Computational Color Naming for Human-Machine Interaction

Kondala Rao Jyothi
Dept. of Electronics and Communications Engg.
National Institute of Technology
Rourkela, India
rao.nitrkl@gmail.com

Manish Okade
Dept. of Electronics and Communications Engg.
National Institute of Technology
Rourkela, India
okadem@nitrkl.ac.in

***Abstract*—In this paper, we present two simple methods for automatic color naming, the first one learns the pixel-wise color name annotations from the color-chips and later method learns the image-wise color names from the real-world weakly labelled images. Color information is an important feature for many computer vision applications. Color name as a descriptor finds it's applications in many real-world tasks such as Color Blind assistance, image retrieval and scene understanding. Color naming in images is a challenging problem due to shadows, view angles, illumination conditions and surface reflections. Manual labelling of color names for real-world images in applications like search engines, fashion parsing and tracking is a tedious task and time consuming. The proposed systems for color naming automates the process and avoids human labelling. These methods are based on superpixels in the CIELAB color space. We trained Random Forests classifier with color-chip dataset for pixel-wise color naming and for image-wise color naming it is trained on weakly color labelled image dataset. Both models are tested on real-world images for color name judgments. Experimental results shows that color names learned through these proposed systems have advantages in terms of implementation costs, speed of execution and can be used in real-time applications with low-cost hardware.**

***Index Terms*—Color Naming, SuperPixels, Color-chips, Image-wise Color naming.**

I. Introduction

Color naming within the computer vision context is mapping image pixels to linguistic color labels. We perceive colors and use color names in our day to day life without pretty much effort to describe the real-world scenario that surrounds us. We have the ability to name the colors reliably and distinguish even minute color changes. Color naming task has been under study of different fields such as anthropology, visual psychology and linguistics. The starting point that motivated for research community about this topic has been studied in the field of anthropology by Berlin and Kay [1]. They studied color naming across different languages and cultures to arrive at striking similarities in the use of color names. They stated the universality of color name categories and defined a set of 11 basic color categories that could be found in the the most evolved languages. These are red, green, blue,brown, yellow, pink, purple, orange, black, gray and white. Since then several others have studied,extended and confirmed their results in different fields [2]–[6], [8]–[10]. Color Naming finds its applications in computer vision tasks ranging from image search engines on the web, image retrieval,visual tracking, object recognition and texture recognition [11]. In other applications state-of-the-art results can be obtained by exploiting color description as a feature.

Color naming requires mapping *RGB* values of image pixels

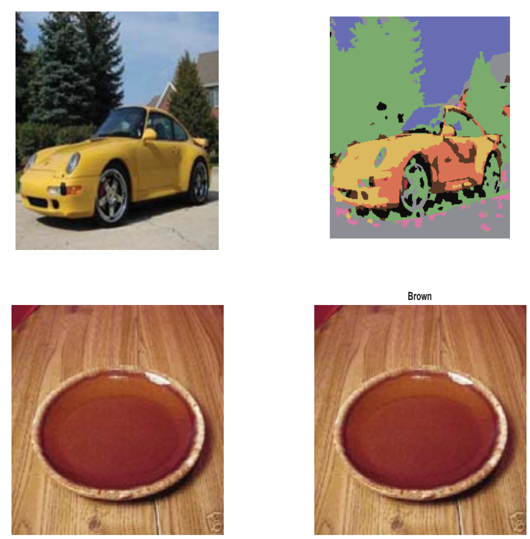

Fig. 1: Examples of Proposed Color Naming systems: First row original image and its pixel-wise color named image. Second row Original Image and its Image-wise color named image

to a set of pre-defined color names [4]–[6], [8]–[10], [12], [13]. Fig.1 shows original images in first column and their pixel-wise color named and image-wise color named images in the second column. Image pixels are labelled with corresponding color name values. Berlin and Kay [1] in their linguistic study chose human subjects for their experiment. Then subjects were asked to give color names to a set of color chips, after that subjects were provided with color terms and asked to choose most relative color chips for the provided color term. Benavente et al. [4] proposed a parametric model that assigns pixel-level color names based on the distributions of lightness and chromaticity. Here, color categories are modeled as fuzzy sets with each category having a parametric membership function.

The parameters of the membership functions are estimated through a fitting process and the training dataset used for the fitting process is derived from psychophysical experiments. Van deWeijer et al. [2] used histogram features of *LAB* color space as 'words' and applied them to Probabilistic Latent Semantic Analysis (PLSA) model to learn for 'topic' color naming. Benavente et al. [10] presented a dataset obtained from a color naming experiments. The data set consists of color reflectances and their corresponding judgments. For each of these reflectances 11 membership values are given corresponding to each color. Several others have reported their results on real-world images by deploying this color-chips based color names [5], [6], [8]–[10], [14].

Serra et al. [15] and Liu et al. [16] using pixel-wise color naming improved region consistency of color names. In particular, Serra et al. [15] used Markov Random Fields (MRF) and color intrinsic components are extracted from images. In their model, intrinsic color features are extracted using Ridge Analysis of color Distribution (RAD) segmentation scheme, and pixels connected within apurticular ridge are assigned with same color label. However, the RAD method only describes the distribution of *RGB* histogram and is vulnerable when handling complex color distribution. Liu et al. [16] applied the similar Markov Random Fields model to built a label propagation model in which color labels of pixels in normal regions propagates to shadows, illumination and highlighted regions within the same objects's surface. Our proposed method is inspired from works of Van de Weijer et al. [2] who trained their model with real-world images from internet to learn the color names and Cheng et al. [3] who proposed Pedestrian color naming Convolution neural network (PCN-CNN) for person re-idenfication.

In this paper we contribute two color naming methods based on superpixels:

- Pixel-wise color naming which annotates each pixel in the image with a color name.
- Image-wise color naming that gives a global color name caption for the main object in the target image.

II. Data set Description

For the proposed color naming systems we have used two training datasets CC-I and Google Color Name dataset and a testing dataset named Ebay dataset. CC-I dataset is a color-chip dataset and remaining two datasets are collection of images from search engines.

CC-I: This data set consists of 387 color named chips with *CIElab* values and Munsell Hue, value and chroma and corresponding judgments [3]. The color-chips are classified into 11 basic color terms by 10 human subjects with no color blindness. Multiple color names can be assigned to a color-chip when it is difficult judge the color name of the chip. Every color patch labelled is represented by its sRGB values (standard default color space) and a probability distribution over all of the color names. We have modified the dataset so that only *CIElab* values exists and the corresponding membership function with highest value is replaced with relative color label. This modification makes the training dataset simple with three features and a color label.

Google Dataset: This data set is formed by Van De Weijer et al. [2] which consists images retrieved by Google image search engine for 11 basic colors. Data set comprises of 100 images for each of the 11 basic colors. It is a weakly labelled set because it contains some images which do not contain the color of the query and in other cases only a small portion of the images represent queried color.

Ebay dataset: For testing the proposed color name systems, we need human labelled set of object images as a ground truth. This data set consists of images and their hand segmented object areas that correspond to the color name [2]. It is compiled by images from EBay auction website. It consists of four categories of objects: pottery, cars, dresses and shoes. 10 images were collected for each color name which makes 110 images for each category of objects. Fig.2 shows examples images of this data set and their hand segmented masks for each category.

III. Feature Extraction

Color is an important cue in the computer vision and its naming plays a vital role in many computer vision tasks. There are many features that describe the color such as *RGB* values of the pixels, color histograms, Hue, saturation and values. Here in our approach we have taken the *CIELAB* and *RGB* values of the superpixels as features to test our models on images. Superpixel algorithms cluster pixels into tiny and perceptually meaningful regions depending on their spatial and color properties. These extracted regions are used to replace the rigid structure of image pixel grid. They capture redundancy in the image, provide simple structures from which leads to capture image features conveniently, and alleviate the complexities from subsequent computer vision tasks. Many algorithms were proposed to generate superpixels each with it's own advantages. *Simple Linear Iterative Clustering* (SLIC) superpixel algorithm presented by Achanta et al. [9] is faster than other existing superpixel methods, more storage efficient, results in state-of-the-art performance on boundary adherence and exhibits better segmentation performance. The simplicity of SLIC lies in it's use and understanding. A default and only parameter to be set in the algorithm is the desired number of superpixels L. When an image employs this algorithm approximately equally sized superpixels (regions) are created. By default color images are converted to *CIELAB* color space which is device independent, de-correlation among it's color channels and perceptually uniform. Superpixel generation begins with clustering procedure, where L initial clusters centers are initialized to $\lambda_i = [l_i \quad a_i \quad b_i \quad x_i \quad y_i]^T$. These cluster centers are formed uniformly on a image pixel grid, spaced G pixels apart from each other. To get desired number of approximately equally sized tiny superpixels, the grid interval should be taken as $S = \sqrt{N/L}$. The centers of initialized clusters are moved over a 3×3 neighborhood in the direction of lowest gradient position to avoid superpixels on the edges becoming center and to limit the chance a superpixels seeding with a noisy pixel. In the next step algorithm searches for a nearest cluster center within the 2G grid so that it can assign pixel 'i' to it. Because, the size of the search space is limited to 2G grid the number of distance calculations are significantly reduced speeding up of SLIC algorithm compared to conventional k-means clustering in which each pixel has to be compared with all other cluster centers created in the image. This speed up advantage is a direct consequence of the introduction of new distance measure D, which assigns each pixel to it's nearest cluster center. Since the search space expected of a superpixel is a region of size $G \times G$ approximately, the search for similar pixels is done in a region of $2G \times 2G$ grid around the center of superpixel. After each pixel is assigned to the nearest cluster center, cluster centers are updated to the new mean value calculated from all pixels belonging to the cluster and mean vector $[l \quad a \quad b \quad x \quad y]^T$ is updated. To compute a residual error E between the previous cluster center locations and new cluster center locations $L2$ norm is used. Assignment and update steps are repeated iterative manner until error becomes minimum, but from the reports of research community which used this algorithm, it can be stated that 10 iterations are sufficient for most of the images. In the end, disjoint pixels are reassigned to nearby superpixels by a postprocessing which ensures connectivity.

Distance Measure: SLIC superpixels are represented as clusters in the color and image plane space *labxy*, where *lab* color information and xy spatial location of pixel. When the color terms and spatial terms are presented simultaniously defining the distance measure D becomes problematic which can not be found immediately. D computes the distance between a pixel 'i' and cluster center λ_i. A pixel's color is represented in the *CIELAB* color space as $[l \quad a \quad b]^T$, whose range of possible values is known. The pixel's position $[x \quad y]^T$, on the other hand may take a range of values depending on the size of the image. If the distance measure D is not properly defined in *lab* space, it causes inconsistencies in the cluster behavior for different superpixel sizes. In large superpixels, color proximity is outweighed by spacial distances causing more importance to spatial proximity than color, consequent results produces superpixels that do not align to image boundaries. When distance measure is small the number of superpixels created are more and boundaries are perfectly aligned and results in better segmentation performance but the complexity increases. To obtain a single measure from two distances, it is necessary to normalize color proximity and spatial proximity by their respective maximum distances within a cluster, N_s and N_c. Doing so, D' is written as

$$\delta_s = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \tag{1}$$

$$\delta_c = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{2}$$

$$D' = \sqrt{(\frac{\delta_s}{N_s})^2 + (\frac{\delta_c}{N_c})^2} \tag{3}$$

The expected maximum spatial distance within a given cluster should correspond to the sampling interval,$N_s = \sqrt{N/L}$. Determining the maximum color distance N_c is not so straight forward, as color distances significantly vary from cluster to cluster and image to image. This problem can be avoided by fixing N_c to a constant k so that (3) becomes

$$D' = \sqrt{(\frac{\delta_s}{G})^2 + (\frac{\delta_c}{k})^2} \tag{4}$$

IV. Proposed Color Naming Systems

A. Pixel-wise Color Name Annotation

Given an image **I**, our goal is to assign each pixel of **I** with a specific color name c, $\forall$ $c \in$ C = {1,2... ,11}, representing the set of 11 basic color names. Proposed system for color naming takes an image **I**, as an input and outputs an image with pixel-level annotation of color. We attempte pixel-by-pixel color labelling and found that it is a tedious task and not scalable. To overcome this problem, we over segmented the image with superpixel algorithm. These superpixels are given as the feature vector for the classifier trained on color-chip dataset. The algorithm with it's steps is briefly described in the **Algorithm 1**.

Algorithm 1 Pixel-wise Color Naming Algorithm

1: Extracting Features
   - i. Read the input Image I.
   - ii Over Segment the image I with SLIC superpixel algorithm.
   - iii Construct the feature vector with mean values of cluster.

2: Training classifiers
   - i. Train the classifiers with CC-I dataset.
   - ii Tune the classifier parameters with for optimum performance.

3: Color Labelling Image Pixels
   - i. Above trained classifier is tested on feature vectors obtained from test image.
   - ii. Reconstruct Image with color name predicted pixels.
   - iii. Replace pixels within the same cluster with corresponding color palettes.

1) Training: The pixel level classification for a given image is based on the *LAB* values of superpixels, and is carried out by Random forests classifier which has number of decision trees as the only external parameter (hyper parameter). We trained classifier with CC-I color-chip dataset and testing is done on given images from Ebay dataset. Number of decision trees taken are only 150 as the data consists few dimensions i.e *lab* values in CC-I. Trained classification models are fed with superpixels that are taken extracted from image and is given as testing data. Color label is predicted for each superpixel and predicted color labels replace the pixels in the respective superpixels. Out-Of-Bag (OOB) classification error for Random forests classifier is show in Fig.7 which achieves minimum error at 100 decision trees.

TABLE I: Performance of Pixel-wise Color Naming System

Random Forests	Training Time(Sec)	Avarage Execution Time per Image(Sec)
CC-I	0.5	0.3

TABLE II: Pixel annotation Score of proposed Pixel-wise Color naming system and other Methods

Method	cars	shoes	dresses	pottery	overall
Proposed **CC-I**	62	75	82	75	**73.5**
PLSA-std	54	74	75	66	67.30
PLSA-bg	56	76	79	68	70.00
PLSA-ind	56	77	80	70	70.60

2) Results: In our first experiment for pixel-wise color naming, the classifier is trained on samples of CC-I dataset and tested on real-world images for performance analysis. Fig.2 shows the results of our proposed methods. We have even tested our method on complex images with multiple colors. Fig.3 shows the results show that our method works in complex scenarios with images containing many colors and have significant accuracy. Some images with complex backgrounds are tested to analyze the performance of proposed system. In Fig.3 shoes image that has complex background and Fig.1 car image has trees, houses and sky as the background which was color named with 90% accuracy. Table.I shows the performance analysis of the proposed system. The time taken for the training of the model is 0.5 seconds. Testing time is calculated on the 440 images of the ebay datset which contas cars, dresses, pottery and shoes. We have calculated average testing time for the dataset since it contains images of different sizes. The average time taken for each image is 0.3 seconds which suits for color naming in real-world images. The performance metric to calculate the accuracy is pixel annotation score. Pixel annotation score gives the percentage of correctly classified pixels. For calculating the pixel annotation score we have used mask images as ground truth. Ebay dataset provides the masks along with the images. Fig.4 shows the original image and its mask (Ground truth) and color named image with applied mask. Table.II shows the pixel annotations scores of the proposed method and it out performs the PLSA methods [2] in terms of performance.

B. Image-wise Color Naming

The outcome of this method is to predict the best color name possible for the prime object in the image irrespective of background and local variations. Manual annotation of color names for search engines and fashion datasets is a laborious task and time consuming. Automating this process saves time and cost. The method needs the pre-processing as images are different sizes and taken from different devices under different illumination conditions and different backgrounds. Sometimes images are domain specific like fashion datasets where color plays vital role in retrieval process and uploading new content to search engines. Image wise color labelling makes the task simple easy to access and uploading contents [18]. Our method is trained on a weakly-labeled data collected from Google image search engine [2], weak-label here means a color name is given for the entire image with out giving the description of complex backgrounds and other surroundings. Sometimes given color name represents a small portion of the image and no segmentation mask or bounding box is provided to identify the principal object. Fig.5 shows outcome of proposed method where first image is the original image second and third are background subtracted and color named image respectively. The steps involved in this method are briefly discussed in Algorithm 2.

Algorithm 2 Image-wise Color Naming

1: Extracting Features
   i. Load Google Color Name dataset
   ii. Pre-process and Exclude background.
   iii. Oversegment each image to N superpixels.
   iv. Rank the superpixels according to their weights.
   v Take first M ranked superpixels and construct a Feature Vector.
2: Training classifiers
   i. Train the classifiers with feature vector of step 1.
   ii Tune the classifier parameters for optimum performance.
3: Output Color Labelled Image
   i. Read a test Image to be color labelled.
   ii. Above trained classifier Predicts the color label for image.
   iii. Output: Color named image.

1) Pre-Processing: The Google data set contains different images from different sources and different sizes so we have resized the image to 70% of the original image and applied a gamma correction of factor 1.5 to compensate the non-linear luminance of display devices. Background of the images are removed by morphological dilation and erosion process. Background causes for false positives in many cases and plays significant role in color naming as some times the principal object in the image occupies small portion of the entire image, So we removed the background in the feature extraction pipeline and trained our model.

2) Weighted Ranking of Superpixels: Since the dataset consists of different sizes of images, different backgrounds and weakly-labelled images, taking all the superpixels of the image directly results in false positives. To alleviate this problem we have removed the background given the ranks for the superpixels according to their weights i.e the superpixel that contains highest number of pixels gets first rank. If an image **I** is over segmented into L superpixels, it can treated as a set of superpixels.

$$I = \{S_1, S_2, ...S_L\}$$

$$I = \sum_{i=1}^{L} S_i$$

$S_1, S_2...S_L$ are superpixels and $n_1, n_2...n_L$ are the number of pixels in each superpixel and $r_1, r_2...r_L$ are ranks given to the superpixels based on the number of pixels a superpixel contains. The number of pixels in a superpixel are calculates as follows

Fig. 2: Examples of Images from *Ebay dataset* test dataset and their corresponding color named images. First row shows original Images and second row shows pixel-level annotated images

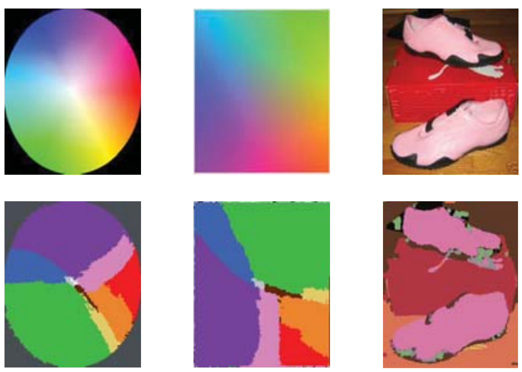

Fig. 3: Proposed Pixel-wise color naming Method on complex images and images with complex background

Fig. 4: Ebay test datset for calculating pixel annotation scores. Orginal image its mask and color named Image with applied mask

$$n_i = \sum_k S_i(k)$$

Where k is the size of the superpixel S_i. Among these L supixels first M ranked pixels are taken as feature vector for training the model.

3) Training: In our proposed method, the training of Randomforests classifier is done on first M (M $\leq$ L) ranked superpixels. The mean value of superpixel with its L, a and b are taken as feature vector, making it $M \times 3$ dimensions. In our experiment we have taken L=500 and M=200 making each sample to have 600-Dimensions. Since our dataset contains 1100 images 100 images for each of the 11 basic colors, training dataset results in 1100×600 dimensions. Number of decision trees taken are 500. The time taken by the model to train on these features is given in the Table.III. The OOB error for the training set is shown in the Fig.7. Increasing the number of decision trees results in increase of complexity of the algorithm significantly.

TABLE III: Performance of Image-wise Color Naming System

Random Forests	Training Time(Sec)	Avarage Execution Time per Image(Sec)
Google Color Name dataset	19.98	0.91

Fig. 5: Image-wise Color Naming system: Original image, Background subtracted image and Image-wise color named image

4) Results: In our second experiment for image-wise color naming, we have trained the model with superpixels extrcted from Google color name dataset and tested on ebay datset. Fig.6 shows some of the results of the image-wise color named images. Color name is predicted by the proposed method for each of the image. Table.III shows the image-wise annotation score. Our method is compared with the human accuracy [18] and needs to improved. It is analyzed that Google color name dataset consists many false positives and causes significant

Fig. 6: Proposed Image-wise color naming Method on Ebay dataset

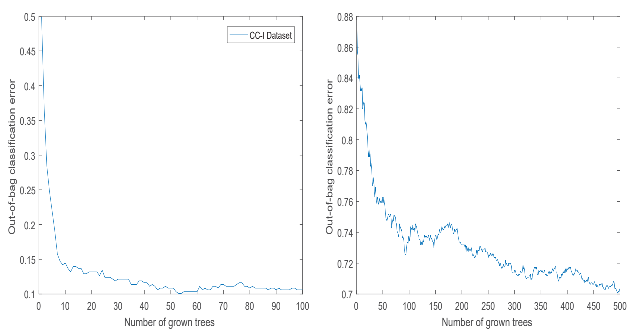

Fig. 7: Out-Of-Bag classification Error for Random Forests with CC-I and Google Color name datsets

derogation in the performance. Other image-wise color naming methods are discussed in [10] which are trained on different large datasets and improved the results. The improved results are at the computational cost as they deployed Covolution Neural Networks. The average time taken for our proposed method is shown in the Table.III.

V. Conclusion

In this paper, we have proposed two color naming systems one for pixel-wise color naming and other for image-wise color naming. In first method, Random Forests classifier that trained on color-chip dataset is used for pixe-wise color naming system based on superpixels gives pixel-level annotation with significant accuracy and out performs the some of the state-of-the-art methods and is computationally efficient to deploy for real-world applications. The second method for image-wise classification finds its applications in Fashion parsing and search engines is presented. Image-wise color naming badly effected by the weakly-labelled Google color name dataset which has many false positives. Pruning the dataset for removing false positives can result in more accurate naming. We showed the pixel-wise and image-wise color name predictions through Random forests and obtained significant results on Ebey color name datset.

TABLE IV: Image-wise Color Naming classification scores

Method	cars	shoes	dresses	pottery	overall
Proposed **Image-wise CN**	65	81	82	72	**74.25**
Human	–	–	–	–	88.98

References

[1] Berlin, B. and Kay, P.: Basic Color Terms: Their Universality and Evolution. University of California Press (1969)

[2] Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Transactions on Image Processing, 18(7), 1512–1523, (2009)

[3] Cheng, Z., Li, X., Loy, C. C.: Pedestrian color naming via convolutional neural network. In Asian Conference on Computer Vision, Springer, Cham, 35–51 (2016)

[4] Benavente, R., Vanrell, M., Baldrich, R.: Parametric fuzzy sets for automatic color naming. JOSA A, 25(10), 2582–2593 (2008)

[5] Menegaz, G., Le Troter, A., Sequeira, J., Boi, J. M.: A discrete model for color naming. EURASIP Journal on Advances in Signal Processing, 029125 (2007)

[6] Menegaz, G., Le Troter, A., Boi, J. M., Sequeira, J.: Semantics driven resampling of the osa-ucs. In Image analysis and processing workshops, 2007. iciapw 2007. 14th international conference on (pp. 216-220) IEEE (2007)

[7] Wyszecki, G., Stiles, W. S.: Color science (Vol. 8). New York: Wiley. (1982)

[8] Lammens, J. M.: Computational Model of Color Perception and Color Naming (1994)

[9] Mojsilovic, A.: A computational model for color naming and describing color composition of images. IEEE Transactions on Image Processing, 14(5), 690-699 (2005)

[10] Benavente, R., Vanrell, M., Baldrich, R.: A data set for fuzzy colour naming. Color Research and Application, 31(1), 48–56 (2006)

[11] Van De Weijer, J., Khan, F. S.: An overview of color name applications in computer vision. In International Workshop on Computational Color Imaging, Springer, Cham, 16–22 (2015)

[12] Conway, D.: An experimental comparison of three natural language colour naming models. In Proc. east-west int. conf. on human-computer interaction. 328–339 (1992)

[13] Griffin, L. D.: Optimality of the basic colour categories for classification. Journal of the Royal Society Interface, 3(6), 71–85 (2006)

[14] Tominaga, S.: A colour-naming method for computer color vision. In Proceedings of the 1985 IEEE International Conference on Cybernetics and Society (Vol. 573) 577 (1985)

[15] Serra, M., Penacchio, O., Benavente, R., Vanrell, M.: Names and shades of color for intrinsic image estimation. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 278-285). IEEE (2012)

[16] Liu, Y., Yuan, Z., Chen, B., Xue, J., Zheng, N.: Illumination robust color naming via label propagation. In Computer Vision (ICCV), 2015 IEEE International Conference on (pp. 621-629). IEEE,(2015, December).

[17] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence, 34(11), 2274-2282 (2012)

[18] Yu, L., Cheng, Y., van de Weijer, J.: Weakly Supervised Domain-Specific Color Naming Based on Attention. arXiv preprint arXiv:1805.04385(2018).