OPEN ACCESS

Volume 9 (3), July-September 2025, 1024-1034

E-ISSN:2580-1643

Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

DOI: https://doi.org/10.35870/jtik.v9i3.3802

Classification of Skin Color Detection System to Determine Color Selection in Cosmetics (Foundation) Using Modified Chamfer Matching Algorithm Method

Dadang Iskandar Mulyana [1], Indah Rosmalina [2*]

[1,2*] *Informatics Engineering Study Program, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.*

article info

Article history:
Received 10 January 2025
Received in revised form 10 February 2025
Accepted 10 March 2025
Available online July 2025.

Keywords:
Detection; Skin Color; CMA.

Kata Kunci:
Deteksi; Warna Kulit; CMA.

abstract

Skin color detectionis one of the segmentation processes that separates object regions in an image based on color differences. Objects that have a certain color are separated from objects that have other colors. The segmentation results can be used for further processes such as feature extraction or image classification. In this example, skin color is defined in the YCbCr color space with a Cb value between 77 and 127 and a Cr value between 133 and 173. Skin color detection is one of the initial stages in computer vision to detect things related to humans (people detection). Skin color detection can be used as a segmentation method for face recognition or recognition of other body organs. The system can be further developed for biometric systems.

abstrak

Deteksi warna kulit merupakan salah satu proses segmentasi yang memisahkan daerah objek dalam suatu citra berdasarkan perbedaan warna. Objek yang memiliki warna tertentu dipisahkan dari objek yang memiliki warna lain. Hasil segmentasi dapat digunakan untuk proses selanjutnya seperti ekstraksi fitur atau klasifikasi citra. Dalam contoh ini, warna kulit didefinisikan dalam ruang warna YCbCr dengan nilai Cb antara 77 dan 127 serta nilai Cr antara 133 dan 173. Deteksi warna kulit merupakan salah satu tahap awal dalam computer vision untuk mendeteksi hal-hal yang berhubungan dengan manusia (people detection). Deteksi warna kulit dapat digunakan sebagai metode segmentasi untuk pengenalan wajah atau pengenalan organ tubuh lainnya. Sistem tersebut dapat dikembangkan lebih lanjut untuk sistem biometrik.

Corresponding Author. Email: indahrosmalina@gmail.com [2].

ACM Computing Classification System (CCS)
EBSCOhost
Communication and Mass Media Complete (CMMC)

1. Introduction

When selecting makeup products, great care must be taken, as skin color significantly influences the choice of product color. If the chosen makeup product does not match the skin color and skin condition of the face, it will likely have adverse effects, causing skin irritation and a mismatch in color with the user's skin type. The face is one of the most important human identifiers due to its unique nature. Through the face, a person can be easily recognized, typically by remembering the face of someone previously known. Facial skin is also the most important organ and is constantly cared for by humans. Humans generally recognize the type or color of facial skin by directly observing it. However, this is not as straightforward for computers, which require more complex mechanisms to achieve the same. To recognize facial skin, computers must use specific processes, including face detection, feature extraction, feature matching, and face identification. Face detection is a specialized form of pattern detection and is typically the first stage in face recognition. Face detection in an image is done by determining the location and size of each face (if present) in the image.

Many methods have been developed for face detection. According to Viola and Jones (2001), face detection methods can be grouped into four categories: (1) knowledge-based methods, (2) invariant feature-based methods, (3) template matching methods, and (4) appearance-based methods. The template matching method is one that has been widely developed. Several algorithms are used to compute the distance between the image model and the template in the template matching method, such as Euclidean distance, Hausdorff distance, and Chamfer distance. Template matching using Chamfer distance is referred to as Chamfer matching. Chamfer matching was first introduced by Barrow *et al.* (1977). The original idea of Chamfer matching has several advantages, such as its ability to handle imperfect data, though errors still occur in the matching process. As the matching problem has been studied intensively, several solutions have emerged, including developments of Chamfer matching itself, such as the hierarchical Chamfer matching algorithm (HCMA) and the modified Chamfer matching algorithm (MCMA) (Aziz *et al.*, 2019b; Cahyaningsih *et al.*, 2021). The face has a different color compared to other objects in the environment. The face has a color similar to other body parts, specifically skin color. Skin color, like other colors, can be represented by a range of values indicating that the color is skin color, but the color may change depending on brightness levels or the influence of light. Thus, the skin color range changes depending on the brightness conditions. YCbCr color space can separate the brightness value, called luminance, from the original color value. The color displayed by YCbCr is the original color without the influence of luminance, so the skin color range can be determined. By utilizing YCbCr, objects in an image can be categorized as skin and non-skin. The objects to be matched in the template matching process are only those with skin color, thus saving processing time and reducing the risk of matching errors. The face has a specific skin color, shape, and size. The skin color can be detected by determining the range of values that classify a color as skin color, and this can be achieved by utilizing the YCbCr color transformation. After undergoing the skin color segmentation process, the face is not the only object detected as having skin color; other objects such as body parts may also be detected. To distinguish a face from other objects that also have skin color, the shape of the face can be used. The shape of the face can be recognized using a specific method, one of which is the template matching method (Daniati & Nugroho, 2017; Jufri, 2022).

Digital image representation refers to a function of light intensity f(x, y), where the values of x and y represent spatial coordinates and the value of the function at each point (x, y) corresponds to the level of brightness of the image at that point. A digital image is represented by discretizing spatial coordinates (sampling) and discretizing brightness levels (quantization). In this context, the digital image is essentially a matrix where the row and column indices represent a point in the image, and the matrix elements, known as picture elements or pixels, represent the brightness level at that point. An 8-bit grayscale image has 256 possible color intensities ranging from 0 (black) to 255 (white) (Cahyaningsih *et al.*, 2021). Face detection can be seen as a pattern classification problem where the input is an image and the output is the class label of the image, which could either be "face" or "non-face" (Aziz *et al.*, 2019b).

Many facial recognition techniques assume that facial data have the same size and a uniform background, but in real-world scenarios, faces can appear in various sizes, positions, and with varying backgrounds (Daniati & Nugroho, 2017). Research areas related to face processing include face recognition, which involves comparing an input facial image to a database of faces to find the best match; face authentication, which tests the similarity of a face to previously stored data; face localization, which assumes there is only one face in the image; face tracking, which estimates the location of a face in a video; and facial expression recognition, which is used to identify human emotional states (Firman *et al.*, 2016). Face detection involves various challenges, including variations in face position (e.g., upright, tilted, or turned), the presence of facial components like mustaches or glasses, the impact of facial expressions (e.g., smiling or talking), obstruction by other objects or faces, and the influence of image capture conditions, such as lighting or camera characteristics (Gede Astuti *et al.*, 2021). Face detection methods can be categorized into four types: knowledge-based methods, feature-invariant approaches, template matching methods, and appearance-based methods (Homepage *et al.*, 2019).

The YCbCr color model, also known as the CCIR 601 color space, was developed to support the growing demand for video-based information and is widely used in digital video. Unlike analog color systems such as YUV and YIQ, YCbCr is a digital color system that separates RGB values into luminance and chrominance information, which is useful for compression applications (Krismawati, 2021). When RGB values are transformed into YCbCr, they are split into three components: Y (luminance), Cb (chrominance), and Cr (chrominance). This transformation helps to manage color information more efficiently, especially in video transmission, where compression and storage are important (Kumarahadi *et al.*, 2020). By using YCbCr, skin color can be effectively segmented, distinguishing skin from non-skin objects, which is crucial for applications like face detection and recognition (Jufri, 2022). In conclusion, understanding digital image representation and face detection techniques is essential for accurately identifying faces and processing visual data. The YCbCr color model plays a key role in segmenting skin colors, which is a critical step in face detection algorithms. Through methods such as template matching and recognition of facial expressions, these technologies contribute to advancements in biometric systems and real-time applications such as video surveillance and human-computer interaction.

$Y = 0.299900R + 0.58700G + 0.11400B$
$CB = -0.16874R – 0.33126G + 0.50000B$
$CR = 0.50000R – 0.41869G –0.08131B$ (2)
$R = 1.00000Y + 1.40200CR$
$G = 1.00000T – 0.34414CB – 0.71414CR$
$B = 1.00000Y +1.77200CB$ (3)

The RGB-YCbCr conversion formulation in other formats is shown as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 64.481 & -37.757 & 112 \\ 128.553 & -74.203 & -93.786 \\ 24.996 & 112 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The color transformation from CIE RGB to CIE XYZ base can be done as follows, given the RGB triplet (Ri, Gi, Bi) for pixel i, then the XYZ triplet (Xi, Yi, Zi) is calculated as.

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.99 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}$$

The reverse transformation from CIE XYZ to CIE RGB can be done with the equation.

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 2.365 & -0.3896 & -0.468 \\ -0.515 & 1.425 & 0.088 \\ 0.005 & -0.014 & 1.009 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

The RGB and XYZ color models are CIE standards widely used in various imaging and display systems. In addition to these, specific hardware platforms have proposed their own color models to optimize image representation. For example, the National Television Systems Committee (NTSC) developed a specialized RGB color model used to display color images on CRT screens. This format is primarily used on televisions in the United States, with one key advantage being that it separates grayscale data from color data. This separation allows the same signal to be used for both color and black-and-white displays. Regarding edge detection in digital images, it refers to the identification of sudden changes in the intensity of gray levels across short distances. There are three main types of edges found in images: steep edges, which exhibit sharp intensity changes and typically have a direction around 90 degrees; ramp edges, which have a gradual intensity transition and can be viewed as a series of local edges closely grouped together; and edges containing noise, which often occur in real-world images. To handle such noisy edges, image enhancement operations are performed before applying edge detection. The purpose of edge detection is to identify boundaries between homogeneous regions of an image that have differing brightness levels. This operation helps in marking parts of the image that form crucial details, correcting blurred image details caused by errors or acquisition processes, and converting 2D images into more complex, curved shapes. A point (x, y) in an image is considered an edge if it exhibits a significant intensity difference compared to its neighboring points.

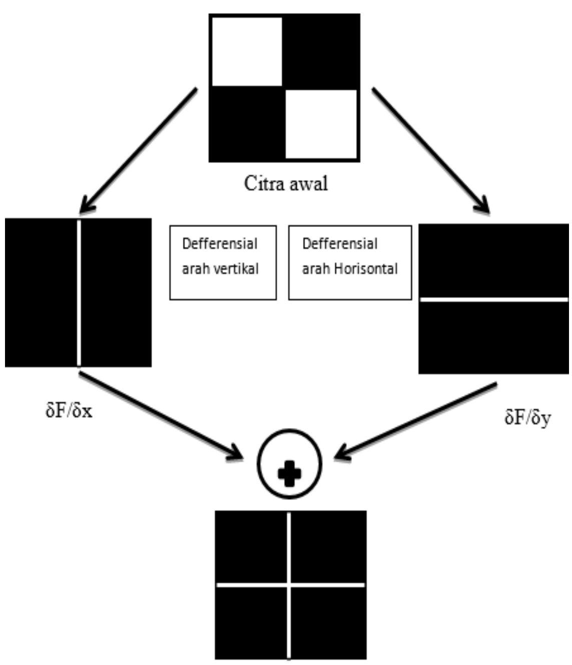

Figure 1. Edge Detection Process

Based on the principles of filters on images, the edges of an image can be obtained using a High Pass Filter (HPF), which has the following characteristics:

$$\sum_y \sum_x H(x, y) = 0$$

Operatorwhich are often used in the edge detection process are usually Robert cross operators, prewitt operators, sobel operators and laplace operators (Cahyo, 2010). In this study, the sobel operator will be used for the edge detection process. The process used by the sobel operator is a process of a convolution that has been set against the detected image. In the sobel operator, a 3 X 3 convolution matrix is used and the arrangement of the pixels around the pixel (x, y) as in Figure 2 below:

$$\begin{bmatrix} P1 & P2 & P3 \\ P8 & (x, y) & P4 \\ P7 & P6 & P5 \end{bmatrix}$$

Sobel operator is a development of Robert operator using High Pass Filter (HPF) filter which is given one zero buffer. This operator takes the principle of laplacian and gaussian functions which are known as functions to generate HPF. The advantage of this Sobel operator is the ability to reduce noise before performing edge detection calculations. So the gradient value can be calculated using the equation:

$$Sx = (p3 + cp4 + p5) - (p1 + cp8 + p7)$$

$$Sy = (p1 + cp2 + p3) - (p7 + cp6 + p5)$$

$$G = \sqrt{Sx^2 + Sy^2} \quad (8)$$

$$Sx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The Sobel operator typically emphasizes or weights pixels that are closer to the center point of the window, ensuring that the influence of neighboring pixels differs based on their position relative to the point where the gradient is calculated. The arrangement of the weighting values in the window also reveals that the gradient calculation combines both horizontal and vertical positions. In image processing, binarization refers to converting grayscale images into binary images, which have two gray-level values: black and white. The binarization process involves determining a threshold value (T) to decide whether a pixel is black or white. There are three common methods for determining this threshold: a global threshold where T depends on the gray level of the pixel at position (x, y), a local threshold based on the properties of neighboring pixels, and a dynamic threshold that depends on the pixel coordinates and surrounding information. In this study, the threshold value was derived from the YCbCr color range indicating skin color, and pixels that met this threshold were assigned a value of one, while others were assigned zero.

Chamfer matching, introduced by Barrow *et al.* in 1977, is a technique used in object detection and classification, renowned for its tolerance to mismatches in position, scale, and rotation. This method is used to find the best match between the vertices of two different images by minimizing the common distance between them. The vertices of an image are transformed through a set of parametric transformation equations, which describe how the images can geometrically deviate from one another. The Chamfer distance between two shapes is calculated as the average of the distances between each point in one set and the nearest point in the other set. This distance can be computed more efficiently using a distance transform (DT), which assigns each pixel in the image the distance to its nearest feature, allowing for faster calculation of the Chamfer distance between the template and the edge map.

2. Research methodology

The primary problem addressed in this study is how to detect faces within an image by utilizing the YCbCr color model and the Modified Chamfer Matching Algorithm (MCMA). This section will explore the analysis of using the YCbCr color space to identify human skin color and the application of the MCMA for template matching in face detection. The system diagram depicting the face detection process is shown in Figure 2. Through this methodology, we aim to optimize the accuracy of face detection by leveraging the combination of color segmentation and geometric matching techniques.

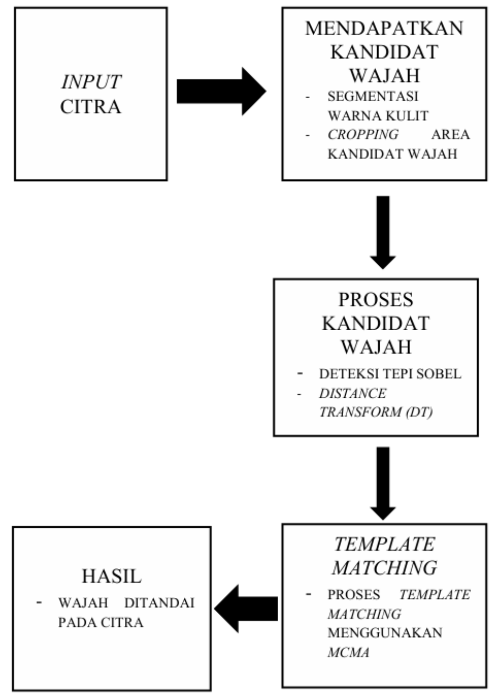

Figure 2. Face Detection System Diagram

Color Analysis

After the user selects the image to be tested, the input image undergoes the skin color segmentation process. This segmentation is carried out using the YCbCr color model. Initially, the input image is converted from the RGB color model to YCbCr using the RGB to YCbCr conversion equation. The color conversion from RGB to YCbCr will follow the formula provided in equation (2). To perform the conversion, the image is first stored in the BufferedImage class in Java, and the Color class is used to obtain the RGB values of the image, which consist of red, green, and blue components. These RGB values are then used in the conversion equation to obtain the corresponding YCbCr values. For example, if the Red (R) value is 3, the Green (G) value is 2, and the Blue (B) value is 3, the YCbCr value of the pixel is calculated using the given formula. The RGB to YCbCr color conversion process is illustrated in Figure III-2 below.

$Y = 0.299900R + 0.58700G + 0.11400B$
$= 0.299900*3 + 0.58700*2 + 0.11400*3$
$= 0.8997 + 1.174 + 0.342 = 2.4157$
$Cb = -0.16874R – 0.33126G + 0.50000B$
$= -0.16874*3 – 0.33126*2 + 0.50000*3$
$= -0.50622 – 0.66252 + 1.5$
$= 0.33126$

Cr = 0.50000R – 0.41869G – 0.08131B
= 0.50000*3 – 0.41869*2 – 0.08131*3
= 1.5 – 0.83738 – 0.24393 = 0.41869

Color Analysis and Face Detection Process

After the user selects an image for analysis, the input image undergoes the skin color segmentation process. This process involves converting the image from the RGB color model to the YCbCr color model. The first step in this conversion is to use the RGB to YCbCr conversion equation, as shown in Equation (2). To perform the conversion, the image is stored in the BufferedImage class in Java, and the Color class is used to obtain the RGB values, which consist of red, green, and blue components. These RGB values are then used to calculate the corresponding YCbCr values. For example, if the Red (R) value is 3, the Green (G) value is 2, and the Blue (B) value is 3, the resulting YCbCr values are calculated, with Y being 2.4157, Cb being 0.33126, and Cr being 0.41869. These values are then rounded according to the 0-255 pixel color range, so Y becomes 2, Cb becomes 0, and Cr becomes 0. These values are then used in the skin detection process.

After the color conversion, the next step is to identify potential face candidates. The image is segmented into two categories: skin and non-skin. The YCbCr color range used to classify skin color is based on the following criteria: Y must be greater than 80, Cb must lie between 85 and 135, and Cr must fall between 135 and 180. If the pixel values meet these criteria, the pixel is marked as white, indicating it is likely skin. If the values do not meet the criteria, the pixel is marked as black. The next step involves using the connected component labeling (CCL) method to detect regions of the image that contain white pixels, which represent skin. The areas are labeled, and if an area contains more than 100 white pixels, it is considered a candidate for face detection. The identified face candidate is then cropped, resized to 80 x 119 pixels to match the face template, and stored in the BufferedImage class.

Edge Detection and Distance Transform

Once the face candidate is obtained, edge detection is performed using the Sobel operator. The pixel values of the candidate face are used to calculate the gradient values in the x and y directions (Sx and Sy) through the Sobel equation. The resulting gradient values are then checked and adjusted if they fall outside the allowable range (0-255). If the gradient value is below 0, it is set to 0, and if it exceeds 255, it is set to 255. The adjusted gradient values are used to update the pixel values, completing the edge detection process. Following edge detection, the distance transform (DT) process is applied to the face candidate. The pixel values of the candidate face are obtained and stored in an array, which is then processed to compute the distance transform using the equations provided in Chapter 2. After the processing, the resulting values are used to update the pixel values of the face candidate, transforming it into a distance transform format.

Template Matching and Face Identification

The final step in the face detection process is template matching. This process compares the face candidate with a pre-existing face template. The goal is to determine whether the candidate is a valid face or not. If the distance between the face candidate and the face template is less than 2%, the candidate is declared a face. The Chamfer distance equation, as described in Chapter 2, is used to calculate the distance. The pixel values of both the face candidate and the face template are stored in arrays, which are then compared using the Chamfer distance calculation. If the distance is below the threshold, a red box is drawn around the face candidate to mark it as a detected face.

Software Analysis

The software developed for this study is designed to detect human faces by utilizing skin color segmentation and template matching using the Modified Chamfer Matching Algorithm (MCMA). The software accepts input in the form of RGB images in JPEG format. The functionality of the developed system is depicted in Figure 3. After the pixel values of the image are obtained, they are processed using the distance transform and Chamfer matching algorithms to detect faces. If the results of the template matching indicate a match, the face candidate is marked with a red box in the image, confirming it as a detected face. The software developed in this study is designed for human face detection, utilizing skin color segmentation to identify candidate faces, followed by template matching using the Modified Chamfer Matching Algorithm (MCMA). The software accepts input in the form of RGB images in JPEG format. The system's functionality is depicted in Figure 3. Once the pixel values are obtained, they are stored in an array. This array is then processed to calculate the distance transform, using equations (12) and (13) from Chapter 2. After processing, the distance transform value is applied to update the pixel values in the face candidate, resulting in a distance-transformed face candidate.

Next, the system performs template matching to determine whether the face candidate matches the template. If the results of the template matching process indicate a match, the candidate is declared a face and is marked with a red box. The candidate face is considered a valid match if the calculated distance between the candidate face and the template face is less than 2%. The distance calculation follows equation (11) in Chapter 2, which uses Chamfer distance, a key component of the Chamfer Matching method. Each pixel value from both the face candidate and the template is stored in arrays, where the candidate face array represents variable "u" and the face template array represents variable "v." The Chamfer distance is then calculated based on these values. After obtaining the calculation results, the system checks if the Chamfer distance is below 2%. If this condition is met, the face candidate is confirmed as a face and a red box is drawn around the detected face in the tested image. The software is specifically designed to work with RGB JPEG images, ensuring that the system processes the images accurately through these steps. The developed system's functionality is visualized in Figure 3, showing the overall flow from input image processing to face detection output.

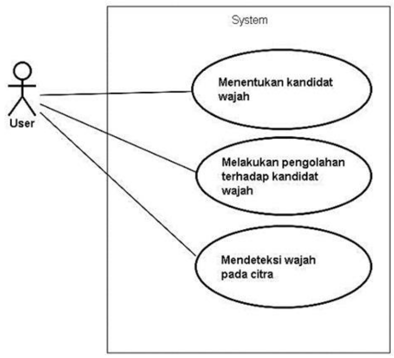

Figure 3. Usecase Diagram

3. Results and Discussion

Results

The developed application was tested to evaluate the alignment between the software design and its implementation. The results indicate that the tested unit and interface function as expected. In the test shown in Figure 7, face candidates are successfully identified and marked with red boxes. These candidates are then processed and used in the template matching process to determine whether they represent a face. Conversely, the test in Figure 8 demonstrates that no face candidates were detected, as evidenced by the absence of red boxes marking any face candidates. The reason for this is that the input image was in grayscale, which prevented the system from identifying skin regions necessary for face detection. Figure 4 illustrates the successful completion of the face candidate processing, where the image of the face candidate is displayed as a distance transform. This processed image is then ready for the template matching process. These tests validate the software's ability to correctly identify and process face candidates, laying the foundation for further refinements in face detection and matching accuracy.

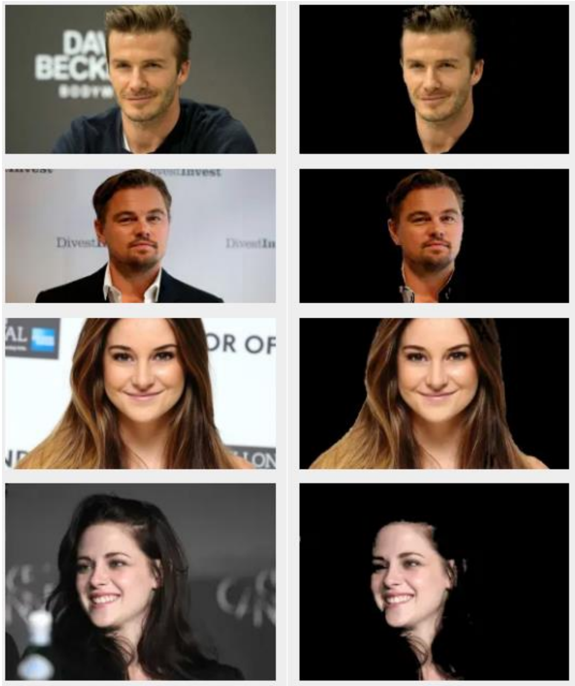

Figure 4. Testing determines candidate faces

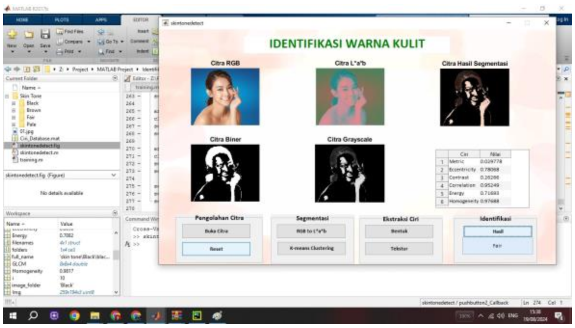

Figure 5. Testing to determine skin color

The test in Figure 10 demonstrates that the face detection process has been completed successfully. The face in the image is marked with a red box, which indicates that the template matching results between the candidate face and the face template are appropriate. However, the presence of other non-face objects also marked with red boxes suggests that errors still exist in the matching process, as these results do not fully align with the expected output. These discrepancies indicate that the template matching algorithm needs further refinement to improve its accuracy in distinguishing faces from other objects. The results from the tests are summarized in Table 1, which includes four key factors: 1) the entire face was detected, 2) parts of the face were not detected, 3) objects other than the face were detected, and 4) the face was not detected. These detection outcomes can be quantified using the calculation provided in Equation 4.1, allowing for a more detailed analysis of the software's performance in various conditions.

Table 1. Face Skin detection result

No	Name File	Face Detection Results			
		Detected all over the face	There are some not detected	Detected besides face object	Not detected
1	S1	✓			
2	S2	✓			
3	S3	✓			
4	S4	✓		✓	
5	S5	✓		✓	
6	S6	✓			
7	S7	✓		✓	
8	S8	✓			
9	S9	✓			
10	S10	✓		✓	
11	S11		✓	✓	
12	S12		✓	✓	
13	S13	✓		✓	
14	S14		✓	✓	
15	S15		✓	✓	
16	S16		✓	✓	
17	S17		✓	✓	
18	S18			✓	
19	S19		✓	✓	
20	S20		✓	✓	
21	S21	✓			
22	S22	✓		✓	
23	S23		✓	✓	
24	S24		✓	✓	

25	S25			✓	✓
26	S26			✓	✓
27	S27	✓		✓	
28	S28		✓	✓	
29	S29	✓			
30	S30	✓			

$$\text{Persentase (\%)} = \frac{Jumlah\ citra\ kategori}{Jumlah\ seluruh\ citra\ uji} \times 100\%$$

After performing calculations using Equation 12 above, the following results were obtained:

1) The test image successfully detected the entire face and there were no other objects besides the face detected at 30%.
2) The test image successfully detected the entire face and there were other objects besides the face that were detected at 23.33%.
3) Test images that contain undetected faces and other objects besides detected faces are 40% 4. Test images that fail to detect faces are 6.66%.

Discussion

The results from the testing of the face detection software indicate a mixed performance, with both strengths and areas for improvement. As seen in Figure 10, the face detection process successfully identified the face in the image, marking it with a red box. This confirms that the system's template matching method, combined with the Modified Chamfer Matching Algorithm (MCMA), works as expected in detecting faces under ideal conditions. However, the presence of non-face objects also marked with red boxes suggests that the system still struggles with distinguishing faces from other objects. This issue highlights one of the major challenges in face detection: the presence of similar color or shape characteristics between faces and other objects, which leads to false positives. This is consistent with findings in previous research, such as Aziz *et al.* (2019b), who noted that face detection systems often face difficulties with distinguishing between faces and similar-looking objects in cluttered environments. The results summarized in Table 1 provide additional insight into the software's performance.

The outcomes ranging from successful face detection to partial detection and misidentification of non-face objects as faces suggest that while the system works well in simple scenarios, its accuracy diminishes when faces are partially obstructed or when the image contains noise or complex backgrounds. Similar challenges have been addressed in other studies, such as those by Cahyaningsih *et al.* (2021), who discussed the importance of refining detection algorithms to minimize errors in complex environments. These errors, as observed in the test results, suggest that additional pre-processing steps, such as noise reduction or enhanced background segmentation, could help improve detection accuracy. To quantify the performance of the system, the detection results can be calculated using Equation 4.1, which offers a more detailed view of how the system performs across different detection scenarios. By analyzing the results using this equation, it becomes evident that the software is effective in ideal conditions but requires further refinement to handle more complex real-world images, such as those with partial faces, occlusions, or cluttered backgrounds. Research by Daniati and Nugroho (2017) also points out that improving the segmentation process can significantly enhance the accuracy of face detection, especially when dealing with such challenging cases. Furthermore, the errors observed in the detection process can be attributed to the limitations of the template matching method used in this study. As noted by Jufri (2022), template matching can struggle with images where faces are rotated, scaled, or obscured. To address this, further refinement of the MCMA could be explored, possibly by incorporating advanced techniques such as machine learning or deep learning, which have been shown to provide more robust face detection performance (Krismawati, 2021; Kumarahadi *et al.*, 2020). These techniques could help the system better handle variations in facial appearance and background noise.

In conclusion, while the face detection system demonstrates strong potential, especially in controlled environments, it requires further improvements to handle more diverse and complex images effectively. Enhancing the accuracy of the template matching process, refining skin color segmentation, and incorporating more advanced detection methods could significantly improve the system's robustness, aligning it with the performance goals set in related studies (Firman *et al*., 2016).

4. Conclusion

In conclusion, this study highlights several key findings regarding the use of YCbCr color space and the Modified Chamfer Matching Algorithm (MCMA) in face detection. Firstly, YCbCr color proves to be effective for skin color segmentation by utilizing the range of skin color values in YCbCr, specifically $Y > 80$, $85 \leq Cb \leq 135$, and $135 \leq Cr \leq 180$. This allows the system to accurately distinguish skin regions from non-skin areas in an image. Secondly, the Modified Chamfer Matching Algorithm is shown to be a viable method for the template matching process, facilitating the detection of face candidates. However, the accuracy of MCMA in determining faces needs improvement, as evidenced by the results: the system achieved 30% accuracy in detecting the entire face, 23.33% accuracy when other objects were mistakenly detected as faces, 40% accuracy when some faces were not detected, and 6.66% accuracy when faces were entirely missed. These results indicate that while the system has potential, further refinements are needed to enhance its performance, especially in complex or cluttered images.

5. References

Aziz, A., & Karpen, K. (2019). Diagnosa Penyakit Kulit Wajah Menggunakan Metode Decession Tree Dan Algoritma C4. 5. *Jurnal Teknologi Dan Open Source*, *2*(1), 74-86. https://doi.org/10.36378/jtos.v2i1.148.

Cahyaningsih, S., Triayudi, A., & Sholihati, I. D. (2021). Kombinasi metode certainty factor dan forward chaining untuk identifikasi jenis kulit wajah berbasis Android. *Jurnal Media Informatika Budidarma,* *5*(1), 74. https://doi.org/10.30865/Mib.V5i1.2591.

Daniati, E., & Nugroho Skom, A. (2017). Aplikasi perawatan wajah berdasarkan jenis kulit wajah. *Jurnal Simki-Techsain*, *1*, 12.

Firman, A., Wowor, H. F., & Najoan, X. (2016). Sistem informasi perpustakaan online berbasis web. *Jurnal Teknik Elektro dan Komputer*, *5*(2), 29-36.

Gede Astuti Gede Aditya Mahardika Pratama, L. (2021). The expert system design to identify laptop damage by applying certainty factor method. *International Research Journal of Engineering, IT & Scientific Research, 7*(4), 165–175. https://doi.org/10.21744/Irjeis.V7n4.1898.

Hamrani, A., Leizaola, D., Reddy Vedere, N. K., Kirsner, R. S., Kaile, K., Trinidad, A. L., & Godavarty, A. (2024). AI Dermatochroma Analytica (AIDA): Smart Technology for Robust Skin Color Classification and Segmentation. *Cosmetics*, *11*(6), 218.

Hsiao, S. W., Yen, C. H., & Lee, C. H. (2017). An intelligent skin‐color capture method based on fuzzy C‐means with applications. *Color Research & Application*, *42*(6), 775-787.

Jufri, M. (2022). Designing an Expert System for Diagnosing Otitis Disease Using Forward Chaining and Certainty Factor Methods. *IJISTECH (International Journal of Information System and Technology)*, *6*(2), 282-289. https://doi.org/10.30645/ijistech.v6i2.240.

Kakumanu, P., Makrogiannis, S., & Bourbakis, N. (2007). A survey of skin-color modeling and detection methods. *Pattern recognition*, *40*(3), 1106-1122. https://doi.org/10.1016/j.patcog.2006.06.010.

Krismawati, Y. (2021). EXPERT SYSTEM FOR DETECTION OF SKIN ON THE FACE IN THE D'ANGELS BEAUTY CLINIC. *International Journal of Artificial Intelligence and Robotic Technology (IJAIRTec)*, *1*(1), 17-21.

Kumarahadi, Y. K., Arifin, M. Z., Pambudi, S., Prabowo, T., & Kusrini, K. (2020). Sistem pakar identifikasi jenis kulit wajah dengan metode certainty factor. *Jurnal Teknologi Informasi dan Komunikasi (Tikomsin), 8*(1).

Pebrianto, R., Nugraha, S. N., & Gata, W. (2020). Perancangan Sistem Pakar Penentuan Jenis Kulit Wajah Menggunakan Metode Certainty Factor. *IJCIT (Indonesian Journal on Computer and Information Technology)*, *5*(1), 83-93.

Sitinjak, D. D. J. T., & Suwita, J. (2020). Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang. *Insan Pembangunan Sistem Informasi dan Komputer (IPSIKOM)*, *8*(1). https://doi.org/10.58217/ipsikom.v8i1.164.