Facial detection and recognition

Advanced Image Processing

Mila Shahnavaz, milsh107
Julia Ronnefalk, julro676
Anna Granberg, anngr950
Filip Hamrelius, filha243

Linköping University

SE-581 83 Linköping

013-28 10 00, www.liu.se

Summary

This report covers the implementation of facial recognition through a MATLAB-based system. The system employs techniques including masking, morphological operations, segmentation and detection. PCA, eigenfaces and feature vectors were used for detection methods. The purpose of this project was to explore different image processing methods and understand their weaknesses and potential when working together. Segmentation complexities and the reliance on static thresholds, were challenges that had to be improved for a more robust approach. Color correction proved to be a challenge as well which required some trial and error, the first approach was implemented with the method Gray World, later to be changed to the White Patch method, which improved the result. Other challenges included the normalization process, specifically identifying the real eyes from all the false positives. This resulted in a quite complex algorithm which iteratively goes through the potential eye candidates. In summary the system performs with a 81% accuracy on a dataset containing 16 images under eight different transformations, i.e 144 query images including the non transformed images.

Table of contents

1. Introduction

This report explores the implementation process and method pipeline of a facial recognition program within the course *Advanced Image Processing*. The program endeavors to employ relevant techniques in the field of image processing, including masks, morphological formulas, and detection methods. Additionally, it incorporates data science methods such as Principal Component Analysis (PCA), eigenfaces, and feature vectors.

1.1. Background

Facial recognition has gained popularity in security systems and authentication due to advancements in technology. Systems can swiftly identify individuals and cross-reference them with databases, enhancing security in various settings. Additionally, facial recognition has become a commonly used authentication method in mobile devices, offering secure and convenient unlocking and transaction authorization. However, discussions about privacy, ethics, and potential misuse are essential as the technology continues to evolve. Balancing enhanced security with individual rights remains a critical consideration.

1.2. Requirements

The facial recognition program aimed to identify or reject portrait images of various individuals based on a predefined dataset containing 16 different individuals. It would return either the index of the particular person or zero for a non-existing person. Furthermore, the system was supposed to be able to adjust for alterations and distortions in these images. These included:

- Translation of the face in the image (30 pixels),
- rotation (+/- 5°),
- scaling (+/- 10 %) and
- change of tone values in the image (max - 30 %, +10%).

1.3. Purpose

The purpose of the project revolves around leveraging fundamental image processing methods and their potential to facilitate large-scale programs. Additionally, the project aims to delve into the realm of PCA, eigenfaces, and feature vectors. This forms the basis of an advanced image processing project where we created a facial recognition program, combining theoretical understanding with hands-on implementation successfully.

2. Method

The methodology combines color correction, segmentation and transformation to optimize facial recognition. To ensure consistency of the system, techniques such as rotation, scaling and cropping are conducted. Additionally, Principal Component Analysis (PCA) together with eigenfaces are employed to identify key features and subsequently perform the classification.

2.1. Color Correction

In a facial recognition program, color correction plays a significant role as it directly influences the accuracy and reliability of the recognition process. It is crucial for a real-world system to handle images with different environments, such as varying lighting conditions and color tones. To handle these variations, color correction methods such as Gray World and White Patch are used to maintain consistency across the images.

2.1.1 Gray World

Gray World has the function to fulfill equality between the R, G and B-channels, and thereby result in a scene that is balanced to a neutral gray. Gray world is recommended when the image has light interference [2]. To achieve Gray World, an average of the image is calculated with the help of the R,G and B-channels, this is then used to calculate the adjustment factor for each channel. Adjustment factor helps to get an achromatic result and the final value for each channel, see equations (1).

$$Gray_{avg} = (R_{avg} + G_{avg} + B_{avg})\frac{1}{3}$$
$$aR = \frac{Gray_{avg}}{R_{avg}}$$
$$R = R \cdot aR$$

(1) [2]

2.1.2 White Patch

Another key component of the color correction is the White Patch method. The method works by calculating the maximum intensity values in the red, green, and blue channels, and thus identifying the brightest regions of the image. Using these maximum values, normalization is then applied to the entire image. This approach ensures that the brightest areas become pure white, and thus the method becomes a practical solution to enhancing the overall image quality, and also the accuracy for steps later on in the pipeline, such as skin detection [1].

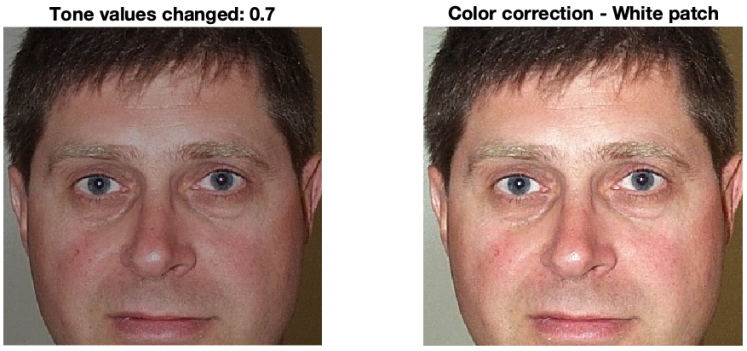

Figure 1 - White patch on manipulated tone values.

2.2. Segmentation

The purpose of segmentation is to eventually standardize all images. The segmentation aims to extract vital features from the face, so the later steps in the facial recognition program can have the same reference point for all images. In this instance the program aims to extract the eyes, face and mouth. And with this information finding the eye coordinates.

2.2.1 Cropping

Before any real segmentation is done, all images undergo an initial crop. The cropping ratio can be found heuristically and is unique to the dataset in question. All images are cropped with the same ratio, and aims to remove some of the unnecessary data surrounding the portrait.

2.2.2 Face mask

The facial mask operates within the YCbCr color space, where the program iteratively processes the image and binarizes it based on skin color. This involves applying specific thresholds for each channel in the YCbCr space, determined through trial and error. To remove outlying objects, morphological operations like dilation and erosion are used.

2.2.3 Eye map

The eye map is based on two separate eye maps derived from the color space YCbCr where the first one is constructed with the chrome channel and the other the luminance channel.

$$EyeMapC = \frac{1}{3}\left\{(C_b^2) + (\tilde{C}_r)^2 + (C_b/C_r)\right\}, \tag{2}$$

$$EyeMapL = \frac{Y(x,y) \oplus g_\sigma(x,y)}{Y(x,y) \ominus g_\sigma(x,y) + 1}, \tag{3}$$

Where ”$\oplus$“ and “$\ominus$” represent dilation and erosion operations. The combination between the two eye maps is constructed with equation (4).

$$EyeMap \;=\; (EyeMapC)\,\&\,(EyeMapL) \tag{4}$$

[4]

Where the “&” represents, in our case, element-wise multiplication. The eye map is based on theory from the paper “Face Detection in Color Images” by Hsu, Abdel-Mottaleb and K.Jain [4].

2.2.4 Mouth map

In order to locate the mouth an understanding about the YCbCr channel values must be considered, especially the relationship between the Cr and Cb channel. Since in most cases the Cr component will be greater than the Cb. The first draft of the mouth map is constructed with equation (5) and (6) [4]. Morphological operations are then used to refine and enhance clarity of the mouth map. Normalization and thresholding ensures standardized values and clear distinction between mouth and non-mouth regions.

$$MouthMap = C_r^2 \cdot \left(C_r^2 - \eta \cdot C_r/C_b\right)^2, \tag{5}$$

$$\eta = 0.95 \cdot \frac{\frac{1}{n}\sum_{(x,y)\in FG} C_r(x,y)^2}{\frac{1}{n}\sum_{(x,y)\in FG} C_r(x,y)/C_b(x,y)}, \tag{6}$$

[4]

2.2.5 Skin color detection

To minimize false positives related to the eyes and mouth, the skin color can be taken into consideration. Utilizing this as a mask subsequently should enhance the accuracy in detecting the eyes. The skin color detection operates in the YCbCr color space. The process involves extracting the chrominance components (Cb and Cr). Pixels with Cb values between 77 and 200 and Cr values between 134 and 173 are considered potential skin pixels, these values can be found heuristically and ultimately depend on the dataset in question. The identified skin pixels are then processed using morphological operations to enhance the accuracy of the skin region. The resulting binary mask is applied to the original image, isolating the detected skin

region. The final output includes a grayscale representation of this region, which is then binarized.

The decision to opt for the YCbCr color space was influenced by the findings presented in the article "*Detecting skin in face recognition systems: A color spaces study*" [3]. While YCbCr demonstrated comparable results to HSV and RGB, its noteworthy performance in the Cr-channel seemed better for this use case since alot of images in the dataset contained more red tones than other tones in the skin colors.

2.2.6 Eye and mouth coordinates

Extracting the eye coordinates is a crucial step in the pipeline, as the eye map alone does not exclusively identify the eyes. Therefore the eye extraction process needs to effectively differentiate between false positives and the actual eyes.

The coordinate function initiates by using the built-in function *regionprops* to identify the mouth. With this information the coordinate function then eliminates everything located beneath the mouth, including the mouth itself, resulting in an image cropped to accentuate the eyes. Employing MATLAB's built-in *imfindcircles*, which relies on the circular Hough transform with a low threshold, the algorithm identifies all potential circles within the image, including numerous false positive eyes.

The algorithm then selects the two most promising candidates, the two circles found first by the *imfindcircles* function, assuming them to be the left and right eyes. If the left eye is excessively positioned to the right, the order is flipped, designating the initially assumed left eye as the right eye, even if it involves a false positive for the right eye.

Upon confirming the accuracy of the left eye, the algorithm proceeds to evaluate the right eye. It checks for conditions such as the steepness of the presumed eyes slope, their proximity to each other, or excessive separation. If any of these conditions are not met, the function iterates through potential right eyes until a match with the left eye's conditions is found. Finally, the function verifies the correct positioning of the left and right eyes; if necessary, it swaps them before concluding the process and returning the coordinates for the eyes.

The different steps within the segmentation pipeline can be seen in figure 2.

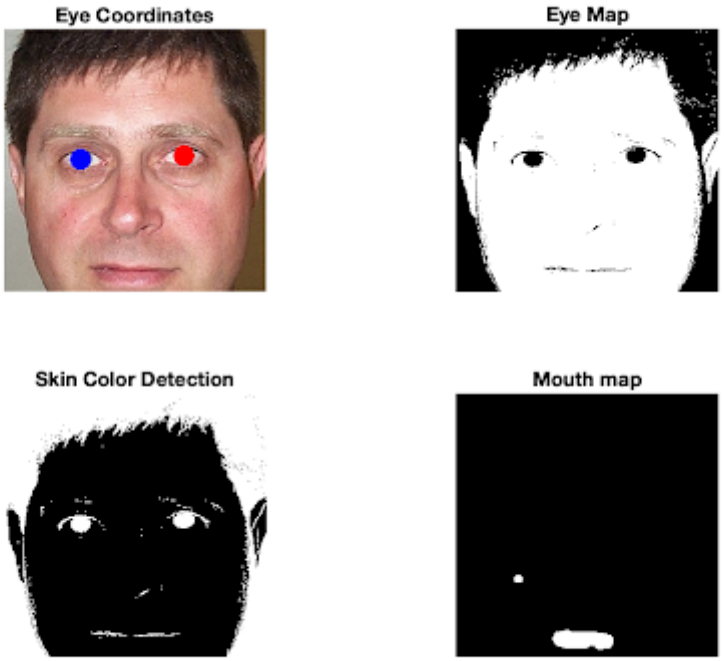

Figure 2 - Eye, Mouth and skin Detection.

2.4 Transformations

The transformations in the pre-processing phase of the facial recognition system are necessary operations that addresses variabilities in facial appearances. This makes sure that the subsequent stages of the pipeline can work effectively and consistently when analyzing and identifying facial features.

2.4.1 Padding and translation

As a first step in the transformation phase, the image is padded to a predetermined target size. This step is for handling variations in the dimensions of the images and providing a standardized size for the subsequent steps. Following the padding and with help of the calculated eye coordinates, the left eye is translated to the center of the image to receive a standard reference point for all images and for the following operations.

2.4.2 Rotation

To handle any tilts or angular disparities in the facial image, a rotation is performed. This is done using the angle between the line connecting the eye coordinates and the horizontal axis. This angle serves as a representation of the relative positioning of the eyes within the facial structure. The calculation ensures that the rotation aligns the eyes horizontally, which is essential for the consistency of the system.

2.4.3 Scaling

The following step includes scaling the images to standardize the distance between the eyes across all images. A desired eye distance is specified, and a scaling factor is calculated based on the actual distance between the eyes. The image is then resized using this factor to achieve a consistent eye distance across all processed images.

2.4.4 Cropping

As a last step in this face, the image is cropped to eliminate irrelevant information like hair and the neck. Margins are added around the eyes to define a region of interest, and the cropping fine-tunes the image, ensuring that the following steps of the program focus on the most relevant facial features, such as the eyes and mouth.

The difference between the original image and the normalized image can be seen in figure 3.

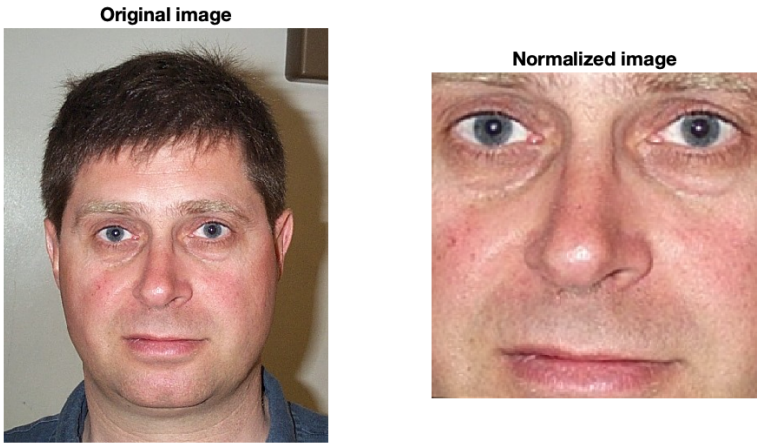

Figure 3 - Before and after normalization.

2.5 Features and matching

The subsequent phase in the facial recognition pipeline involves generating feature vectors, which are essential in extracting eigenfaces. Through the application of pre-processing methodologies, the training of the provided dataset is conducted to derive eigenvectors for the matching process.

2.5.1 Structure

To identify a given face from a dataset the program needs to be trained on the given dataset. The program therefore has two pipelines, one for training and one for testing the query image. The training phase iterates through the given dataset and saves the data in a *.mat* file. The test phase on the other hand takes a given image and tests it against the saved dataset, and decides which is the most similar. If no match is found, i.e the match is not close enough, it returns zero; nothing found.

2.5.2 PCA & Matching

Principal component analysis (PCA) is a dimensionality reduction method to extract the most significant features from a feature space. It can be suitable for a facial recognition system since it reduces the complexity of facial images while retaining essential information needed for accurate classification.

Before applying PCA, the images in the training phase are vectorized and combined into a matrix, where each column represents a vectorized facial image. The mean face is then calculated, representing the average facial structure across the dataset. PCA identifies the principal components in the image space where the data varies the most. When talking about face recognition, these components are referred to as eigenfaces. The eigenfaces are generated by applying PCA to the covariance matrix of the centered image vectors. By representing each image as a linear combination of these eigenfaces, the facial data is compressed which leads to a reduction of the complexity. In the training phase, PCA is conducted to create these eigenfaces which characterizes the variability within the dataset. The test phase involves comparing the facial features of a query image with the eigenfaces from the training phase. By projecting the query image onto the eigenface subspace, a set of weights are computed, which represents the contribution of each eigenface to the query image. Using these weights, the query image are then classified based on its resemblance to the training set. This is done by calculating the euclidean distance. If the difference in distance falls below a predefined threshold, the query image is classified as the corresponding image in the training set. If the distance is above the threshold, the query image is not considered to be a part of the set.

The saved eigenfaces as well as the mean face from the training phase can be seen in figure 4 and 5.

Figure 4 - Eigenfaces from the DB1 dataset.

Figure 5 - Mean face from DB1 dataset.

4. Results

Table 1 - Accuracy of the system - 144 cases from the DB1 dataset

Method	Accuracy with threshold	Accuracy without threshold
Implemented PCA	81%	90%
Built-in PCA	78%	90%

** Euclidean distance threshold = 19.8 based on rejecting the DB0 dataset.*

The accuracy was calculated for the following cases, totaling 144 cases from the dataset named DB1 containing 16 images:

- Translation of the face in the image (+/- 30 pixels),
- rotation (+/- 5°),
- scaling (+/- 10 %),
- change of tone values in the image (max - 30 %, +10%) and
- no transformations, original dataset.

5. Discussion

2.1. Color correction

The implementation of color correction to images that were altered with -30% and +10% change in tone values led to unexpected challenges in our facial recognition process. While this correction typically improved the images, it notably caused difficulties for the algorithms responsible for identifying facial features, especially in the case of images transformed with a 1.1 factor. The issues with these transformed images were due to subtle variations in color tones and shadows present in the images. These variations became more amplified or diminished following color correction. Consequently, this normalization process resulted in the loss of details necessary for accurately pinpointing and identifying facial features. Furthermore, the lack of contrast in lighter images post-color correction caused certain facial features to blend into the overall facial tone, making it difficult for algorithms, dependent on specific color thresholds or contrasts, to accurately detect and locate these features. The color correction process on images altered by a reduction factor of 0.7 were better compared to images altered by a factor of 1.1. This could be due to the fact that the darker images can result in enhanced contrast, reduced noise, and better-preserved details.

Building upon the previous discussion, Gray World's application in color correction posed greater challenges when recognizing facial features. On the other hand, White Patch gave better results. Therefore the decision was made to only utilize White Patch.

2.2. Segmentation

The entire program would benefit from a more thoroughly designed face mask and skin detection method, a lot of time and effort was put towards the eye coordinate function, to find the actual eyes. With a more robust solution from the start this step would not have to be as complex as it ended up being. This would probably also make the program more efficient. The next step in face mask and corresponding skin detection would involve employing a probabilistic model to achieve a more comprehensive approximation of skin color. Currently, the system relies on static thresholds, which are not an ideal solution for an entirely different dataset or other color models.

2.3. PCA & Eigenfaces

The final accuracy was heavily influenced by the criterion of not misclassifying images not contained in the original dataset. This is ultimately a trade-off between the false acceptance rate (FAR) and false recognition rate (FRR). If our program aimed only to recognize images from the dataset, the accuracy would be much higher since the program would not have to consider entirely different images; it could assume that the image exists, just a question of which one. However, this program tries to strike a balance between FAR and FRR. Ultimately it resulted in having a 100% FRR meant that the FAR becomes only 79% instead of closer to 90%. The problem lies with the distance calculation, where images not contained

in the dataset have a smaller distance than those that do but have undergone transformations. This meant that if the program were to reject false images the threshold had to be quite low, so low that it also removed some of the correct classifications. But, on the other hand, this meant that the program classified images correctly but based on a very weak criterion. Having a lower threshold, even though it lowered the overall accuracy, meant the program became more robust.

6. Conclusions

The implementation of color correction on images with tone value changes presented challenges in facial recognition, especially for images with a 1.1 factor. Darker images with a 0.7 reduction were notably better post color correction. White Patch was, in the end, favored over Gray World. Segmentation could be improved with a more robust face mask and skin detection method. In the matching phase, a trade-off between false acceptance and recognition rates was made for program robustness, leading to a lower overall accuracy but better FAR. In summary, it can be concluded that a robust facial recognition program can be developed using relatively simple methods individually, and by combining them, achieving satisfactory results. However, in this case, it comes with the tradeoff that the system will heavily depend on the predefined dataset. If one were to change the dataset, adjustments would be necessary to achieve the same accuracy levels presented in this report. It is also heavily dependent on the choice to prioritize FRR or FAR, which will ultimately be decided depending on the use case and the importance of each factor.

Sources

[1] D. Nyström. TNM034 *Advanced Image Processing - Lecture 2: Face Detection in color images.* [PowerPoint slides - TNM034_2023_Le2.pdf]. 2023.

[2] B. Wang, X. Chang and C. Liu. *Skin Detection and Segmentation of Human Face in Color Images*. International Journal of Intelligent Engineering and Systems. 2011. https://liuonline.sharepoint.com/sites/Lisam_TNM034_2023HT_F6/CourseDocuments/References/Skin%20Detection%20and%20Segmentation%20of%20Human%20Face%20in%20Color%20Images.pdf?CT=1701251433264&OR=ItemsView

[3] J.M. Chaves-González, M.A. Vega-Rodríguez, J.A. Gómez-Pulido, J.M Sánchez-Pérez. *Detecting skin in face recognition systems: A colour spaces study*. Univ. Extremadura, Dept. Technologies of Computers and Communications. 2009. https://liuonline.sharepoint.com/sites/Lisam_TNM034_2023HT_F6/CourseDocuments/References/Detecting%20skin%20in%20face%20recognition%20systems%20-%20A%20colour%20spaces%20study.pdf?CT=1702042673614&OR=ItemsView

[4] R.L. Hsu, M. Abdel-Mottaleb, A.K. Jain. *Face detection in color images*. Pattern Analysis and Machine Intelligence. IEEE Transactions, 2002. https://liuonline.sharepoint.com/sites/Lisam_TNM034_2023HT_F6/CourseDocuments/References/Face%20Detection%20in%20Color%20Images.pdf?CT=1701687537574&OR=ItemsView