Enhanced Segmentation-CNN based Finger-Vein Recognition by Joint Training with Automatically Generated and Manual Labels

Ehsaneddin Jalilian and Andreas Uhl
Department of Computer Science, University of Salzburg
Jakob-Haringer-Str.2, Salzburg, Austria
ejalilian@cs.ac.at,Uhl@cs.ac.at

Abstract

Deep learning techniques are nowadays the leading approaches to solve complex machine learning and pattern recognition problems. For the first time, we utilize state-of-the-art semantic segmentation CNNs to extract vein patterns from near-infrared finger imagery and use them as the actual vein features in biometric finger-vein recognition. In this context, beside investigating the impact of training data volume, we propose a training model based on automatically generated labels, to improve the recognition performance of the resulting vein structures compared to (i) network training using manual labels only, and compared to (ii) well established classical recognition techniques relying on publicly available software. Proposing this model we also take a crucial step in reducing the amount of manually annotated labels required to train networks, whose generation is extremely time consuming and error-prone. As further contribution, we also release human annotated ground-truth vein pixel labels (required for training the networks) for a subset of a well known finger-vein database used in this work, and a corresponding tool for further annotations.

1. Introduction

Finger-vein recognition is a biometric method in which a person's finger-vein patterns, captured under tissue-penetrating near-infrared (NIR) illumination, are used as a basis for biometric recognition. This technique is considered to offer significant advantages compared to classical biometric modalities (e.g. fingerprint, iris, and face recognition), as the vein patterns can be captured in touch-less manner, are not influenced by finger surface conditions, are acquired typically in non-invasive manner and only when the subject is alive, and cannot easily get forged. While plenty of finger-vein recognition methods have been proposed in recent years, yet extracting accurate vein patterns from NIR finger-vein images remains far from being trivial. This is mainly due to the often poor quality of the acquired imagery. Improperly designed scanner devices, close distance between finger and the camera (causing optical blurring), poor NIR lighting, varying thickness of fingers, ambient external illumination, varying environmental temperature, and light scattering represent different aspects which can degrade the finger-vein images' quality and cause the images to contain low contrast areas and thus ambiguous regions between vein and non-vein areas. The intensity distributions in these areas are complicated, and it is very hard to propose a mathematical model which can described them. Nevertheless, even manual annotation of the actual vein patterns (required as ground-truth to train segmentation CNN networks) in such ambiguous areas is extremely difficult, time-consuming, and error-prone process.

In this paper, for the first time in literature, we utilize three different the-state-of-the art CNN-based semantic segmentation architectures to segment finger-vein patterns from NIR finger imagery and use the extracted patterns for the recognition process, proposing an efficient training and configuration setting for these networks. In particular, beside inspecting the impact of training data volume, we investigate three automatic label generation techniques, and use the obtained labels together with manual labels (in varying quantity combinations) in joint training of the networks, primarily to improve the networks' feature extraction capability, and also eventually to eliminate the need for manual labels, whose annotation (especially in the ambiguous areas mentioned above) is extremely time-consuming and cumbersome. After training the networks with these labels and obtaining corresponding vein patterns, we evaluate the recognition performance in terms of receiver operating characteristics and relate the results to those obtained

2019 IEEE 5^{th} International Conference on Identity, Security, and Behavior Analysis (ISBA)
978-1-7281-0532-1/19/$31.00

by classical vein feature extraction techniques. We further publicly release human annotated ground-truth used in network training (and a corresponding tool to generate further vein-pattern labels) for a subset of a well known finger-vein database for the first time.

2. Related work

Classical finger-vein recognition techniques (using model-based, aka "hand-crafted" features) generally fall into two main categories: feature-based methods and profile-based methods. Feature-based methods assume that in the clear contour of finger-vein images, the pixels located in the vein regions have lower values than those in the background and that the vein pattern has a line-like shape in a predefined neighborhood region. E.g. "Repeated Line Tracking" (RLT [19]) tracks the veins as dark lines in the finger-vein image. A tracking point is repeatedly initialized at random positions, and then moved along the dark lines pixel by pixel. The number of times a pixel is traversed is recorded in a matrix. Pixels that are tracked multiple times have a high likelihood of belonging to a vein. The matrix is then binarized using a threshold.

Profile-based approaches consider the cross-sectional contour of vein pattern which shows a valley shape. E.g. "Maximum Curvature" (MC [20]) traces only the center lines of the veins and is insensitive to varying vein width. To extract the center positions, first the local maximum curvature in the cross-sectional profiles of vein images is determined. Next, each profile is segmented as being concave or convex, where only local maxima in concave profiles are specified as valid center positions. Then according to width and curvature of the vein region a score is assigned to each center position, and recorded in a matrix called locus space. Eventually, the matrix is binarized using the median of the locus space. Similarly, in "Deformation-Tolerant Feature-Point" (DTFP [32], a more recent approach), curvature of image-intensity profiles is used to extract feature points that are robust against irregular shading and vein deformation. Another profile-based method, exploiting the line-like shape of veins in a predefined neighborhood region is termed "Gabor Filter" (GF [2]). A filter bank consisting of several 2D even symmetric Gabor filters with different orientations is created. Several feature images are extracted using different filters from the filter bank, and consequently fused to generate the final feature image. There are many other techniques which often apply classical feature extraction techniques to the finger-vein pattern generation task such as Local Binary Pattern (LBP [6]), Region Growth [13], Principal Component Analysis (PCA [7]), etc. For a general overview on finger-vein recognition techniques up to 2014, please refer to e.g. [18].

2.1. CNN based finger-vein recognition

Recent techniques in deep learning, and especially CNNs, are gaining increasing interest within the biometric community. However, in finger-vein recognition prior art is relatively sparse and the extent of sophistication is quite different. The simplest approach is to extract features from certain layers of pre-trained classification networks and feed those features into a classifier to determine similarity to result in a recognition scheme. This approach is suggested by Li et al. [29] who apply VGG-16 and AlexNet feature extraction and KNN classification for recognition. Extracting vein features as such rather than the binary masks, hinders the application of more advanced training techniques such as label fusion.

Another approach to apply classical classification networks is to train the network with the available enrollment data of certain classes (i.e. subjects). Radzi et al. used a model of four-layered CNN classifier with fused convolutional-subsampling architecture for finger-vein recognition [1]. Itqan et al. performed finger-vein recognition using a CNN classifier of similar structure [16], and Das et al. [9] correspondingly proposed a CNN classifier for finger-vein identification. This approach however has serious drawbacks in case new users have to be enrolled as the networks should be re-trained, which is not practical.

Hong et al. [10] used a more sensible approach, employing fine-tuned pre-trained models of VGG-16, VGG-19, and VGG-face classifiers, which is based on determining whether a pair of input finger-vein images belongs to the same class (i.e. subject) or not. Likewise, Xie et al. [30] used several known CCN models (namely: light CNN (LCNN) [28], LCNN with triplet similarity loss function [24], and a modified version of VGG-16) to learn useful feature representations and compare the similarity between finger-vein images. Doing so, they eliminated the need for training in case of new enrolled users. However utilizing raw images, the system possesses a potential security threat.

Qin et al. [12], being the only approach so far focusing on explicit segmentation of vein patterns, applied a two-step procedure to extract the finger-vein patterns from NIR finger images. First, they used a CNN classifier to compute the probability of patch center pixels to belong to vein patterns, one by one, and labeled them according to the winning class (based on a probability threshold of 0.5). In the next step, in order to reduce finger-vein mismatches (as they had the problem of missing vein pixels) they further used a very shallow Fully Convolutional Neural Network (FCN) to recover those missing vein pixels. The approach used in the first network is rather simplistic and computationally demanding compared to the state-of-the-art segmentation networks as used in this work. Moreover, using a further network to recover the missing pixels, additional processing time is added to the feature extraction process.

3. Finger-vein Pattern Extraction using Semantic Segmentation CNNs

The first computer vision tasks for which initial CNN architectures were developed include classification [15], bounding box object detection [31], and key point prediction [4]. More recently, CNN architectures have been developed enabling semantic segmentation, in which each pixel is labeled separately with the class of its enclosing object or region. The primary techniques, classifying the center pixel of an entire image patch required immense time and computation resources, especially when used for large scale (whole image) segmentation. Fully convolutional neural networks are a rich class of architectures, which extend simple CNN classifiers to efficient semantic segmentation engines. Improving the classical CNN design with multi-resolution layer combinations, the resulting architectures are proven to be much better performing than their counterparts consisting of fully connected (FC) layers [14]. As the key distinction, typically the FC layer is replaced in FCN with a decoding mechanism, which uses the down-sampling information to up-sample the low resolution output maps to the full resolution of the input volumes in a single step, reducing computational cost and improving segmentation accuracy. There have been already attempts to use FCNs to extract vessel patterns from different human organs. For example, in [3] an FCN is used for segmentation of retinal blood vessels in fundus imagery, or in [21] an FCN is used for vessel segmentation in cerebral DSA series. However, there are significant differences as compared to this work. First, the networks have been trained with manually annotated labels provided by human experts only, and second, evaluation has been done with respect to segmentation accuracy relative to the ground truth labels while in our context segmentation results are indirectly evaluated by assessing recognition performance using the generated vein patterns.

In this work we used three different FCN architectures to extract the finger-vein patterns from NIR finger images. The first network architecture we used to extract the finger-vein patterns is "Unet" by Ronneberger et al. [22]. The network consists of an encoding part, and a corresponding decoding part. The encoding architecture consists of units of two convolution layers, each followed by a rectification layer (ReLU) and a 2×2 down-sampling (Pooling) layer with stride 2. At each down-sampling step, feature channels are doubled. The corresponding decoding architecture consists of units of 2×2 up-convolution layers (up-sampling layers, which halve the number of feature channels), a concatenation operator with the cropped feature map from the corresponding encoding unit, and two 3×3 convolutions, each followed by a ReLU. At the final layer a 1×1 convolution is used to map the component feature vectors to the desired number of segmentations. The network's softmax layer generates the final segmentation as a probability map, whose pixel values reflect the probability of a particular pixel to belong to a vein or not. The network implementation[1] was realized in the TensorFlow and Keras.

Network	Unet	RefineNet	SegNet
Optimizer	Stochastic gradient descent	Adam	Stochastic gradient descent
Learning rate	0.08	0.0001	0.003
Momentum	0.9	-	0.01
Weight decay	0.0005	0.1	0.000001
Iteration	300	40,000	30,000

Table 1: Networks' training parameters.

The second network architecture we used to extract the finger-vein patterns is "RefineNet" [17]. RefineNet is a multi-path refinement network, which employs a 4-cascaded architecture with 4 RefineNet units, each of which directly connects to the output of one Residual net [11] block, as well as to the preceding RefineNet block in the cascade. Each RefineNet unit consists of two residual convolution units (RCU), whose outputs are fused into a high-resolution feature map, and then fed into a chained residual Pooling block. The implementation[2] of this network was also realized in the TensorFlow and Keras.

The third network architecture we used in our work is identical to the "Basic" fully convolutional encoder-decoder network proposed by Kendall et al. [27], and is termed "SegNet" subsequently. However, we redesigned the network's softmax layer to segment only the vein pattern. The whole network architecture is formed by an encoder network, and a corresponding decoder network. The network's encoder architecture is organized in four stocks, containing a set of blocks. Each block comprises a convolutional layer, a batch normalization layer, a ReLU layer, and a Pooling layer with kernel size of 2×2 and stride 2. The corresponding decoder architecture, likewise, is organized in four stocks of blocks, whose layers are similar to those of the encoder blocks, except that here each block includes an up-sampling layer. In order to provide a wide context for smooth labeling in this network the convolutional kernel size is set to 7×7. The decoder network ends up to a softmax layer which generates the final segmentation map. The network implementation[3] was realized in Caffe deep learning framework. Table 1 summarizes the training parameters (which turned out to deliver best results) we used to train each network in our experiments.

4. Experimental Framework

Database: We used the UTFVP database [26][4], acquired by the University of Twente with a custom sensor, in our

[1] https://github.com/orobix/retina-unet.
[2] https://github.com/eragonruan/refinenet-image-segmentation.
[3] http://mi.eng.cam.ac.uk/projects/segnet/tutorial.html.
[4] Available at: http://scs.ewi.utwente.nl/downloads.

experiments. The UTFVP database contains 1440 finger-vein images (with resolution of 672×380 pixels), collected from 60 volunteers. For each volunteer, the vein pattern of the index, ring, and middle finger of both hands have been collected twice at each session (each individual finger has been captured four times in total). We resized the images to the corresponding networks' input volumes, using bicubic interpolation method, as specified in the Table 2.

Training Labels Generation: We established and utilized an annotation tool (implemented as ImageJ plugin) to generate the manual labels for a subset (400 samples) of the UTFVP dataset (including at least one sample per subject). Using this tool, the vein structure is marked using polylines. Each line segment is assigned with a width representing the vein thickness. In order to diminish variances introduced by different persons all annotations were accomplished by the same person. We release the tool and annotated labels for further usage under the link: (blinded for review).

With primary aim of improving network performance and also considering the fact that data labeling is an expensive and time-consuming task, especially due to the significant human effort involved, we also use the approach of automatically generating labels and training the networks with different proportion of such labels jointly with manual labels. Addressing the automatic training label generation, in some works (i.e. [25]), it has been suggested to generate training ground-truth labels utilizing available classical algorithms within the same field. In [12], authors used several algorithms to generated a set of finger-vein masks and then applied a probabilistic algorithm to each pixel (within the masks) to assign it as being vein or not. However, to the best of the authors' knowledge, this approach: (i) has not yet been investigated systematically, and (ii) has not been used jointly with manual labels in network training process so far. Subsequently, we generated the same number of corresponding automated labels (using the identical images), utilizing the following classical binary vein-pattern extraction algorithms: Maximum Curvature (MC), Gabor Filter (GF), and Repeated Line Tracking (RLT). The technical details of these algorithms are already discussed in Section 2. For MC and RLT we used the MATLAB implementation of B.T. Ton [5], and for GF we used the implementation in [23][6].

Network Training and Finger-vein Recognition Evaluations: We divided the whole database into 2 parts, each containing a disjoint set of training (200 samples, set by experiment to obtain the networks' full capacity) and testing (720 samples) divisions. Then we created 6 disjoint subdivisions (containing 5, 20, 60, 100, 140, 180 labels) within each manual label division, and 5 disjoint subdivisions (containing 40, 80, 120, 160, 200 labels) within each automatically generated label division respectively. In the first stage of our experiments, we trained each network with the subdivisions within the first manual label division, and evaluated the networks on the corresponding subdivisions in the second testing division. Next we trained networks with the subdivisions in the second manual label division, and evaluated the networks on the corresponding subdivisions in the first testing division. In the second stage of our experiments, we repeated the same training and evaluation procedure using automatically generated subdivisions (while adding 40 manual labels to each training subdivision). Doing so, we tested the networks on the whole database in each experiential stage, without overlapping training and testing sets. Table 2 shows the segmentation run-time per input volume for each network, on TITAN-X (Pascal) GPUs.

Network	Unet	RefineNet	SegNet
Input volume size	584×565	584×565	360×480
processing time	$3.164s$	$0.138s$	$0.0398s$

Table 2: Run-time per input volume for each network.

To quantify the recognition performance of the networks (using their vein pattern outputs), as well as the classically generated vein patterns in comparison, receiver operator characteristic behavior is evaluated. In particular, the equal error rate EER as well as the FMR 1000 (FMR) and the ZeroFMR (ZFMR) are used. For their respective calculation we followed the test protocol of the FVC2004 [8]. For matching the binary feature maps, we adopted the approach by Miura et al. [20], which is essentially the calculation of the correlation between an input and reference image. As the input maps are not registered to each other and only coarsely aligned (using LeeRegion [5] background removal), the correlation between the input image $I(x, y)$ and the reference one is calculated several times while shifting the reference image $R(x, y)$, whose upper-left position is $R(c_w, c_h)$ and lower-right position is $R(w - c_w, h - c_h)$, in x- and y-direction.

$$N_m(s,t) = \sum_{y=0}^{h-2c_h-1} \sum_{x=0}^{w-2c_w-1} I(s+x, t+y) R(c_w+x, c_h+y) \quad (1)$$

where $N_m(s, t)$ is the correlation. The maximum value of the correlation is normalized and used as matching score:

$$S = \frac{N_{m_{max}}}{\sum_{y=t_0}^{t_0+h-2c_h-1} \sum_{x=s_0}^{s_0+w-2c_w-1} I(x,y) + \sum_{y=c_h}^{h-2c_h-1} \sum_{x=c_w}^{w-2c_w-1} R(x,y)} \quad (2)$$

where s_0 and t_o are the indexes of $N_{m_{max}}$ in the correlation matrix $N_m(s, t)$, and S values are: $0 \leq S \leq 0.5$.

5. Results

Table 3 displays EER, FMR, and ZFMR results obtained by each network in the first stage of our experiments using varying number of manual training labels. As it can

[5] Publicly available on MATLAB Central.

[6] Available at: http://www.wavelab.at/sources/Kauba16e.

Networks	Unet			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
180 pcs	0.87	1.85	5.18	2.73	5.83	11.85	2.91	6.75	12.63
140 pcs	1.15	2.08	4.30	2.73	6.62	9.02	3.09	8.79	16.94
100 pcs	1.04	1.88	3.47	3.09	8.61	18.24	**2.21**	6.20	17.03
60 pcs	1.71	3.65	11.52	2.32	6.01	9.39	2.35	6.66	11.25
20 pcs	**0.64**	1.94	6.34	2.26	5.83	8.19	7.26	25.09	53.70
5 pcs	3.80	11.75	24.30	**1.76**	4.12	6.34	9.71	25.69	31.57

Table 3: Networks' performance, trained with different number of manual labels.

be seen in the table, Unet performs better than the other two networks in terms of almost all parameters (EER, FMR and ZFMR). The network shows the best performance when trained with 20 labels only, while increasing the number of training labels (specially between 60 to 140) erodes the network performance considerably. SegNet and RefineNet show rather similar performance, as the EER, FMR and ZFMR results obtained by these networks demonstrate. Yet it is interesting to note that while RefineNet achieves the best performance when trained with minimum of 5 training labels, the performance of SegNet improves as the number of training labels increases (at least up to 100 pcs).

Method	MC			GF			RLT			DTFP		
Database	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
UTFVP	0.41	0.55	1.29	1.11	2.45	4.12	2.17	5.87	9.35	1.68	2.91	5.18

Table 4: Classical algorithms' performance.

In order to assess the recognition performance of the vein patterns generated by the different network training approaches considered, we compared the corresponding recognition performance to that of the classical algorithms as presented in Table 4. As it can be observed in the tables, Unet shows better performance than the GF, RLT, and DTFP algorithms, when train with certain number (i.e. 20, 180) of labels, while RefineNet outperforms only RLT algorithm when trained with a limited (5) pcs of labels. SegNet generally does not perform well on the dataset and falls behind all the classic algorithms.

Next, we look into the results we obtained in the second stage of our experiments, where we trained networks jointly with different proportion of automatically generated labels, and 40 pcs of manual labels). As Table 5, and also the corresponding DET (Detection Error Trade-off) curves in Figure 1 illustrate, training networks jointly with labels generated by MC algorithm and the manual labels significantly improves the networks performance. Furthermore, the networks' performance continuously increases with increasing the quantity of training labels (up to a certain saturation point). Note that this behavior is only observed for SegNet on manual labels. The most interesting results are

Networks	Unet			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
200 pcs	**0.32**	0.60	0.92	**0.28**	0.37	1.57	1.43	2.45	5.64
160 pcs	0.51	1.20	5.18	**0.28**	0.69	1.29	1.34	2.31	3.42
120 pcs	0.41	0.64	1.25	0.36	0.69	1.11	**0.73**	1.66	2.91
80 pcs	0.41	0.55	0.78	0.47	0.97	1.25	1.15	3.47	9.12
40 pcs	1.25	1.38	2.17	1.43	2.50	12.91	4.44	12.96	16.80

Table 5: Networks' performance, trained with different number of labels generated by MC algorithm.

obtained by RefineNet, when trained with sufficient (160, 200) pcs of training samples, scoring: 0.28, 0.37, 1.57, and 0.28, 0.69, 1.29, in ERR, FMR, and ZFMR parameter respectively. These results clearly outperforms the best classical algorithms results (obtained by MC algorithm) in all terms (see Table 4). Likewise, Unet outperforms MC algorithm (and all other algorithms) when trained with 200 pcs of automatically generated labels, while generally outperforms GF, RLT, and DTFP algorithms when trained with more than 40 labels. SegNet outperforms GF, RLT, and DTFP algorithms when trained with 120 labels, while increasing the number of training labels for this network (up to 180 pcs) generally erodes its performance.

As Table 6 shows, the results for training networks jointly with labels generated by GF algorithm and the manual labels shows just limited improvements (i.e. SegNet trained with 160 or more pcs of automatically generated labels), as compared to those obtained when training networks jointly with labels generated by MC algorithm and the manual labels (see Table 5, and the corresponding DET curves in Figure 1 for more details). Nevertheless, Unet and SegNet outperform the GF, RLT, and DTFP algorithms when trained with distinct number (i.e. 80, 16 receptively) of GF labels, and RefineNet only outperforms the RLT algorithm when trained with 80 pcs of this type of labels.

Networks	Unet			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
200 pcs	0.79	2.73	3.79	2.13	5.04	8.75	1.20	2.68	5.55
160 pcs	1.02	3.14	12.77	2.77	6.85	10.13	**0.78**	2.59	5.74
120 pcs	3.33	64.30	95.23	3.74	9.35	12.08	1.47	3.37	5.60
80 pcs	**0.74**	2.08	6.71	**1.84**	5.00	8.33	2.21	6.25	12.82
40 pcs	1.61	3.56	6.15	2.36	4.07	7.50	6.57	12.31	17.31

Table 6: Networks performance, trained with different number of labels generated by GF algorithm.

Similarly, as it can be seen in Table 7, training networks jointly with labels generated by RLT algorithm and manual labels just results in limited improvement in SegNet's preference (when trained with 160 or more pcs of automatically generated labels), as compared to the results in Table 5. However, comparing the results to those obtained when

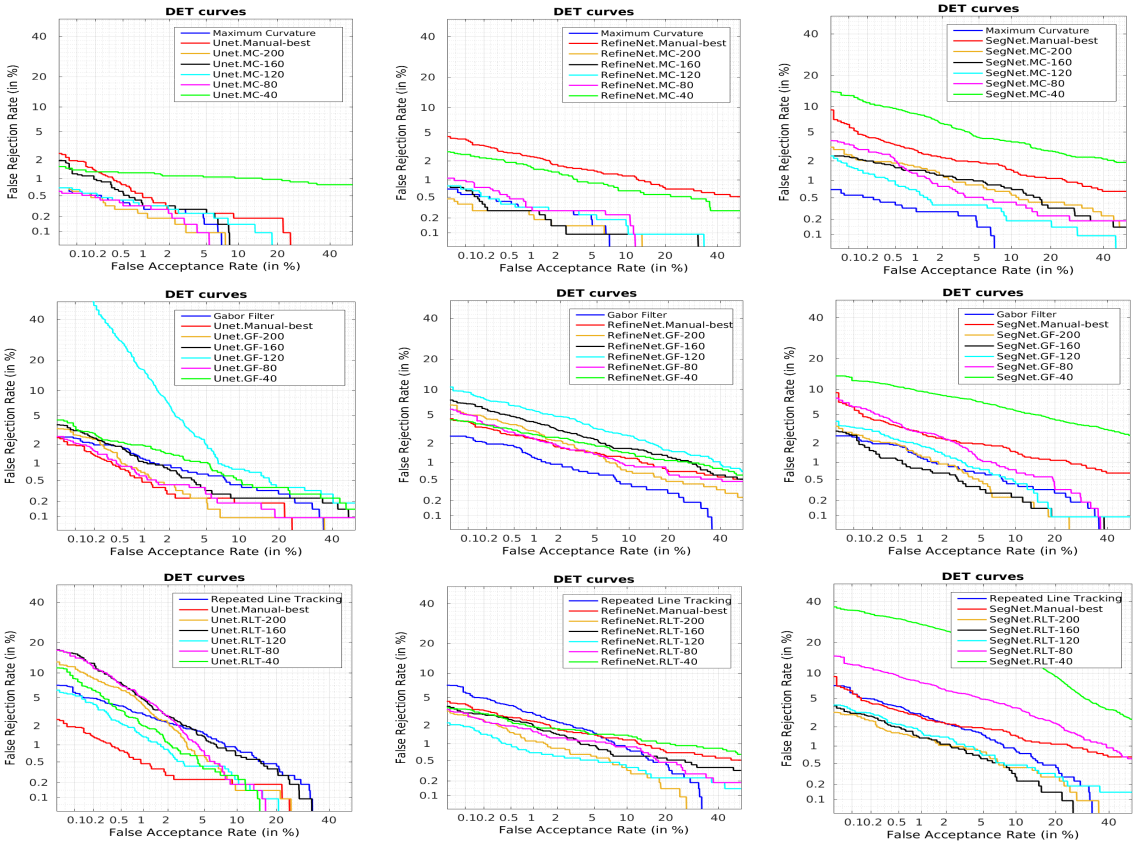

Figure 1: DET curves for: Unet (left column), RefineNet (middle column), and SegNet(right column) networks, trained whit automatically generated labels using: MC (first row), GF (second row), and RLT (third row) algorithms.

Networks	Unet			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
200 pcs	2.09	11.62	24.86	1.10	2.82	3.75	**1.29**	3.00	7.59
160 pcs	2.45	15.78	23.33	1.61	3.05	5.00	**1.29**	3.14	6.48
120 pcs	**1.16**	5.46	11.20	**0.78**	1.89	3.51	1.43	3.51	5.69
80 pcs	2.36	14.95	35.09	1.38	3.00	4.90	4.87	12.31	19.02
40 pcs	1.57	8.61	17.96	1.80	3.47	6.20	13.41	35.23	43.19

Table 7: Networks' performance, trained with different number of labels generated by RLT algorithm.

training networks jointly with labels generated by GF algorithm and manual labels (Table 6), interestingly we can observe that, while RefineNet gains up to 50% improvement, yet Unet suffers a considerable degradation up to the same order of magnitude. This is mostly due to the effect of labels quality (accuracy) and the architectural specifications of the networks, which will be discussed later in section 6.

6. Discussion

When analyzing our results, the first issue to be discussed is the quality/accuracy of the manual labels (see Figure (2b) for an example). Human annotators have been instructed to only annotate vein pixels without any ambiguity in order to avoid false positive annotations. Thus, manual labels are restricted to rather large scale vessels, while fine grained vasculature is entirely missed/avoided. The correspondingly segmented vein patterns are rather sparse and it may be conjectured that these patterns simply do not contain sufficiently high entropy to facilitate high accuracy recognition. In contrast, more accurate labels such as MC labels, and their corresponding outputs of CNNs trained with these labels, exhibit much more fine grained vasculature details, reflected in much better recognition accuracy.

As reflected in the tables, the performance of the networks is quite different using a changing number of manual training labels. RefineNet maintains a certain level of performance almost in all cases and seems to stay invariant with respect to the quantity of the training labels. The network converges well and exhibits its optimal performance even with a limited number of (5) manual training labels. Nonetheless, the network's capability to learn the target pattern significantly improves in case of introducing a higher quantity of more precise labels (i.e. MC labels). This seems to be owed to the multi-path refinement architecture used in

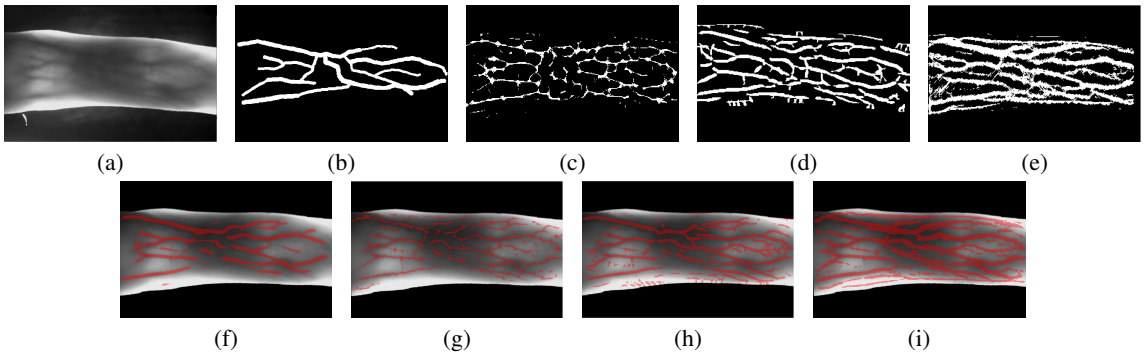

Figure 2: Sample finger-vein image (2a), its manual ground-truth (2b), and the corresponding automatically generated labels using: MC (2c), GF (2d), and RLT (2e) algorithms, along with outputs of RefineNet when trained with manual (2f), and in joint with these lables (2g), (2h), (2i) respectively.

this network, which exploits the information available all along the down-sampling process to enable high-resolution prediction, emphasizing on preservation of veins' edges and boundaries, and thus retaining the veins' main structure.

Unet's architecture is designed to converge fast with a limited number of training labels. When trained with precise labels (i.e. MC labels), this network is able to deal well with the ambiguous boundary issue between vein and non-vein regions in finger-vein images. The network benefits from the large number of feature channels built into its architecture, which allow for propagating key context information to higher resolution layers. However, such an architecture seems to be very sensitive to the quality of the input images. A simple comparison of the very different results obtained by this network when trained with labels generated by MC, and RLT algorithms underpins this fact clearly.

SegNet enjoys a stable (however not optimal) performance, reflecting the network's ability to deal with low quality training labels (i.e. RLT labels). Meanwhile, the network's performance considerably improves by introducing actual vein pixel labels, and removing outliers (non-vein pixels) using automatically generated labels. This ability of the network is mainly owed to the up-sampling mechanism used in this network, which uses max-pooling indices from the corresponding encoder feature maps to generate the up-sampled feature maps without learning. Nevertheless, the network seems to be comparably more sensitive to the quantity of the training labels, and regardless of the quality of the input labels, requires a minimum of 60 to 120 labels to converge to its optimal performance.

7. Conclusion

In the context of training three different FCN architectures, utilizing a varying number of manual and additional automatically generated labels, we have found that results vary significantly among the different networks. First, the number of required training labels is highly network architecture dependent and second, only Unet and RefineNet are able to outperform the best considered classical recognition technique (MC). We have demonstrated that using automatically generated labels in training in addition to manual ones can significantly improve the networks' performance in terms of achieved recognition accuracy and only in this configuration clearly outperforms classical feature extraction schemes. Furthermore, we observed that the quality of training labels has a significant impact, also when comparing the usage of different automatically generated labels.

In future works we will assess the strategy to use labels generated by different automated feature extraction techniques in a single training process. Also, an evaluation of cross-vessel type (using training data of different vessel types, e.g. retinal vasculature) training will be conducted. Finally, we will look into augmentation techniques specifically tailored to the observed problem with the manual labels, i.e. scaling the data to model finer vessel structures.

Acknowledgements

This project received funding from the European Union's Horizon 2020 research and innovation program under the grant agreement No 700259.

References

[1] R. S. Ahmad, H. M. Khalil, and B. Rabia. Finger-vein biometric identification using convolutional neural network. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(3):1863–1878, 2016.

[2] K. Ajay and Z. Yingbo. Human identification using finger images. *IEEE Transactions on Image Processing*, 21(4):2228–2244, 2012.

[3] D. Avijit and S. Sonam. A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation. In *Proceedings of 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 248–251. IEEE, 2017.

[4] L. Ce, Y. Jenny, and T. Antonio. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011.

[5] L. E. Chul, L. H. Chang, and P. K. Ryoung. Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. *International Journal of Imaging Systems and Technology*, 19(3):179–186, 2009.

[6] L. E. Chul, J. Hyunwoo, and K. Daeyeoul. New finger biometric method using near infrared imaging. *Sensors*, 11(3):2319–2333, 2011.

[7] W. J. Da and L. C. Tsiung. Finger-vein pattern identification using principal component analysis and the neural network technique. *Expert Systems with Applications*, 38(5):5423–5427, 2011.

[8] M. Dario, M. Davide, C. Raffaele, W. Jim, and J. Anil. Fvc2004: Third fingerprint verification competition. In *Lecture notes in Biometric Authentication*, pages 1–7. Springer, 2004.

[9] R. Das, E. Piciucco, E. Maiorana, and P. Campisi. Convolutional neural network for finger-vein-based biometric identification. *IEEE Transactions on Information Forensics and Security*, pages 1–1, 2018.

[10] H. H. Gil, L. M. Beom, and P. K. Ryoung. Convolutional neural network-based finger-vein recognition using nir image sensors. *Sensors*, 17(6):1297, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] Q. Huafeng and M. ElYacoubi. Deep representation-based feature extraction and recovering for finger-vein verification. *IEEE Transactions on Information Forensics and Security*, 12(8):1816–1829, 2017.

[13] Q. Huafeng, Q. Lan, and Y. Chengbo. Region growth-based feature extraction method for finger-vein recognition. *Optical Engineering*, 50(5):057208, 2011.

[14] L. Jonathan, S. Evan, and D. Trevor. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E.Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, volume 1 of *NIPS'12*, pages 1097–1105. USA, 2012.

[16] I. KS, S. AR, G. FG, M. N, W. YC, and I. MM. User identification system based on finger-vein patterns using convolutional neural network. *ARPN Journal of Engineering and Applied Sciences*, 11(5):3316–3319, 2016.

[17] G. Lin, M. Anton, S. Chunhua, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5168–5177, 2017.

[18] Y. Lu, Y. Gongping, Y. Yilong, and Z. Lizhen. A survey of finger vein recognition. In S. Zhenan, S. Shiguang, S. Haifeng, Z. Jie, W. Yunhong, and Y. Weiqi, editors, *Lecture notes in Chinese Conference on Biometric Recognition*, pages 234–243. Springer International Publishing, 2014.

[19] M. Naoto, N. Akio, and M. Takafumi. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Machine Vision and Applications*, 15(4):194–203, Oct 2004.

[20] M. Naoto, N. Akio, and M. Takafumi. Extraction of finger-vein patterns using maximum curvature points in image profiles. *IEICE Transactions on Information and Systems*, 90(8):1185–1194, 2007.

[21] C. Neumann, K. D. Tnnies, and R. P. Frhlich. Angiounet - a convolutional neural network for vessel segmentation in cerebral dsa series. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 331–338. INSTICC, SciTePress, 2018.

[22] R. Olaf, F. Philipp, and B. Thomas. U-net: Convolutional networks for biomedical image segmentation. In *Lecture notes in International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[23] E. Piciucco, E. Maiorana, C. Kauba, A. Uhl, and P. Campisi. Cancelable biometrics for finger vein recognition. In *Proceedings of the 1st Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE 2016)*, pages 1–6, Aalborg, Denmark, 2016.

[24] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[25] J. Shaima, D. Charles, H. Nicholas, and C. Edward. Using convolutional neural network for edge detection in musculoskeletal ultrasound images. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pages 4619–4626. IEEE, 2016.

[26] B. T. Ton and R. N. J. Veldhuis. A high quality finger vascular pattern dataset collected using a custom designed capturing device. In *Lecture notes in 2013 International Conference on Biometrics (ICB)*, pages 1–5, June 2013.

[27] B. Vijay, K. Alex, and C. Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[28] X. Wu, R. He, Z. Sun, and T. tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, Nov 2018.

[29] L. Xiaoxia, H. Di, and W. Yunhong. Comparative study of deep learning methods on dorsal hand vein recognition. In *Lecture notes in Chinese Conference on Biometric Recognition*, pages 296–306. Springer, 2016.

[30] C. Xie and A. Kumar. Finger vein identification using convolutional neural network and supervised discrete hashing. *Pattern Recognition Letters*, 2018.

[31] L. Yann, B. Bernhard, D. John, H. Donnie, H. Richard, H. Wayne, and J. Lawrence. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[32] M. Yusuke, M. Naoto, N. Akio, K. Harumi, and M. Takafumi. Finger-vein authentication based on deformation-tolerant feature-point matching. *Machine Vision and Applications*, 27(2):237–250, Feb 2016.