JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 17, ISSUE 3, 2023, 9542 – 9551
DOI: https://doi.org/10.15282/jmes.17.3.2023.1.0755

اونيۏرسيتي مليسيا ڤهڠ السلطان عبدالله
UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

RESEARCH ARTICLE

# Crowd counting algorithm based on face detection and skin color recognition

**Y.N. Hao[1,2,*], V.C. Tai[2], Y.C. Tan[2]**

[1] Department of Electrical Engineering, Taiyuan Institute of Technology, No.31 Xinlan Road, Taiyuan, Shanxi 030008, China
[2] Centre for Modelling and Simulation, Faculty of Engineering, Built Environment and Information Technology, SEGI University, 47810 Petaling Jaya, Selangor, Malaysia
Phone: +60361451777; Fax.: +60361452725

**ABSTRACT -** This paper introduces an innovative crowd counting algorithm using skin color information. Through stages of color space transformation, threshold segmentation, morphological processing, and region filtering, the algorithm successfully conducts crowd counting in images. The study encompasses analyses of images with diverse crowd densities, skin colors, backgrounds, and lighting intensities, revealing the algorithm's robustness to various factors. It remains unaffected by skin color and crowd size and exhibits minimal sensitivity to background and lighting intensity. Furthermore, the paper explores image feature analysis and uses MATLAB programming for simulation and initial crowd counting, considering images with different actual crowd sizes. Despite minor issues such as the insufficient separation of faces from clothing and the influence of lighting intensity, the algorithm performs reliably in most scenarios, demonstrating high crowd counting accuracies. To bolster the accuracy and robustness of the algorithm, optimization of the separation step and control of the lighting effect on images is suggested. The key focus of this study is the application of the Gaussian model in the YCbCr color space for face detection and examining its impact on the efficiency and accuracy of crowd counting algorithm. The research not only provides a novel approach for crowd counting in images but also offers insightful perspectives for future studies and potential improvements. Thus, the study proves to be a significant contribution to face detection and recognition technology, enhancing its application in fields like public safety, crowd management, and surveillance systems.

**ARTICLE HISTORY**

| | | |
|---|---|---|
| Received | : | 19th Jan. 2023 |
| Revised | : | 09th July 2023 |
| Accepted | : | 04th Aug. 2023 |
| Published | : | 28th Sept. 2023 |

**KEYWORDS**
*Crowd counting algorithm*
*Face detection*
*YCbCr color space*
*Skin color recognition*
*Image processing*

## 1.0 INTRODUCTION

Face detection and recognition technology, a form of biometric identification, uses computer algorithms to automatically acquire and analyze facial features, facilitating the identification and verification of individuals. This technology, incorporating fields such as image processing [1, 2], neural networks and machine learning [3–6], finds extensive applications in diverse areas like access control systems, payment systems, and computer security. Unlike other biometric identification methods, face recognition offers unique advantages: it is widely accepted by the public and provides rich personal information. As science and technology progress, face detection and recognition technology has increasingly broad applications in social life. It not only brings convenience to people's lives but also holds significant economic value. In a surveillance system, face detection and recognition are often used together with crowd counting to identify specific individuals within a crowd.

Crowd counting algorithms are crucial for enhancing public safety and security, providing an efficient means for crowd management. However, diverse crowd distribution and background noise have posed challenges to crowd counting algorithms [7]. Ge and Collins [8] proposed a Bayesian marked point process model is used to detect and count people in crowded scenes by combining spatial and conditional mark processes. Hudaib et al. [9] proposes a hybrid model for counting people based on video streams, utilizing human skin detection and geometric head recognition. Zan et al. [10] proposed to apply deep learning to extract meta-information for improving the accuracy and convergence speed of crowd counting tasks. Stec et al. [11] presents a robust people counting algorithm that utilizes multi-sensor data fusion to achieve accurate results for logistical planning and efficient assignment of transport vehicles in public transportation. Saxena et al. [12] employed the Viola-Jones algorithm to create a people counting system in MATLAB [13]. Ledda et al. [14] proposes to use temporal information between adjacent frames to improve the accuracy of crowd counting. All these studies show a diverse range of strategies were utilized to answer the crowd counting problems, from statistical models to machine learning algorithms, to sensor fusion.

Color space plays an important role in face detection [15]. Humans are sensitive to color information, and rich color information can enhance image features and make them more recognizable. There are various methods of representing colors in computers, forming different color spaces such as RGB, HSI, CMY/CMYK, YUV, and YCbCr [15, 16]. Yang et al. [17] presented a face detection method for high-resolution images with complex backgrounds. YCbCr color model was used to construct the skin color model and an improved AdaBoost algorithm were used to identify potential face

*CORRESPONDING AUTHOR | Y. N. Hao | ✉ yananhao2020@163.com areas, alongside with Particle Swarm Optimization (PSO) algorithm was employed to optimize the neural network parameters, resulting in high detection rates, improved speed, and reduced false and missed detections. Kang et al. [18] proposed a face detector that combined skin color-based face detection and sliding-based face detection methods to reduce the face detection time. In their study, YCbCr color model was used in the study to construct the skin color classifier. Fachrurrozi and Afif [19] presented a study that applied skin color segmentation in YCbCr color space to speed up the face detection process using Viola-Jones algorithm. These studies indicate that skin color model using YCbCr color space is effective in face detection.

This paper introduces the application of the Gaussian model in the YCbCr color space to detect faces in crowd counting algorithms. The study aims to answer the question whether the YCbCr color model can be used to improve the efficiency and accuracy of crowd counting algorithms. Hence, the main objectives of this study are to apply and the Gaussian model in the YCbCr color space for face detection and investigate the impact of the proposed approach on the efficiency and accuracy of crowd counting algorithms.

## 2.0 METHODOLOGY

This study employed the Gaussian model in the YCbCr color space and applies threshold segmentation for image processing, completing the automatic counting and identification of the number of people based on skin color images.

### 2.1 YCbCr Color Space

In the YCbCr color space, $Y$ represents the luminance component, while Cb refers to the difference between the blue chroma component and a reference value, and Cr represents the difference between the red chromaticity component and a reference value. In this color space, it is also possible to separate the luminance component and color feature information, and this space has fast computational speed, making it highly time efficient. It is commonly used for color image compression, transmission, video encoding, and more. The conversion relationship between the YCbCr color space and the RGB color space is as follows [20]:

$$\begin{bmatrix} Y \\ \text{Cb} \\ \text{Cr} \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.996 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{1}$$

Due to the fact that the difference in skin color between individuals is primarily influenced by brightness rather than chromaticity, it is preferable to select a color space that can effectively separate luminance from chromaticity when choosing a color space. Extensive statistical data analysis has shown that skin color exhibits excellent clustering characteristics in the YCbCr color space. Therefore, the YCbCr color space is chosen for processing skin color information. This color space allows for the separation of luminance and chromaticity, with luminance being the primary factor in distinguishing skin color variations. By utilizing the clustering properties of skin color in the YCbCr space, accurate identification and analysis of skin color can be achieved.

### 2.2 Gaussian Model

In the simple Gaussian model, there is a premise that assumes all skin color distributions follow a unimodal Gaussian distribution. Under this premise, statistical analysis is performed on the skin color region to predict the parameters (mean and variance) of the Gaussian distribution. A skin color model is established based on these parameters, which is then used to determine whether each new pixel in an image contains skin color information. The pixels that meet the criteria for skin color information are subsequently clustered together to form a connected region. Compared to the region model, the simple Gaussian model better aligns with the distribution of skin colors, and its model parameters are relatively easy to compute. As a result, it enables the establishment of a more efficient and accurate skin color model [21].

### 2.3 Threshold Segmentation

The threshold segmentation method involves setting a threshold based on the difference in grayscale values between the object to be segmented in the image and other pixels in the image. All pixels in the image are then filtered using this threshold. If a pixel's grayscale value falls within the threshold range, it is classified as a skin color pixel, whereas if it falls outside the threshold range, it is classified as a non-skin color pixel. The formula for threshold segmentation is as follows [22]:

$$g(i,j) = \begin{cases} 1, & f(i,j) \leq T \\ 0, & f(i,j) > T \end{cases} \tag{2}$$

where, $g(i,j)$ is the pixel value of the segmented image, $f(i,j)$ represents the grayscale value of a pixel in the image, $(i,j)$ denotes any pixel in the image, and $T$ is the selected threshold range. This process allows for the division of all pixels in the image into skin tone areas and non-skin tone areas. In practical applications of threshold segmentation, due to the instability of the image acquisition environment, it is not feasible to use a fixed threshold for accurate image segmentation. Therefore, it becomes necessary to divide the image into multiple regions and select thresholds specific to the local range of pixels within these regions. This approach helps achieve effective image segmentation. In the YCbCr color space, skin color can be identified by meeting the chromaticity information requirements of 77≤Cb≤127 and 133≤ Cr≤173. Utilizing the threshold segmentation method, the skin color area and non-skin color area can be rapidly separated.

### 2.4 Crowd Counting Algorithm

The process flow of the proposed crowd counting algorithm is as follows:

**Step 1: Image acquisition**

i. The image containing crowd is uploaded to the system.

**Step 2: Color space conversion**

i. Process images in the RGB color space, isolating the skin color regions.

ii. Convert these images from RGB to the YCbCr color space to take advantage of the better clustering properties of skin color in the YCbCr space.

iii. Conduct statistical analysis on the Cb and Cr components of these skin color regions. This analysis helps to differentiate skin color from non-skin color regions.

iv. Fiber compensation processing:

   a. Understand that skin color chrominance is often influenced by brightness or color differences in imaging devices, resulting in a chrominance deviation in the images.

   b. Sort all pixels in the test image in descending order based on their brightness.

   c. Extract the higher-ranked pixels as a reference white.

   d. Transform the remaining pixels proportionally based on this reference white to compensate for deviations. This step helps to mitigate errors in color information extraction and analysis caused by such color differences.

v. Nonlinear segmented color transformation:

   a. Recognize the nonlinear relationship between chrominance and luminance values upon converting the image to the YCbCr color space.

   b. Notice that the clustering properties of skin color gradually diminish within specific ranges of luminance values.

   c. Apply a nonlinear segmented color transformation to the YCbCr color space to eliminate redundant components during skin color information extraction.

vi. Convert the processed images back to the RGB color space after all the transformations and analyses.

**Step 3: Skin color modeling**

i. Use the Gaussian probability formula for multivariate data to determine the likelihood of a pixel being a skin color pixel [23]:

$$p(x) = \frac{1}{2\pi\sqrt{|\Sigma|}} exp[-0.5(x-m)^T\Sigma^{-1}(x-m)] \tag{3}$$

where, $x = (Cb, Cr)^T$, $m = E(x)$ is the mean and $\Sigma = E(x-m)(x-m^T)$ is the covariance matrix.

ii. Conduct statistical analysis on a large number of images to calculate the mean and covariance matrix. The results are:

$m = (117.4316, 148.5599)$

$\Sigma = \begin{pmatrix} 97.0946 & 24.4700 \\ 24.4700 & 141.9966 \end{pmatrix}$

**Step 4: Skin color segmentation**

i. Apply low-pass filter to eliminate high-frequency noise using the followings impulse response matrix [24]:

$$h = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{4}$$

ii. Obtain a likelihood map:

   a. In the context of the Gaussian distribution skin tone model, calculate the similarity between each pixel and the skin color pixels contained in the image. The similarity algorithm is presented in the next section.

   b. Determine the maximum similarity among these calculated values to obtain the maximum skin tone likelihood across the entire image.

c. Divide the skin tone likelihood of each pixel in the image by the maximum skin tone likelihood obtained in the previous step to provide the grayscale value of each pixel in the image.

ii. Binarize the image to achieve a better separation between the skin tone area and the non-skin tone area. Skin color areas are represented by "1" (white), while the non-skin areas are represented by "0" (black).

iii. Extract the skin color area using the fixed threshold segmentation method.

**Step 5: Morphological treatment**

i. Define the Neighborhoods:

a. Define p(x,y) as the coordinates of a specific pixel in the image. The four neighboring pixels of pixel (x,y), namely, (x-1,y), (x+1,y), (x,y-1), (x,y+1) form a pixel set, known as the 4-neighborhood of pixel p, denoted as N4(p).

b. Additionally, pixel p has four pixels that are diagonally adjacent, with coordinates (x-1,y-1), (x-1,y+1), (x+1,y-1), (x+1,y+1). These form the diagonal neighborhood of pixel p, denoted as ND(p).

c. The combination of these two-pixel sets is referred to as the 8-neighborhood of pixel p, denoted as N8(p), which contains 8 pixels.

ii. Binary Pixel Manipulation:

a. Binary pixels are the fundamental units used in morphological image processing. If translations of a binary image A by B = {$b_1$, $b_2$, ..., $b_n$} are given, the dilation of A by B refers to the translation of A by B. Here, B is called the structuring element:

$$\boldsymbol{A} \oplus \mathrm{B} = \bigcup_{b_2} Ab_1 \tag{5}$$

b. In the case of a binary image, dilation bridges the gaps between closely located connected regions, connecting them together. It can also fill gaps, holes, and other imperfections between connected regions in a binary image. In this experiment, the four-directional dilation method is used.

iii. Erosion:

a. Erosion is the opposite of dilation. For a binary image, erosion eliminates boundary points of regions and regions that do not meet the requirements of the structuring element, thereby smoothing the boundaries of certain areas.

b. The area of the eliminated regions varies depending on the choice of the structuring element. If the selected structuring element is larger than the connected portion between two regions, erosion can separate these two regions. In this experiment, four-directional erosion is used.

iv. Opening and Closing Operations:

a. In practical applications, opening and closing operations are commonly used morphological operations.

b. The opening operation, involving dilation followed by erosion, removes noise points from the image while preserving the shape of objects.

c. The closing operation, involving erosion followed by dilation, eliminates rough edges in the image, resulting in a smoother appearance.

**Step 6: Filtering**

If the target aspect ratio is less than 1.0 or greater than 2.1, it is considered a non-face area.

i. If the area is too large or too small, it is also considered a non-face area [25].

**Step 7: Face counting and marking**

i. Obtain the total number of face regions through the previous statistical filtering step.

ii. Use this total number to determine the number of individuals present in the image.

iii. Mark the position and serial number of each face using a rectangular box.

### 2.5 Similarity Algorithm

The followings steps of similarity algorithm are applied in *Step 4 (ii)* to calculate the similarity between each pixel and the skin color pixels contained in the image:

**Step 1: Grayscale Conversion**

The similarity between skin tone pixels and non-skin tone pixels is calculated, with the similarity represented by grayscale values.

1. Convert an image from RGB format to a grayscale chart using the following conversion equation:

$$\mathrm{Y} = 0.39 \times \mathrm{R} + 0.50 \times \mathrm{G} + 0.11 \times \mathrm{B} \tag{6}$$

**Step 2: Grayscale Value Normalization:**

1. The range of converted gray values can be very wide, making subsequent operations difficult. To manage this, constrain the gray values to the range of [0, 255].
2. Grayscale linear expansion of an image can significantly enhance its quality and clarity.

**Step 3: Grayscale Linear Transformation to map the original gray intensity to new gray intensity:**

1. The calculation for grayscale linear transformation is as follows:

$$I' = \begin{cases} \dfrac{I - I_{min}}{I_{max} - I_{min}}(I'_{max} - I'_{min}) + I'_{min}, & \text{if } \ I \in [I_{min}, I_{max}] \\ I \quad , & else \end{cases} \tag{7}$$

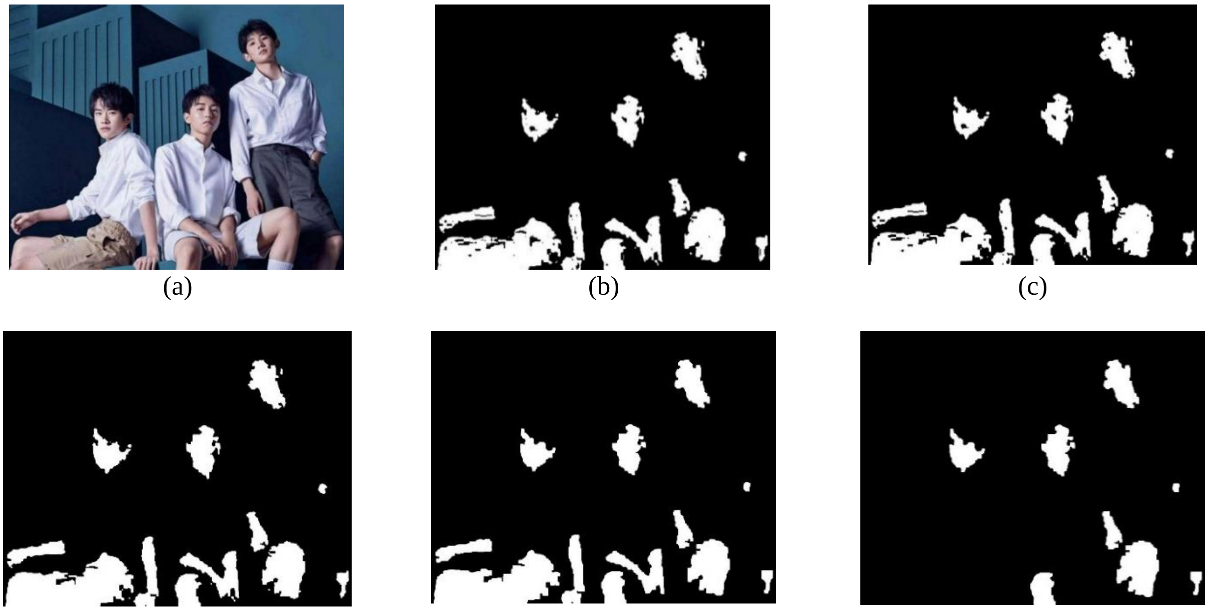
where, $I$ and $I'$ are the original and new gray intensities of a pixel, and the subscripts min and max denote the minimum and maximum intensities, respectively.

## 3.0 RESULTS AND DISCUSSON

Five different images featuring individuals with various skin tones, diverse lighting intensities, and different backgrounds were used, as depicted in Figures 1 to 5. The number of people in each image was counted, and the accuracy of the crowd counting is calculated as follows:

$$\text{crowd counting accuracy } = \frac{\text{actual number of people detected}}{\text{actual number of people}} \times \ 100\% \tag{8}$$

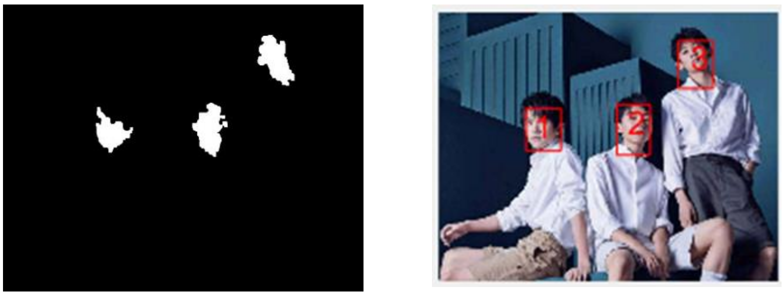
Fixed threshold segmentation is employed to achieve image binarization and dimensionality reduction processing, as demonstrated in Figure 1(b). To eliminate noise, the image is subjected to opening and closing operations, as shown in Figure 1(c). Gaps between regions in the image are bridged through filling, as depicted in Figure 1(d). Erosion and dilation operations are then performed to disconnect adjacent areas, as displayed in Figure 1(e). Upon the completion of morphological processing on the image, the number of faces can be discerned by filtering out connected regions that match the proportion and size of a face area. The first filtering stage, as illustrated in Figure 1(f), excludes areas that do not conform to face proportions, such as arms and legs, based on the ratio of the length and width of the connected region. The second filtering stage, as displayed in Figure 1(g), excludes areas that do not conform to the size of a face, such as larger skin-tone backgrounds or clothes resembling skin tone, based on the image size. Once the connected regions are filtered based on the aforementioned conditions, the total number of people is calculated by counting the connected areas. The positions of the faces are outlined in the original image, and each face is assigned a serial number, as depicted in Figure 1(h).



(a) (b) (c)

(d) (e) (f)

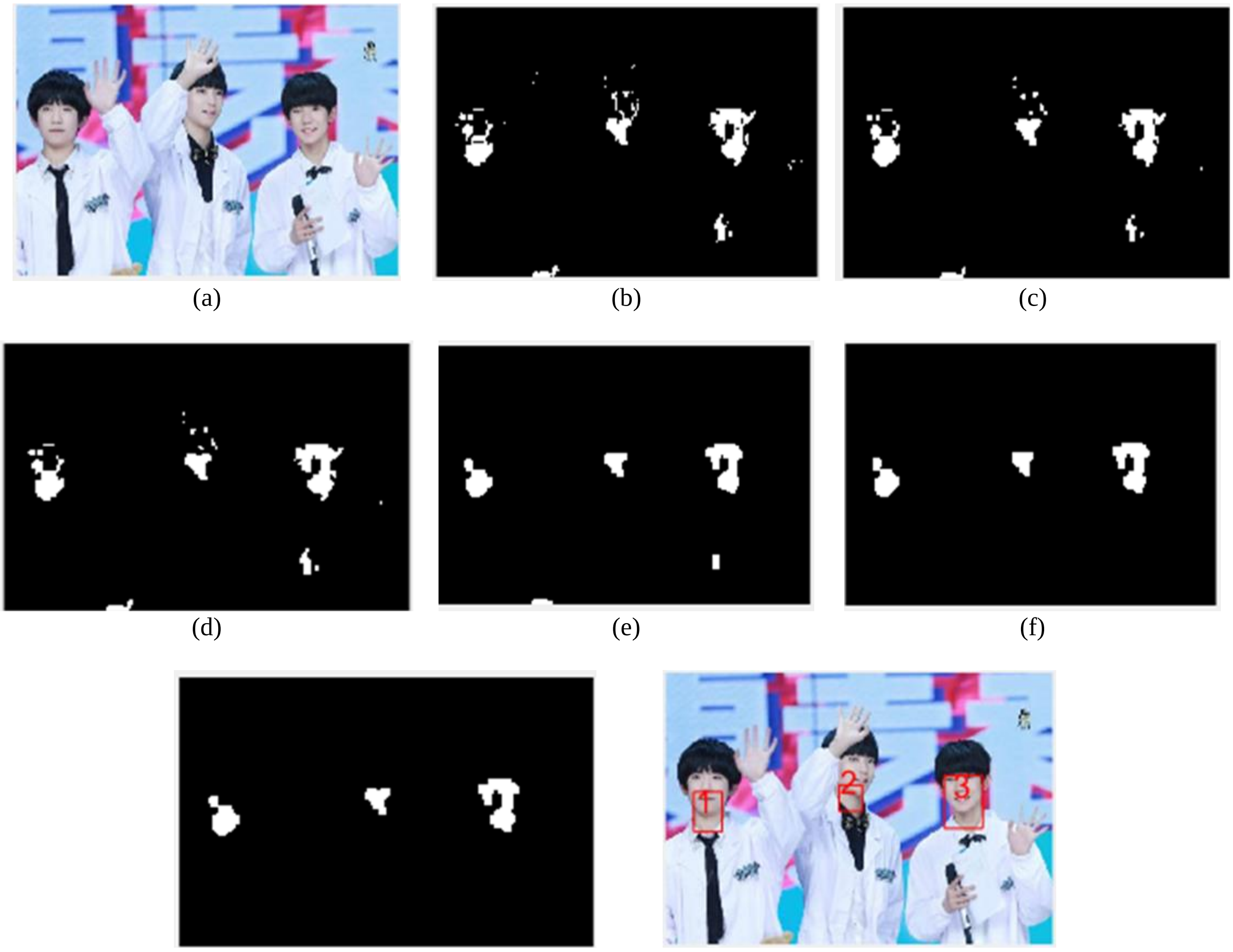
Figure 1. Crowd counting process of image 1: (a) Original image, (b) binarization, (c) noise cancellation treatment, (d) hole filling, (e) disconnect processing, (f) 1st filtering result,



(g) (h)

Figure 1. (cont.) (g) 2nd filtering result and (h) face marking

It is observed in the first filtering stage of this image that, in addition to areas corresponding to face regions, leg regions and hand regions that resemble the shape and proportion of a face are also treated as face regions. This necessitates the second filter based on the size of the face. The crowd counting accuracy is 100% after two filtering stages. For Figure 2, in the upper left corner of the second face, there is a hand obscuring the face, and the second face is in a profile position. This resulted in a smaller detected facial area during the subsequent image processing steps. However, it did not affect the detection of the face, yielding a crowd counting accuracy of 100%.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 2. Crowd counting process of image 2: (a) Original image, (b) binarization, (c) noise cancellation treatment, (d) hole filling, (e) disconnect processing, (f) 1st filtering result, (g) 2nd filtering result and (h) face marking

In Figure 3, variations in skin color are found not to impact the face detection results, yielding a crowd counting accuracy of 100%. The effectiveness of the algorithm is demonstrated through statistical analysis of people under different skin colors, backgrounds, and light intensities. The algorithm is minimally affected by skin color, number of people, background, or light intensity, yielding satisfactory people counting results.

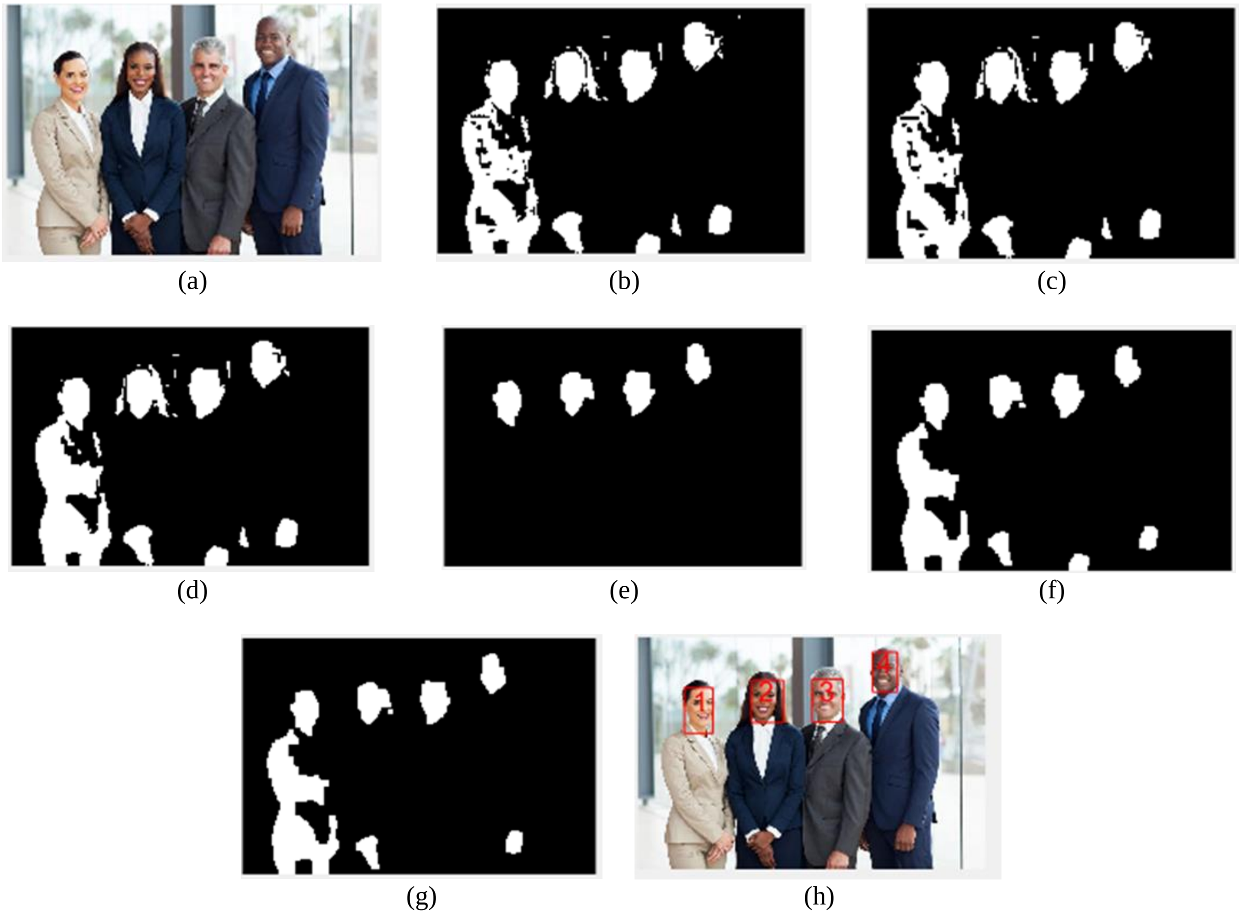

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3. Crowd counting process of image 3: (a) Original image, (b) binarization, (c) noise cancellation treatment, (d) hole filling, (e) disconnect processing, (f) 1st filtering result, (g) 2nd filtering result and (h) face marking.

Figure 4 presents a discrepancy between the actual number of people detected and the true number of people. This discrepancy is caused by the impact of the disconnect operation and lighting intensity on the subsequent image processing. Due to the disconnection processing, the linkage between faces and clothing was not severed, leading to the exclusion of this portion during the initial filtering. Additionally, a portion of the clothing was mistakenly identified as facial features, resulting in a discrepancy between the detected and actual number of people (the regions 5 and 6 as shown in Figure 4(h)). Furthermore, the impact of varying light intensities caused one side of the face with stronger lighting to register as 1 after binarization, while the weaker side registered as 0, affecting subsequent image processing tasks. As a result, a crowd counting accuracy of 66.67% is observed for this image.

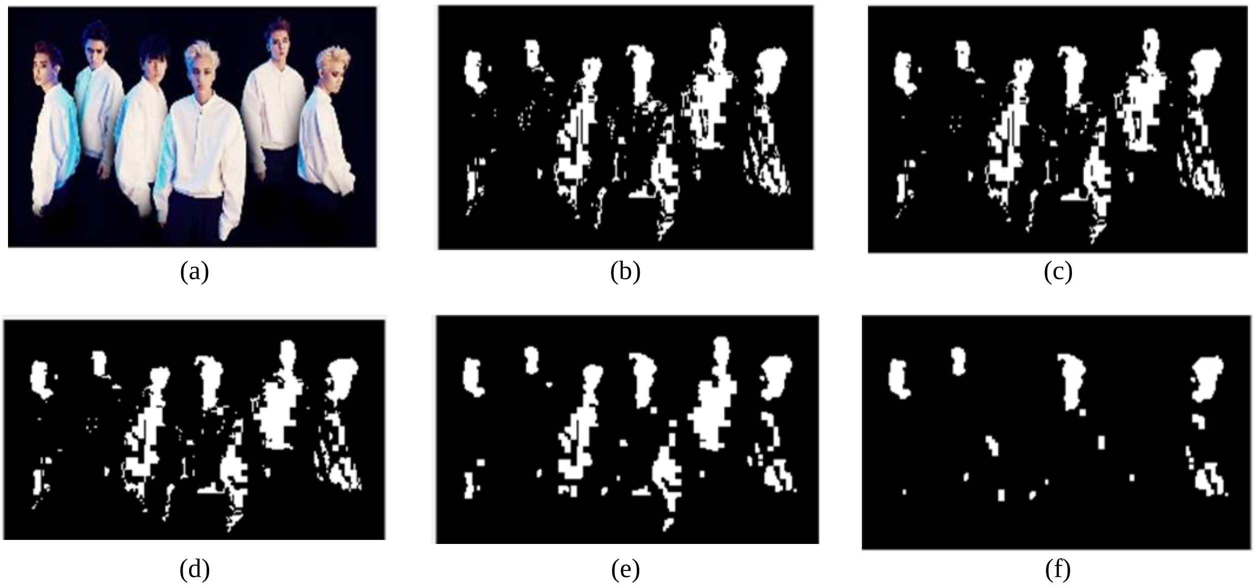
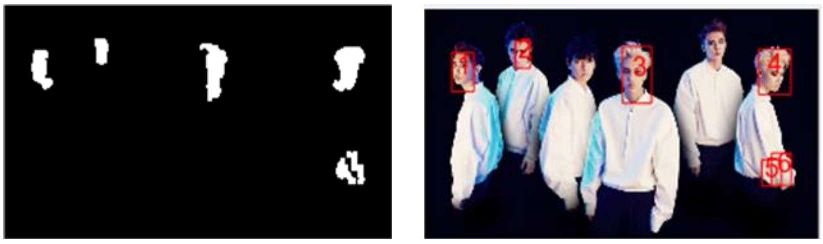

(a) (b) (c) (d) (e) (f)

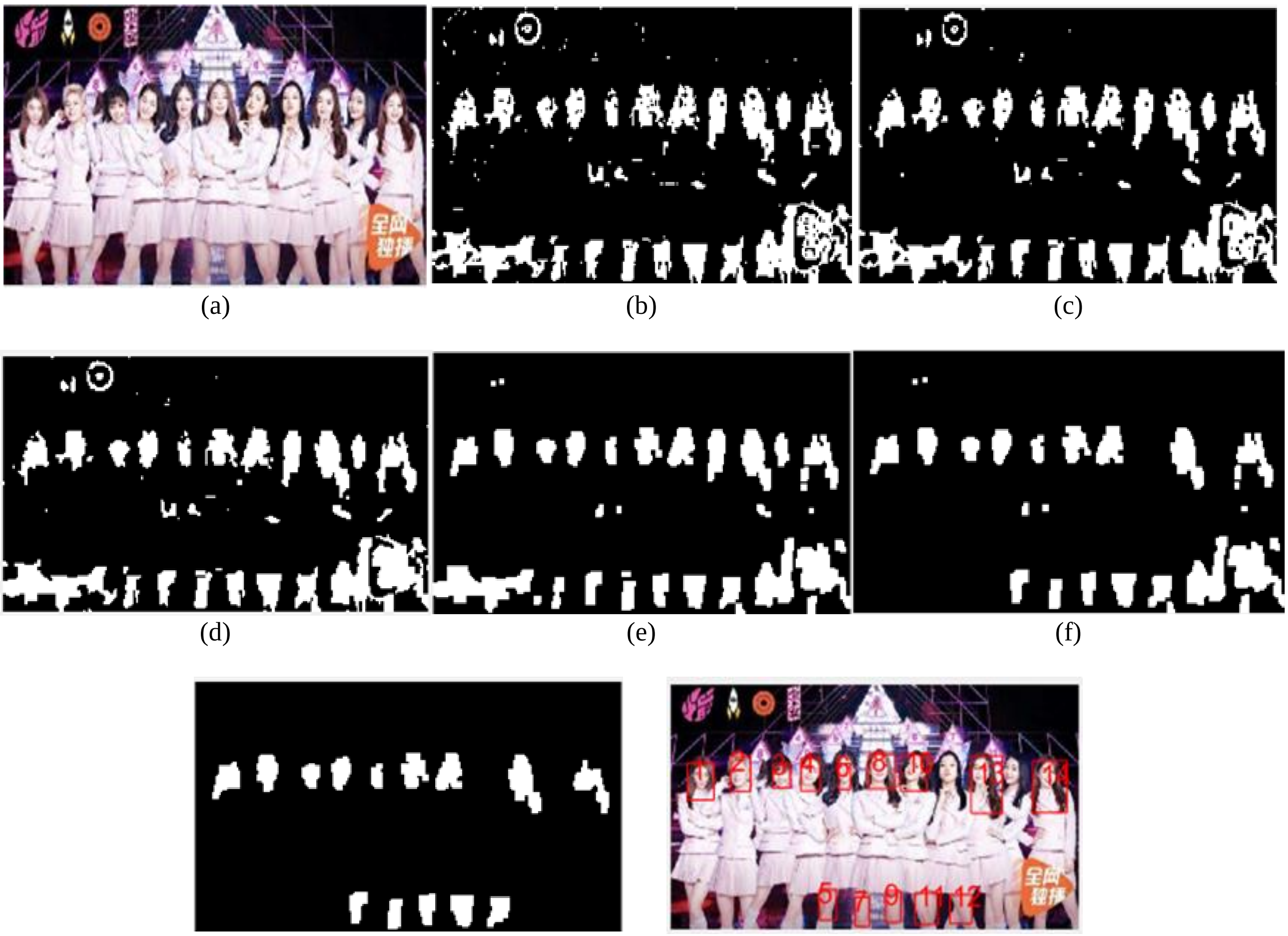
Figure 4. Crowd counting process of image 4: (a) Original image, (b) binarization, (c) noise cancellation treatment, (d) hole filling, (e) disconnect processing, (f) 1st filtering result,



(g) (h)

Figure 4. (cont.) (g) 2nd filtering result and (h) face marking

In Figure 5, which contains a high number of people, the detection is compromised by non-disconnected facial and arm regions, and by distorted facial regions due to a side view. These factors, in addition to knee areas mistaken for faces, (the regions 5,7,9,11,12 as shown in Figure 4(h)), result in a discrepancy between the actual number of people detected and the true number of people. So, a total of 14 individuals were detected, while the actual number of individuals present was 9. Consequently, a crowd counting accuracy of 81.82% is observed.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 5. Crowd counting process of image 5: (a) Original image, (b) binarization, (c) noise cancellation treatment, (d) hole filling, (e) disconnect processing, (f) 1st filtering result, (g) 2nd filtering result and (h) face marking

Table 1 provides a summary of the people counting results. The proposed algorithm is shown to have several advantages. It remains unaffected by skin color or the number of people, ensuring accurate people counting results irrespective of these factors. The algorithm is also minimally impacted by changes in background and lighting conditions. However, limitations are noted when faces overlap, when individuals wear clothing of similar skin tones, or when exposed limbs are present in the image. Despite these potential inaccuracies, the algorithm consistently yields reasonably accurate results across a range of scenarios, thereby fulfilling practical needs.

Table 1. The results of crowd counting

| No. | Actual number of people | Detected number of people | Counting accuracy (%) |
|---|---|---|---|
| 1 | 3 | 3 | 100.00 |
| 2 | 3 | 3 | 100.00 |
| 3 | 4 | 4 | 100.00 |
| 4 | 6 | 4 | 66.67 |
| 5 | 11 | 9 | 81.82 |

## 4.0 CONCLUSIONS

An algorithm for crowd counting in images, utilizing skin color information, is presented in this paper. The algorithm successfully accomplishes crowd counting through stages that include color space transformation, threshold segmentation, morphological processing, and region filtering. Analyses are conducted on images featuring diverse skin colors, varying crowd densities, backgrounds, and lighting intensities, leading to certain conclusions. The algorithm exhibits various advantages in crowd counting: it remains unaffected by skin color and crowd size, and is minimally impacted by background and lighting intensity, thereby producing reliable results.

Image feature analysis is specifically carried out in the study, with features being extracted based on skin colour recognition. Image processing techniques and MATLAB programming are utilized to simulate and conduct initial crowd counting. When examining images with varying actual crowd sizes, certain phenomena and issues are observed. For instance, when the actual crowd size is three, regions resembling facial shapes and proportions, such as legs and hands, are also classified as facial regions. However, crowd counting accuracy reaches 100% after two filtering steps. Moreover, when the actual crowd size is four, despite the variety in skin colors, the results of face detection remain unaffected, yielding a crowd counting accuracy of 100%. Conversely, when the actual crowd sizes are six and eleven, certain clothing is misclassified as facial regions due to insufficient separation of faces from clothing during the processing step. The impact of lighting intensity introduces errors into the image processing, resulting in crowd counting accuracies of 66.67% and 88.82%, respectively.

The proposed algorithm for crowd counting, based on skin color information, performs well in most scenarios. However, certain issues remain to be addressed, including the separation of faces from clothing and the influence of lighting. To further enhance the accuracy and robustness of the algorithm, optimization of the separation step and controlling the lighting effect on images are recommended. This research presents a novel approach for crowd counting in images and provides valuable insights for future studies and potential improvements.

## 5.0 ACKNOWLEDGMENTS

The project is funded by Cooperative Education Program of the Ministry of Education (No. 220504309231428)

## 6.0 REFERENCES

[1] S. Sharma and V. Kumar, “Performance evaluation of 2D face recognition techniques under image processing attacks,” *Modern Physics Letters B*, vol. 32, no. 19, p.1850212, 2018.

[2] J. Zeng, X. Qiu, S. Shi, “Image processing effects on the deep face recognition system,” *Mathematical Biosciences and Engineering*, vol. 18, no. 2, pp. 1187–1200, 2021.

[3] J. Liu and M. A. Ashraf, “Face recognition method based on GA-BP neural network algorithm,” *Open Physics*, vol. 16, no. 1, pp. 1056–1065, 2018.

[4] Q. Wang and G. Guo, “Benchmarking deep learning techniques for face recognition,” *Journal of Visual Communication and Image Representation*, vol. 65, p.102663, 2019.

[5] J. Lu, X. Yuan, and T. Yahagi, “A method of face recognition based on fuzzy clustering and parallel neural networks,” *Signal Processing*, vol. 86, no. 8, pp. 2026–2039, 2006.

[6] G. Guo and N. Zhang, “A survey on deep learning-based face recognition,” *Computer Vision and Image Understanding*, vol. 189, p.102805, 2019.

[7] H. Mo *et al.*, “Background noise filtering and distribution dividing for crowd counting,” *IEEE Transactions on Image Processing*, vol. 29, pp. 8199–8212, 2020.

[8] W. Ge and R. T. Collins, “Marked point processes for crowd counting,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, vol. 1-4, pp. 2913–2920, 2009.

[9] A. Hudaib and K. Kaabneh, “Hybrid model for people counting in a video stream,” *12th WSEAS International Conference on Computers*, Heraklion, Greece, pp. 482-486, 2008.

[10] C. Zan, B. Liu, W. Guan, K. Zhang, and W. Liu, "Learn from object counting: Crowd counting with Meta-learning," *IET Image Processing*, vol. 15, no. 14, pp. 3543–3550, 2021.

[11] M. Stec, V. Herrmann, and B. Stabernack, "Multi-sensor-fusion system for people counting applications," *1st International Conference on Societal Automation (SA)*, Krakow, Poland, pp. 1–4, 2019.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features" *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA. vol. 1, pp. 511-518, 2001.

[13] S. Saxena and D. Songara, "Design of people counting system using MATLAB," *10th International Conference on Contemporary Computing (IC3)*, Noida, pp. 345-347, 2017.

[14] E. Ledda, L. Putzu, R. Delussu, G. Fumera, and F. Roli, "On the evaluation of video-based crowd counting models," *Image Analysis and Processing – ICIAP 2022*, in *Lecture Notes in Computer Science*, vol. 13233, pp. 301–311, 2022.

[15] H. A. Nugroho, R. D. Goratama, and E. L. Frannita, "Face recognition in four types of colour space: a performance analysis," *IOP Conference Series: Materials Science and Engineering*, vol. 1088, no. 1, p.012010, 2021.

[16] D. J. Sawicki and W. Miziolek, "Human colour skin detection in CMYK colour space," *IET Image Processing*, vol. 9, no. 9, pp. 751–757, 2015.

[17] X. Yang, N. Liang, W. Zhou, and H. Lu, "A face detection method based on skin color model and improved AdaBoost algorithm," *Traitement du Signal*, vol. 37, no. 6, pp. 929–937, 2020.

[18] S. Kang, B. Choi, and D. Jo, "Faces detection method based on skin color modeling," *Journal of Systems Architecture*, vol. 64, pp. 100–109, 2016.

[19] M. Fachrurrozi and K. K. Afif, "Face detection using the Viola-Jones method with segmentation of skin color on face images," *Journal of Engineering Science and Technology*, vol. 15, no. 4, pp. 2149–2162, 2020.

[20] R. Mohanty and M. V. Raghunadh, "A new approach to face detection based on YCgCr color model and improved AdaBoost algorithm," *International Conference on Communication and Signal Processing (ICCSP)*, Tamilnadu, India, vol. 1, pp. 1392–1396, 2016.

[21] M. Ryan and N. Hanafiah, "An examination of character recognition on ID card using template matching approach," *Procedia Computer Science*, vol. 59, pp. 520–529, 2015.

[22] T. Chen, T. Chen, and Z. Chen, "An intelligent people-flow counting method for passing through a gate," *IEEE Conference on Robotics, Automation and Mechatronics*, Bangkok, Thailand, vol. 1-2, pp. 1–6, 2006.

[23] S.-K. Ueng and C.-Y. Chang, "An improved skin color model," *International Conference on Applied System Innovation (ICASI)*, Okinawa, Japan, pp. 1–4, 2016.

[24] H. Foroughi, N. Ray, and H. Zhang, "Robust people counting using sparse representation and random projection," *Pattern Recognition*, vol. 48, no. 10, pp. 3038–3052, 2015.

[25] A. J. Ma, P. C. Yuen, and Jian-Huang Lai, "Linear dependency modeling for classifier fusion and feature combination," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 35, no. 5, pp. 1135–1148, 2013.