COLOR RECOGNITION METHODS IN DIGITAL IMAGES: A REVIEW OF CLASSICAL AND DEEP LEARNING APPROACHES

Ashik James, Ansa Anto, Ann Maria Pauly, Ahmed Razal K M, Soumya M K, Sanam E Anto

Ashik James Student, Computer Science and Engineering, Holy Grace Academy of Engineering, Kerala, India
Ansa Anto Student, Computer Science and Engineering, Holy Grace Academy of Engineering, Kerala, India
Ann Maria Pauly Student, Computer Science and Engineering, Holy Grace Academy of Engineering, Kerala, India
Ahmed Razal K M Student, Computer Science and Engineering, Holy Grace Academy of Engineering, Kerala, India
Soumya M K Mentor, Computer Science and Engineering, Holy Grace Academy of Engineering, Kerala, India
Sanam E Anto Head of the Department, Computer Science and Engineering, Holy Grace Academy of Engineering, Kerala, India

ABSTRACT

Color detection is a fundamental computer vision task essential for applications spanning object recognition, robotics, image retrieval, and accessibility. Its wide applicability necessitates accurate, efficient, and robust methods capable of identifying colors under diverse conditions, including lighting variations and image complexity. This survey aims to comprehensively review and analyze techniques for color detection in digital images, emphasizing implementations utilizing Python and prevalent computer vision libraries like OpenCV. We examine numerous approaches, from foundational methods involving color space analysis (RGB, HSV, LAB) and thresholding techniques (global, adaptive) to more advanced strategies employing clustering algorithms (K-Means, Fuzzy C-Means). The review further covers the use of standard machine learning classifiers, such as K-Nearest Neighbors (KNN), and the increasing adoption of deep learning models, including Convolutional Neural Networks (CNNs), often combined with feature extraction based on color histograms, contrast, and saturation. The literature indicates that methods like HSV thresholding and KNN/K-Means clustering provide effective solutions for many common color identification tasks. Concurrently, deep learning approaches demonstrate significant potential for addressing challenging scenarios, particularly regarding lighting invariance, albeit with higher computational demands. This survey seeks to guide researchers in selecting suitable methodologies by comparing various techniques and highlights areas for future research, such as real-time implementation and enhanced robustness for complex visual environments. The summarized algorithms offer valuable tools for integration into software applications requiring automated color analysis.

Keyword: - *Color Detection, Image-Based Color Recognition, OpenCV, HSV Thresholding, K-Means Clustering, Fuzzy C-Means, K-Nearest Neighbors (KNN).*

1. INTRODUCTION

Color is a fundamental visual cue, essential for interpreting the world around us. Automating the detection and recognition of color within digital images is a core task in computer vision, underpinning numerous applications from industrial quality control and robotics to content-based retrieval and assistive technologies. Despite its apparent simplicity to human perception, computational color detection faces significant hurdles, including sensitivity to varying lighting conditions, material properties, and the inherent subjectivity in color naming. Over the years, a diverse range of computational methods has been developed to address these challenges. These methodologies encompass foundational image processing techniques like thresholding and color space manipulation (RGB, HSV, LAB), unsupervised clustering algorithms such as K-Means for identifying dominant colors, and various supervised machine learning approaches including K-Nearest Neighbors and deep learning models like Convolutional Neural Networks. Many contemporary implementations utilize the flexibility of Python and the comprehensive tools provided by libraries such as OpenCV. This literature survey aims to present a structured overview of these prevalent techniques, analyzing their principles, common strategies, and relative effectiveness for tackling the multifaceted problem of color detection in digital images, providing a consolidated view of the current landscape.

2. MILESTONES

The paper titled “Color Detection using OpenCV” by Nevathetha, R.A.; Fathima, K.K.; Niveditha, S.; Selvavathi, M. published in 2023 [1] details a practical application designed for identifying color names within digital images using fundamental computer vision tools. Leveraging the capabilities of the widely-used OpenCV library, the project aims to provide users with an intuitive way to determine the color of any point in an image simply by clicking on it. The core methodology relies on comparing the RGB (Red, Green, Blue) color values of the selected pixel against a predefined, comprehensive database. This database contains an extensive list of 865 distinct color names, each meticulously mapped to its corresponding RGB values and potentially hexadecimal codes. When a user interacts with the displayed image (e.g., via a double-click event, handled by OpenCV's event listeners), the system captures the coordinates of the click, extracts the RGB triplet of the pixel at that location, and then compares this triplet to the entries in the color database. The algorithm likely calculates a distance metric (e.g., Euclidean distance in RGB space) between the sampled pixel's color and each color in the database, ultimately identifying and displaying the name of the color with the smallest distance (the closest match). The authors highlight the system's simplicity and its reliance on foundational computer vision concepts, suggesting its utility in various applications such as robotics, assisting self-driving vehicles in identifying signal colors, or as a tool within image editing software, demonstrating a direct and accessible approach to color detection.

The article titled “Color Detection using Pandas and OpenCV” authored by Gomathy, C. K.; Chaitanya, C.; Anvitha, B. J. S.; Jahnavi, C. was published in October 2021 [2]. This work presents a straightforward method for color identification in digital images, specifically designed to overcome the limitations and inaccuracies noted in previous basic OpenCV implementations where color extraction yielded incorrect results. The authors explicitly utilize Python, integrating the Pandas library for data handling and the OpenCV library for image processing and interaction. Their proposed system relies on a curated database ("CV database") containing 865 distinct color names, each associated with its specific RGB and hexadecimal values. The core functionality allows a user to load an image using OpenCV's imread function. When the user interacts with the displayed image by clicking the cursor on a specific point, the system captures this event and extracts the RGB color values of the pixel at the cursor's location. These extracted RGB values are then compared against the predefined color entries in the Pandas-managed database. The system automatically identifies the closest match, likely by calculating the minimum color distance (e.g., Euclidean distance in RGB space), and then displays the corresponding color name and its RGB values to the user. This approach leverages the combination of OpenCV's image display and event handling capabilities with Pandas' efficient data management for the color lookup process, aiming for accurate color name identification.

The paper “Computer Vision for Color Detection” by Joy, D. T.; Kaur, G.; Chugh, A.; Bajaj, S. B. published in May 2021 [3] proposes a computer vision application aimed at enabling a computer to detect and define colors within an image, mirroring the human process but using computational methods. The core concept involves using image input, potentially from a camera feed, and processing it to identify the color name associated with specific pixel values, primarily focusing on the RGB color space. The algorithm described involves obtaining the RGB values for a pixel (likely selected by user interaction, such as clicking). It then iterates through a dataset containing predefined color names and their corresponding RGB values. Within this iterative process (described as a loop), a distance calculation is performed between the input pixel's RGB values and each entry in the dataset. The algorithm identifies the color name associated with the "nearest match," meaning the entry yielding the minimum calculated distance. This identified color name, along with the source RGB values, is then presented as the output. The authors emphasize that this method, based on RGB comparisons and a shortest distance metric, achieves high ("peaking") accuracy in defining colors. They position this work as foundational for various useful applications that rely on accurate color detection, demonstrating a clear, algorithmically defined approach to teaching color recognition to a computer using standard computer vision principles.

The article titled “Real-Time Color Detection Using Python and OpenCV” authored by Kaushal, M.; Singh, B. was published in May 2022 [4]. This work focuses on the development of a system capable of identifying specific colors (Red, Green, and Blue) in a real-time video stream captured via webcam, explicitly utilizing Python programming language along with the OpenCV and NumPy libraries. The methodology addresses the challenge of accurately identifying colors under dynamic conditions. The process begins by capturing video frames using OpenCV's VideoCapture(). Each frame is then converted from the default BGR color space to the HSV (Hue, Saturation, Value) color space using cv2.cvtColor(). This conversion is crucial as the HSV space allows for more robust color segmentation under varying lighting conditions compared to RGB. For each target color (Red, Green, Blue), specific minimum and maximum HSV range thresholds are defined. These ranges are used with cv2.inRange() to create binary masks, isolating pixels within the desired color range. To refine these masks and reduce noise, morphological operations, specifically dilation using cv2.dilate() with a kernel, are applied. A bitwise AND operation is performed between the original frame and the mask to visually isolate the detected color regions. Finally, contour detection is employed on the masks to potentially outline and distinguish the areas corresponding to each detected color. The system outputs the identified color names in real-time, demonstrating a practical application of fundamental computer vision techniques for live color analysis.

The research titled “Image recognition software geometry with Python and OpenCV” by Mar-Hernández, P. G.; Ibarra-Angulo, P. L.; Grijalva-Acuña, J. C.; Abril-García, J. H. published in 2023 [5] describes the development of a software application aimed at geometric shape and colour recognition within images, explicitly utilizing Python alongside several key computer vision and processing libraries. Employing a structured, sprint-based methodology, the authors detail the integration of tools including OpenCV for core vision tasks, NumPy for numerical operations, Imutils and Pillow for image manipulation conveniences, and Tkinter for the graphical user interface. A significant part of their work, detailed in their third and fourth development sprints, focused specifically on colour and shape detection. The colour detection phase involved converting image segments (potentially after initial processing like thresholding to black and white or using HSV space) and applying filtering techniques to isolate pixels within specific colour ranges. The subsequent sprint integrated this with geometric shape detection, enabling the software to identify and outline objects that conform to both a user-selected geometric shape (e.g., triangle, square, circle) and a user-selected colour (e.g., red, green, blue). This demonstrates a practical application combining multiple recognition tasks, relying on OpenCV's functions for image loading, colour space conversion, thresholding (defining colour ranges, likely using cv2.inRange), filtering, and potentially contour analysis to outline the final detected objects based on both shape and colour criteria, showcasing a multi-faceted approach within a Python environment.

The paper “Specific Color Detection in Images using RGB Modelling in MATLAB” by Goel, V.; Singhal, S.; Jain, T.; Kole, S. published in March 2017 [6] presents an approach implemented in MATLAB for recognizing specific user-selected colors within a static two-dimensional image, leveraging the Image Processing Toolbox and the RGB color model. The methodology centers around color thresholding techniques applied within the RGB space. The process involves reading an input RGB image, which is represented as an MxNx3 matrix. To isolate a specific color, the technique involves several steps: converting the original 3D RGB image into a 2D grayscale image (rgb2gray) and then subtracting this grayscale representation from the individual red, green, or blue color channel of the original image (imsubtract). This subtraction aims to enhance the components of the target color. Median filtering (medfilt2) is applied to the resulting image to remove noise while preserving edges. The filtered image is then converted into a binary image (im2bw) using a thresholding process, likely based on the intensity histogram of the subtracted image, to segment the regions corresponding to the target color. Connected components labeling (bwlabel) is used to identify distinct regions in the binary image, and bounding box properties (regionprops) are employed to analyze the metrics (like centroid, area) of these labeled regions. The system confirms the color by analyzing the RGB values of the detected pixels, with the ultimate goal of identifying objects based on a specific, user-defined color for applications like object tracking or segregation.

The paper titled “Color Based Image Segmentation Using Adaptive Thresholding**”** authored by Sharma, P.; Abrol, P. was published in 2016 [7]. This research investigates the application of adaptive thresholding techniques for segmenting colored images, presenting it as a more effective alternative to global thresholding, particularly when dealing with images exhibiting variations in illumination or contrast. The authors provide a comparative analysis, highlighting the limitations of using a single threshold for the entire image and advocating for methods that dynamically select thresholds based on local image characteristics. Their proposed methodology integrates properties from both RGB and HSV color models to enhance segmentation accuracy. The process involves initially dividing the input RGB image into smaller blocks and applying Otsu's global thresholding method to each block independently, thereby creating an adaptive thresholding scheme. Crucially, the thresholded RGB values are then converted to the Hue component of the HSV color space, leveraging Hue's robustness to intensity and saturation variations. Additional thresholding is applied to the Saturation and Value bands to prevent the selection of very dark/black or very light/white regions. Morphological operations are subsequently used to refine the resulting masks, ensuring smoother and cleaner segmentation. The study experimentally demonstrates that this combined approach, utilizing adaptive thresholds derived from image blocks and exploiting HSV color space properties, yields superior segmentation results compared to standard global thresholding or using RGB information alone.

The article titled “Statistical Approach for Color Image Detection” authored by Kar, S. K.; Mohanty, M. N. was published in 2013 [8]. This paper presents a method for recognizing the color of a specific object within an image, employing a foundational statistical technique combined with image processing steps implemented in MATLAB. The core idea is to isolate the object of interest and then apply thresholding based on pixel statistics to determine its dominant color. The process begins with image acquisition, followed by boundary detection to distinguish the target object from its background, effectively defining a Region of Interest (ROI). Once the ROI is established, further processing is confined to this area. The authors utilize an iterative thresholding method applied separately to the Red, Green, and Blue (RGB) color planes of the image within the ROI. This iterative approach likely involves choosing an initial threshold, segmenting pixels based on it, calculating statistics (like mean intensity) of the resulting groups, and refining the threshold until convergence, similar to basic statistical thresholding techniques (though the exact iterative method isn't specified in the abstract, it contrasts with more complex ML approaches). After thresholding the R, G, and B planes, the resulting binary or segmented information is analyzed to recognize the primary color of the object contained within the initial ROI. The study aims to demonstrate the performance of this direct, statistical thresholding method for basic color recognition tasks.

The research paper “Unveiling Colors: a K-means Approach to Image-Based Color Classification” by Suryadevara, C. K. published in 2022 [9] explores color classification from digital images by leveraging the K-Means clustering algorithm, a fundamental unsupervised learning technique. This approach treats the color information of image pixels as data points within a multi-dimensional color space (typically RGB or a perceptually uniform space like LAB). The primary goal is to partition these pixel data points into a predefined number (K) of distinct clusters, where each cluster represents a dominant or significant color group present in the image. The K-Means algorithm iteratively assigns pixels to the nearest cluster centroid (mean color) and recalculates the centroids based on the assigned pixels, minimizing the within-cluster variance until convergence. This unsupervised method allows for the automatic discovery of inherent color groupings without requiring pre-labeled data. The study outlines the systematic methodology, including data preparation (image acquisition, preprocessing like resizing, color space conversion), feature extraction (pixel color values), the K-Means clustering process itself (determining K, applying the algorithm), and post-processing steps like cluster assignment and optional color labeling based on mean cluster values. The author emphasizes the significance of this technique for applications like object recognition, content-based image retrieval, and scene understanding, showcasing K-Means as a simple yet effective tool for harnessing color information in image analysis.

The research titled “Background dominant colors extraction method based on color image quick fuzzy c-means clustering algorithm” by Liu, Z.-y.; Ding, F.; Xu, Y.; Han, X. was published in 2021 [10]. This paper proposes a novel algorithm, named Color Image Quick Fuzzy C-Means (CIQFCM), designed to efficiently extract dominant colors from large background images, particularly relevant for applications like adaptive camouflage design. The core challenge addressed is the high computational cost of standard Fuzzy C-Means (FCM) clustering when applied directly to all image pixels, especially as image size increases. The CIQFCM method introduces a crucial pre-processing step: *clustering spatial mapping*. Instead of using every pixel as a sample, the algorithm first maps the pixels to a quantized CIE Lab color space. This quantization significantly reduces the number of unique color points (samples) that need to be clustered. The method further refines the sample set and employs an improved pedigree clustering technique to determine effective initial cluster centers for the subsequent fuzzy clustering step. The standard FCM objective function and update rules are then adapted to operate on this reduced set of quantized color samples, weighted by the number of original pixels corresponding to each quantized color. Experimental results demonstrate that CIQFCM maintains clustering accuracy comparable to improved FCM variants (PFCM) while drastically reducing computation time, achieving speed improvements of over 36 times for 1024x1024 images, showcasing its significant advantage for large-scale dominant color extraction tasks.

The article titled “Real-Time object color Identification” authored by Borawake, A.; Kulkarni, N.; Ghorse, A. was published in 2021 [11]. This study presents a significant step towards automated color recognition in dynamic environments by focusing on real-time identification using the OpenCV library and machine learning techniques. The core of their approach involves utilizing the K-Nearest Neighbors (KNN) classification algorithm, a well-established supervised learning method. To prepare the model, the authors performed feature extraction on a training dataset containing numerous color images, specifically focusing on generating RGB Color Histogram attributes for each color. These histogram attributes, which represent the distribution of red, green, and blue pixel intensities, serve as the feature vectors used to train the KNN classifier. The trained KNN model is then deployed to classify colors observed in live webcam streams. The system processes incoming video frames scene by scene, extracts the corresponding RGB Color Histogram features from objects within the frame, and feeds these features into the trained KNN classifier. The classifier identifies the color by comparing the input features to the learned patterns in the training dataset, assigning the color label corresponding to the nearest neighbors in the feature space. The authors mention a substantial dataset capable of identifying over 800 different colors and briefly contrast their ML-based approach with simpler methods like extreme channel comparison or basic color space conversions, positioning their work as a more robust solution for real-time applications in areas like AI-based systems or robotics where accurate color perception is required.

The article titled "Multiple Colour Detection of RGB Images using Machine Learning Algorithm" authored by Awotunde, J. B.; Misra, S.; Florez, H.; Obagwu, D. O. was published in 2022 [12]. This research specifically addresses the challenge of detecting multiple distinct colors within standard RGB images by applying a machine learning approach. The authors propose utilizing the K-Nearest Neighbour (KNN) algorithm, a prominent supervised classification technique known for its simplicity and instance-based learning. Central to their methodology is the use of Color Histograms for feature extraction. They process images to generate RGB color histograms, which quantify the distribution of pixel intensities across the red, green, and blue channels. These histograms, likely represented as numerical vectors, serve as the primary features that define the color characteristics of different image regions or potentially the entire image. The KNN classifier is trained using a dataset where these histogram features are associated with predefined color labels (the paper mentions classifying eight distinct colors: White, Black, Orange, Green, Yellow, Red, Blue, and Violet). During the classification phase, the color histogram of a new image or region is extracted and compared to the histograms in the training set using a distance metric (Euclidean distance is mentioned). The KNN algorithm assigns the color label based on the majority class among the 'k' nearest neighbors in the feature space. The study emphasizes that this feature extraction step is crucial for improving the efficacy and accuracy of the KNN classifier for distinguishing multiple colors within RGB images.

The paper "An efficient color detection in RGB space using hierarchical neural network structure" by Altun, H.; Sinekli, R.; Tekbas, U.; Karakaya, F.; Peker, M. published in 2011 [13] proposes a novel neural network architecture designed to improve the efficiency and accuracy of color detection directly within the RGB color space. The key innovation presented is a *hierarchical* structure composed of multiple neural networks. Instead of a single, large network classifying all colors, this approach utilizes a set of specialized "expert" neural networks in the first layer of the hierarchy, with each network trained specifically to recognize a particular color class (e.g., an expert network for red, another for green, etc.). These expert networks analyze the input RGB pixel values (potentially normalized) and produce outputs indicating the likelihood of the pixel belonging to their specific color. The outputs from all these specialized first-layer networks are then aggregated and fed as input into a subsequent "main" neural network at the second layer of the hierarchy. This main network integrates the information from the expert networks to make the final classification decision, determining the definitive color of the input pixel. The authors compare this hierarchical approach against a conventional, single Multi-Layer Perceptron (MLP) structure trained for the same task. Through experiments using various datasets, they demonstrate that their proposed hierarchical structure achieves better classification results than the classical single-network architecture, suggesting advantages in specialization and potentially handling the complexities of color classification in RGB space more effectively.

The article titled "Color Detection using Deep Learning Techniques" authored by Sainath, J.; Saketh Reddy, M.; Saketh, S.; Sakshitha, Ch.; Akshay, K.; Mohammed, T. K.; Kalyani, A. was published in 2024 [14]. This study explores the integration of deep learning methodologies with established computer vision tools for the task of color detection in digital images. The authors utilize a combination of techniques including deep learning frameworks alongside libraries such as OpenCV and Pillow, and incorporate clustering algorithms, specifically mentioning K-Means. The process aims to achieve robust color identification by leveraging clustering for color quantization and segmentation, effectively analyzing image data to extract dominant color information. The abstract highlights the use of these combined techniques to address challenges inherent in automated color recognition, which finds applications in diverse fields like image processing, computer vision, and robotics. While not detailing the specific deep learning architecture, the work positions itself at the intersection of advanced neural network capabilities and traditional image processing steps, such as preprocessing, color space conversion (potentially HSV or LAB as mentioned in the introduction), thresholding, and feature extraction handled by the clustering stage. The study emphasizes the potential of combining these different technological approaches (deep learning, library-based processing, unsupervised clustering) to enhance the accuracy and efficiency of understanding and interpreting color information within the visual domain, moving beyond simpler rule-based or single-algorithm methods.

The research titled "Color Recognition in Challenging Lighting Environments: CNN Approach" by Maitlo, N.; Noonari, N.; Ghanghro, S. A.; Duraisamy, S.; Ahmed, F. published in 2024 [15] addresses the significant problem of robust color recognition under varying and difficult lighting conditions, which often hampers the performance of traditional computer vision techniques. The authors propose a solution leveraging deep learning, specifically a Convolutional Neural Network (CNN). Their methodology involves a two-stage process: first, image segmentation is performed using an edge detection technique to accurately isolate and specify the object of interest from its background. This segmented object region, presumably containing primarily the object's pixels, is then fed as input into a specially trained CNN. The CNN architecture is designed and trained to classify the color of the object, with a key focus on achieving robustness against diverse lighting scenarios (e.g., changes in intensity, color temperature, or direction of illumination). The study experimentally validates this approach, comparing its performance against existing methods and demonstrating that the proposed CNN-based technique substantially enhances color detection accuracy and reliability across different lighting environments. By combining object segmentation with a CNN classifier trained for lighting invariance, the work aims to bridge the gap left by simpler color detection methods that are often sensitive to environmental lighting changes, offering a more resilient solution for practical applications.

The research paper “New Method for Extreme Color Detection in Images” by Forero, M. G.; Ávila-Navarro, J.; Herrera-Rivera, S. published in 2020 [16] proposes a distinct method focused specifically on detecting "extreme" colors within images, defined as primary RGB (Red, Green, Blue) or complementary CMY (Cyan, Magenta, Yellow) colors exhibiting very high saturation. The motivation stems from applications where such highly saturated colors serve as important landmarks or identifiers. The core of the proposed method involves direct comparison between the color channel values of a pixel. For instance, to detect extreme red, the Red channel value (R) is compared against the Green (G) and Blue (B) values by calculating R-G and R-B. An extreme red is identified only if both differences are significantly high, which is evaluated using a minimum (or fuzzy AND) operation on these differences (min(R-G, R-B)). This logic is extended analogously to detect extreme Green, Blue, Cyan (min(G-R, B-R)), Magenta, and Yellow. A key advantage highlighted is that this method does not require setting or empirically finding a threshold value for each color channel, unlike some normalization-based approaches. The authors compare their method against three other techniques found in the literature (including normalization methods and a subtraction-based method for yellow), using synthetic color palettes and real photographs. Their results demonstrate that the proposed method is uniquely capable of selectively detecting all six extreme colors (R, G, B, C, M, Y) with higher purity and selectivity than the compared methods, while also being computationally fast.

The paper titled “Color Feature Based Dominant Color Extraction” authored by Chang, Y.; Mukai, N. was published in 2022 [17]. This research addresses a limitation in conventional dominant color extraction methods (like basic K-means or histogram approaches) which often prioritize colors occupying large areas and fail to capture significant colors present only in small image regions. The authors propose a novel method that extracts dominant colors based on features humans typically consider when analyzing color schemes. The process begins by applying the K-means algorithm in the perceptually uniform CIELAB color space to obtain initial cluster centers. To account for spatial coherency, these initial segments are refined using a Region Adjacency Graph (RAG) and graph cut method to merge similar adjacent regions. For each resulting candidate color (cluster), the algorithm calculates specific color features: Saturation (derived from a*b* values), Area (number of pixels), and Contrast (based on Itti's saliency/attention map). An integrative feature score is computed for each candidate color, combining these individual feature values. A crucial step involves eliminating boundary colors caused by mixing, which is achieved by applying erosion to the mask image of each cluster; clusters that disappear after erosion are discarded. Finally, the dominant colors are selected sequentially based on saturation, area, and the combined feature score, with a weighting mechanism applied to avoid selecting colors with similar hues repeatedly. This feature-driven approach demonstrated improved ability to extract perceptually important colors, even from small image areas.

The article titled “Dominant Color Detection For Online Fashion Retrievals” authored by Zeybek, S.; Çelik, M. was published in 2024 [18]. This study introduces a novel framework specifically tailored for extracting dominant colors from online fashion imagery, a critical task for improving product retrieval and analysis in e-commerce. Addressing challenges like overlapping clothing items and the computational cost of some methods, the authors combine K-Means clustering with advanced segmentation techniques. The methodology involves using a fashion apparel detection model (specifically comparing YOLOv4 and U-Net architectures) to first identify and segment clothing items from the background and potentially other elements within the image. This segmentation precisely isolates the regions of interest. Subsequently, the K-Means clustering algorithm is applied to the pixels within these segmented clothing regions to identify and quantify the dominant colors present. The framework incorporates an adaptive weighting strategy during color extraction to enhance accuracy. Experimental results presented compare the performance of K-Means when coupled with YOLOv4 versus U-Net for the initial segmentation. The findings indicate that while both segmentation models can be effective, the K-Means with YOLOv4 combination generally outperforms K-Means with U-Net in this context, suggesting YOLOv4's suitability for the object detection phase preceding dominant color analysis in fashion images. The work provides a structured approach for robust dominant color extraction relevant to fashion image processing and retrieval systems.

The article "A color extraction algorithm by segmentation" by Wu, Q. E.; Fang, Z.; Song, Z.; Chen, H.; Lu, Y.; Zhou, L.; Qian, X. published in 2023 [19] presents a comprehensive algorithm for extracting color features (edges and corners) from digital images by employing segmentation techniques. Recognizing that many existing methods are grayscale-based or lose color information, this work focuses on processing three-dimensional color parameters. The proposed algorithm consists of several stages. Initially, image denoising is performed using a fuzzy operator (specifically, an Ordered Weighted Average operator variant) to handle noise and image fuzziness while preserving essential features. Subsequently, color feature segmentation is applied using a probabilistic method that assesses the proportion of specific color pixels within a Region of Interest (ROI), allowing for segmentation based on color distribution. The final stage involves feature extraction, where edges and corner points are detected using a color level jump algorithm, potentially incorporating gray-level mutation techniques. The study provides experimental comparisons with traditional edge detection operators (Robert, Sobel, Prewitt, Laplace, Canny), demonstrating that the proposed segmentation-based extraction algorithm achieves higher accuracy, significantly faster processing speed (0.157s vs. several seconds for a 362x500 image), stronger anti-noise capabilities, and better performance, particularly on divergent color edges where other methods might struggle. This research emphasizes a multi-stage, segmentation-centric approach for robust and efficient color feature extraction.

The article titled "Colour Detection from Image" authored by Arali, R.; Sumitha M.; Swathi P.; Halawal, V.; Naik, A. was published in 2021 [20]. This work focuses on developing a user-friendly application for identifying the names of colours within a digital image through direct interaction. The methodology centrally employs the OpenCV computer vision library and relies on a predefined dataset that maps common colour names to their corresponding RGB (Red, Green, Blue) values. The core functionality allows a user to simply click on any point within a displayed image, triggering the system to automatically retrieve the RGB values of the selected pixel. These extracted values are then compared against the entries in the stored dataset. By calculating the distance (likely Euclidean distance in RGB space) between the clicked pixel's colour and each colour in the dataset, the algorithm identifies the entry with the minimum distance, thus determining the closest matching colour name. This name is then presented to the user. The authors highlight several advantages of using OpenCV for this task, including its ability to handle various image processing steps potentially needed for preprocessing (like filtering or colour space adjustments, though not detailed extensively) and its capability to integrate with live video streams, allowing for colour detection in real-time webcam feeds. The system architecture described involves modules for webcam activation, object scanning, frame matching (comparing webcam patterns to defined RGB colour models), and displaying results, emphasizing a practical approach to colour identification.

The article titled "Integrating OpenCV and Pandas for Enhanced Image Filtering and Color Detection" authored by Kandala, A.; Damisetty, R.; Kamal D N, V. was published in 2024 [21]. This work demonstrates the synergistic use of two powerful Python libraries, OpenCV and Pandas, to perform common computer vision tasks focusing on image filtering and color detection. The methodology involves leveraging OpenCV for core image manipulation and analysis functions. Images are loaded using cv2.imread, and various filtering techniques are applied to enhance quality or extract features; the paper mentions cv2.filter2D for general filtering and potentially others like Gaussian or Median filters for noise reduction, and Canny or Sobel for edge detection. For color detection, the primary approach involves converting the image to a suitable color space (e.g., RGB or HSV using cv2.cvtColor) to facilitate the isolation of specific colors. This is typically achieved through thresholding techniques, such as using cv2.inRange to create binary masks for pixels falling within a specified color range. Further refinement using morphological operations like erosion and dilation might also be applied. A key aspect highlighted is the integration with the Pandas library. While OpenCV handles the image processing, Pandas is employed for data analysis and visualization of the results. Pandas DataFrames provide a flexible structure for managing extracted image data (like pixel values, color statistics, or feature measurements), enabling easier manipulation, filtering, grouping, and visualization through plots or histograms, thus enhancing the interpretation and presentation of the image filtering and color detection outcomes.

The work titled "Detection and Extraction of Colours in Digital Images and Computer Vision" by Obagbuwa, I. C.; Klaaste, K.; Jasmin, A. J. published in 2023 [22] explores and compares two distinct models for color detection and extraction. The first model, implemented in MATLAB, focuses on extracting specific, predefined colors from an image by allowing users to select pixel value ranges within the RGB color space (e.g., using impixel). This approach involves defining approximate threshold ranges for primary colors (Red, Green, Blue) based on their typical pixel intensity values and manipulating the image matrices (e.g., using imsubtract, im2bw) to isolate pixels falling within the desired range. The second model, developed in Python, utilizes the OpenCV and Pandas libraries for interactive color identification. This model enables a user to click on any location within an image; it then extracts the RGB value of that pixel and compares it against an extensive preloaded dataset (containing 865 color names mapped to RGB and hex values). Using a shortest distance calculation (likely Euclidean distance), the algorithm identifies the closest matching color name in the dataset and displays it along with the pixel's RGB values. The authors conclude from their experiments that red, green, and yellow are relatively easy to extract accurately, while colors like purple present challenges due to being combinations of extremes in the RGB space. They demonstrate the practical utility of both targeted color extraction and interactive color naming.

The paper "Shirt-color recognition for the color-blindness" by Wong, Q.-Q.; Ng, K.-W.; Haw, S.-C. published in 2024 [23] addresses the practical challenge faced by color-blind individuals in identifying shirt colors, proposing a machine learning-based system to assist them. Recognizing the limitations of simple color visualization tools which often suffer from incorrect recognition, this study focuses on developing and evaluating a robust classification model. The methodology involves retrieving RGB pixel values from input shirt images and converting them into the HSV (Hue, Saturation, Value) color space. The HSV space is chosen for its relative robustness to lighting variations, which commonly affect perceived color. Using the H, S, and V values as features, the authors train and compare the performance of five standard machine learning classification algorithms: K-Nearest Neighbors (KNN), Naive Bayes (NB), Random Forest (RF), Logistic Regression (LR), and Support Vector Machine (SVM). The goal is to predict the corresponding color name based on the extracted HSV features. The study includes hyperparameter tuning using GridSearchCV to optimize the models, although results indicate that tuning sometimes decreased accuracy for certain models like KNN and RF in their experiments. Ultimately, KNN was identified as the best-performing model before tuning. The system is implemented as a web application, likely using Python, which allows users to input an image and click on a point to get the predicted color name along with suggested matching colors, aiming to provide a practical aid for color-blind individuals during activities like shopping.

The article titled "Extraction of Color Information and Visualization of Color Differences between Digital Images through Pixel-by-Pixel Color-Difference Mapping" authored by Yoo, W. S.; Kang, K.; Kim, J. G.; Yoo, Y. was published in 2022 [24]. This work introduces a comprehensive method for extracting quantitative color information from digital images and visualizing variations between them, facilitated by custom image processing software named PicMan. Addressing the limitations of traditional colorimeters (low spatial resolution) and manual analysis (labor-intensive), the authors demonstrate pixel-by-pixel or region-of-interest (ROI) based color extraction across multiple standard color spaces, including RGB, HSV, XYZ, CIE L*a*b*, Munsell color, and hexadecimal codes. A key contribution is the visualization of color differences between images (e.g., before and after restoration) through pixel-level mapping of difference values like ΔL*, Δa*, Δb*, and overall color difference ΔE*. The study showcases practical applications using examples such as analyzing an aged oil painting pre- and post-restoration, performing statistical color analysis on an 18th-century silk portrait, and extracting/superimposing a historical seal stamp. The research also investigates the color distortion effects resulting from image file format conversion (PNG vs. JPG) using both statistical summaries and the proposed color-difference mapping. PicMan software is presented as a user-friendly tool integrating image processing, analysis, modification, and statistical functions, enabling detailed quantitative color characterization for art and cultural heritage applications.

3. CONCLUSIONS

This survey has reviewed a wide spectrum of techniques employed for color detection in digital images, reflecting an active and evolving area within computer vision. The analysis highlighted the progression from traditional image processing methods, such as thresholding and color space analysis, towards more data-driven approaches utilizing clustering, machine learning, and deep learning. Python, often in conjunction with the OpenCV library, emerged as a prevalent platform for implementing these diverse strategies. While foundational methods offer simplicity, they often struggle with robustness, particularly under variable illumination. Clustering techniques like K-Means provide effective means for dominant color extraction, whereas machine learning classifiers offer flexible solutions when sufficient training data is available. Deep learning, notably CNNs, shows considerable promise for achieving higher accuracy and invariance, especially in complex scenarios, though typically at a greater computational cost. Despite significant advancements, accurately replicating human color perception, especially in real-time and across diverse conditions, remains a key challenge. Future efforts will likely focus on developing more robust, efficient, and context-aware hybrid models, further enhancing the capabilities and applications of automated color detection across various scientific and technological domains, underscoring its continued importance.

4. REFERENCES

[1] R. A. Nevathetha, K. K. Fathima, S. Niveditha, and M. Selvavathi, "Color detection using OpenCV," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 7, no. 2, pp. 1–7, Feb. 2023.

[2] C. K. Gomathy, C. Chaitanya, B. J. S. Anvitha, and C. Jahnavi, "Color detection using Pandas and OpenCV," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 5, no. 10, pp. 1–5, Oct. 2021.

[3] D. T. Joy, G. Kaur, A. Chugh, and S. B. Bajaj, "Computer vision for color detection," *Int. J. Innov. Res. Comput. Sci. Technol. (IJIRCST)*, vol. 9, no. 3, pp. 53–59, May 2021.

[4] M. Kaushal and B. Singh, "Real-time color detection using Python and OpenCV," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 9, no. 5, pp. 1399–1400, May 2022.

[5] P. G. Mar-Hernández, P. L. Ibarra-Angulo, J. C. Grijalva-Acuña, and J. H. Abril-García, "Image recognition software geometry with Python and OpenCV," *J. Comput. Technol.*, vol. 7, no. 19, pp. 1–9, Dec. 2023.

[6] V. Goel, S. Singhal, T. Jain, and S. Kole, "Specific color detection in images using RGB modelling in MATLAB," *Int. J. Comput. Appl.*, vol. 161, no. 8, pp. 38–41, Mar. 2017.

[7] P. Sharma and P. Abrol, "Color based image segmentation using adaptive thresholding," *Int. J. Sci. Tech. Adv.*, vol. 2, no. 3, pp. 151–156, 2016.

[8] S. K. Kar and M. N. Mohanty, "Statistical approach for color image detection," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, 2013.

[9] C. K. Suryadevara, "Unveiling colors: A K-means approach to image-based color classification," *Int. J. Innov. Eng. Res. Technol. (IJIERT)*, vol. 9, no. 9, pp. 47–54, Sep. 2022.

[10] Z.-Y. Liu, F. Ding, Y. Xu, and X. Han, "Background dominant colors extraction method based on color image quick fuzzy c-means clustering algorithm," *Def. Technol.*, vol. 17, no. 1, pp. 291–300, Feb. 2021.

[11] A. Borawake, N. Kulkarni, and A. Ghorse, "Real-time object color identification," *Int. J. Innov. Res. Technol. (IJIRT)*, vol. 7, no. 8, pp. 191–195, Jan. 2021.

[12] H. Altun, R. Sinekli, U. Tekbas, F. Karakaya, and M. Peker, "An efficient color detection in RGB space using hierarchical neural network structure," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, pp. 154–158, Jun. 2011.

[13] H. Altun, R. Sinekli, U. Tekbas, F. Karakaya, and M. Peker, "An efficient color detection in RGB space using hierarchical neural network structure," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, pp. 154–158, Jun. 2011

[14] J. Sainath, M. S. Reddy, S. Saketh, Ch. Sakshitha, K. Akshay, T. K. Mohammed, and A. Kalyani, "Color detection using deep learning techniques," *Int. J. Res. Publ. Rev.*, vol. 5, no. 5, pp. 9529–9532, May 2024.

[15] N. Maitlo, N. Noonari, S. A. Ghanghro, S. Duraisamy, and F. Ahmed, "Color recognition in challenging lighting environments: CNN approach," *arXiv Prepr.*, arXiv:2402.04762, Feb. 2024.

[16] M. G. Forero, J. Ávila-Navarro, and S. Herrera-Rivera, "New method for extreme color detection in images," in *Pattern Recognit.*, Lect. Notes Comput. Sci., vol. 12088, pp. 89–97, Jun. 2020

[17] Y. Chang and N. Mukai, "Color feature based dominant color extraction," *IEEE Access*, vol. 10, pp. 93055–93061, 2022.

[18] S. Zeybek and M. Çelik, "Dominant color detection for online fashion retrievals," *Batman Univ. Yasam Bilim. Derg.*, vol. 14, no. 1, pp. 69–80, Jul. 2024.

[19] Q. Wu, Z. Fang, Z. Song, H. Chen, Y. Lu, L. Zhou, and X. Qian, "A color extraction algorithm by segmentation," *Sci. Rep.*, vol. 13, no. 21261, Dec. 2023.

[20] R. Arali, S. M, S. P, V. Halawal, and A. Naik, "Colour detection from image," *Int. J. Adv. Eng. Manag. (IJAEM)*, vol. 3, no. 7, pp. 2164–2171, Jul. 2021.

[21] K. K. Abhigna, R. Damisetty, and V. K. D. N., "Integrating OpenCV and Pandas for enhanced image filtering and color detection," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 8, no. 7, pp. 1–16, Jul. 2024.

[22] I. C. Obagbuwa, K. Klaaste, and A. J. Jasmin, "Detection and extraction of colours in digital images and computer vision," in *Proc. Int. Conf. Sci., Eng. Bus. Sustain. Dev. Goals (SEB-SDG)*, 2023, pp. 1–13.

[23] Q.-Q. Wong, K.-W. Ng, and S.-C. Haw, "Shirt-color recognition for the color-blindness," *MethodsX*, vol. 13, Art. no. 102866, 2024.

[24] W. S. Yoo, K. Kang, J. G. Kim, and Y. Yoo, "Extraction of color information and visualization of color differences between digital images through pixel-by-pixel color-difference mapping," *Heritage*, vol. 5, no. 4, pp. 3923–3945, Dec. 2022.