Biometric Systems

Identification Using Principal Lines on a Palm

Author: Samuel Olekšák (xoleks00)
Academic year: 2023/24

Contents

1 Introduction

This project explores biometric identification using principal lines on human palms. The model extracts rotation-invariant region of the palm from the source image, filters the region to highlight ridges and troughs and extracts a set of 9 floats according to which the classification is performed. This project only focuses on three main lines on the palm and tries to ignore smaller creases and imperfections (Fig. 1).

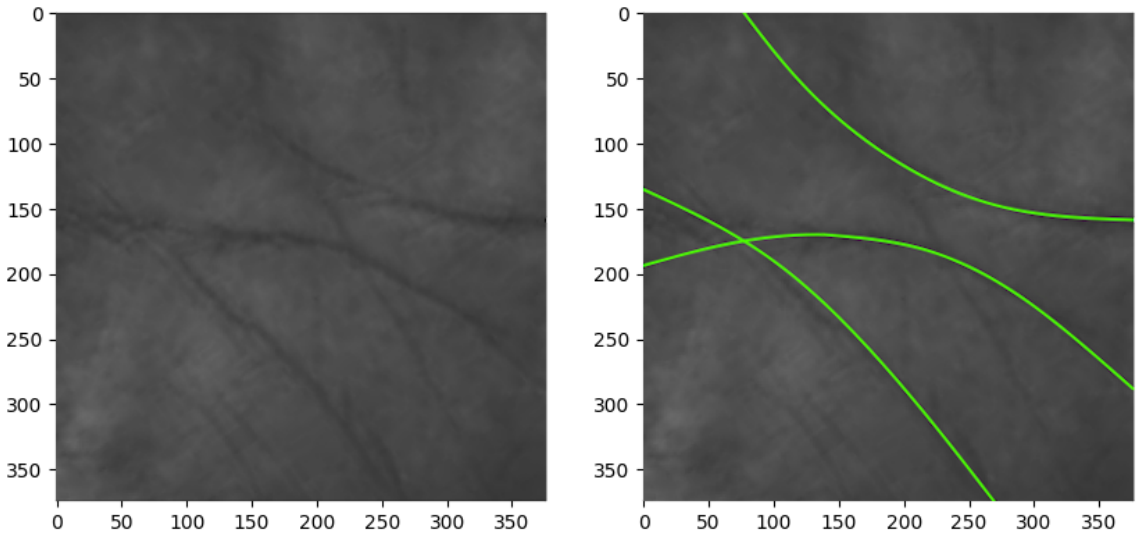

Figure 1: Left image shows region of the palm with visible main lines and several small ridges. Right image shows three major lines highlighted.

1.1 Implementation details

For the implementation of this project Python 3.10 along with the libraries `opencv-python`, `numpy` and `matplotlib` was used. To see the description of the attached files see Section 5.

2 Database

This project is based on the *CASIA Multi-Spectral Palmprint Database (CASIA-MS-PalmprintV1)* dataset and is partly inspired by Multispectral palm image fusion for accurate contact-free palmprint recognition [1]. The dataset provides 7200 palm images of 100 different people captured in two sessions separated by more than a

month. Each sample is captured with six different illumination wavelengths – 460 nm, 630 nm, 700 nm, 850 nm, 940 nm and white light (Fig. 2). The images are 8 bit gray-level JPEG files.

For the purposes of this project only images with wavelength 460 nm are used and only left hands are used. The sidedness of a hand could be easily distinguished by comparing x-coordinates of extracted interdigital points. If the algorithm were to work for right hands the only change would be in the sorting of interdigital points by y-value – their sorting would be reversed.

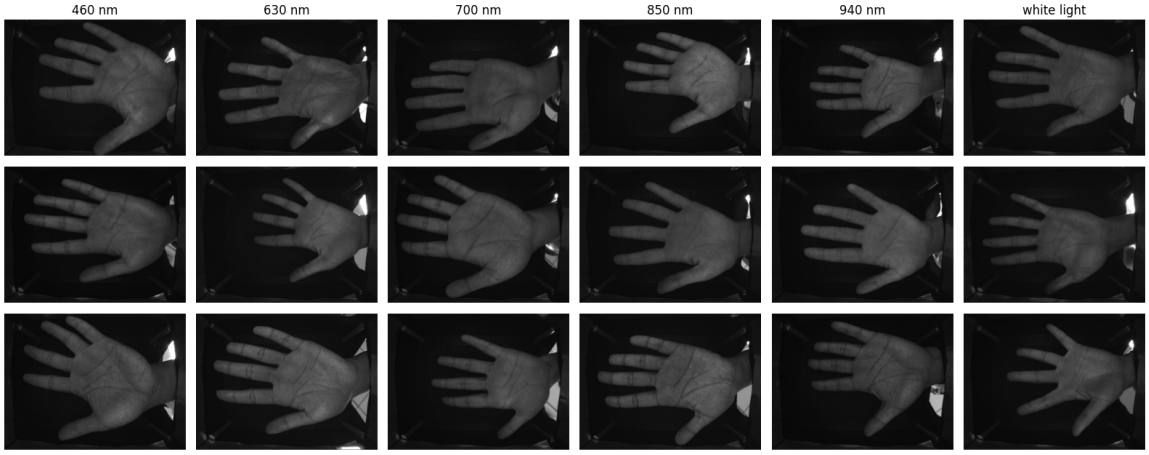

Figure 2: Dataset sample shows palm images of three individuals (one for each row) in six different wavelengths.

3 Feature Extraction

Feature extraction consists of three stages. Firstly, region of interest (ROI) is extracted. ROI is a square area of palm found by locating the point between index and middle finger and point between ring and little finger. If ROI is extracted correctly it should be invariant to rotation around axis perpendicular to the image plane. If ROI width is calculated according to the width of the palm it is also scale-invariant. ROI extraction stage is followed by contour extraction and polynomial fitting stages described later.

3.1 Region of Interest Extraction

Original image (shown in Fig. 3) contains distracting features around the wrist where light seeps from above the black blanket above the hand. Due to this, cropping

is performed where right 1/6 of the image is removed.

Figure 3: Dataset image captured under 460 nm light.

Binary thresholding is used to differentiate between hand and background. Experimentally gathered lightness value working best for the particular dataset is 40.

Contours detected in the binary image are sorted by their arc length and the longest contour is assumed to wrap the whole hand. During the contour detection part of the preprocessing two problems occurred:

1. In some images a thumb or a little finger can reach outside of the image (for example see tip of the thumb in Fig. 3). This causes hand contours to be separated. The fix entails adding a black border few pixels thick around the whole image so that the contour detector does not encounter image edge when running on the perimeter of the hand.

2. Sometimes the thresholded image color boundaries are jagged – this is remedied by applying morphological dilatation and Gaussian blur to the thresholding image prior to contour detection.

The input and output of the contour detection algorithm are shown in Fig. 4 and Fig. 5 respectively.

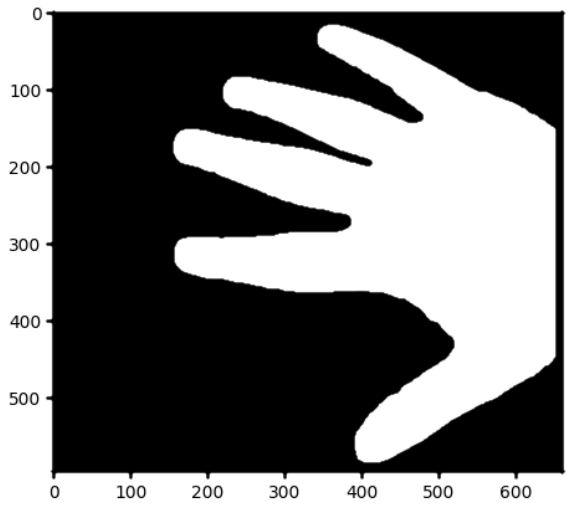

Figure 4: Image after slicing off the wrist portion, applying binary thresholding and with black border added.

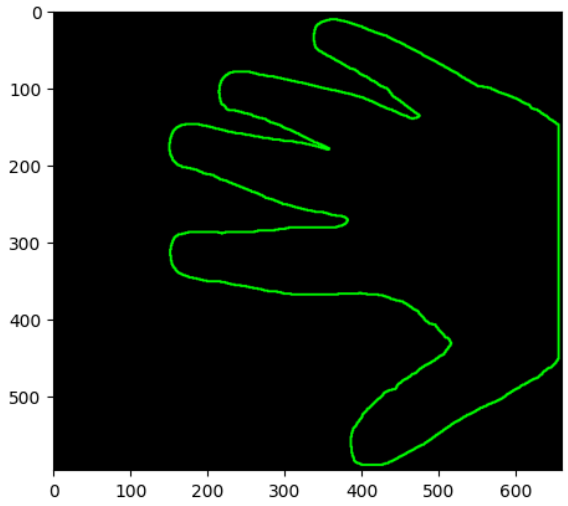

Figure 5: Highlighted Longest contour captures the whole hand.

Points between fingers are located using the convex hull construction and convexity defect detection. Four largest detected convexity defects correspond to four interdigital points (Fig. 6).

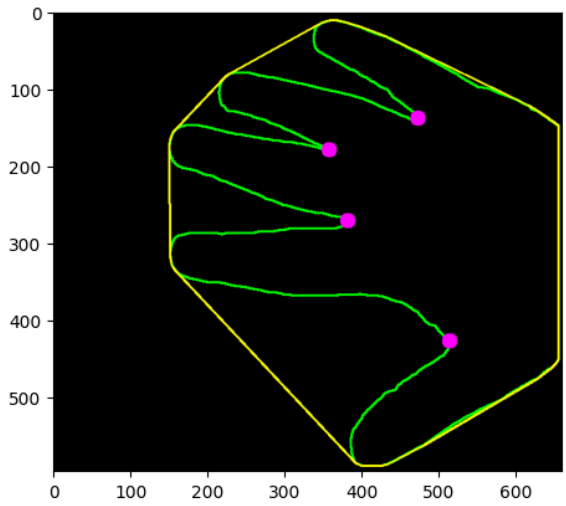

Figure 6: Longest contour wrapped in convex hull with four largest convexity defects highlighted.

If defect points are sorted by y-coordinate, the point between index and middle finger (point A) and point between ring a little finger (point B) can be identified. These points are essential to the location of the ROI. The line running through them forms the reference line. ROI is extracted as a square region starting 30 pixels below the reference line and with width of $1.2 * w$ where w is the distance between point A and point B. The process of identifying the ROI in the image is shown in Figures 7 – 12.

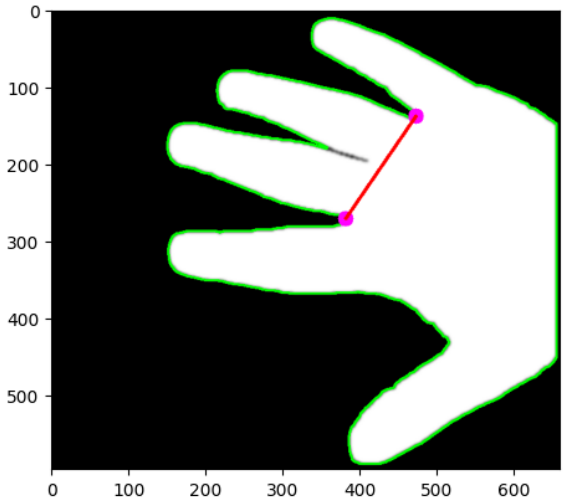

Figure 7: Points between index and middle finger and between ring and little finger define the placement of the reference line.

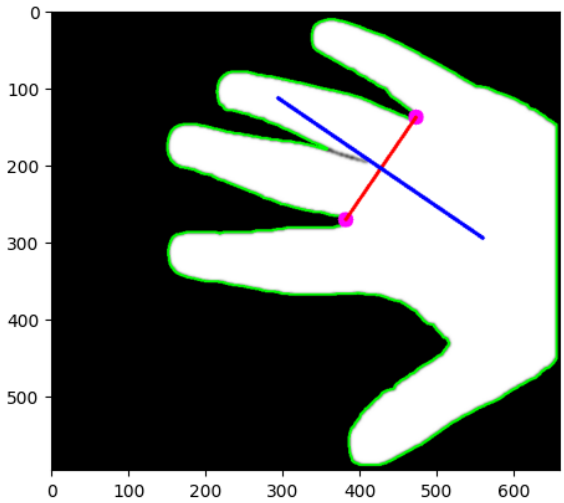

Figure 8: Blue line is perpendicular to the reference line and runs through the center of it. The blue line will be running throught the centroid of the ROI.

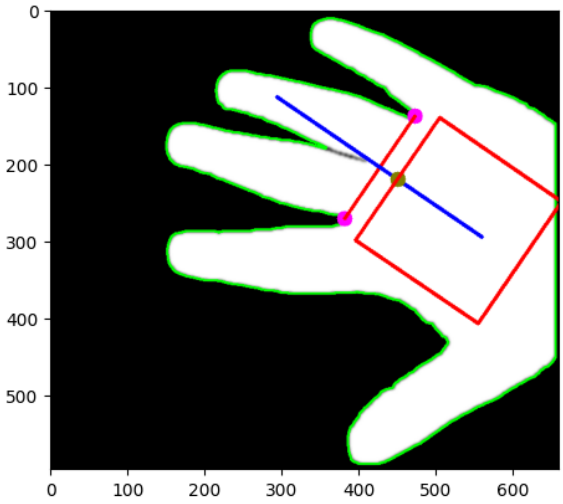

Figure 9: The red square starting 30 pixels below the reference line represents the ROI. The ROI clipping outside of the images does not pose any problems since ROI will be extracted from the original image without the wrist cropped out.

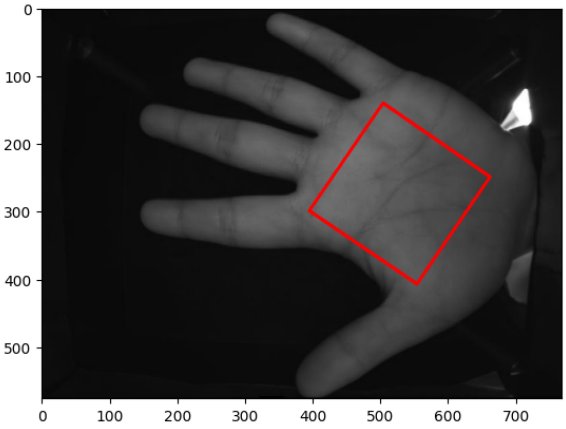

Figure 10: ROI projected over the original image.

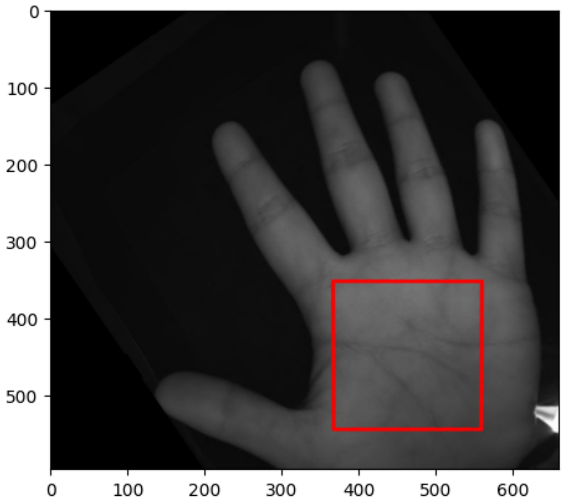

Figure 11: Original image rotated to align ROI to the axes.

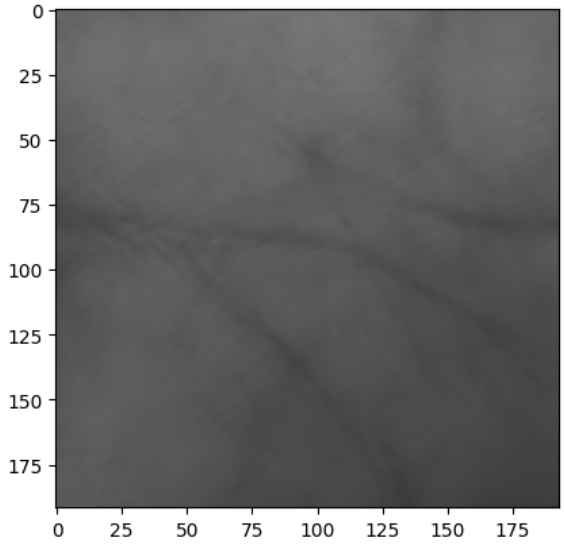

Figure 12: Extracted ROI.

3.2 Principal Lines Extraction

This stage converts ROI cropped out of original image (Fig. 12) to 3 contours using various filters. Firstly, a series of 24 Gabor filters with various angle θ ranging

from $\pi/2$ to 2π is applied and the resulting images are averaged.

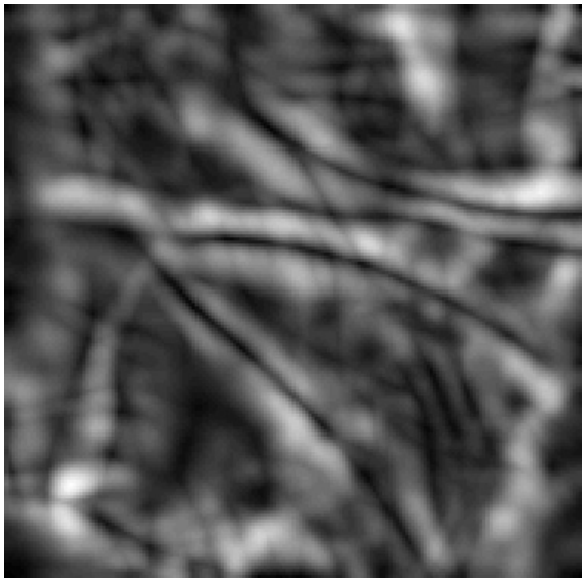

Figure 13: ROI with Gabor filters applied.

Next, the image is scaled to twice the resolution to better work with contour detection. Contrast limited adaptive histogram equalization (CLAHE) [2] and adaptive thresholding (`cv2.adaptiveThreshold`) are used to help bring out principal lines detail in the image. In the resulting image contours are detected and sorted by their arc length. The three longest contours are selected to represent the palm (Fig. 14).

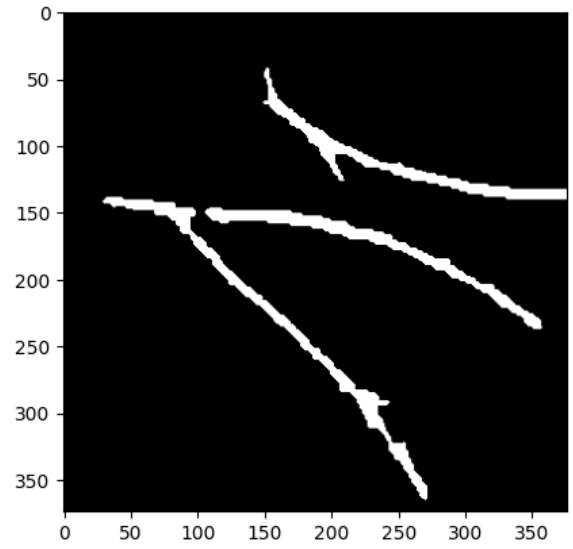

Figure 14: Three longest contours found in the Gabor filtered image with applied CLAHE and adaptive thresholding.

3.3 Analytical Description of Lines

Quadratic polynomial was selected as an analytical description of a single principal line. Each of the three longest contours is converted to quadratic polynomial coefficients therefore each palm image can be represented by 9 floats. Each contour point is inserted into `np.polyfit` function which tries to fit quadratic function onto it and return three coefficients. Before saving, the contours are sorted by y-coordinate of their centroid such that the topmost contour is stored first.

```
[
    [  -0.00287985,     1.76726984,    -137.73551719  ],
    [   0.00159217,    -0.40672544,     177.95731736  ],
    [   0.00183072,     0.38661045,     116.54046866  ],
]
```

Y-coordinate of contours is extracted by calculating moments with function `cv2.moments` and dividing 1st-order moment about the y-axis (`m01`) by 0th-order moment (`m00`).

As shown in Figure 15, the lines are not constrained but run border to border, which is helpful since the ends of principal lines are often fuzzy and hard to consistently identify.

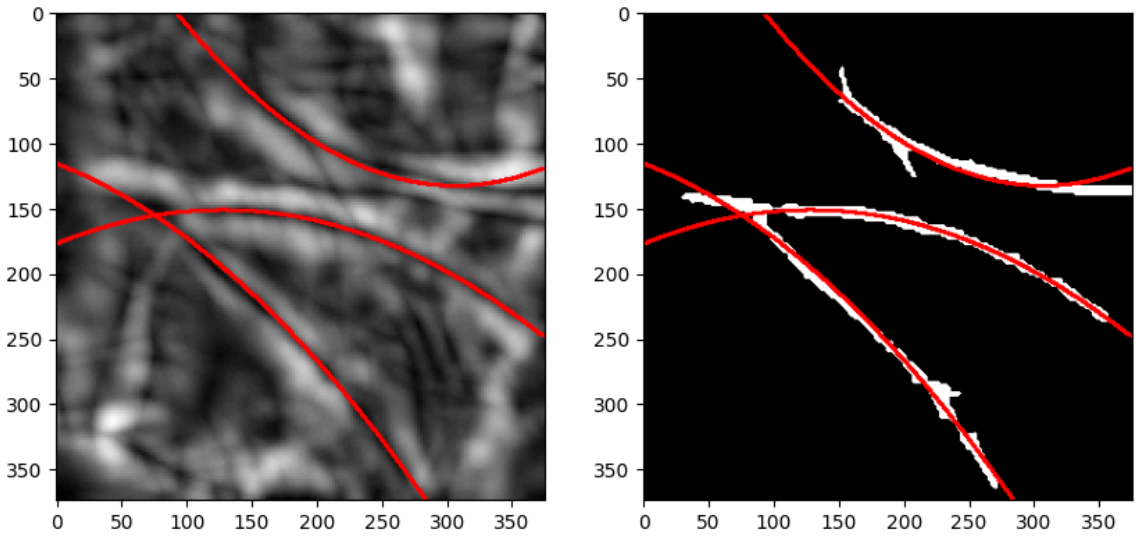

Figure 15: Fitted quadratic functions drawn over output of Gabor filtering (left) and over 3 longest contours (right).

4 Classification

Classification is performed using a 1-NN classifier with Euclidean (L2) distance. The distance function between two samples a and b is defined as

$$\frac{1}{C}\sum_{i=1}^{C}\sqrt{\sum_{j=1}^{D}\left(\frac{a_{i,j}-b_{i,j}}{n_j}\right)^2}$$

where C is the count of contours representing the image (in our case $C = 3$), D is the degree of the polynomial representing one contour (in our case $D = 3$), $n = (n_1, n_2, ..., n_D)$ is the median polynomial coefficient vector for all contours over the whole training dataset. Vector n works as a normalizer so that constant coefficient does not overshadow linear and quadratic coefficients that are typically multiple magnitudes smaller.

5 Attached files

The `python3 evaluate.py` command will load dataset of all left hand images in 460 nm light, extract their features and compare 300 testing samples to 300 training

samples.

Folder `figures/` contains all figures from this PDF file.

The archive does not contain dataset files – the images need to be placed directly into the `images/` folder, as shown by filenames wrapped in brackets in the following tree.

```
xoleks00/
├── evaluate.py
├── requirements.txt
├── doc.pdf
├── images/
│   ├── [001_l_460_01.jpg]
│   ├── [001_l_460_02.jpg]
│   └── [...]
└── figures/
    └── ...
```

6 Conclusion

The project explored classification of human palms using the three main lines. Comparison of two palms is performed by extracting features – quadratic polynomial coefficients for each of the three main lines on a palm.

Unfortunately, due to the poor quality of contour detection only 18.33 % accuracy was able to be achieved. By looking at the images, the ROI extraction stage works 99 % of the time and the polynomial comparison accuracy for images with correctly identified contours was very good. I believe the only stage which did not perform well is the contour detection.

Average time to extract features from original image was around 15 ms. Average time to compare 1 testing samples to the whole training database was 8 ms.

6.1 Possible improvements

Ideally a 3-NN classifier would be used since each individual has three training samples. Since edge and contour detection was not reliable enough, this was dropped in favour of 1-NN classifier.

Some additional detail might be gained by using a higher degree polynomial for principal line representation.

7 Acknowledgment

Portions of the research in this paper use the CASIA-MS-PalmprintV1[1] collected by the Chinese Academy of Sciences' Institute of Automation (CASIA).

Bibliography

[1] Y. Hao, Z. Sun, T. Tan **and** C. Ren, "Multispectral palm image fusion for accurate contact-free palmprint recognition," **in***2008 15th IEEE International Conference on Image Processing* 2008, **pages** 281–284. DOI: 10.1109/ICIP.2008.4711746.

[2] A. Reza, "Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement," *VLSI Signal Processing*, **jourvol** 38, **pages** 35–44, **august** 2004. DOI: 10.1023/B:VLSI.0000028532.53893.82.

[1] http://biometrics.idealtest.org/dbDetailForUser.do?id=6#/datasetDetail/6