Finger-vein Recognition using Deep Fully Convolutional Neural Semantic Segmentation Networks: The Impact of Training Data

Ehsaneddin Jalilian and Andreas Uhl
Department of Computer Science, University of Salzburg
Jakob-Haringer-Str.2, Salzburg, Austria
ehsaneddin.jalilian@sbg.ac.at, andreas.Uhl@sbg.ac.at

Abstract

We propose a novel approach for finger-vein recognition, focused on direct extraction of actual finger-vein patterns from NIR finger images without any specific pre- or post-processing, using semantic segmentation convolutional neural networks (CNNs). We utilize three network architectures and besides identifying efficient training and configuration settings for these networks, using manually annotated training data, we present a training model based on automatically generated labels to improve the networks' performance. Based on our experimental results, the proposed model can achieve superior performance over traditional finger-vein recognition algorithms. As further contribution, we also release human annotated ground-truth vein pixel labels (required for training the networks) for a subset of two well known finger-vein databases 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 traditional biometric modalities (e.g. fingerprint, iris, and face recognition) [3], 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 meanly 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 [24], varying environmental temperature [16], and light scattering [9] 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 an extremely difficult and error-prone process.

In this work, we propose a new finger-vein recognition model using fully convolutional deep neural networks designed for semantic segmentation for direct extraction of actual finger-vein patterns (i.e. pixels labeled as vein or non-vein) present in NIR finger-vein samples. The generated binary labels (i.e. constituting the stored template) are directly used in the finger-vein recognition process. Recent methods utilizing convolutional neural networks (CNNs, see Section 2) typically differ from this approach, since they encompass the whole system. In some of these approaches, the biometric template used in biometric comparison is the NIR finger image itself rather than a derived set of features. While this is allowed and even standardized, it prevents the use of cancelable biometrics, since the image is used as is, creating a significant security risk. Furthermore, the advances made in any of the three major steps of traditional biometric systems (pre-processing, feature extraction, alignment and matching) can not be utilized when CNNs are used for all steps jointly. Finally, our approach allows to employ the generated binary labels in existing multiple features fusion [10] or multi-sample ([28, 30]) recognition techniques, which is not straightforward for all but the last CNN-based technique reviewed in Section 2 ([8]).

We consider three different CNN architectures and propose efficient training and configuration settings. In particular, we investigate the (minimal) number of training samples required to train these networks as the generation of manual annotation labels of the actual vein patterns (especially in the ambiguous areas mentioned above) is extremely time-

consuming and cumbersome. In this context, we also publicly release human annotated ground-truth used in network training (and a corresponding tool to generate further vein-pattern labels) for subsets of two well known finger-vein databases for the first time. Furthermore, in order to eventually eliminate the need for (further) manually annotated labels, we present training models based on (i) automatically generated labels and (ii) apply cross-dataset training.

2. Related work

For a general overview on finger-vein recognition techniques up to 2014, please refer to e.g. [14]. Traditional finger-vein recognition techniques (using model-based, aka "hand-crafted" features) generally fall into two main categories: Profile-based methods and feature-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. Profile-based approaches consider the cross-sectional contour of a vein pattern which shows a valley shape. "Maximum Curvature" (MC [16], being of profile-based type) is used as the recognition performance baseline in this work as it has turned out to deliver robust results on a wide range of datasets (e.g. [24], [10]) and is publicly available as open source software, thus fostering reproducible research. MC tracks the veins as dark lines in the finger-vein image, initializing random positions, and then moving along the dark lines pixel by pixel. Additionally, we compare results to those of "Repeated Line Tracking" (RLT [15]), "Gabor Filters" (GF [12]), and "Deformation-tolerant Feature Point" (DTFP[29], a more recent approach) algorithms, for which also open software is available.

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 vein pattern similarity to result in a recognition scheme. This approach is suggested by Li et al. [26] who apply VGG-16 and AlexNet for feature extraction, and a KNN classifier for recognition.

Another apporach to apply traditional 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, reduced-complexity, four-layered CNN classifier with fused convolutional-subsampling architecture for finger-vein recognition [18]. Itqan et al. performed finger-vein recognition using a CNN classifier of similar structure [11]. This approach however has significant drawbacks in case new users have to be enrolled as the networks have to be re-trained, which is not practical. Hong et al. [5] 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 images belongs to the same subject or not. Thus, they eliminated the need for training in case of new enrollment. Similarly, Xie et al. [27] used several known CCN models (namely: light CNN (LCNN) [25], LCNN with triplet similarity loss function [21], and a modified version of VGG-16) to learn useful feature representations and compare the similarity between finger-vein images.

Qin et al. [8], 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 CNNs

There have been already attempts to use FCNs to extract vessel patterns from different human organs. For example, in [4] an FCN is used for segmentation of retinal blood vessels in fundus imagery, or in [17] 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 use three different FCN architectures to extract the finger-vein patterns from NIR finger images. The first network architecture is the U-net by Ronneberger et al. [19]. 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. The corresponding decoding architecture consists of units of 2×2 up-convolution layers (up-sampling), 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 is implemented[1] in the TensorFlow using the Keras library.

The second network architecture we used to extract the finger-vein patterns is RefineNet [13]. 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 [7] 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 using the Keras library.

The third network architecture we used in our work is identical to the "Basic" fully convolutional encoder-decoder network proposed by Kendall et al. [2] and is termed "SegNet" subsequently. 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. The decoder network ends up to a softmax layer which generates the final segmentation map. The network implementation[3] was realized in the Caffe deep learning framework.

4. Experimental framework

Database: For our experiments, we used the UTFVP database [23][4], which contains 1440 finger-vein images (with resolution of 672×380 pixels), collected from 60 volunteers, with 2 hands and 3 fingers (index, middle, and ring fingers), and 4 images per finger. We further considered the SDUMLA-HMT (SDUMLA) database[5], originally containing 3816 images, with a resolution of 320×240 pixels, acquired from 106 subjects, with 2 hands and 3 fingers (index, middle, and ring fingers), and 6 images per finger. For our work, to enable fair comparison between the datasets, we used a subset (1440 images with the same finger / subject structure as the UTFVP database) of this database. Generally, the latter database is considered as a more difficult one (compared to the UTFVP database), as most images in this database exhibit shadings and low image contrast in the vein regions (see Figures 2a and 2d for exemplary images).

Training Labels Generation: We established and utilized an annotation tool (implemented as ImageJ plugin) to generate the manual labels for a subset (360 samples) of each 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. CLAHE [31] is used in the tool to optimize the visibility of the vein patterns.

An exemplary original sample image and annotated label from the UTFVP and SDUMLA datasets are depicted in Figures 2a, 2b and 2d, 2e respectively. We release the tool and annotated labels for further usage under the link: (blinded for review). As 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 as a comparison. It is not surprising that generating ground-truth labels automatically to train CNNs has been suggested for some CNN-based segmentation tasks in medical imaging (e.g. [20] [22] [6] [1]), as it is particularly difficult to convince physicians to come up with a sufficient number of manually annotated labels. To the best of the authors' knowledge, this approach has not yet been investigated for segmentation of vascular data.

To enable a fair comparison, we generated the same number of corresponding "automated" labels (i.e. 360, also using the identical images) utilizing the Max Curvature (MC) feature extraction algorithm. The technical details of this algorithm are already discussed in Section 2. We used the MATLAB implementation of this algorithm [6].

Network Training and Finger-vein Recognition Evaluations: We divided each dataset into two parts, creating two disjoint testing sets (containing 720 samples each) in each dataset. Then within each testing set, we created a training set (corresponding to the available annotated samples), forming two disjoint training sets (containing 180 samples each) as well. Then we created 5 disjoint training subsets (containing 180, 140, 100, 60, 20 and 5 labels, respectively) within each training set. In the first stage of our experiments, we trained the networks with each training subset belonging to the first training set of each dataset using manual labels, and evaluated the networks on the second testing set of the same dataset. We repeated the same training process using the training subsets in the second training set in each dataset, and then evaluated the networks on the

[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.
[5] Available at: http://mla.sdu.edu.cn/sdumla-hmt.html.
[6] http://www.mathworks.com/matlabcentral/fileexchange/35716-miura-et-al-vein-extraction-methods.

Networks	U-net			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
180	5.14	11.52	16.01	3.28	7.68	12.59	2.79	6.85	11.43
140	4.53	10.04	13.61	2.95	7.08	13.61	2.54	5.18	14.53
100	4.95	11.15	14.44	3.05	7.59	14.12	3.28	7.50	11.62
60	5.98	11.62	14.58	2.95	6.48	15.55	4.77	11.38	15.37
20	4.25	9.16	12.26	2.63	7.17	12.12	4.99	11.01	22.08
5	5.93	13.33	21.38	2.45	6.20	11.80	6.06	16.99	22.36
UTFVP 360	4.95	12.59	17.22	2.87	8.93	19.25	4.94	12.31	23.93
UTFVP 180	5.28	12.68	17.59	3.88	10.50	16.11	6.61	16.34	24.62

Table 1: Networks performance on SDUMLA dataset, trained with different manual labels.

Networks	U-net			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
180	0.87	1.85	5.18	2.73	5.83	11.85	2.91	6.75	12.63
140	1.15	2.08	4.30	2.73	6.62	9.02	3.09	8.79	16.94
100	1.04	1.88	3.47	3.09	8.61	18.24	2.21	6.20	17.03
60	1.71	3.65	11.52	2.32	6.01	9.39	2.35	6.66	11.25
20	0.64	1.94	6.34	2.26	5.83	8.19	7.26	25.09	53.70
5	3.80	11.75	24.30	1.76	4.12	6.34	9.71	25.69	31.57
SDUMLA 360	0.41	0.78	2.22	2.17	4.12	5.32	2.82	6.94	10.50
SDUMLA 180	0.41	0.74	2.63	1.80	3.37	6.20	2.17	3.98	5.09

Table 2: Networks performance on UTFVP dataset, trained with different manual labels.

first testing set of the same dataset. By doing so, we tested the networks on all samples in each dataset without overlapping training and testing sets using different training subsets. As a further part of our experiments, molding cross-dataset training, we also trained networks with the whole and half of total training samples (i.e. 360 and 180) in each dataset and tested them on the other dataset. In the second stage of our experiments, we repeated the same training and testing process on the datasets using the automatically generated labels instead of manual labels (as used in the first stage of our experiment). Also, we preformed the cross-dataset experiments only on the UTFVP dataset. For the sake of comparability the training parameters (epoch, learning rate, etc) kept unchanged for each network (regardless of training label quantity).

As we wanted the comparison to concentrate on the quality of the pure training labels, we deliberately did not apply any data augmentation techniques. To quantify the recognition performance of the networks (using their vein pattern outputs), as well as the traditionally 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[7]. For comparing the binary feature maps, we adopted the approach by Miura et al. [16]. As the input maps are not registered to each other, the correlation between the input image and the reference one is calculated several times while shifting the reference image.

5. Results

Tables 1 and 2 display EER, FMR, and ZFMR results obtained using different manual training labels, for each network on the SDUMLA and UTFVP datasets respectively. As we can see in Table 2, U-net performs better than the other networks on the UTFVP dataset 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. As the values of the EER, FMR and ZFMR parameters demonstrate, the SegNet and RefineNet show rather similar performance on this dataset. Yet it is interesting to note that while RefineNet achieves the best performance when trained with just 5 training labels, the performance of SegNet improves as the number of training labels increases (at least up to 100 labels in terms of EER and FMR).

RefineNet performs far better than the U-net on SDUMLA dataset (more difficult dataset), and again exhibits the best performance with the lowest number of training labels (see Table 1). The performance of SegNet is similar to that of RefineNet also on this dataset, but again the best performance is observed using a high number of training labels (140 labels for both EER and FMR).

Cross-dataset training (last two lines of Tables 1 and 2) delivers interesting results. As it can be seen in the tables, UTFVP samples segmented with the networks trained with SDUMLA data consistently exhibit better recognition performance than trained with data from the same dataset. SegNet is able to take benefit from the full (i.e. 360 labels) training set, while this is not the case for the other networks (where recognition performance is actually decreased, in accordance with results for within-dataset training). For SDUMLA data segmented with networks trained on UTFVP data, we cannot observe improved results as compared to within-dataset training (see Table 1).

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
SDUMLA	5.17	13.70	15.46	13.34	25.60	27.22	10.41	22.22	26.80	5.28	13.96	26.60

Table 3: Classic algorithms' performance on the UTFVP and SDUMLA datasets.

In order to finally assess the recognition performance of the vein patterns generated by the different network training approaches considered, we compare the corresponding recognition performance to that of the classical algorithms as presented in Table 3. As it can be observed, almost all networks outperform the traditional algorithms on the SDUMLA dataset (considered as more difficult dataset), when trained with a sufficient number (100 to 140) of labels (recall that the high number of samples is only relevant for SegNet). U-net shows better performance than GF, RLT,

[7] http://bias.csr.unibo.it/fvc2004/.

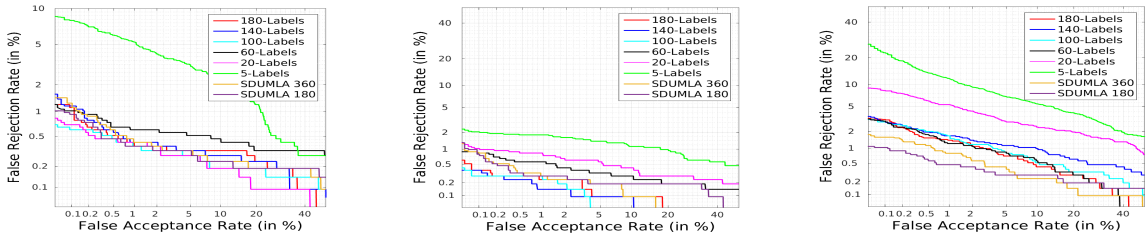

Figure 1: DET curves for the: U-net (left), RefineNet (middle), and SegNet(right) networks' performance on UTFVP.

Networks	U-net			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
180	0.50	1.11	2.03	0.28	0.37	0.92	1.34	3.14	4.44
140	0.55	1.15	1.80	0.27	0.37	0.46	1.53	2.91	4.67
100	0.46	0.60	1.75	0.28	0.27	1.06	1.38	2.87	7.31
60	0.64	1.01	1.89	0.55	0.83	1.38	1.20	2.96	5.04
20	0.46	0.69	2.17	0.83	1.01	1.52	3.52	8.33	13.00
5	3.52	8.00	11.20	1.67	2.03	2.87	6.47	22.36	39.76
SDUMLA 360	0.52	1.25	2.22	0.40	0.83	2.36	0.83	1.57	2.08
SDUMLA 180	0.46	0.92	1.43	0.32	0.78	1.75	0.55	1.01	1.80

Table 4: Networks' performance on UTFVP dataset, trained with labels generated by MC algorithm.

and DTFP algorithms on the UTFVP dataset, while RefineNet only outperforms the RLT algorithm when trained with a limited (5) number of labels. SegNet doesn't perform well on this dataset and falls behind all algorithms used.

Next, we look into results for training networks with automatically generated (MC) labels on UTFVP dataset as in Table 4. When looking at the table the overall impression is that (i) this approach improves the results in all cases as compared to manual labels, and (ii) the network 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 obtained by RefineNet, scoring: 0.27, 0.37, and 0.92 in ERR, FMR, and ZFMR parameters respectively, which clearly outperforms the best classical algorithms results (obtained by MC algorithm) in all terms (see Table 3). U-net outperforms the GF, RLT, and DTFP algorithms, and SegNet generally outperforms the RLT and DTFP algorithms when trained with adequate (60 or more) number of automatically generated labels from UTFVP dataset. As also reflected in related DET (Detection Error Trade-off) plots in Figure 1, cross-dataset training here seems not to deliver particular improvement (beyond the improvement achieved by training networks with automatically generated labels) for U-net and RefineNet. However, using this method, SegNet's performance gets significantly improved (i.e. scoring: 0.55, 1.01, and 1.80 in ERR, FMR, and ZFMR parameters receptively, when trained with 180 labels (see the Table 4)).

Finally, we look into the results for training networks with automatically generated labels on SDUMLA dataset as shown in Table 5. As it can be seen in the table, while

Networks	U-net			RefineNet			SegNet		
Labels	EER	FMR	ZFMR	EER	FMR	ZFMR	EER	FMR	ZFMR
180	6.15	14.49	22.45	2.77	6.11	8.98	5.73	13.14	16.34
140	6.29	13.42	19.44	2.73	7.12	9.49	6.47	17.31	25.04
100	6.62	14.76	20.92	2.45	6.38	10.64	5.50	12.63	20.46
60	6.29	13.37	18.42	2.59	5.87	8.75	5.84	13.40	18.93
20	6.27	13.19	16.80	2.82	6.25	8.65	6.39	13.47	17.40
5	7.59	17.22	24.76	3.57	7.59	8.98	9.77	23.14	29.49

Table 5: Networks' performance on SDUMLA dataset, trained with labels generated by MC algorithm.

results obtained by RefineNet (for all training label groups) outperform the best classical algorithms results (obtained by MC algorithm) in all terms (see Table 3) by far, yet U-net and SegNet can satisfy just a compatible level of performance (compared to the MC algorithm results). It is also interesting to note that the network's performance doesn't improve considerably by increasing the quantity of training labels. This behavior apparently accentuates the fact that while training networks with automatically generated labels improves the networks' performance, yet network architecture plays a key role in overall improvement achieved (specially when dealing with difficult datasets (i.e. SDUMLA)).

6. Discussion

When analyzing our results, the better performance of the networks trained with automatically generated labels is surprising. Thus, the first issue to be discussed is the quality/accuracy of our manual labels. Human annotators have been instructed to only annotate vein pixels without any ambiguity in order to avoid false positive annotations (see Figures 2b and 2e for examples). When looking at the examples, it is obvious that manual labels are restricted to rather large scale vessels, while fine grained vasculature is entirely missed/avoided. The correspondingly segmented vein patterns (e.g., the outputs of U-net trained with the manual labels (Figure 2g)) 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, MC labels (Figures 2c and 2f) and their corresponding outputs of CNNs trained with these labels (Figures 2h, and 2i) exhibit much more fine grained vasculature details, reflected in much better recognition accuracies.

The remarkable results of the cross-dataset training ex-

periments is the next issue that needs to be discussed. As it can be seen in Tables 1 and 2, while training networks with SDUMLA data can generally improve their performance on the UTFVP data, yet this is not true in the reverse order (training networks with UTFVP data and evaluation on the SDUMLA data). Having this fact in mind that SDUMLA is considered as a more difficult dataset compared to the UTFVP dataset (containing images which exhibit strong shading and low contrasts in the vein regions), we can conclude that training networks with comparably more difficult (lower quality) images forces the networks to learn the actual vein patterns with higher precision, and when evaluated on easier (higher quality) images (as those in the UTFVP dataset), networks can leverage a better performance.

As reflected in Tables 1 and 2, the performance of the networks is quite different using a changing number of manual training labels. RefineNet maintains a certain level of performance and seems to stay invariant with respect to the quantity of the training labels. It is interesting to note that this network can converge well even with a limited (5) number of training labels, and exhibits its optimal performance on both datasets with such a limited number of 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. automated labels). This seems to be owed to the multi-path refinement architecture used in this network, which exploits the information available all along the down-sampling process to enable high-resolution prediction, emphasizing on preservation of vein edges, and retaining the veins main structures.

U-net has been proven to excel in many bio-medical applications. The network architecture is designed to converge fast with a limited number of training labels. When trained with precise labels (i.e. automatically generated labels), this network is able to deal well with the ambiguous boundary issue between vein and non-vein regions. The network benefits from 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 on the two datasets with two different levels of difficulty underpins this fact clearly.

The SegNet network enjoys an stable (however not optimal) performance on both datasets, reflecting the network's ability to deal with low quality inputs (as the samples in the SDUMLA dataset). 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 indicts from the corresponding encoder feature maps to generate the up-sampled feature maps without learning. Nevertheless, the network seems to be more sensitive to the quantity of the training labels compared to the other two networks, and regardless of the quality of the input labels, requires a minimum of 60 to 100 training labels to converge to its optimal performance.

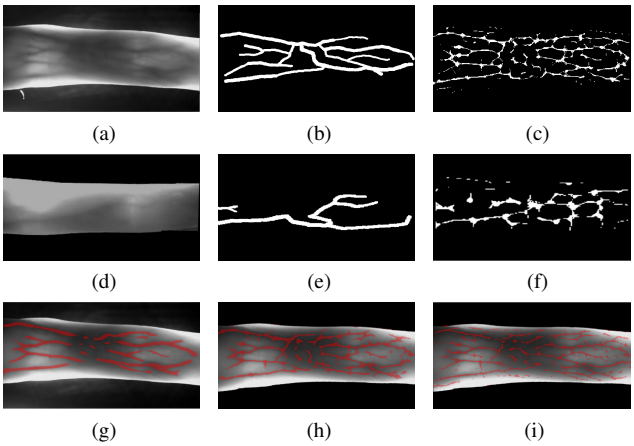

Figure 2: Sample UTFVP and SDUMLA finger-vein images (2a, 2d), their manuals (2b, 2e), and MC generated labels (2c, 2f), along with the segmentation outputs by U-net trained with manual (2g), and MC labels (2h), and a segmentation output by RefinNet trained with MC labels (2i).

7. Conclusion

In this work, we proposed a new model for finger-vein recognition using fully convolutional deep neural networks (FCN), focusing on direct segmentation of actual finger-vein patterns from the finger images, and using them as the binary finger-vein features for the recognition process. In this context, we trained three different FCN architectures, utilizing a varying number of manual and automatically generated labels to figure out efficient training and configuration settings for these networks using this type of data. We evaluated the respective recognition performance of the generated vein patterns in each case. We were able to show that the number of required training labels is highly network architecture dependent and have demonstrated that automatically generated labels can improve the networks' performance in terms of achieved recognition accuracy. It also turned out that cross-dataset training interestingly is able to improve within-dataset training in settings where the training data is more challenging than the evaluation data. The best configurations outperform traditional finger-vein recognition techniques significantly. In any case, we have demonstrated that utilizing automatically generated labels to train the networks is an effective approach and eventually eliminates the need for manual labels.

Acknowledgements

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

References

[1] G. R. Abhijit, S. Conjeti, N. Navab, and C. Wachinger. Fast mri whole brain segmentation with fully convolutional neural networks. In *Bildverarbeitung für die Medizin 2018*, pages 42–42. Springer, 2018. 3

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 3

[3] W. J. Da and Y. S. Huan. Driver identification using finger-vein patterns with radon transform and neural network. *Expert Systems with Applications*, 36(3):5793–5799, 2009. 1

[4] A. Dasgupta and S. Singh. 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. 2

[5] 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. 2

[6] M. Havaei, A. Davy, F. D. Warde, A. Biard, A. Courville, Y. Bengio, C. Pal, P. M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017. 3

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[8] Q. Huafeng and M. A. ElYacoubi. Deep representation-based feature extraction and recovering for finger-vein verification. *IEEE Transactions on Information Forensics and Security*, 12(8):1816–1829, 2017. 1, 2

[9] Y. Jinfeng and S. Yihua. Towards finger-vein image restoration and enhancement for finger-vein recognition. *Information Sciences*, 268:33–52, 2014. 1

[10] C. Kauba, E. Piciucco, E. Maiorana, P. Campisi, and A. Uhl. Advanced variants of feature level fusion for finger vein recognition. In *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'16)*, pages 1–12, Darmstadt, Germany, 2016. 1, 2

[11] S. I. Khalid, S. Radzi, F. Gong, N. Mustafa, Y. C. Wong, and M. M. ibrahim. User identification system based on finger-vein patterns using convolutional neural network. *ARPN Journal of Engineering and Applied Sciences*, 11(5):3316–3319, 2016. 2

[12] A. Kumar and Y. Zhou. Human identification using finger images. *IEEE Transactions on image processing*, 21(4):2228–2244, 2012. 2

[13] 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. 3

[14] Y. Lu, Y. Gongping, Y. Yilong, and Z. Lizhen. A survey of finger vein recognition. In Z. Sun, S. Shan, H. Sang, J. Zhou, Y. Wang, and W. Yuan, editors, *Lecture notes in Chinese Conference on Biometric Recognition*, pages 234–243. Springer International Publishing, 2014. 2

[15] 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, 2004. 2

[16] 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. 1, 2, 4

[17] C. Neumann, K.-D. Tnnies, and R. Pohle-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. 2

[18] S. A. RADZI, M. K. HANI, and R. Bakhteri. Finger-vein biometric identification using convolutional neural network. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(3):1863–1878, 2016. 2

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 2

[20] K. S. Sadanandan, P. Ranefall, S. L. Guyader, and C. Whlby. Automated training of deep convolutional neural networks for cell segmentation. *Scientific Reports (Nature Publisher Group)*, 7:1, 2017. 3

[21] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. 2

[22] 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. 3

[23] 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. 3

[24] S. Wonseok, T. Kim, K. H. Chan, C. J. Hwan, K. H. Joong, and L. S. Rae. A finger-vein verification system using mean curvature. *Pattern Recognition Letters*, 32(11):1541–1547, 2011. 1, 2

[25] 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. 2

[26] L. Xiaoxia, H. Di, and W. Yunhong. Comparative study of deep learning methods on dorsal hand vein recognition. In *Chinese Conference on Biometric Recognition*, pages 296–306. Springer, 2016. 2

[27] C. Xie and A. Kumar. Finger vein identification using convolutional neural network and supervised discrete hashing. *Pattern Recognition Letters*, 2018. 2

[28] G. Yang, X. Xi, and Y. Yin. Finger vein recognition based on a personalized best bit map. *Sensors*, 12:1738–1757, 2012. 1

[29] 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. 2

[30] H. Zou, B. Zhang, Z. Tao, and X. Wang. A finger vein identification method based on template matching. *Journal of Physics*, Conference Series 680:012001, 2016. 1

[31] K. Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994. 3