Liveness Detection on Touchless Fingerprint Scanner - Research Article

Katerina Fortova

March 2020

1 Introduction

2 Extracted vectors from image

Three different vectors were used for research. I worked with vector based on Local Binary Pattern, Sobel and Laplacian operator and Wavelet transformation. The important feature for every vector was Gray Level Cooccurrence Matrix (GLCM), which works as classifier of texture in image. The features which I used and extracted from GLCM were contrast, correlation, energy and homogeneity.

2.1 Extracted vector based on LBP

This vector is focused on image processed by LBP and extraction of characteristics from GLCM based on prepossessed gray level image. The histogram is extracted from LBP image and feature based on histogram is extracted with this technique:

Suppose we have histogram of LBP image which looks for example like this:

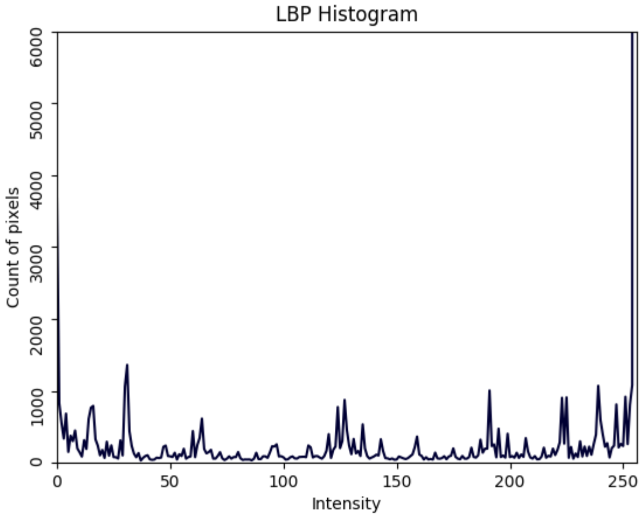


Figure 1: Example of histogram of LBP image

Histogram contains 256 bins and then their pixel count values. Parameter $value_b$ means pixel value of b bin, where b is from set $\{0, 1, 2, ..., 255\}$.

$$histogram = \{value_b, value_{b+1}, ...value_{255}\}$$

I extracted four partial sums of this histogram:

$$partitialLBPsum_1 = \sum_{b=0}^{b=63} value_b$$

$$partitialLBPsum_2 = \sum_{b=64}^{b=127} value_b$$

$$partitialLBPsum_3 = \sum_{b=128}^{b=191} value_b$$

$$partitialLBPsum_4 = \sum_{b=192}^{b=255} value_b$$

The extracted vector v contains these four partial sums of LBP histogram and then characteristics of input gray scale preprocessed image - contrast f_1, homogeneity f_2, energy f_3 and correlation f_4:

$$v = [partitialLBPsum_1, partitialLBPsum_2, partitialLBPsum_3, partitialLBPsum_4, f_1, f_2, f_3, f_4]$$

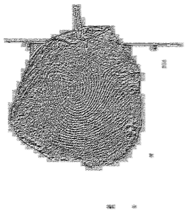


Figure 2: Fake fingerprint processed by LBP

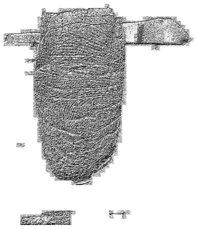


Figure 3: Live fingerprint processed by LBP

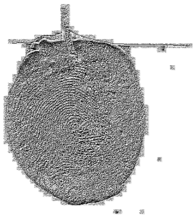


Figure 4: Fake fingerprint processed by LBP

2.2 Extracted vector based on Sobel and Laplacian operator

Sobel and Laplacian operators work well as edge and texture detectors. Kernel $5x5$ was used for this feature extraction. We can extract Sobel only based on x or y axis. So we have two different results of horizontal and vertical Sobel operator. Then Laplacian image was extracted and characteristics were also added to this vector.

Vector contains features f_1 (contrast), f_2 (homogeneity), f_3 (energy), f_4 (correlation) of GLCM matrix for Sobel x image $Sobelx$, Sobel y $Sobely$ image and Laplacian $Laplacian$ image.

This is the final extracted vector:

$$v = [f_{1Sobelx}, f_{2Sobelx}, f_{3Sobelx}, f_{4Sobelx}, f_{1Sobely}, f_{2Sobely}, f_{3Sobely}, f_{4Sobely}, f_{1Laplacian}, f_{2Laplacian}, f_{3Laplacian}, f_{4Laplacian}]$$

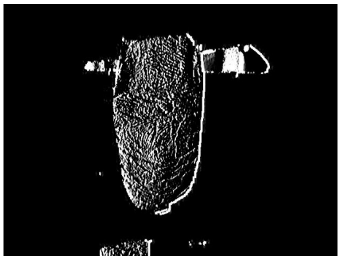


Figure 5: Sobel of x-axis of live fingerprint

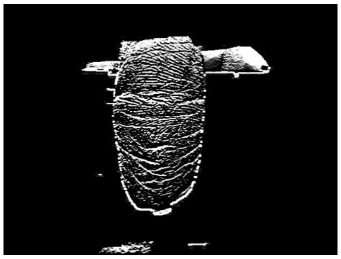


Figure 6: Sobel of y-axis of live fingerprint

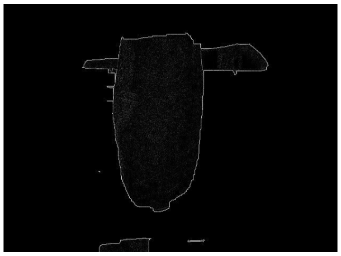


Figure 7: Laplacian of live fingerprint

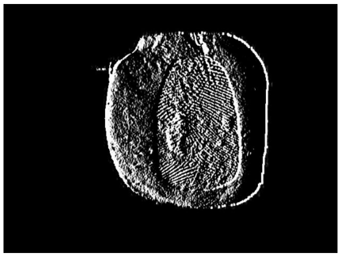


Figure 8: Sobel of x-axis of fake fingerprint

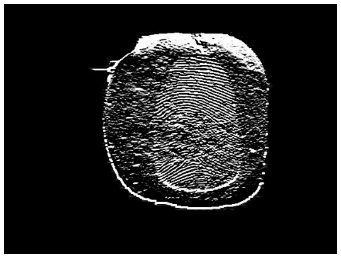


Figure 9: Sobel of y-axis of fake fingerprint

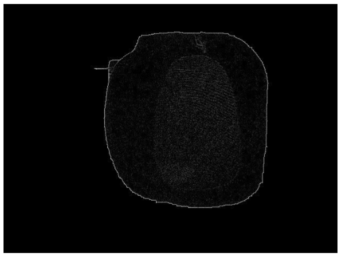


Figure 10: Laplacian of fake fingerprint

2.3 Extracted vector based on Wavelet Transformation

Wavelet processing with different wavelet families such as biorthogonal or Daubechies wavelets is also used in image processing. We have three details from image processed by wavelet transformation - horizontal LH, vertical HL and diagonal HH. We can extract features from GLCM matrix for each detail. Features f_1, f_2, f_3, f_4 stands for same features from GLCM matrix as in previous vectors.

$$v = [f_{1LH}, f_{2LH}, f_{3LH}, f_{4LH}, f_{1HL}, f_{2HL}, f_{3HL}, f_{4HL}, f_{1HH}, f_{2HH}, f_{3HH}, f_{4HH}]$$

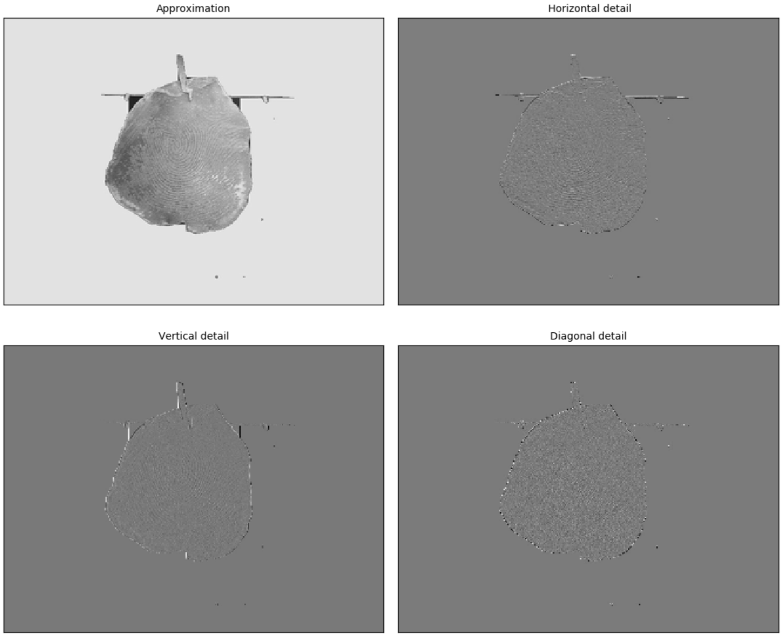


Figure 11: Wavelet transformation of fake image based on bior1.5 method

3 Analyzing the images with same light

During taking the dataset from touchless sensor, we use lights of different wavelength for taking the images. The lights were blue, green and red. The goal was to realize which light is most accurate for liveness detection. The all methods were also compared. Three methods for segmentation were used. First type of segmentation uses Otsu tresholding, which works with constant tresh based on particular image. The other two types of segmentation use Gaussian and Mean adaptive tresholding, which is suitable for images with various light properties. The tresh isn't constant for whole image as with Otsu algorithm, but it changes for every pixel based on their neighbors.

Below are some examples of images, how they were taken with our touchless fingerprint scanner.

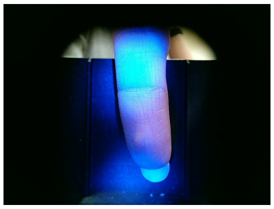


Figure 12: Image of live fingerprint with blue light

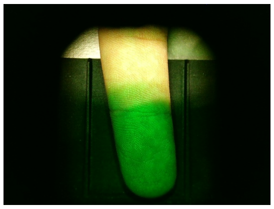


Figure 13: Image of live fingerprint with green light

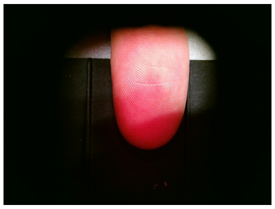


Figure 14: Image of live fingerprint with red light

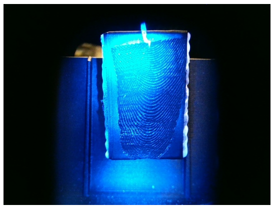


Figure 15: Image of fake fingerprint with blue light

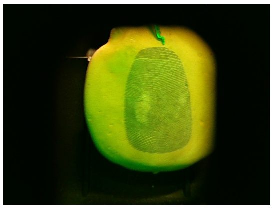


Figure 16: Image of fake fingerprint with green light

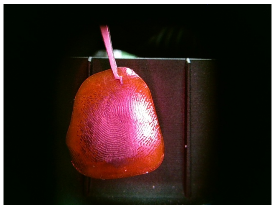
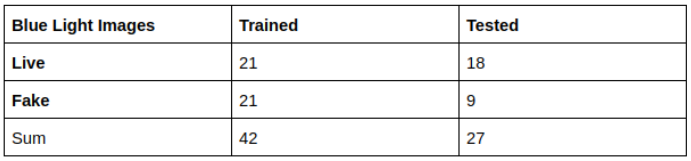


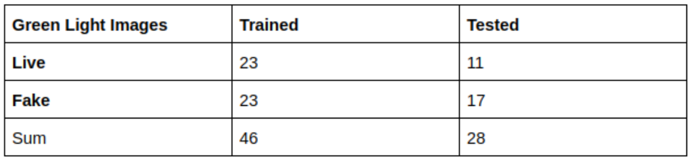
Figure 17: Image of fake fingerprint with red light

Each tested and trained dataset contains images with same light. These are datasets for blue, green and red images:



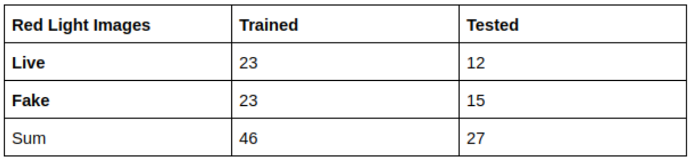
Blue Light Images	Trained	Tested
Live	21	18
Fake	21	9
Sum	42	27

Figure 18: Dataset for images with blue light



Green Light Images	Trained	Tested
Live	23	11
Fake	23	17
Sum	46	28

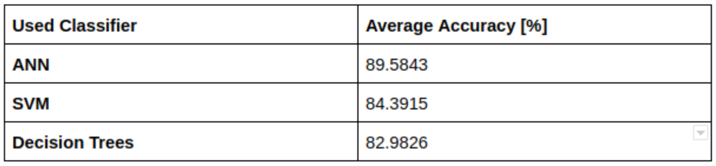
Figure 19: Dataset for images with green light



Red Light Images	Trained	Tested
Live	23	12
Fake	23	15
Sum	46	27

Figure 20: Dataset for images with red light

Each dataset was trained and tested with three different types of segmentation with different types of tresholding. Artificial Neural Network (ANN), Support Vector Machines (SVM) and Decision Trees were used for final classification, training and decision about results for tested images.



Used Classifier	Average Accuracy [%]
ANN	89.5843
SVM	84.3915
Decision Trees	82.9826

Figure 21: Table of average accuracy of various classifier

These are examples of various segmentation type based on three methods of thresholding:

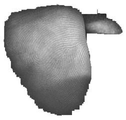


Figure 22: Segmentation with Otsu tresholding

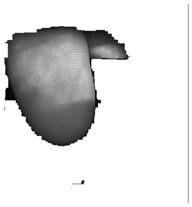


Figure 23: Segmentation with adaptive Gaussian tresholding

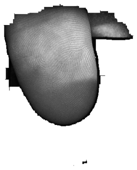
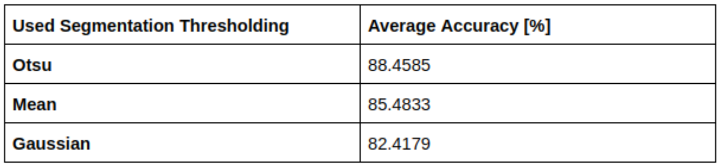


Figure 24: Segmentation with adaptive Mean tresholding

This is the rating of different segmentation based on their accuracy:

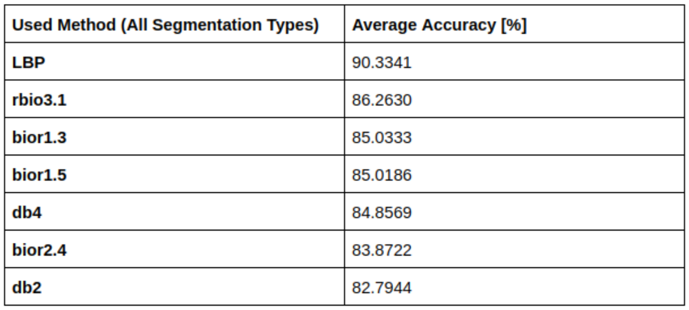


Used Segmentation Thresholding	Average Accuracy [%]
Otsu	88.4585
Mean	85.4833
Gaussian	82.4179

Figure 25: Table of average accuracy of various segmentation type

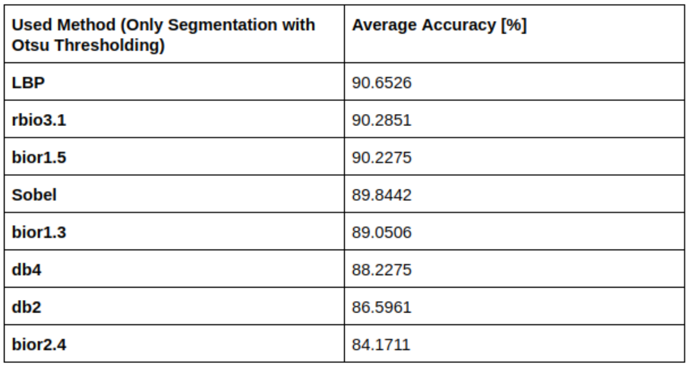
Different wavelet families were used for Wavelet transformation. I used two wavelet transformations based on Daubechies wavelet family ($db2$, $db4$), three wavelets based on biorthogonal wavelet family ($bior1.3$, $bior2.4$, $bior1.5$) and one wavelet type based on reverse biorthogonal wavelet family ($rbio3.1$). We can compare not only different methods but also different types of wavelet transformation.

This is average accuracy of various used methods. For every method except Sobel method were used all three different types of segmentation, Sobel uses only Otsu tresholding because of better accuracy and ANN error function based on epochs. So we can compare all methods except Sobel with all segmentation types and then all methods including Sobel with only Otsu segmentation.



Used Method (All Segmentation Types)	Average Accuracy [%]
LBP	90.3341
rbio3.1	86.2630
bior1.3	85.0333
bior1.5	85.0186
db4	84.8569
bior2.4	83.8722
db2	82.7944

Figure 26: Average accuracy for methods with use of all segmentation techniques

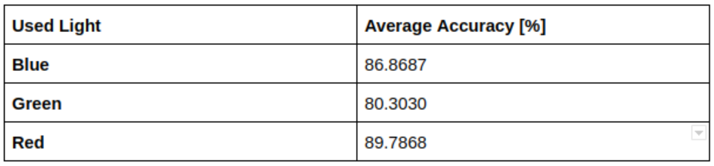


Used Method (Only Segmentation with Otsu Thresholding)	Average Accuracy [%]
LBP	90.6526
rbio3.1	90.2851
bior1.5	90.2275
Sobel	89.8442
bior1.3	89.0506
db4	88.2275
db2	86.5961
bior2.4	84.1711

Figure 27: Average accuracy for all methods with use of segmentation with Otsu thresholding

We can see that for separate datasets with different lights is most suitable to use a vector based on LBP. Method based on Sobel and Laplacian is also quite accurate and faster than processing LBP algorithm. The advantage of wavelet transformation is wide list of various wavelet families (biorthogonal, reverse biorthogonal, Daubechies, Symlets, Coiflets) but the processing the image with wavelet and get horizontal, vertical and diagonal detail is slower especially with large datasets.

Finally we can find out which color gives the best accuracy and is the most reliable for liveness detection on touchless device.



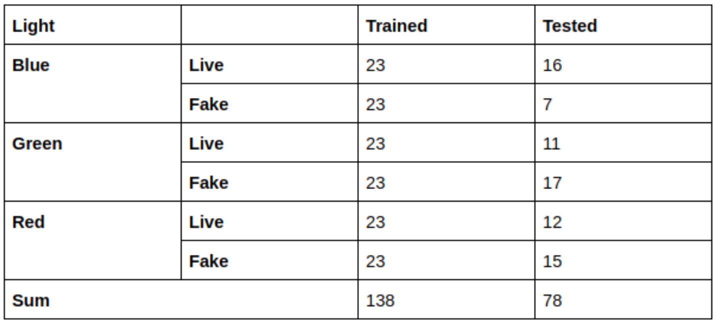
Used Light	Average Accuracy [%]
Blue	86.8687
Green	80.3030
Red	89.7868

Figure 28: Average accuracy for all tested datasets with blue, green and red lights

We can see that the best accuracy have the images with red light. Therefore they are most suitable for liveness detection on our device. It is commonly known that green light reflects the biggest light amount back, so it is most suitable for measuring reflected light with some specialized device, but when it comes to software texture analysis, the red light can shine through finger with least reflected light back, so the texture of fingerprint is most accurate with minimal reflections.

4 Analyzing images with mix of images with all lights

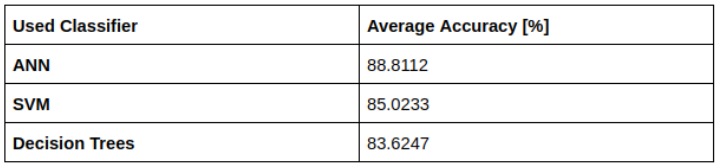
The all datasets from previous chapter are combined into one dataset, so we have dataset with mix of images with all three different lights. This is the largeness of dataset, amount of images for testing and training and their properties.



Light		Trained	Tested
Blue	Live	23	16
	Fake	23	7
Green	Live	23	11
	Fake	23	17
Red	Live	23	12
	Fake	23	15
Sum		138	78

Figure 29: Properties of analyzed dataset with mix of all three lights

Similar to analyzing data set with various lights, we want to figure out what type of segmentation, classifier and method is most suitable for liveness detection, when we have large dataset with various lights.

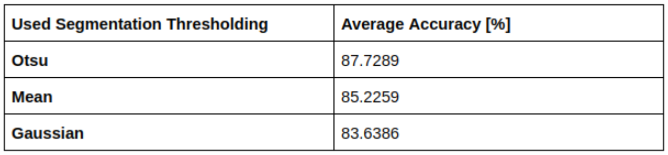


Used Classifier	Average Accuracy [%]
ANN	88.8112
SVM	85.0233
Decision Trees	83.6247

Figure 30: Average accuracy of classifiers with dataset containing mix of images with various lights

We can see that similar to previous datasets with separate lights, here is also same order of average accuracy of classifiers. Artificial Neural Network is really accurate, but it is slower in evaluation, especially when it is trained for a larger amount of epochs. Support Vector Machines and Decision Trees are quicker in evaluation and they also gave quite accurate and good results.

Now it is time to decide which type of segmentation works best for this larger dataset.

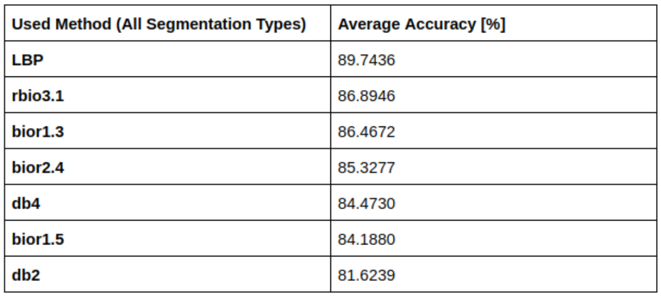


Used Segmentation Thresholding	Average Accuracy [%]
Otsu	87.7289
Mean	85.2259
Gaussian	83.6386

Figure 31: Average accuracy of various types of segmentation techniques

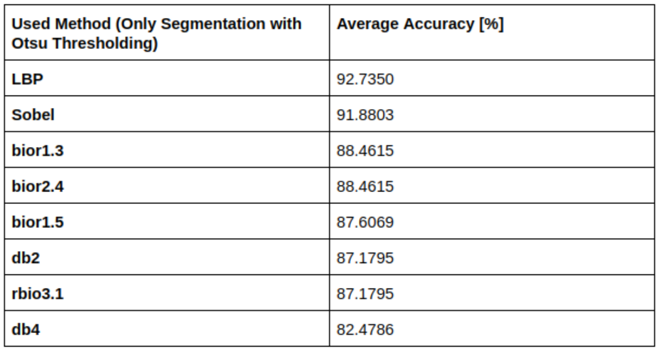
Segmentation with used Otsu tresholding also works best for this dataset. But it is important to mark that all segmentation methods have accuracy around 85%, so they are all really usable and get good results in our research.

This is the ranking of all methods and methods with only segmentation based on Otsu tresholding:



Used Method (All Segmentation Types)	Average Accuracy [%]
LBP	89.7436
rbio3.1	86.8946
bior1.3	86.4672
bior2.4	85.3277
db4	84.4730
bior1.5	84.1880
db2	81.6239

Figure 32: Average accuracy of methods



Used Method (Only Segmentation with Otsu Thresholding)	Average Accuracy [%]
LBP	92.7350
Sobel	91.8803
bior1.3	88.4615
bior2.4	88.4615
bior1.5	87.6069
db2	87.1795
rbio3.1	87.1795
db4	82.4786

Figure 33: Average accuracy of methods with segmentation based on Otsu tresholding

LBP is most accurate method for this analyzed dataset and also with previous datasets with same light. Sobel has accuracy around 90% and this method is also the fastest for analyzing. Best methods of Wavelet transformation are methods of Reverse Biorthogonal Wavelet Family and Birthogonal Wavelet Family, Daubechies are overall least accurate.