Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230171

A new color classification method based on multiple machine learning models

Kuang Gao[1,†,4], Zhiyin Yang[2,†] and Chengyu Yu[3,†]

[1]Shanghai Meihua International School, Shanghai, China
[2]The Second Affiliated Middle School of Beijing Normal University, Beijing, China
[3]Shanghai Experimental Foreign Language School, Shanghai, China

[4]india.nolac1@student.wwcc.edu
[†]These authors contributed equally.

Abstract. Color classification is a well-established field with everyday applications in areas such as object recognition, face recognition, skin color recognition, and materials recognition. With the advent of machine learning, various algorithms, including Artificial Neural Network (ANN), Supporting Vector Machine (SVM), decision trees, and Bayesian networks, have been employed for color classification. In this paper, we employ two distinct techniques, K-nearest Neighbors (KNN) and logistic regression, for the purpose of classification. Our study primarily focuses on simplifying image processing procedures to reduce time consumption. We utilize RGB values to train the KNN and logistic regression models and gather statistics on different colors. The results of our tests indicate that the KNN model has an advantage in terms of accuracy due to its straightforward and efficient computations. While logistic regression performs well on raw image data, it is outperformed by KNN on processed images. The proposed model can provide a color table for complex images in a relatively short amount of time.

Keywords: color classification, machine learning, deep learning.

1. Introduction

Color is the property possessed by an object of producing different sensations on the eye as a result of the way the object reflects or emits light. Moreover, color classification holds significant importance across multiple domains, as color perception is a fundamental aspect of human perception. For example, individuals with color blindness experience difficulty in distinguishing between colors, thereby emphasizing the utility of color classification algorithms to facilitate daily activities. the automated identification of color in images is crucial for technological advancements, such as the classification of traffic signals for autonomous vehicles. Therefore, being able to accurately determine color in images is truly crucial.

The earliest history of color classification is from human eyes. Green, blue, red, all from our experiences. As technology develops, computer starts to take place in this field, sorting colors into RGB format. Nowadays, as more and more Artificial Intelligence (AI) models establishes and upgrades, applying neural network on the field of color classification became possible. For the classification process, many classification algorithms can be used. Among them, the most popular machine learning algorithms are neural networks [1-3], decision trees [4, 5], k-nearest neighbors [6, 7], Bayesian networks

[8, 9], and support vector machines [10]. However, even though neural networks are broadly used and have a high accuracy, most classifiers like this can only determine limited colors at one time, bringing inconvenience for users. Typically, this kind of color classifier requires a high amount of calculation while given a rather long period of time for training, which means listing all colors out of an image is of great difficulty. That's why it is important to find a more efficient and convenient method of color classification. Thus, for this research, the objective is to optimize existing color classification techniques, rendering them more sophisticated, refined, and practical. The proposed method is expected to be especially beneficial for individuals with color blindness and machines requiring color classification capabilities.

In order to realize the classification mentioned above, we use two different machine learning method, k-nearest neighbors (KNN) and Logistic Regression. In addition, we also compared the basic time use in classifying the image with different sizes and the accuracy with each module.

2. Methodology

2.1. K-nearest neighbor

KNN is a passive learning, it does not calculate a special function which used as distinguishing the how the predictive object is different from others. It only calculates Minkowski distance (all the feather extracted take the same weight in the calculation), so it takes all the data into the memory and only used during the prediction.

KNN is a straightforward classification method, in which the number of nearest samples (K) is counted, and the catalogue with the highest number of samples is chosen for classification. Adding weight for each sample around as inversely proportional to the distance to the predictive object.

KNN is outer than others method, as it can always adding more data into the dataset without recreating the module.

However, a significant limitation of KNN is it requires considerable time to calculate the distance between the prediction and other data. The dataset needs to be equally distributed for each catalogues, otherwise the test data will be more easier fit into the one had more sample.

2.2. Logistic regression

Logistic Regression is a statistical technique that transforms the output of Linear Regression into a range of values between 0 and 1 using an activation function, commonly the sigmoid function. The algorithm estimates the probability of an observation belonging to a particular class by applying a threshold, typically 0.5, above which the predicted class is assigned.

Due to the simplicity of the model architecture and the low number of parameters, Logistic Regression has a relatively fast training time and requires less memory compared to other complex models. However, the simplicity of the model may result in difficulty fitting the data, leading to lower accuracy compared to more advanced models.

2.3. NLMeans denoising

NLMeans denoising is a popular method employed in image processing to mitigate the effects of noisy pixels. The core principle of this technique involves evaluating the similarity between the target pixel and its neighboring pixels, typically in the form of a patch, due to the inherent unreliability of single pixels. Each pixel is assigned a weight based on its Euclidean distance from the target pixel. Subsequently, a search is conducted throughout the image to identify similar regions, and the true target pixel is estimated by computing the mean of all similar regions. This approach has been shown to be effective in reducing noise and enhancing image quality.

2.4. The overview of the proposed method

The initial step in the color analysis process involves converting the input image to RGB mode, without the inclusion of the alpha channel. The image is then preprocessed to reduce overall complexity. After

DOI: 10.54254/2755-2721/8/20230171

image processing, the color table containing RGB value and corresponding names are opened and filtered, which goes into models as training data. Upon completion of the training process, the model will iterate through the image, while returning names of each color and it's reoccurrence. The resulting output is a set of recorded colors, along with their respective counts, which can be expressed as percentages if desired.

2.5. Implementation details

This study opens image and converts it into RGB style while ignoring alpha channel. If given any method, original image will be processed by that method. Results of the given method must be a two-dimensional array, while the smallest item being RGB values. After image manipulation, the array of image is being iterated. Each unique pixel and their reoccurrence are counted, in format of python dictionary.

The color-table is opened as DataFrame object from pandas module, and color json file loaded as python dictionary by json module. If a specific color object is identified in the color JSON file, a new DataFrame object with filtered colors is set. Else, unedited table is set. Pandas module then extracts RGB value and its name from table.

Two models, KNeighborsClassifier and LogisticRegressionClassifier, are utilized in the color analysis process, with each model having non-default parameters. For the KNeighborsClassifier, the result is limited to one, and the algorithm is set to brute. For the LogisticRegressionClassifier, the maximum iteration is set to the length of the preprocessed image, and the algorithm is set to liblinear. After initialization, the RGB values and names are fitted into the model, which is returned upon completion.

During the main processing sequence, each unique pixel color is passed through the selected model. The closest color name and its frequency of occurrence are recorded and saved as a Python dictionary. At this point, all computations are completed.

3. Result and discussion

Table 1. The inference time of the proposed model.

	Example Image	Normal Image
200×200	0.17 s	-
400×400	0.30 s	-
800×800	0.65 s	-
2480×3508	-	53.66 s
849×1200	-	34.01 s

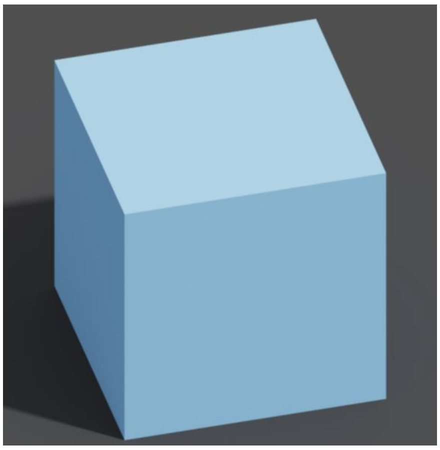

Figure 1. The example image used.

DOI: 10.54254/2755-2721/8/20230171

```
--- START ---
{'Dark liver': 200833, 'Wenge': 78, 'Beaver': 441, 'Pale taupe': 147, 'Tan':
189267, 'Shadow': 95, 'Light French beige': 215, 'Dark vanilla': 321, 'Desert
sand': 138539, 'Light taupe': 135, "Davy's grey": 31, 'Khaki (HTML/CSS) (Khaki)':
26, 'Pastel brown': 303, 'Deep Taupe': 17, 'Granite Gray': 8, 'Café au lait': 8853,
'Chamoisee': 24923, 'Camel': 446, 'Deer': 5, 'Dirt': 16449, 'Quartz': 4836, 'Outer
Space': 606, 'Raw umber': 32, 'Arsenic': 101, 'Black olive': 691, 'Umber': 23,
'Onyx': 114, 'Jet': 755, 'Coffee': 198, 'Charleston green': 17665, 'Raisin black':
33570, 'Dark lava': 22, 'French bistre': 32, 'Donkey brown': 10, 'Pale brown': 55,
'Bistre': 25, 'Coyote brown': 36, 'Old burgundy': 21, 'Dark liver (horses)': 18,

'Dark brown-tangelo': 4, 'Olive Drab #7': 6, 'Van Dyke Brown': 3, 'Café noir': 1,
'Rifle green': 42, 'Liver chestnut': 2}
--- END ---
Time taken: 0.6805756092071533 second(s).
```

Figure 2. The experimental result based on the provided example image.

Table 1 reveals that the classification of a normal image is significantly slower compared to an example image that contains fewer color variations. This observation suggests that the complexity of an image's color scheme significantly impacts its classification speed.

In our study, the K-Nearest Neighbors (KNN) model yields high accuracy rates owing to its basic architecture, which is well-suited for fuzzy searching. However, when classifying high-resolution images with rich color schemes, the KNN model still requires considerable time, particularly if the color set is expanded to enable more precise color identification. Meanwhile, Logistic Regression incurs less prediction-related overhead following model training, which primarily involves color set selection. Nonetheless, its classification accuracy declines when presented with images of high complexity.

For denoising an image, grouping similar pixels into classes and computing their mean color values can significantly reduce classification time. Subsequently, the KNN model can classify each color class to estimate the corresponding segment of the image. Nevertheless, given the natural complexity of light distribution and variations in brightness, identifying colors accurately remains challenging.

Figure 1 and Figure 2 depict the classification outcomes, where most color names are uncommon. This may pose a challenge that can be addressed by adding appropriate prefixes and suffixes to convert them into commonly used color names. However, this study did not explore this issue in detail.

4. Conclusion

This study describes a color analysis method that can be applied in various contexts. The method involves preprocessing image files to generate RGB values, which reduces the overall time needed for analysis while optimizing data density. The analysis is customizable, allowing for greater control over the precision and amount of results. The preprocessing step can be fine-tuned to add complexity and information to the image analysis, and custom color sets can be defined to limit results and improve recognition. The method employs a K-Nearest Neighbor Classifier, which ensures that each color is matched with its closest counterpart. This approach takes advantage of the simplicity of the classifier, saving computational resources. Although a Logistic Regression Classifier is also implemented, it performs better with raw image data and cannot benefit from preprocessing. The final results are presented in a simplified format of color names and corresponding amounts, which facilitates further processing.

References

[1] Bishop C M 1994 Neural networks and their applications Review of scientific instruments 65(6): 1803-1832

[2] Al-Shayea Q K 2011 Artificial neural networks in medical diagnosis International Journal of Computer Science Issues 8(2): 150-154

DOI: 10.54254/2755-2721/8/20230171

[3] Qiu Y Yang Y Lin Z et al. 2020 Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV China Communications 17(3): 46-57
[4] Myles A J Feudale R N Liu Y et al. 2004 An introduction to decision tree modeling Journal of Chemometrics: A Journal of the Chemometrics Society 18(6): 275-285
[5] Quinlan J R 1996 Learning decision tree classifiers ACM Computing Surveys (CSUR) 28(1) 71-72
[6] Peterson L E. 2009 K-nearest neighbor Scholarpedia 4(2): 1883.
[7] Laaksonen J Oja E 1996 Classification with learning k-nearest neighbors Proceedings of international conference on neural networks (ICNN'96) IEEE 3: 1480-1483
[8] Heckerman D 1998 A tutorial on learning with Bayesian networks Springer Netherlands
[9] Darwiche A 2010 Bayesian networks Communications of the ACM 53(12): 80-90
[10] Wang H Hu D 2005 Comparison of SVM and LS-SVM for regression 2005 International conference on neural networks and brain 2005 1: 279-283