmentation database, which exceed previous approaches such as GrabCut [32], the Random Walker [15], Santner's approach [35], TV-Seg [39] and interactive graph cuts [4] in accuracy. Our results demonstrate that taking into account the spatial variation of color models leads to drastic improvements for interactive image segmentation.

3. PROPOSED SYSTEM

In the proposed system, we are introducing the CV database and according to it the number of shades that can be identified using 865 color names along with their RGB and hex values. Whenever the cursor clicks the image, it automatically shows the RGB shades color values.

4. MODULES

Image Capture:

The first step is to fetch a high quality image with resolution. To load an image from a file we use Cv2.imread ().Image should be in working directory or full path of the image should be given. Img=cv2.imread (img path).

Extraction of RGB colors:

In this phase, the 3layered colors are extracted from the input image. All the color images on the screens such as televisions, computer, monitors, laptops and mobile screens are produced by the combination of Red, Green and Blue light. Each primary color takes an intensive value 0 (lowest) to 255 (highest). When mixing three primary colors at different intensity levels a variety of colors are produced. For example: If the intensity value of the primary color is 0, this linear combination corresponds to black. If the intensity value of the primary colors is 1,this linear combination corresponds to white. Index= (color, colorname, hex, R, G, B) Calculate minimum distance from coordinates:The rectangle window is used to display the image with the shades of color. After the double click is triggered, the RCB values and color name is updated. To display an image Cv2.imshow () method is used. By using cv2.rectangle and cv2.putText () functions, the color name and its intensity level can be obtained .text=getColorName (r,g,b)+'R='+str(r)+G=+str(g)+ B=+Str (r).

5. CONCLUSION

In this paper we defined to get the required color field from an RGB image. In this various steps are implemented using openCV platform. The main positive point of this method is its color differentiation of a mono color.

6. RESULT

Fig 1: Waterspout

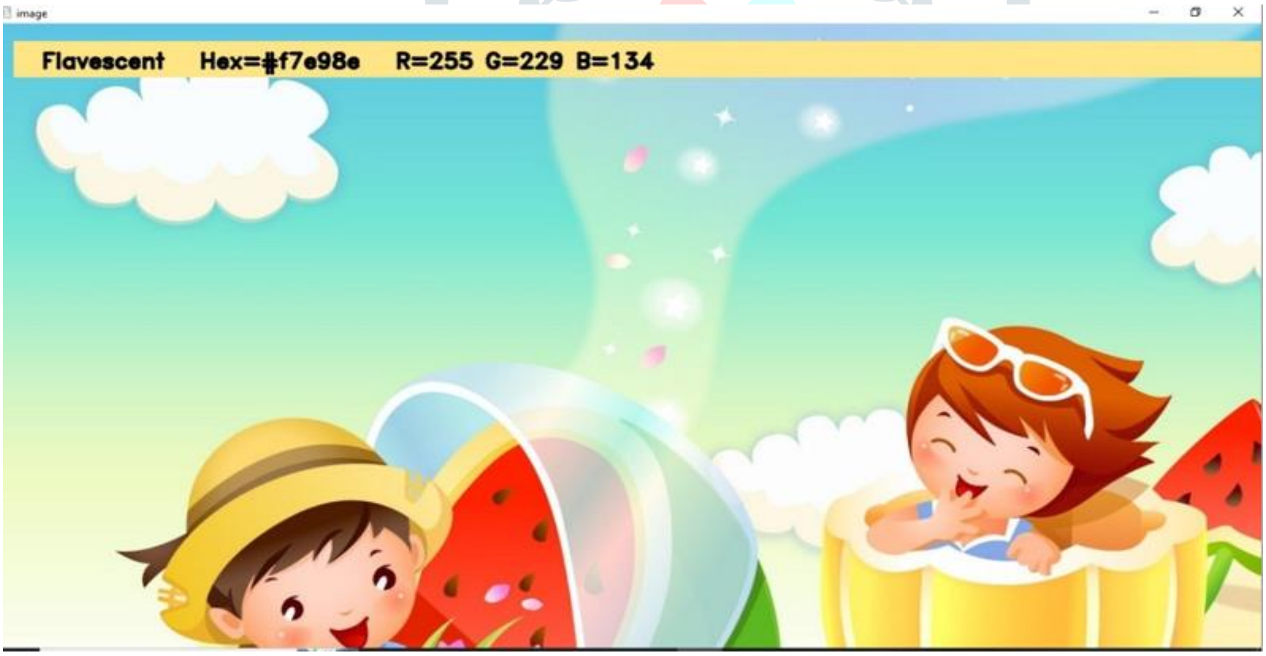

Fig 2: Flavescent

Fig 3: Earth Yellow

7. FUTURE SCOPE

In the future scope, the detection of the edge detection techniques has different other applications like facial detection color conversion for grey scale image etc. That can also be implemented.

8. REFERENCE

[1] Weiming Hu, Xue Zhou, “Active Counter Based Visual Tracking by Integrating Colors, Shapes and Motions”, IEEE Transactions on Image Processing, Vol. 22, No. 5, May 2013

[2] G.M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 9, September 2010

[3] Claudia Nieuwenhuis, “Spatially Varying Color Distributions for Interactive Multi Label Segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 35, No. 5, May 2013

[4] Rosebrock, A., 2020. OpenCv and Python Color Detection - Pyimagesearch. [online]PyImageSearch.

[5] cv-python-color-detection/> J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73

[6] vkisku/Face-Detection-OpenCV. GitHub. (2020). Retrieved 23 June 2020, from https://github.com/vkisku/Face-Detection- OpenCV.

[7] RGB Color Codes Chart Rapidtables.com. (2020). Retrieved 23 June 2020, from https://www.rapidtables.com/web/color/RGB_Color.html.

[8] Rosebrock, A. (2020). Measuring distance between objects in an image with OpenCV - PyImageSearch. PyImageSearch. Retrieved23 June 2020, from https://www.pyimagesearch.com/2016/04/04/measuring- distance-between-objects-in-an-image-with-opencv/.

[9] Zelinsky, A. (2009). Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008) [On the Shelf]. IEEE Robotics & Automation Magazine, 16(3), 100-100. https://doi.org/10.1109/mra.2009.933612.

[10] Naive Bayes Classifier in Machine Learning - Javatpoint. www.javatpoint.com. (2020). Retrieved 23 June 2020, from https://www.javatpoint.com/machine-learning-naive-bayes-classifier.