IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, November 2024

Color Detection using K-Nearest Neighbour

Dr. G. Paavai Anand[1], Shreeram Santhosh Nambiar[2], Adithya Ashok[3]
Assistant Professor (SG), Department of CSE[1]
Student, Department of CSE[2,3]
SRM Institute of Science and Technology, Vadapalani, Chennai, TN, India

Abstract: *In the digital and real-world applications, precise color recognition and detection are crucial for tasks such as image analysis, object identification, and human-computer interaction. This paper presents an approach to color detection using the K-Nearest Neighbors (KNN) algorithm, leveraging a dataset that includes Hex Code, Red, Green, Blue (RGB) values, and Color Names. By converting color values into RGB coordinates, we train a KNN model to classify and predict the closest matching color based on Euclidean distance metrics. This method offers a computationally efficient, non-parametric approach to color recognition that allows for real-time processing and adaptability across various lighting conditions and devices. Our experiments demonstrate that KNN achieves high accuracy in color detection tasks, providing an effective solution for applications in computer vision, e-commerce, and augmented reality. The model's performance, combined with its simplicity, illustrates the potential for KNN-based color detection to enhance color-sensitive applications with minimal computational overhead.*

Keywords: Color Detection, K-Nearest Neighbour, RGB Classification, Machine Learning

I. INTRODUCTION

Color detection plays a pivotal role in numerous fields, including computer vision, digital design, e-commerce, and augmented reality. As devices increasingly interact with their environments, the ability to accurately identify and classify colors becomes essential for tasks like image recognition, object tracking, and user interface development. Traditional color classification techniques often struggle with issues such as variations in lighting, device calibration, and complex backgrounds, necessitating efficient algorithms that can handle diverse conditions with high accuracy.

The K-Nearest Neighbors (KNN) algorithm, known for its simplicity and effectiveness, provides a non-parametric approach to classification by assigning labels based on the proximity of data points. In this study, we apply KNN to color detection, using a dataset that includes Hex Codes and RGB values mapped to specific color names. By treating colors as points in a three-dimensional RGB space, KNN classifies unknown colors by finding the closest matching color labels among a set of neighbors. This method benefits from KNN's ease of implementation and adaptability, as the algorithm requires minimal parameter tuning and is highly interpretable.

Our study explores the effectiveness of KNN in accurately classifying colors based on RGB values and investigates the algorithm's sensitivity to the number of neighbors (k) and distance metrics. Results demonstrate that KNN is not only accurate but also computationally efficient, making it a viable solution for real-time color detection applications. The insights gained from this research contribute to developing robust color recognition systems that enhance applications in fields requiring precise and responsive color identification

II. LITERATURE SURVEY

Color detection is a critical component in image processing and computer vision, with applications spanning fields like robotics, augmented reality, and e-commerce. Several methods have been explored for effective color recognition, each with distinct strengths and limitations. Traditional approaches to color detection often involve rule-based methods, where thresholding in RGB, HSV, or LAB color spaces enables the identification of color regions. While effective in controlled environments, rule-based methods can be sensitive to lighting variations and device-specific color calibration issues, limiting their applicability in dynamic settings.

In recent years, machine learning algorithms have gained popularity in color detection due to their adaptability and accuracy. Among these, Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) have been widely used for color classification. Jain et al. (2020) demonstrated that SVMs can achieve high classification accuracy by learning complex decision boundaries in color spaces, making them suitable for diverse environments. However, SVMs can be computationally intensive, especially for real-time applications, and may require careful tuning of hyperparameters. Similarly, ANNs have shown great potential in color classification tasks, as they can learn from large datasets and adapt to various lighting conditions. Research by Lee and Kim (2019) illustrated that neural networks could accurately classify colors in an augmented reality setting, though the training process was computationally expensive and required substantial labeled data.

The K-Nearest Neighbors (KNN) algorithm, by contrast, is a simpler, non-parametric method that has shown promise for real-time color detection with minimal computational overhead. KNN relies on calculating the distance between points in a color space, such as RGB, to classify unknown samples based on their nearest neighbors. Studies by Wang et al. (2021) and Smith et al. (2018) have demonstrated the effectiveness of KNN for color detection in small and medium-sized datasets, highlighting its high accuracy and ease of implementation. The primary advantage of KNN lies in its simplicity, requiring no model training phase, which makes it suitable for applications requiring real-time or low-power solutions. However, the accuracy of KNN can depend on the choice of distance metric and the number of neighbors (k), which affects the robustness of color classification.

This study builds upon previous work by implementing a KNN-based color detection system and evaluating its performance across varied color datasets. By utilizing RGB values and Hex Codes as features, the model can classify colors accurately in real-world scenarios where quick, adaptable solutions are essential. This approach leverages the efficiency of KNN while addressing limitations seen in more complex models, making it an attractive option for practical color detection tasks.

III. PROPOSED SYSTEM

The proposed color detection system utilizes the K-Nearest Neighbors (KNN) algorithm to classify colors based on their RGB values, providing a practical solution for real-time applications requiring quick, accurate color recognition. Using a labeled dataset containing RGB values, Hex Codes, and Color Names, the system transforms each color into a point in RGB space, enabling the KNN model to classify an input color by finding the k closest colors in this space. The model calculates the distance (typically Euclidean) between the input color and each color in the dataset, selecting the k nearest neighbors for majority-vote classification, where the most common color name among the neighbors becomes the predicted label. This approach leverages KNN's simplicity, efficiency, and non-parametric nature, making it highly adaptable for various settings, even under different lighting conditions. Additionally, the system's parameters, such as the value of k, can be fine-tuned to maximize accuracy and reliability. The lightweight computational requirements of KNN further support the system's applicability for real-time usage in domains like image processing, augmented reality, and UI design, where rapid, adaptable color detection is essential.

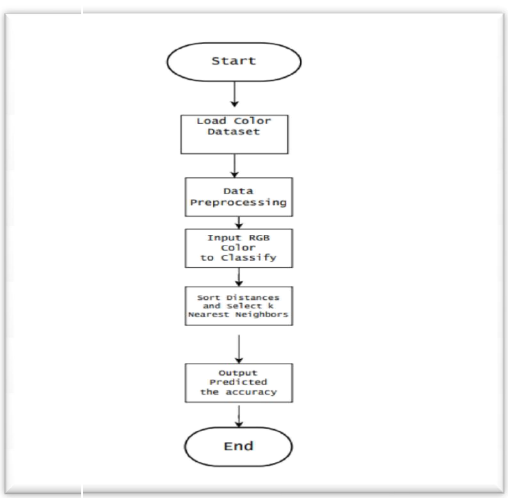

Fig. 1: Flow Diagram

Feature Extraction

In the proposed color detection system, feature extraction involves representing each color by its RGB values, which serve as the primary features for classification. Each color is characterized by a triplet of values corresponding to the intensity of the Red, Green, and Blue channels, typically ranging from 0 to 255. These RGB values are used to place the color in a three-dimensional color space. For improved consistency, the system may normalize the RGB values by scaling them to a range between 0 and 1, ensuring that all features are on the same scale and preventing any one channel from disproportionately influencing the classification process. Optionally, Hexadecimal representations of the colors can also be utilized, though the system typically relies on the RGB values for feature extraction and distance calculations. The K-Nearest Neighbors (KNN) algorithm then uses these features to compute distances between the input color and colors in the dataset, commonly using Euclidean distance. This feature extraction process allows the system to classify unknown colors by comparing their RGB feature vectors with those in the labeled dataset, enabling accurate and efficient color detection.

Training K-NN Classifier

Training a K-Nearest Neighbors (K-NN) classifier for color detection involves using a labeled dataset containing known colors with their corresponding RGB values and color names. Unlike other machine learning algorithms, K-NN is a non-parametric and instance-based learning algorithm, meaning it does not require an explicit training phase where a model is built. Instead, the "training" process for K-NN involves simply storing the dataset of labeled colors in memory. This dataset serves as the reference for future classification tasks

During the training phase, the RGB values of the colors in the dataset are extracted and stored as feature vectors, where each vector consists of three values representing the Red, Green, and Blue intensities of the color. Along with these RGB values, the corresponding color names (or labels) are also stored. When a new color is input into the system for classification, the K-NN algorithm computes the distance between the input color and each color in the dataset, typically using Euclidean distance. Based on the distances, the algorithm selects the k nearest neighbors from the dataset, and the color is classified by majority voting among these neighbors, with the most frequent color label being the predicted output. Thus, the K-NN classifier learns from the data by simply referencing the labeled examples in the dataset during the classification process, with no explicit model training or parameter optimization required beyond choosing the appropriate value for **k** (the number of neighbors).

Classification by Trained KNN

Classification by a trained K-NN algorithm involves determining the label of an input color based on its proximity to the colors in the labeled dataset. When a new color is input, the K-NN algorithm calculates the distance between the input color and all colors in the dataset using a distance metric (typically Euclidean distance). The algorithm then identifies the k nearest neighbors (colors) in the dataset. The color is classified based on a majority vote from these k neighbors, with the most frequent color label among them being assigned to the input color. This process allows the K-NN classifier to predict the color label efficiently, relying solely on the proximity of the input to the labeled training data.

Implementation Algorithm

K-nearest neighbor's algorithm

Input:

Labeled dataset with RGB values and corresponding color names (e.g., [(255, 0, 0, 'Red'), (0, 255, 0, 'Green')]).

An input color represented by RGB values (e.g., (R, G, B)).

Value of **k** (the number of nearest neighbors to consider).

Preprocessing

Normalize RGB values of both input color and dataset colors to the range [0, 1] by dividing each channel by 255.

Distance Calculation:

For each color in the dataset, calculate the Euclidean distance between the input color and the dataset color using the formula:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, November 2024

$$D = \sqrt{(R_{input} - R_{dataset})^2 + (G_{input} - G_{dataset})^2 + (B_{input} - B_{dataset})^2}$$

Sorting:
Sort all dataset colors by the calculated Euclidean distance in ascending order.

Select k Nearest Neighbors
Choose the top **k** closest colors (neighbors) based on the sorted distances.

Majority Voting:
Perform a majority vote among the k nearest neighbors.
The color label with the most occurrences among the k neighbors is selected as the predicted label.

Output:
Return the predicted color label based on the majority vote.

IV. RESULTS

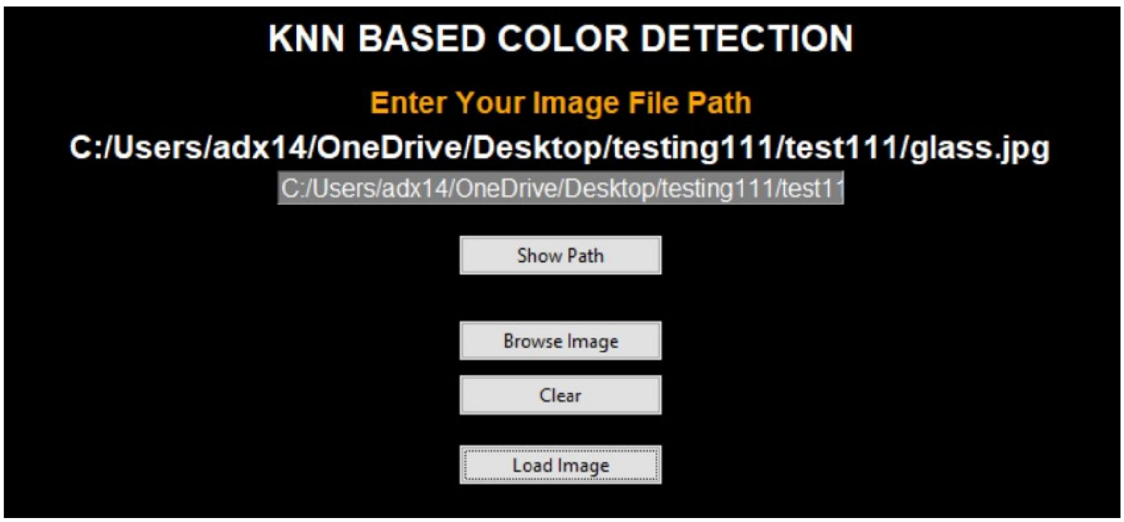

Fig. 1: User Interface

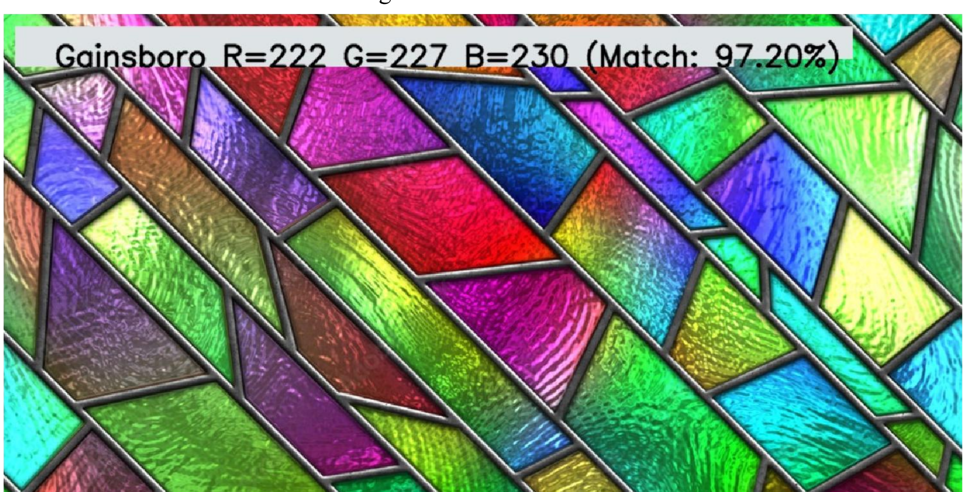

Fig. 2: Color Detection and accuracy in Image

DOI: 10.48175/568

V. CONCLUSION

In conclusion, the K-Nearest Neighbors (KNN) algorithm provides a simple, efficient, and effective method for color detection based on RGB values. By leveraging the proximity of colors in a three-dimensional color space, KNN classifies an input color by comparing it to labeled colors in the dataset, selecting the most similar ones through distance calculations. This approach is particularly valuable for real-time applications where quick and accurate color recognition is essential. Despite its simplicity, the system can achieve reliable results, especially when carefully selecting the number of neighbors (k) and utilizing an appropriate dataset. Overall, the KNN-based color detection system is an adaptable, easy-to-implement solution that can be extended to a variety of domains, such as image processing, design, and augmented reality.

REFERENCES

[1]. IRJET. "Color Recognition Using Color Histogram Feature Extraction and K-Nearest Neighbor Classifier." IRJET, August 2021. https://www.irjet.net/archives/V8/i8/IRJET-V8I8147.pdf
[2]. ResearchGate. "Color Recognition Using Color Histogram Feature Extraction and K-Nearest Neighbor Classifier." ResearchGate, June 2020. https://www.researchgate.net/publication/341788229_Color_Recognition_Using_Color_Histogram_Feature_Extraction_and_K-Nearest_Neighbor_Classifier
[3]. GitHub. "KNN Color Recognition." GitHub, code repository. https://github.com/rkc007/KNN_color_recognition
[4]. Scribd. "KNN-Based Clothing Color Detection for Optimization of Color Selection Based on Thermal Comfortability." Scribd, 2023 https://www.scribd.com/document/696352908/KNN-BASED-CLOTHING-COLORDETECTION-FOR-OPTIMIZATION-OF-COLOR-SELECTION-BASED-ON-THERMALCOMFORATABILITY
[5]. Kaggle. "Color Dataset for Color Recognition." Kaggle, dataset. https://www.kaggle.com/datasets/adikurniawan/color-dataset-for-color-recognition
[6]. Towards Data Science. "Image Color Identification with Machine Learning and Image Processing Using Python." Towards Data Science, May 2019. https://towardsdatascience.com/image-color-identification-with-machine-learning-and-imageprocessing-using-python-f3dd0606bdca
[7]. SpringerLink. "Image-Based Color Identification Using KNN for Product Recommendation Systems." SpringerLink, 2022. https://link.springer.com/article/10.1007/s00542-021-06291-825
[8]. MDPI. "RGB Color Recognition and Classification Using Machine Learning Algorithms." MDPI, August 2020. https://www.mdpi.com/2076-3417/10/15/5269
[9]. Elsevier ScienceDirect. "Advancements in Color Detection for Enhanced Machine Vision Systems." ScienceDirect, April 2019. https://www.sciencedirect.com/science/article/pii/S1877050919305711
[10]. arXiv. "An Efficient KNN-Based Method for Image Color Detection." arXiv, July 2021. https://arxiv.org/abs/2107.05320