Quadtree vs. K-Means for Color Palette Extraction: A Comparative Study

An Evaluation Based on Perceptual Error Metrics and Computational Efficiency

Noumisyifa Nabila Nareswari - 13523058
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: noumisyifa@gmail.com , 13523058@std.stei.itb.ac.id

***Abstract*— Color palette extraction plays a fundamental role in digital image processing, with applications in image compression, visualization, and design. This study presents a comparative analysis between two contrasting algorithms for palette extraction: Quadtree decomposition and K-Means clustering. The Quadtree algorithm, based on a divide-and-conquer strategy, recursively segments the image based on perceptual color uniformity using the CIEDE2000 Delta E metric. In contrast, K-Means performs global optimization in LAB color space to identify dominant colors. Both methods are implemented and evaluated on a diverse set of images, measuring execution time, perceptual color accuracy, and structural fidelity using metrics such as Mean Squared Error (MSE), Delta E (CIEDE2000), and Structural Similarity Index (SSIM). The results demonstrate that Quadtree outperforms K-Means in terms of computational efficiency with significantly reduced execution time, while K-Means yields higher color and structural fidelity. The findings suggest that Quadtree is well-suited for time-critical or low-fidelity applications, whereas K-Means is preferable for tasks requiring high perceptual accuracy.**

Keywords—Color palette extraction; Quadtree decomposition; K-Means clustering,

I. Introduction

Digital image processing continues to grow in significance across various fields, from digital art and photography to data visualization and machine learning. One of the crucial component in this domain is color palette extraction, which referred as the computational task of reducing the color space of an image while preserving perceptual similarity and visual coherence with the original. This process extends beyond simple color reduction; it requires sophisticated understanding of human color perception, spatial color relationships, and computational efficiency considerations. The primary focus of this paper is to comprehensively evaluate and compare the effectiveness of two fundamentally different algorithmic approaches, which are Quadtree decomposition and K-Means clustering, in performing color palette extraction, with particular emphasis on their perceptual accuracy, computational performance, and spatial color awareness capabilities.

Traditional methods such as K-Means clustering, while widely adopted due to their simplicity and global optimization properties, encounter significant limitations in spatial color awareness and initialization sensitivity. In contrast, the Quadtree algorithm employs a divide and conquer strategy that recursively partitions an image into quadrants based on spatial location and color homogeneity criteria. The implementation of Quadtree practiced in this paper utilizes the Delta E (CIEDE2000) perceptual color difference metric to evaluate whether a spatial region exhibits sufficient color uniformity thus preserving both spatial coherence and perceptual accuracy. Our approach addresses the fundamental challenge of balancing palette accuracy and computational efficiency by implementing both methodologies and subjecting them to comparative analysis using real-world images of varying complexity and color distributions.

In the following sections, the author will present the theoretical foundations of both algorithms and perceptual color metrics (Section II), describe our proposed implementation methodology (Section III), demonstrate the practical implementation using Python (Section IV), conduct performance evaluation and comparative analysis (Section V), analyze the causes of performance differences (Section VI), and synthesize our findings with practical recommendations (Section VII). Our research demonstrates that while both approaches offer valuable capabilities, their optimal application depends on the specific balance required between perceptual accuracy, computational efficiency, and spatial color awareness.

II. Theoretical Backgroung

A. Divide and Conquer

~~A.~~

Commented [NN3]: idk

Commented [NN1]: cari sitasinya

Formatted: Normal

Commented [NN2]: you sure?

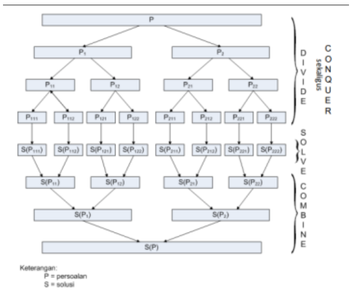


Fig. 1. ~~Example of a figure caption. (figure caption)~~Divide and Conquer Algorithm Visualization
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/07-Algoritma-Divide-and-Conquer-(2025)-Bagian1.pdf

~~Fig. 1.~~

Divide and conquer is an algorithm design that solves complex problems by recursively breaking them down into smaller, simpler sub-problems. This approach is particularly usefule when the original problem exhibits self-similar structure, allowing it to be decomposed into smaller instances of the same problem type.

The strategy consists of three main phases:

1) *Divide:* The original problem of size n is divided into r smaller sub-problems, ideally of approzimately equal size. Each sub-problem must resemble the original in structure but is reduced in complexity.

2) *Conquer:* Each sub-problem is solved independently. If the sub-problem is small enough (i.e., below a base case size n_0), it is solved directly. Otherwise, the divide and conquer strategy is applied recursively to break it down further.

3) *Combine:* The solutions of all sub-problems are merged to construct the solution for the original problem. However, in some cases, the combine step may be minimal or even unnecessary.

This recursive strategy is often expressed using a recurrence relation:

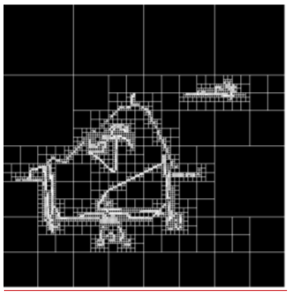
$$T(n) = \begin{cases} g(n), & if\ n \le n_0 \\ T(n_1) + T(n_2) + \cdots + T(n_r) + f(n), & if\ n > n_0 \end{cases}$$

Where:

- T(n) is the time complexity to solve a problem of size *n*.
- g(n) is the cost to directly solve small-sized sub-problems.
- $T(n_1)$, $T(n_2)$, ..., $T(n_r)$ are the costs to solve each sub-problem.
- f(n) is the cost to combine the sub-solutions into the final result.
- The divide step is typically negligible (O(1)) and often omitted in the complexity formula.

This recursive behavior also makes divide and conquer methods well-suited for implementation using recursive functions in programming.

B. *Quadtree Decomposition*



~~B.~~Fig. 2. Quadtree Image Decomposition Visualization
Source: https://uk.mathworks.com/help/images/quadtree-decomposition.html

Quadtree decomposition is a tree-based spatial partitioning technique used primarily in two-dimensional space. It operates by recursively subdividing a rectangular region into four quadrants. This method is particularly well-suited for applications involving hierarchical decomposition of data such as image compression and spatial indexing.~~The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.~~

The core idea is to represent a region with a tree, where:

- The root node represents the entire area.
- If the region does not satisfy a predefined criterion (e.g., uniformity or simplicity), it is divided into four equal-sized subregions.
- Each subregion is associated with a child node, and the subdivision process is applied recursively to these children.

This decomposition process continues until:

- The region meets the stopping criterion, or
- A minimum allowed size is reached

Because each node either has zero or four children, the resulting structure is a type of non-binary tree, typically referred to as a proper quadtree.

The time complexity of quadtree decomposition depends on the input and stopping condition. In the average case, only some regions are subdivided, giving a complexity between O(log n)

Formatted: Normal, Indent: First line: 0,51 cm, No bullets or numbering

Formatted: Font: Not Bold

Formatted: Body Text

Formatted: Indent: First line: 0 cm

Formatted: figure caption, Centered

Formatted: List Paragraph, Justified, Bulleted + Level: 1 + Aligned at: 1,14 cm + Indent at: 1,78 cm

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: List Paragraph, Justified, Bulleted + Level: 1 + Aligned at: 1,14 cm + Indent at: 1,78 cm

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Indent: Left: 1,78 cm, No bullets or numbering and $O(n^2)$. In the worst case, where all regions are split to the smallest unit, the complexity reaches $O(n^2)$ for an $n \times n$ input.

In this study, the Quadtree decomposition algorithm applies divide and conquer principles by recursively dividing the image into four quadrants (divide), checking color uniformity and stopping if homogeneous (conquer), and collecting the results as a hierarchical palette structure (combine).

C. *K-Means Clustering*

K-Means is an iterative clustering algorithm used to partition a dataset into *K* distinct, non-overlapping groups, where each data point belongs to the cluster with the nearest mean. It is widely used for vector quantization, compression, and unsupervised learning tasks, including color quantization.~~The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.~~

The algorithm follows these general steps:

- Initialize *K* cluster centroids, often randomly.
- Assign each data point to the nearest centroid based on a distance metric.
- Update the centroid of each cluster as the mean of all assigned points.
- Repeat steps 2 and 3 until convergence.

The process is guaranteed to converge to a local minimum of the clustering objective, but not necessarily a global minimum. The final result can vary depending on the initial centroid positions.

K-Means has a time complexity of $O(n \times k \times i \times d)$, where:

- n is the number of data points,
- k is the number of clusters,
- i is the number of iterations until convergence,
- d is the dimensionality of the data.

D. *CIE LAB Color Space*

CIE LAB is a perceptually uniform color space defined by the International Commission on Illumination (CIE). It is designed so that the Euclidean distance between two colors in this space approximates human perception of color differences more closely than device-dependent color spaces such as RGB.

The space consists of three components:

- L* for lightness,
- a* for the green–red axis,
- b* for the blue–yellow axis.

E. *Delta E (CIEDE2000)*

Delta E (ΔE) is a quantitative metric used to measure the perceptual difference between two colors. In color science, it is computed as the Euclidean distance between two colors in a perceptually uniform color space, most commonly CIE LAB. The CIEDE2000 variant is a refined formula that accounts for non-uniformities in human color perception, including differences in lightness, chroma, and hue interactions.

The CIEDE2000 formula is widely regarded as the most accurate representation of how humans perceive color differences. A Delta E value close to 0 indicates that the two colors are nearly indistinguishable to the human eye, while higher values signify more noticeable differences.

F. *Structural Similarity Index (SSIM)*

The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation based on changes in structural information, luminance, and contrast. Unlike pixel-wise metrics such as MSE, SSIM considers how humans perceive changes in image structure, making it more aligned with subjective visual quality.

SSIM is typically calculated over local windows and returns a value between -1 and 1, where 1 indicates perfect structural similarity. It is commonly used to assess the fidelity of compressed or reconstructed images and is suitable for comparing perceptual similarity between original and simplified images.

G. *Mean Squared Error (MSE) in LAB Space*

Mean Squared Error (MSE) is a pixel-wise metric that measures the average squared difference between corresponding pixel values of two images. When computed in the CIE LAB color space, MSE becomes more perceptually meaningful, as LAB better reflects human sensitivity to color differences.

By calculating MSE across all the color channels in LAB Space, the metric captures both color and brightness discrepancies in a way that aligns more closely with human vision. Although MSE does not consider spatial structure like SSIM, it remains a useful and interpretable measure of overall color fidelity between images.

III. Proposed Algorithm

Before applying either algorithm, the input image is first converted into a matrix of color values in CIE LAB color space where each pixel is represented as a 3-dimensional vector (L*, a*, b*). To ensure a fair comparison, the user specifies the desired number of output colors k, and the Quadtree algorithm adaptively adjusts its homogeneity threshold to produce approximately k dominant colors, matching the output size of K-Means. After palette extraction, the original image is recolored using each method and compared using Delta E (CIEDE2000), SSIM, and MSE in LAB space.~~Before you begin to format your~~

Formatted: Indent: Left: 1,78 cm, No bullets or numbering
Formatted: Indent: Left: 0 cm, Hanging: 0,51 cm, Tab stops: Not at 5,08 cm
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Normal, Justified
Formatted: Indent: First line: 0,51 cm
Formatted: Indent: Left: 0 cm, Hanging: 0,51 cm, Tab stops: Not at 5,08 cm
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Outline numbered + Level: 2 + Numbering Style: A, B, C, … + Start at: 1 + Alignment: Left + Aligned at: 0 cm + Tab after: 0,63 cm + Indent at: 0,51 cm, Tab stops: 0,51 cm, List tab + Not at 0,63 cm + 5,08 cm
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: List Paragraph, Justified, Bulleted + Level: 1 + Aligned at: 1,14 cm + Indent at: 1,78 cm
Formatted: Left
Formatted: Indent: Left: 0 cm, Hanging: 0,51 cm, Tab stops: Not at 5,08 cm
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold

~~paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.~~

~~Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:~~

A. ~~*Abbreviations and Acronyms*~~*Quadtree Decomposition (main purpose)*

Quadtree is a recursive spatial decomposition algorithm that subdivides the image into smaller rectangular regions based on color homogeneity. It follows the divide and conquer paradigm:~~Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.~~

1) *Divide*
   - The entire image is treated as one region.
   - For each region, the algorithm computes the maximum pairwise Delta E (CIEDE2000) between pixels.
   - If the color variation exceeds a certain threshold, the region is divided into four equal quadrants.
2) *Conquer*
   - The entire image is treated as one region.
   - *If the region is considered homogeneous, the average LAB color is computed and stored as part of the final palette.*
   - Otherwise, the process is repeated recursively.
3) *Combine*
   - The final color palette is formed by collecting the average colors from all terminal regions (leaf nodes).
4) *Threshold Adaptation*
   - To match the target number of colors k, the algorithm adaptively adjusts the Delta E threshold using a binary search loop.
   - It searches for a threshold value that yields a palette size closest to k.
5) *Evaluation*
   - The image is recolored using the extracted palette by replacing each region with its average color.
   - The recolored image then compared to the original using:
     - Delta E (CIEDE2000) to assess perceptual color difference,
     - SSIM to measure structural similarity,
     - MSE in LAB space to quantify color fidelity

~~Identify applicable sponsor/s here. If no sponsors, delete this text box (sponsors).~~

B. ~~*Units*~~*K-Means Clustering (comparison purpose)*

Quadtree is a recursive spatial decomposition algorithm that subdivides the image into smaller rectangular regions based on color homogeneity. It follows the divide and conquer paradigm:

1) *Preprocessing*
   - The image is flattened into a 1D array of LAB pixel vectors.
   - Each pixel becomes a data point in a 3D color space.
2) *Initialization*
   - k pixels are randomly chosen as the initial cluster centroids.
3) *Assignment*
   - Each pixel is assigned to the closest centroid using Euclidean distance in LAB space.
4) *Update*
   - To match the target number of colors k, the algorithm adaptively adjusts the Delta E threshold using a binary search loop.
5) *Repeat*
   - The assignment and update steps repeat until convergence (no changes or max iterations reached).
6) *Result*
   - The final k centroids form the color palette.
7) *Evaluation*
   - The image is recolored using the extracted palette by replacing each region with its average color.
   - The recolored image then compared to the original using:
     - Delta E (CIEDE2000) to assess perceptual color difference,
     - SSIM to measure structural similarity,
     - MSE in LAB space to quantify color fidelity

~~*B.*~~

- ~~Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5 inch disk drive."~~
- ~~Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.~~
- ~~Do not mix complete spellings and abbreviations of units: "Wb/m2" or "webers per square meter," not "webers/m2." Spell units when they appear in text: "...a few henries," not "...a few H."~~
- ~~Use a zero before decimal points: "0.25," not ".25." Use "cm3," not "cc." (*bullet list*)~~

~~*Equations*~~

~~The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font).~~

Formatted: Bulleted + Level: 1 + Aligned at: 0,95 cm + Indent at: 1,59 cm
Formatted: Indent: First line: 0 cm
Formatted: Bulleted + Level: 1 + Aligned at: 0,63 cm + Indent at: 1,27 cm
Formatted: Indent: First line: 0,32 cm, Tab stops: 0,95 cm, List tab + Not at 6,86 cm
Formatted: Bulleted + Level: 1 + Aligned at: 0,63 cm + Indent at: 1,27 cm
Formatted: Indent: First line: 0,32 cm, Tab stops: 0,95 cm, List tab + Not at 6,86 cm
Formatted: Font: Not Italic
Formatted: Indent: Left: 0 cm, Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0,32 cm + Tab after: 0,95 cm + Indent at: 0 cm
Formatted
Formatted
Formatted: Font: Not Italic
Formatted: Heading 3
Formatted
Formatted: Font: Not Bold, Not Italic
Formatted: Font: Not Italic
Formatted
Formatted
Formatted: Font: Not Italic
Formatted
Formatted
Formatted
Formatted: Font: Italic, English (Indonesia)
Formatted: Normal, Justified
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted: Font: Not Bold
Formatted

~~To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.~~

~~Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in~~

~~a + b = γ (1)~~

~~α + β = χ.~~

~~(1) (1)~~

~~Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use "(1)," not "Eq. (1)" or "equation (1)," except at the beginning of a sentence: "Equation (1) is ..."~~

~~*Some Common Mistakes*~~

- ~~The word "data" is plural, not singular.~~
- ~~The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o."~~
- ~~In American English, commas, semi-/colons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)~~
- ~~A graph within a graph is an "inset," not an "insert." The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).~~
- ~~Do not use the word "essentially" to mean "approximately" or "effectively."~~
- ~~In your paper title, if the words "that uses" can accurately replace the word using, capitalize the "u"; if not, keep using lower-cased.~~
- ~~Be aware of the different meanings of the homophones "affect" and "effect," "complement" and "compliment," "discreet" and "discrete," "principal" and "principle."~~
- ~~Do not confuse "imply" and "infer."~~
- ~~The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.~~
- ~~There is no period after the "et" in the Latin abbreviation "et al."~~
- ~~The abbreviation "i.e." means "that is," and the abbreviation "e.g." means "for example."~~

~~An excellent style manual for science writers is [7].~~

~~XXVII.~~IV. PROGRAM IMPLEMENTATION

The program implementation is divided into three main components. The first and primary focus is the Quadtree-based algorithm, which serves as the core method for adaptive color palette extraction. The second component is the K-Means clustering approach, implemented for comparison purposes to evaluate the effectiveness of the Quadtree method. Lastly, a set of supporting functions handles tasks such as image reconstruction, palette visualization, and quality evaluation using metrics like MSE, Delta E, and SSIM. These components are illustrated in the following code segments:~~After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.~~

A. Prerocessing

These functions are responsible for preparing the image data for analysis, including loading the image, transforming its color space, and basic utility computations.

```
def load_image(path):
    """Load image and convert to LAB."""
    img = Image.open(path).convert('RGB')
    img_np = np.array(img) / 255.0  # normalize to [0, 1]
    lab_img = rgb2lab(img_np)
    return lab_img, img_np

def mean_color(block):
    return np.mean(block.reshape(-1, 3), axis=0)

def block_deltaE(block, mean):
    flat_block = block.reshape(-1, 3)
    mean_arr = np.tile(mean, (flat_block.shape[0], 1))
    delta = deltaE_ciede2000(flat_block, mean_arr)
    return np.mean(delta)
```

~~Fig. 2.~~Fig. 3. Preprocessing Helper Functions Implementation

B. K-Means Based Extraction

```
def extract_palette_kmeans(lab_img, k):
    """Extract color palette using K-Means."""
    h, w, _ = lab_img.shape
    img_array = lab_img.reshape((-1, 3))
    img_array = shuffle(img_array, random_state=0)[:10000]  # sample for speed
    kmeans = KMeans(n_clusters=k, n_init='auto').fit(img_array)
    return kmeans.cluster_centers_

def recolor_image(lab_img, palette):
    """Map each pixel to the closest color in the palette using Delta E 2000 (batch)."""
    h, w, _ = lab_img.shape
    img_flat = lab_img.reshape(-1, 3)
    palette_repeated = np.repeat(palette[np.newaxis, :, :], img_flat.shape[0], axis=0)  # (N, K, 3)
    pixels = img_flat[:, np.newaxis, :]  # (N, 1, 3)

    # Hitung Delta E antara semua pixel dan semua warna dalam palette
    delta = deltaE_ciede2000(pixels, palette_repeated)  # (N, K)
    nearest_indices = np.argmin(delta, axis=1)
    recolored_flat = palette[nearest_indices]

    return recolored_flat.reshape(h, w, 3)
```

~~Fig. 3.~~Fig. 4. K-Means Usage Functions Implementation

C. Quadtree Based Extraction

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

```python
def quadtree_decompose(img, threshold, min_size=4):
    h, w, _ = img.shape
    blocks = []

    def split(x, y, w, h):
        region = img[y:y+h, x:x+w]
        mean = mean_color(region)
        dE = block_deltaE(region, mean)
        if dE < threshold or w <= min_size or h <= min_size:
            blocks.append((x, y, w, h, mean))
        else:
            w2, h2 = w // 2, h // 2
            split(x, y, w2, h2)
            split(x + w2, y, w - w2, h2)
            split(x, y + h2, w2, h - h2)
            split(x + w2, y + h2, w - w2, h - h2)

    split(0, 0, w, h)
    return blocks
```

```python
def extract_palette_quadtree(img, threshold, num_colors):
    blocks = quadtree_decompose(img, threshold)

    all_colors = np.array([b[4] for b in blocks])

    from scipy.spatial.distance import pdist, squareform

    def pairwise_delta_e(colors):
        dists = []
        for i in range(len(colors)):
            for j in range(i + 1, len(colors)):
                dists.append(deltaE_ciede2000(colors[i], colors[j]))
        return np.array(dists)

    condensed_dist = pairwise_delta_e(all_colors)
    dist_matrix = squareform(condensed_dist)

    clustering = AgglomerativeClustering(n_clusters=num_colors, metric='precomputed', linkage='average')
    labels = clustering.fit_predict(dist_matrix)

    palette = []
    for i in range(num_colors):
        cluster_colors = all_colors[labels == i]
        mean = np.mean(cluster_colors, axis=0)
        palette.append(mean)
    palette = np.array(palette)

    return palette, blocks
```

```python
def recolor_from_blocks(img, blocks):
    recolored = np.zeros_like(img)
    for x, y, w, h, mean in blocks:
        recolored[y:y+h, x:x+w] = mean
    return recolored
```

~~Fig. 4.~~Fig. 5. Quadtree Palette Extractor Functions Implementation

D. *Evaluation and Visualization*

```python
def mse_lab(original, recolored):
    """Compute MSE in LAB space."""
    return np.mean((original - recolored) ** 2)

def average_delta_e(original, recolored):
    """Compute average Delta E 2000 between two LAB images."""
    h, w, _ = original.shape
    total_delta = 0.0
    for i in range(h):
        for j in range(w):
            total_delta += deltaE_ciede2000(original[i, j], recolored[i, j])
    return total_delta / (h * w)

def ssim_rgb(original_rgb, recolored_lab):
    """Convert LAB to RGB and compute SSIM."""
    recolored_rgb = lab2rgb(recolored_lab)
    original_gray = np.mean(original_rgb, axis=2)
    recolored_gray = np.mean(recolored_rgb, axis=2)
    return ssim(original_gray, recolored_gray, data_range=1.0)
```

```python
def compare_algorithms_with_quadtree(image_path, num_colors, qt_threshold=10.0):
    results = {}

    # Load image
    lab_img, rgb_img = load_image(image_path)

    # --- K-Means ---
    start_kmeans = time.time()
    kmeans_palette = extract_palette_kmeans(lab_img, num_colors)
    recolored_kmeans = recolor_image(lab_img, kmeans_palette)
    time_kmeans = time.time() - start_kmeans

    mse_kmeans = mse_lab(lab_img, recolored_kmeans)
    delta_kmeans = average_delta_e(lab_img, recolored_kmeans)
    ssim_kmeans = ssim_rgb(rgb_img, recolored_kmeans)

    results['kmeans'] = {
        'time': time_kmeans,
        'mse': mse_kmeans,
        'deltaE': delta_kmeans,
        'ssim': ssim_kmeans
    }

    # --- Quadtree ---
    start_qt = time.time()
    qt_palette, qt_blocks = extract_palette_quadtree(lab_img, threshold=qt_threshold, num_colors=num_colors)
    recolored_qt = recolor_from_blocks(lab_img, qt_blocks)
    time_qt = time.time() - start_qt

    mse_qt = mse_lab(lab_img, recolored_qt)
    delta_qt = average_delta_e(lab_img, recolored_qt)
    ssim_qt = ssim_rgb(rgb_img, recolored_qt)

    results['quadtree'] = {
        'time': time_qt,
        'mse': mse_qt,
        'deltaE': delta_qt,
        'ssim': ssim_qt
    }
    results['kmeans_palette'] = kmeans_palette
    results['quadtree_palette'] = qt_palette

    return results
```

```python
def save_palette_image(palette, filename, swatch_size=50):
    """Save a horizontal color palette to an image file."""
    num_colors = len(palette)
    palette_rgb = lab2rgb(palette[np.newaxis, :, :])[0]  # convert to RGB
    img = np.zeros((swatch_size, num_colors * swatch_size, 3))

    for i, color in enumerate(palette_rgb):
        img[:, i * swatch_size:(i + 1) * swatch_size] = color

    img_uint8 = (img * 255).astype(np.uint8)
    Image.fromarray(img_uint8).save(filename)
```

~~Fig. 5.~~Fig. 6. Evaluation and Visualization Functions Implementation

RESULT AND ANALYSIS

V. ~~AFTER THE TEXT EDIT HAS BEEN COMPLETED, THE PAPER IS READY FOR THE TEMPLATE. DUPLICATE THE TEMPLATE FILE BY USING THE SAVE AS COMMAND, AND USE THE NAMING CONVENTION PRESCRIBED BY YOUR CONFERENCE FOR THE NAME OF YOUR PAPER. IN THIS NEWLY CREATED FILE, HIGHLIGHT ALL OF THE CONTENTS AND IMPORT YOUR PREPARED TEXT FILE. YOU ARE NOW READY TO STYLE YOUR PAPER; USE THE SCROLL DOWN WINDOW ON THE LEFT OF THE MS WORD FORMATTING TOOLBAR.~~

Formatted: Heading 1

A. ~~*Authors and Affiliations*~~*Result*

1) *Test Case 1: Actual Image of a Color Palette With Amount Input the Same as the Actual Color Segregation*

~~Fig. 6.~~Fig. 7. Input Image of Test Case 1

```
=== Color Palette Extraction Comparison ===
Enter the absolute path to the image: C:\Users\Syifa\Desktop\ColorPaletteExtractor\tests\gb-mignola-32x.png
Enter the desired number of colors: 4
```

~~Fig. 7.~~Fig. 8. Text Input of test Case 1

~~Fig. 8.~~Fig. 9. K-Means Color Palette Output of test Case 1

~~Fig. 9.~~Fig. 10. Quadtree Color Palette Output of test Case 1

```
=== Evaluation Results ===

-- KMEANS --
Execution Time: 1.8763 seconds
MSE:            0.0000
Delta E:        0.0000
SSIM:           1.0000

-- QUADTREE --
Execution Time: 0.0131 seconds
MSE:            0.0000
Delta E:        0.0000
SSIM:           1.0000
```

~~Fig. 10.~~Fig. 11. Text Output of test Case 1

2) Test Case 2: Actual Image of a Color Palette With Amount Input Different as the Actual Color Segregation

~~Fig. 11.~~Fig. 12. Input Image of Test Case 1

```
=== Color Palette Extraction Comparison ===
Enter the absolute path to the image: C:\Users\Syifa\Desktop\ColorPaletteExtractor\tests\gb-mignola-32x.png
Enter the desired number of colors: 4
```

~~Fig. 12.~~Fig. 13. Text Input of test Case 1

~~Fig. 13.~~Fig. 14. K-Means Color Palette Output of test Case 1

~~Fig. 14.~~Fig. 15. Quadtree Color Palette Output of test Case 1

```
-- KMEANS --
Execution Time: 1.5568 seconds
MSE:            519.2017
Delta E:        15.1289
SSIM:           0.6622

-- QUADTREE --
Execution Time: 0.0143 seconds
MSE:            0.0000
Delta E:        0.0000
SSIM:           1.0000
```

~~Fig. 15.~~Fig. 16. Text Output of test Case 1

3) Test Case 3

~~Fig. 16.~~Fig. 17. Input Image of Test Case 3

```
=== Color Palette Extraction Comparison ===
Enter the absolute path to the image: C:\Users\Syifa\Desktop\ColorPaletteExtractor\tests\Idle.png
Enter the desired number of colors: 3
```

~~Fig. 17.~~Fig. 18. Text Input of test Case 3

~~Fig. 18.~~Fig. 19. K-Means Color Palette Output of test Case 3

~~Fig. 19.~~Fig. 20. Quadtree Color Palette Output of test Case 3

```
=== Evaluation Results ===

-- KMEANS --
Execution Time: 1.7507 seconds
MSE:            29.5564
Delta E:        6.4862
SSIM:           0.9235

-- QUADTREE --
Execution Time: 0.0545 seconds
MSE:            109.1935
Delta E:        9.4997
SSIM:           0.6126
```

~~Fig. 20.~~Fig. 21. Text Output of test Case 1

~~*Selection (Heading 4):* Highlight all author and affiliation lines.~~

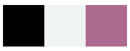
4) Test Case 4

~~Fig. 21.~~Fig. 22. Input Image of Test Case 4

```
=== Color Palette Extraction Comparison ===
Enter the absolute path to the image: C:\Users\Syifa\Desktop\ColorPaletteExtractor\tests\Fall (32x32).png
Enter the desired number of colors: 3
```

~~Fig. 22.~~Fig. 23. Text Input of test Case 4

~~Fig. 23.~~Fig. 24. K-Means Color Palette Output of test Case 4



~~Fig. 24.~~Fig. 25. Quadtree Color Palette Output of test Case 4

```
=== Evaluation Results ===

-- KMEANS --
Execution Time: 1.4181 seconds
MSE:            59.4533
Delta E:        6.5120
SSIM:           0.8844

-- QUADTREE --
Execution Time: 0.0729 seconds
MSE:            234.8104
Delta E:        12.5653
SSIM:           0.4130
```

~~Fig. 25.~~Fig. 26. Text Output of test Case 1

5) Non-Image Result Table

TABLE I. TEST RESULTS

Test Case	Algorithm	Execution Time (s)	MSE	Delta E	SSIM
1	K-Means	1,8763	0	0	1
	Quadtree	0,0131	0	0	1
2	K-Means	1,5568	519,2017	15,1289	0,6622
	Quadtree	0,0143	0	0	1
3	K-Means	1,7507	29,5564	6,4862	0,9235
	Quadtree	0,0545	109,1935	9,4997	0,6126
4	K-Means	1,4181	59,4533	6,5120	0,8844
	Quadtree	0,0729	234,8104	12,5653	0,413
Avg.	K-Means	1,6505	152,05275	7,03	0,8674
	Quadtree	0,0387	86,0001	5,5162	0,7564

~~Fig. 28.~~ *Non-Image results*

~~Fig. 27. Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select "1 Column" from the selection palette.~~

B. Analysis

1) Time Complexity and Execution Time

The Quadtree algorithm and K-Means clustering differ significantly in computational characteristics. The K-Means algorithm iteratively refines cluster centers based on Euclidean distance in color space, resulting in a time complexity of

$$O(n \cdot k \cdot i)$$

where n denotes the number of pixels sampled (in this case, 10,000), k is the number of clusters, and i is the number of iterations until convergence. Despite the use of a subsampling technique to accelerate the process, the iterative nature of K-Means still results in relatively high execution time, measured at average of the four cases, 1.6505 seconds in the conducted experiments.

In contrast, the Quadtree algorithm operates via recursive spatial decomposition. An image region is subdivided into four quadrants if the perceptual color variation (measured using the Delta E 2000 metric) exceeds a predefined threshold. The recursion continues until all blocks meet homogeneity criteria or reach a minimum block size. The average-case time complexity is approximated as

$$O(n \,\mathbf{log}\, n)$$

where n is the number of pixels, due to the recursive splitting and color evaluation over spatial blocks. As no iterative global optimization is required, the Quadtree method achieves significantly faster processing, completing in average of 0.0387 seconds, which has 97,66% time decreases.

This stark difference in execution time demonstrates the Quadtree method's suitability for time-sensitive applications where rapid palette extraction is required.

2) Color Accuracy

Color fidelity is assessed using two metrics: Mean Squared Error (MSE) in LAB space and average Delta E 2000 (ΔE\Delta EΔE), which captures perceptual differences in color.

The K-Means algorithm optimizes cluster centers to minimize intra-cluster variance globally. This results in lower reconstruction error. Such performance indicates effective preservation of the original color distribution.

On the other hand, the Quadtree-based method constructs a palette by averaging color values within spatial blocks and subsequently filtering or clustering the block means. Although perceptually guided, this approach does not globally optimize for color similarity across the entire image. Consequently, it yields a higher MSE and a larger Delta E, signifying lower color accuracy.

While the Quadtree method is faster, it introduces more approximation, leading to a less faithful color reconstruction compared to K-Means.

3) Structural Similarity

To evaluate how well the methods preserve spatial structure and texture, the Structural Similarity Index (SSIM) is utilized. SSIM measures perceptual similarity between the original and recolored images by comparing luminance, contrast, and structural components.

K-Means achieves an average that is indicating high preservation of visual structure. Since each pixel is individually reassigned to its closest cluster center, the technique avoids spatial artifacts and maintains detail.

In contrast, the Quadtree method applies a uniform color to each block, disregarding internal variation. This simplification introduces blockiness and loss of detail, resulting in a lower SSIM score average.

Although this reduction in structural fidelity may be acceptable in use cases prioritizing speed or simplicity (e.g., low-resolution rendering or abstracted visualization), it is less appropriate for high-fidelity image processing tasks.

Formatted: figure caption, No bullets or numbering

~~XXIX.~~ VI. CONCLUSION

AThis paper has presented an in-depth comparison between the Quadtree-based and K-Means clustering approaches for extracting representative color palettes from digital images. Through rigorous experimental evaluations, it was found that the Quadtree algorithm excels in computational efficiency, achieving an average of 97.66% faster execution time compared to K-Means. This is attributed to its divide-and-conquer mechanism which leverages spatial locality and perceptual color thresholds to limit unnecessary processing.

However, the speed advantage of Quadtree comes at the cost of accuracy. K-Means consistently produced lower Mean Squared Error (MSE) and Delta E values, indicating better color reconstruction and perceptual alignment. Furthermore, K-Means achieved higher SSIM scores, preserving more of the original image's structure and texture.

In summary, the Quadtree method is advantageous for real-time or resource-constrained applications, where speed is prioritized over exact visual fidelity. On the other hand, K-Means remains the better choice for applications that require high color accuracy and structural preservation, such as high-quality visualization or digital archiving. Future work may consider hybrid models that combine the spatial sensitivity of Quadtree with the global optimization capability of K-Means to balance speed and accuracy effectively.

> We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi resolution TIFF or EPS file with all fonts embedded) because this method is somewhat more stable than directly inserting a picture.
>
> To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

ACKNOWLEDGMENT ~~(Heading 5)~~

First of all, the author, Noumisyifa Nabila Nareswari, would like to express her deepest gratitude to Mrs. Nur Ulfa Maulidevi, the lecturers in Class 1, for delivering numerous valuable lessons throughout the course. The author would also like to extend her thanks to all the lecturers in ITB, especially Mr. Rinaldi Munir, for the dedication in sharing knowledge, despite the challenges.

Lastly, the author would like to say that the lessons given throughout the semester not only deepened her understanding of the subject but also strengthened her appreciation for the effort and passion of the lecturers.

REFERENCES

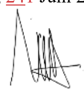
[1] IBM. (n.d.). *K-means clustering*. IBM. https://www.ibm.com/id-id/topics/k-means-clustering

[2] Prakash, A. (2020, June 11). *Transform an image into a quadtree*. Medium. https://medium.com/analytics-vidhya/transform-an-image-into-a-quadtree-39b3aa6e019a

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24~~1~~ Juni 2025



~~Ttd~~

~~Nama~~ Noumisyifa Nabila Nareswari (13523058)~~dan NIM~~

Formatted: Justified

Commented [NN4]: