Fingerprint Damage Localizer and Detector of Skin Diseases from Fingerprint Images

Stepanka Barotova[1], Martin Drahansky[1]

Abstract: This article describes a novel approach for detection and classification of skin diseases in fingerprints using three methods - Block Orientation Field, Histogram Analysis and Flood Fill. The combination of these methods brings a surprising results and using a rule descriptor for selected skin diseases, we are able to classify the disease into a group or concrete name.

Keywords: Fingerprint recognition, skin diseases, image processing, image quality.

1 Introduction

Fingerprint-based systems are the most widely used biometric technology, which is very well accepted by users. Some people might use it literally on a daily basis, others only for civil identification or access systems. However, there is a significant number of people whose fingertip skin is affected of some kind of skin disease. Therefore, they cannot use fingerprint systems since skin diseases cause damages in ridge patterns.

The challenge now is to recognize the presence of skin diseases in fingerprint images and, if possible, eliminate their influence on the fingerprint recognition process, so that people suffering from skin diseases would be able to use fingerprint devices, at least to some extent.

Algorithms developed in our research are now able to locate the damage in the fingerprint image. Moreover, our classifier is able to estimate the possible disease present in the fingerprint image. This can have a great usage in forensic or medical applications, as well as security.

In this text, the methods used for localizing the damage and determining its type are going to be introduced, as well as the classification procedure and results.

2 The Triple-Method Damage Localization

There are three major methods that are used for the damage localization: *Block Orientation Field*, *Histogram Analysis* and *Flood Fill*. What makes the resulting concept so interesting, however, is their combination that provide valuable information about the quality and character of the possible disease.

[1] Faculty of Information Technology at Brno University of Technology, Department of Intelligent Systems, Bozetechova 2, 612 00 Brno, Czech Republic, xbarot00@stud.fit.vutbr.cz, drahan@fit.vutbr.cz

The classification is then based on the features extracted from the image by the Flood Fill algorithm, and their properties.

2.1 Ridge Inconsistence Detection from a Block Orientation Field

The computation of block orientation field is commonly used in the fingerprint recognition process for the purposes of estimating the ridges direction and classifying the fingerprint image into one of the several fingerprint classes [Ma09] [JFR08]. Because a typical fingerprint pattern consists of alternating dark and white lines, this information can be easily processed by a gradient operator that estimates the image gradient for each pixel. This low-level information is gathered and averaged for each $w \times w$ block in the image [HWJ98]. The transformation can result in a relatively smooth and continual image of the ridges direction estimates - for a healthy fingerprint of course - see Figure 1 on the left.

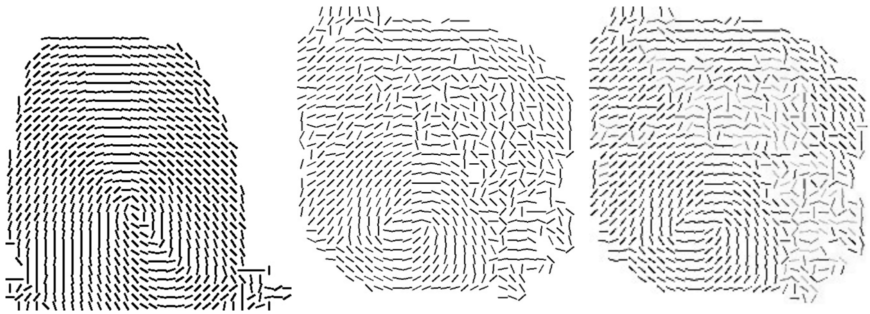

Fig. 1: Examples of block orientation images (left: healthy fingerprint, middle: fingerprint affected by a skin disease, right: detected damaged areas).

If we try to compute the block orientation field for a damaged or a partially damaged fingerprint however, the orientation field in damaged areas will be discontinuous, as displayed in Figure 1 in the middle. Exceptions to this are the peripheral areas and deltas and cores. These discontinuities can be detected by scanning the field for differences in direction angles.

The steps of the gradient-based method of block orientation field computation are as follows [HWJ98]:

1. Compute the gradients ∂_x and ∂_y for each pixel at (i, j) using a gradient operator. In this case a simple Sobel operator was used.
2. Divide the original image into $w \times w$ blocks.
3. Compute the estimation $\theta(i, j)$ of the ridge orientation for every image block centered at (i, j) using the Equations 1, 2 and 3:

$$v_x = \sum_{u=i-\frac{w}{2}}^{u=i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{v=j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v) \tag{1}$$

$$v_y = \sum_{u=i-\frac{w}{2}}^{u=i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{v=j+\frac{w}{2}} \partial_x^2(u,v)\partial_y^2(u,v) \tag{2}$$

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{v_y(i,j)}{v_x(i,j)}\right) \tag{3}$$

The resulting block orientation field is afterwards analyzed for any discontinuities that may occur. The analysis is done using a row-wise and column-wise scanning approach that reveals areas of possible damage in the fingerprint. Neighboring blocks' directions are compared and a block is marked as a discontinuity if $|\theta(i,j) - \theta(i,j+1)| > 45°$, where both estimations $\theta(i,j)$ and $\theta(i,j+1)$ have a value between $0°$ and $180°$. Example detection is shown in Figure 1.

The advantage of this method is that it is already a part of the standard fingerprint recognition pipeline, so the algorithm can be easily implemented into existing methods. Also, it provides a fairly accurate estimate of the fingerprint damage in the sample.

2.2 Fingerprint Damage Detection using Histogram Analysis

This experimental method is based on the presumption that a quality fingerprint image consists of equally distributed ridges and valleys. If we assume that ridges are roughly the same dark color while valleys are light-colored, a histogram computed from each subfield of the fingerprint's area should ideally have a bimodal shape: it should have two peaks of approximately the same height and one valley between them, as displayed in Figure 2 on the left.

On the other hand, the intensity distribution in a fingerprint image part that belongs to a damaged area is not always as equal as in the quality one. Experiments showed that the majority of histograms computed from damaged subfields break the rules of the bimodal histogram. The lower the quality, the less the histogram resembles the ideal one. A non-bimodal histogram always implies a damaged or low-quality area, whereas a damaged subfield does not necessarily imply a non-bimodal histogram because a histogram is a measure for the distribution of intensities only and it does not take into account the pattern or neighborhoods of pixels. Figure 2 shows examples of non-bimodal histograms.

Fig. 2: Left: ideal bimodal histogram, others are examples of histograms computed from damaged areas.

The steps of the algorithm are as follows:

1. Divide the image into $w \times w$ blocks (ROIs = regions of interest), according to the desired resolution.
2. For each ROI, compute a histogram.
3. Check if the histogram is consistent with the bimodal characteristics generally found when finger ridges are healthy.
   a) Find all peaks and valleys of the histogram.
   b) If $peaks == 2$ and $valleys == 1$, histogram is bimodal, so continue with 3c. Otherwise quit: the histogram is non-bimodal.
   c) Check the heights and distances of the peaks and valleys. If the histogram passes these validity tests, it is bimodal, otherwise it is non-bimodal.

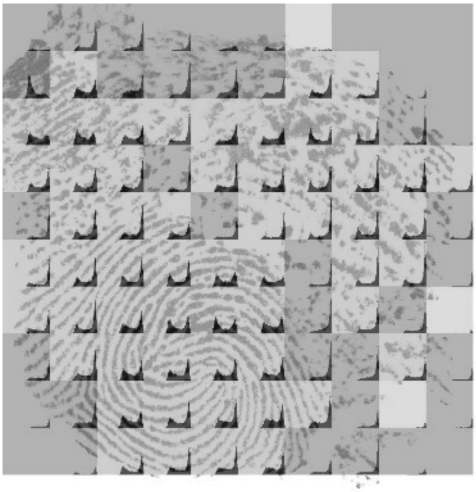

Fig. 3: Histogram Analysis result with details of particular histograms. Red background implies an invalid histogram, green means valid and blue stands for background.

Histogram Analysis is able to detect many areas the Block Orientation Field method might have omitted, therefore it is extremely valuable for the final determination of healthy areas.

Since the Histogram Analysis method is an experimental one, its results are not always accurate. Its drawback is the inability to cope with low-quality, especially dark, images. By implementing appropriate preprocessing steps, the method's performance and accuracy can be improved.

2.3 Features Extraction Based on the Flood Fill Algorithm

Flood Fill is a well known algorithm used for graphical purposes [GG08] and is especially handy for detecting and filling connected single-colored areas of an image. We have used this characteristics to find local features of damaged fingerprints, such as straight lines or spots. These features are later used for the classification process.

In order to use the Flood Fill algorithm, the sample first needs to be preprocessed to obtain a black and white image. The preprocessing steps heavily depend on the image quality, as well as the type of sensor used for the acquisition. Therefore, for each database, they might differ. We have tailored the algorithm for our internal fingerprint database.

There are four types of features the Flood Fill algorithm is programmed to detect: large white spots, thick white lines, small dark spots and oblong dark lines (papillary lines disruptions). This is done using filtering the extracted areas according to specific parameters, such as the area's size or shape.

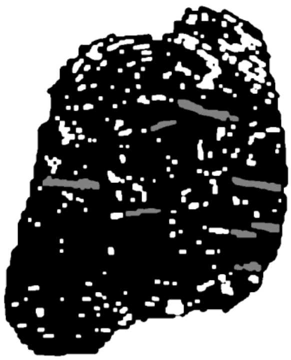

Fig. 4: Extraction of straight white lines.

2.4 Connection of the Methods

Connecting all three of the above-described methods together results in a surprisingly accurate description of the extent of damage in an entire area of a fingerprint image. They complement well as each of them detects a different kind of damage in the image.

At the end of each of the three detection methods, every image pixel is assigned a value 0 (healthy area), or a positive value up to 1 (damaged area). The greater the value, the more damaged the area to which the pixel belongs. Moreover, for the purpose of distinguishing fingerprint area from background, background was extracted separately according to [DH10] and the resulting information was stored in a fourth array with values -1 for (background) and 1 (fingerprint area).

The challenge was to connect these four output matrices together into a so-called Status Map which would give a good overview of the damage state every $w \times w$ block of pixels.

This is the description of the Status Map merging process:

1. Choose the resolution of the resulting Status Map.

2. Get the three output matrices and a background matrix.

3. For each matrix, compute a generalized block matrix (Status Map) that will store the average pixel values from $w \times w$ blocks: m_1, m_2, m_3 and $bckgr$.

4. Assign a weight to each method, according to the desired output and input image quality: w_1, w_2, w_3. Default values are: Orientation Field 2, Histogram Analysis 1 and Flood Fill 3.

5. For each block, compute its damage index. Damage index is a weighted mean of m_1, m_2 and m_3, masked by the value of the $bckgr$ matrix.

$$damageIndex(i,j) = bckgr(i,j) * \frac{w_1 * m_1(i,j) + w_2 * m_2(i,j) + w_3 * m_3(i,j)}{w_1 + w_2 + w_3}$$

6. *damageIndex* now represents the extent of damage in each image block. The resulting Status Map gives an excellent overview of the damage.

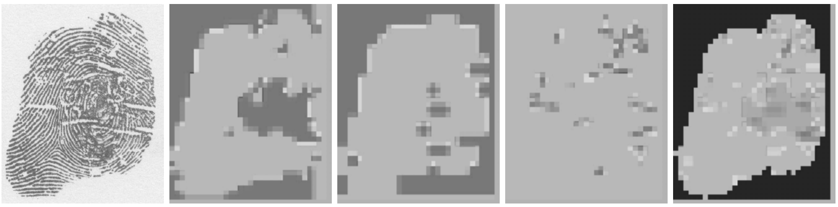

Fig. 5: Example of the pipeline of Status Maps and the final distribution of damage in the image (*atopic eczema*). Green color marks the healthy areas, blue color highlights the background and for the damaged areas a scale from yellow to red is used. Yellow stands for minor damage, whereas red implies extremely damaged places.

3 The Classification Process

The Classifier decides based on features extracted by the Flood Fill method and classifies the fingerprint image, according to the features' numbers and types. We have trained our classifier for 4 diseases: *acrodermatitis*, *atopic eczema*, *psoriasis* and *verruca vulgaris* [Ha09].

The decision rules have been determined with the help of statistics obtained from running the detector on our database of approximately 600 samples - see Table 1.

Tab. 1: Statistics of features extracted from each disease.

	acrodermatitis		atopic eczema		psoriasis		verruca vulgaris	
	med.	**std.dev.**	**med.**	**std.dev.**	**med.**	**std.dev.**	**med.**	**std.dev.**
white spots	5	3.97	5	4.31	8	5.35	1	3.02
white lines	2	1.84	3	3.06	4	2.65	1	1.63
dark spots	47	42.70	29	17.50	21	19.61	18	10.90
dark lines	7	8.37	17	19.80	8	9.22	15	39.76

Also, each disease has been given minimal value for some features (for example, *verruca vulgaris* logically has to have at least one white spot). All these characteristics are used in order to compute an estimated likelihood that a certain set of features belong to a particular disease. The classifier chooses the disease with the highest likelihood.

4 Results

Thanks to the connection of the detection methods, very satisfactory results have been achieved for locating the damaged areas - as an example, see Figure 6. The classifier accuracy reached interesting values as well, as described below.

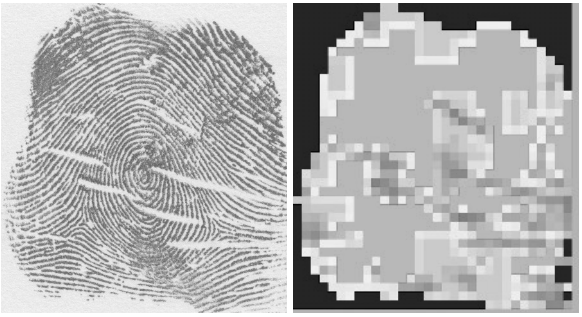

Fig. 6: Example of a final Status Map.

4.1 Classifier Accuracy

The classifier itself relies on the detection results. So far, the following accuracy measures have been computed for each disease class: FAR (*False Accept Rate*) and FRR (*False Reject Rate*) [Po11], ACC (total accuracy) - see Table 2. In this context, to *accept* means to classify a fingerprint into the disease class for which the measurements are being computed, whereas to *reject* means to classify a fingerprint into a different disease class. For the computation we used numbers of TP (*True Positives*), FN (*False Negatives*), FP (*False Positives*) and TN (*True Negatives*).

611 fingerprint images from dactyloscopic cards from the database were used for testing. The images had already been classified into disease classes by medical specialists. Table 2 shows the numbers of fingerprint images for each disease that were correctly/incorrectly classified by the algorithm.

Tab. 2: Classifier accuracy measures.

	TP	FN	FP	TN	FAR	FRR	ACC
Acrodermatitis	10	20	81	500	0.1394	0.6667	0.8347
Atopic eczema	126	297	37	151	0.1968	0.7021	0.4533
Psoriasis	31	87	168	325	0.3408	0.7373	0.5827
Verruca vulgaris	20	20	133	438	0.2329	0.5000	0.7496

The classification accuracy reached high values for for *acrodermatitis* (83.5%) and *verruca vulgaris* recognition (75.0%), whereas it was lower for *atopic eczema* (45.3%) and *psoriasis* (58.3%). The Classifier itself is ready to be further extended and improved.

5 Conclusion

We have developed algorithms that reach great quality in describing the overall extent of damage in a fingerprint image. The following methods were implemented: detection from block orientation field, Histogram Analysis method and an extended Flood Fill method. The best results were achieved by connecting the methods together using a Status Map. Along with the localizer, a classifier of four skin diseases was developed. It reached an accuracy of 83.5% for *acrodermatitis*, 45.3% for *atopic eczema*, 58.3% for *psoriasis* and 75.0% for *verruca vulgaris*.

There is a great potential for improvements and enhancements, and it is assumed that the research will continue. There are opportunities for the results of this research to be used in real-life applications in the future, such as medical applications or programs for police and security purposes.

Acknowledgement

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602. and the university project Secure and Reliable Computer Systems FIT-S-17-4014.

References

[DH10] Dolezel, Michal; Hejtmankova, Dana: Segmentation Procedure for Fingerprint Area Detection in Image Based on Enhanced Gabor Filtering. International Journal of Bio-Science and Bio-Technology, 2(4):39–50, December 2010.

[GG08] Godse, D. A.; Godse, A. P.: Computer Graphics. Technical Publications, 2008.

[Ha09] Habif, Thomas P.: Clinical Dermatology. Edinburgh: Mosby, 5 edition, 2009.

[HWJ98] Hong, Lin; Wan, Yifei; Jain, A.: Fingerprint image enhancement: algorithm and performance evaluation. IEEE Transactions Pattern Analysis and Machine Intelligence, 20(8):777–789, 1998.

[JFR08] Jain, A. K.; Flynn, P.; Ross, A. A.: Handbook of Biometrics. Springer-Verlag, 2008.

[Ma09] Maltoni, Davide; Maio, Dario; Jain, Anil K.; Prabhakar, Salil: Handbook of Fingerprint Recognition. Springer, 2 edition, 2009.

[Po11] Powers, D. M. W.: Evaluation: From Precision, Recall and F-measure to ROC, Informedness & Correlation. Journal of Machine Learning Technologies, 2:37–63, 2011.