Chapter 8
Improved CNN-Segmentation-Based Finger Vein Recognition Using Automatically Generated and Fused Training Labels

Ehsaneddin Jalilian and Andreas Uhl

Abstract We utilise segmentation-oriented CNNs to extract vein patterns from near-infrared finger imagery and use them as the actual vein features in biometric finger vein recognition. As the process to manually generate ground-truth labels required to train the networks is extremely time-consuming and error prone, we propose several models to automatically generate training data, eliminating the needs for manually annotated labels. Furthermore, we investigate label fusion between such labels and manually generated labels. Based on our experiments, the proposed methods are also able to improve the recognition performance of CNN-network-based feature extraction up to different extents.

Keywords Finger-Vein recognition · Finger-Vein segmentation · Convolutional neutral networks · Fused label training · Automated label training

8.1 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 process is considered to offer significant advantages compared to classical biometric modalities (e.g. fingerprint, iris and face recognition). For example, finger vein patterns can be captured in a touchless and non-invasive manner, are not influenced by finger surface conditions, can only be captured when the subject is alive and cannot easily get forged.

While many finger vein recognition techniques have been proposed in recent years and commercial products are readily available (and are even used to authenticate

E. Jalilian (✉) · A. Uhl
Department of Computer Sciences, University of Salzburg,
Jakob Haringer Str. 2, 5020 Salzburg, Austria
e-mail: ejalilian@cs.sbg.ac.at

A. Uhl
e-mail: uhl@cs.sbg.ac.at

A. Uhl et al. (eds.), *Handbook of Vascular Biometrics*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-3-030-27731-4_8

financial transactions in ATMs or home banking), 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. Poorly designed scanner devices, close distance between finger and the camera (causing optical blurring), poor NIR lighting, varying thickness of fingers, ambient external illumination [34], varying environmental temperature [25] and light scattering [16] 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 can hardly be described by a mathematical model. Therefore, proposing a comprehensive algorithmic solution to extract the actual vein patterns from the NIR finger images is not easy. Nevertheless, even the manual annotation of actual vein patterns in such ambiguous areas (required as ground truth for learning-based methods (i.e. segmentation CNN networks) is extremely difficult and time-consuming and therefore an error-prone process.

In this chapter, we employ three different CNN architectures designed for segmentation to extract finger vein patterns from NIR finger imagery and use the extracted features for the recognition process. Furthermore, with the aim of eliminating the need for manually annotated labels and eventually also improving the networks' feature-extraction capability, we investigate several automatic label generating techniques, as well as label fusion methods, to generate more precise labels to train the networks. After training the networks with these labels and the generation of corresponding vein patterns, we evaluate the recognition performance in terms of receiver operating characteristics and relate the results to those obtained by classical finger vein feature-extraction techniques.

The chapter is structured as follows: Sect. 8.2 describes related works and the state of the art in finger vein recognition and Sect. 8.3 describes vein pattern extraction using three different segmentation CNN architectures as used in this work. In Sect. 8.4, we explain different training label generation, and also fusion techniques used in this work. In Sect. 8.5, we describe the experimental framework used in experimentation. Section 8.6 presents the results, while Sect. 8.7 discusses the obtained results, and finally Sect. 8.8 concludes the chapter.

8.2 Related Works

For a general overview of finger vein recognition techniques, please refer to, e.g. [23] and also the book's introduction chapter. In the first subsection of this section, we briefly describe three state-of-the-art schemes also used in experimentation as reference recognition techniques and used to automatically generate labels required for CNN training. The second subsection reviews the previous employment of CNNs in finger vein recognition, while the third subsection discusses works done in other fields to automatically generate data for CNN training.

8.2.1 Classical Finger Vein Recognition Techniques

Classical finger vein recognition techniques 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. "Repeated Line Tracking" (RLT [24], being of feature-based type) 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, where the depth of valley at each position indicates the tracking direction. If no line is detected, a new tracking trail is started. 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 binarised using a threshold.

Profile-based approaches consider the cross-sectional contour of a vein pattern which shows a valley shape. "Maximum Curvature" (MC [25], being of profile-based type) traces only the centre lines of the veins and is insensitive to varying vein width. To extract the centre positions, first the local maximum curvature in the cross-sectional profiles of vein images is determined, using the first and second derivatives. Next, each profile is segmented as being concave or convex, where only local maxima in concave profiles are specified as valid centre positions. Then according to width and curvature of the vein region, a score is assigned to each centre position, and recorded in a matrix called locus space. The centre positions of the veins are connected using a filtering operation subsequently. Eventually, the matrix is binarised using the median of the locus space.

Another profile-based method, exploiting the line-like shape of veins in a predefined neighbourhood region is termed "Gabor Filter" (GF [19]). 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. The final feature image is constructed by fusing all the single images obtained in the previous step, and then morphological operations are used to clear the noise from the image.

Of course, 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 [8]), Region Growth [12] and Principal Component Analysis (PCA [15]). However, also other techniques specifically tailored to the problem have been suggested like using vessel-crossings in a minutiae-type manner [3] or the Principal Curvature [17] approach.

8.2.2 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. [39] 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, as used in this work.

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 reduced-complexity (a four-layered CNN) classifier, with fused convolutional-subsampling architecture for finger vein recognition [35]. Itqan et al. performed finger vein recognition using a CNN classifier of similar structure [29], and Das et al. [5] correspondingly proposed a CNN classifier for finger vein identification. This approach, however, has significant drawbacks in case new users have to be enrolled as the networks have to be retrained, which is not practical.

Hong et al. [13] used a more sensible approach, employing fine-tuned pre-trained models of VGG-16, VGG-19 and VGG-face classifiers, which are based on determining whether a pair of input finger vein images belongs to the same class (i.e. subject) or not. Likewise, Xie and Kumar [40] used several known CCN models (namely, light CNN (LCNN) [38], LCNN with triplet similarity loss function [33], 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, utilising raw images, the system possesses a potential security threat.

Qin and El-Yacoubi [11] applied a two-step procedure to extract the finger vein patterns from NIR finger images. As the first step, they used a CNN classifier to compute the probability of patch centre pixels to belong to vein patterns, one by one, and labelled 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 (the FCN network) to recover the missing pixels, additional processing time is added to the feature-extraction process.

8.2.3 Automated Generation of CNN Training Data

Large amounts of high-quality annotated samples, or ground-truth data, are typically required for CNN training. However, data labelling is an expensive and time-consuming task, especially due to the significant human effort involved. The problem even gets more tedious in case the annotators have to deal with ambiguous images, where clear separation between target regions and the background data is very difficult, as it is the case in many biomedical applications. Given these facts together with the difficulty to persuade medical experts to annotate the required data volumes, 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. In [32], classical techniques were used to segment cells stained with fluorescent markers. The resulting segmentation masks were used as ground-truth labels together with the corresponding bright-field image data to train a CNN. In [14], Canny edge detection was applied to ultrasound images to generate the ground-truth labels required to train a CNN for segmentation of musculo-skeletal ultrasound images. In [9], a part of the ground-truth labels required to train a CNN for brain tumour segmentation was generated by a voted average of segmentation results of top performing classical segmentation algorithms in this field. In [31], a fully convolutional neural network is pre-trained on a large dataset containing ground-truth labels created by existing segmentation tools for brain segmentation, and subsequently fine-tuned with a small dataset containing human expert annotations. In [11], authors used several algorithms to generate 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 or in fusion with manual labels in network training process.

8.3 Finger Vein Pattern Extraction Using CNNs

The first computer vision tasks for which initial CNN architectures were developed include classification [18], bounding box object detection [20] and key point prediction [2]. More recently, CNN architectures have been developed enabling semantic segmentation, in which each pixel is labelled separately with the class of its enclosing object or region. The first techniques, classifying the centre 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 [22]. 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 [6], an FCN is used for segmentation of retinal blood vessels in fundus imagery, or in [26] 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, second the quality of images is by far better than the NIR finger vein images, and third 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. We selected the networks based on diffrent architectural functionalities built in each network, so that we can evaluate endurance of such functionalties in case of finger vein segmentation. The first network architecture used is the U-net by Ronneberger et al. [30]. 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), 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 energy function is computed by a soft-max over the final feature map, combined with the cross-entropy loss function. The cross-entropy then penalises, at each position, the deviation of soft-max ($M_{\lambda(x)}(x)$) from one (1.00) as follows:

$$\varepsilon = \sum_{k'=1} \log(M_{\lambda(x)}(x)), \tag{8.1}$$

where $\lambda : \Omega \rightarrow \{1, \ldots, K\}$ is the true label of each pixel, at the position $x \in \Omega$, with $\Omega \subset \mathbb{Z}^2$. The networks soft-max 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 has a large number of feature channels, which allow it to propagate context information to higher resolution layers, and offers end-to-end training with limited number of training samples. The network implementation[1] was realised in the TensorFlow framework using the Keras library.

The second network architecture we used to extract the finger vein patterns is RefineNet [21]. RefineNet is a multi-path refinement network, which employs a four-cascaded architecture with four RefineNet units, each of which directly connects to the output of one Residual net [10] 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 network has multi-path refinement architecture that explicitly exploits all the information available along the down-sampling process to enable high-resolution prediction using long-range residual connections. In this way, the deeper layers that capture high-level semantic features can be directly refined using fine-grained features from earlier convolutions. The network also uses

[1] https://github.com/orobix/retina-unet.

Table 8.1 Networks' training parameters

Network	U-net	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

a chained residual pooling mechanism to capture rich background context in an efficient manner. The implementation[2] of this network was also realised in the TensorFlow framework 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. [1], named SegNet. However, we redesigned the softmax layer to segment only the vein pattern. The whole network architecture is formed by an encoder network, and the corresponding decoder network. The network's encoder architecture is organised in four stocks, containing a set of blocks. Each block comprises a convolutional layer, a batch normalisation layer, a ReLU layer and a Pooling layer with kernel size of 2 × 2 and stride 2. The corresponding decoder architecture, likewise, is organised 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 labelling, in this network, the convolutional kernel size is set to 7 × 7. A key functionality used in this network is "indices pooling". While several layers of max-pooling and sub-sampling can achieve more translation invariance for robust classification correspondingly there is a loss of spatial resolution of the feature maps. Therefore, it is necessary to capture and store boundary information in the encoder feature maps before sub-sampling is performed. The network utilises a memory-efficient technique for this purpose, storing only the max-pooling indices, i.e. the locations of the maximum feature value in each pooling window is memorised for each encoder feature map. The decoder network up-samples the input feature map(s) using the memorised max-pooling indices from the corresponding encoder feature map(s). The decoder network ends up to a softmax layer which generates the final segmentation map. The network implementation[3] was realised in the Caffe deep learning framework.

Table 8.1 summarises the training parameters (which turned out to deliver best results) we used to train each network in our experiments.

[2] https://github.com/eragonruan/refinenet-image-segmentation.

[3] http://mi.eng.cam.ac.uk/projects/segnet/tutorial.html.

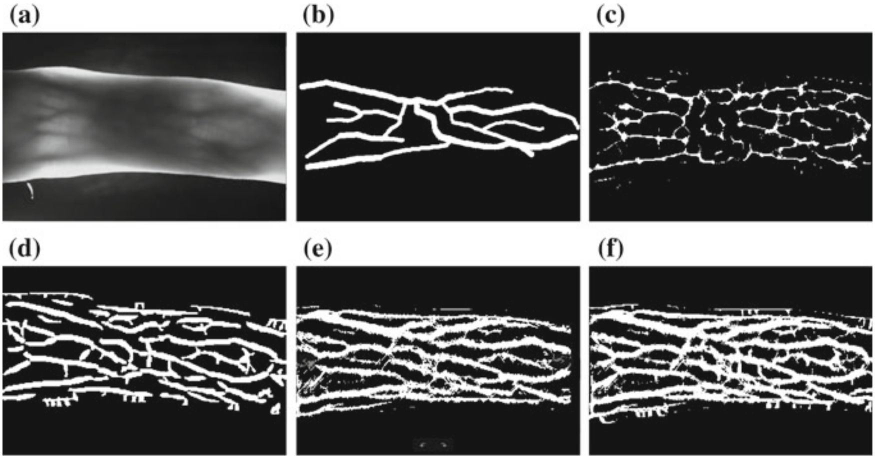

Fig. 8.1 A sample finger vein image (**a**), and its corresponding manual (**b**), MC (**c**), GF (**d**), RLT (**e**), and fused (MC-GF-RLT) (**f**) labels respectively

8.4 Training Label Generation and Setups

From the total samples available in our database (check Sect. 8.5 for database details), we have 388 NIR finger images (covering all subjects in the database) manually annotated (i.e. vein versus non-vein pixels, see Fig. 8.1b for an example) available for training the CNNs. To enable a fair comparison, we generated the same number of corresponding automated labels (also using the identical images), utilising each of 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 Sect. 8.2. For MC and RLT, we utilised the MATLAB implementation of B. T. Ton,[4] and for GF we used a custom implementation as used in [28][5] (see Fig. 8.1c, d and e for corresponding example using each algorithm).

As one of the main objectives of this work, we investigated several training label scenarios, aiming to improve the networks' feature-extraction capabilities, and also eventually eliminating the need for the manually annotated labels. In this way, first we used automatically generated labels adding only 40 pcs of corresponding manual labels to train the networks in each training session. We termed this approach as "automated" training. Next we considered to train the network using automatically generated labels jointly with equivalent number of (i) corresponding manual labels, and also (ii) other (corresponding) automatically generated labels to train the networks. We termed this approach as "joint" training. In particular, in this approach, in each training session, instead of using just one type of label (i.e. manual, MC, GF or

[4] Available on MATLAB Central.

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

Table 8.2 Runtime per input volume for each network

Network	U-net	RefineNet	SegNet
Input volume size	584 × 565	584 × 565	360 × 480
Processing time (s)	3.164	0.138	0.0398

RLT) we used combinations of two or more types of the labels to train the networks. We kept the input data shuffling on during the training process to preserve uniform distribution of training samples.

In an alternative approach, we considered to fuse in between different types of the labels available, to generate single training labels. For this purpose, we utilised the "STAPLE" (Simultaneous Truth And Performance Level Estimation [37]) algorithm to fuse between the binary labels. STAPLE is an algorithm developed for performance analysis of image segmentation approaches in medical imaging based on expectation–maximisation. It takes in a collection of labels and computes a probabilistic estimate of the true labels and a measure of the performance level represented by each label. In our work, we applied STAPLE to fuse between (i) automatically generated labels, (corresponding) manual labels and also (ii) different types of automatically generated labels (see Fig. 8.1f for an example). We termed this approach as "fusion" training.

8.5 Experimental Framework

Database: We used the UTFVP database [36],[6] acquired by the University of Twente with a custom sensor, in our experiments. The UTFVP database contains 1440 finger vein images (with resolution of 672 × 380 pixels), collected from 60 volunteers. The images were captured in two identical sessions with an average time lapse of 15 days. For each volunteer, the vein pattern of the index, ring and middle finger of both hands has been collected twice at each session (each individual finger has been captured four times in total). The percentage of male volunteers was 73% and the percentage of right-handed volunteers was 87%. The width of the visible veins ranges from 4 to 20 pixels which corresponds to vein widths of approximately 0.3–1.6 mm. These vein widths are approximate numbers because the pixel density was determined assuming a flat surface. We resized the images to the corresponding networks' input volume, using bicubic interpolation method, as specified in Table 8.2 (see Fig. 8.1a for a sample of finger vein images in the database).

Network training and finger vein recognition evaluations: We trained each network with different label groups (manual, automated, fused and joint) using a cross-fold training method with disjoint training and testing sets. For this purpose, first we

[6] Available at:https://scs.ewi.utwente.nl/downloads/.

partitioned the whole training set (388 labels) for each label group into two parts (194 labels each). Next, we trained the networks with the first training part, and tested the networks on the remaining samples in the database (1246 samples). Then we did the training with the second part this time, and tested the networks on the remaining samples in the database. Doing so, we tested the networks on all samples in the database without overlapping training and testing sets.

As we wanted the comparison to concentrate on the quality of the pure training labels, we deliberately did not apply any data augmentation technique. Also, while a different number of training samples were used by the network developers (e.g. 35, 376, 1449 samples for U-Net, SegNet and RefineNet, respectively), we selected this number of training samples (194 samples) based on our experimental trials and also the availability of labels. Moreover, as the NIR finger images are acquired under standardised conditions in the sensor, no additional normalisation techniques have been applied. The RefineNet and the SegNet already generate their final outputs as binarized maps, but as the final output of the U-net is in form of probability maps, we binarised these maps using a grey-level threshold function based on the Otsu's algorithm [27]. The algorithm chooses the best threshold to minimise the intra-class variance of the black and white pixels while maximising inter-class scatter. Table 8.2 shows the segmentation runtime per input volume for each network, using TITAN-X (Pascal) GPUs. It is interesting to note that U-net is relatively much slower than the other two networks, which is primarily due to the overlapping-tile strategy, and also the large number of feature channels used in this network.

Finger vein Recognition Evaluations: 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 behaviour is evaluated. In particular, the equal error rate EER as well as the FMR1000 (FMR) and the ZeroFMR (ZFMR) are used. For their respective calculation, we followed the test protocol of the FVC2004 [4]. All possible genuine comparisons are performed, i.e. each sample is compared against all remaining samples of the same finger/hand while no symmetric comparisons are performed. So, in total 2160 genuine comparisons are carried out. For the impostor comparisons, only the first sample of each finger/hand is compared against the first sample of all remaining fingers/hands, while no symmetric comparison is performed, resulting in total 64520 impostor comparisons. For matching the binary output features, we adopted the approach by Miura et al. [25], 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 [7] 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.

$$N_m(s, t) = \sum_{y=0}^{h-2c_h-1} \sum_{x=0}^{w-2c_w-1} I(s + x, t + y) R(c_w + x, c_h + y), \tag{8.2}$$

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

$$score = \frac{N_{m_{max}}}{\sum_{y=t_0}^{t_0+h-2c_h-1} \sum_{x=s_0}^{s_0+w-2c_w-1} I(x, y) + \sum_{y=c_h}^{h-2c_h-1} \sum_{x=c_w}^{w-2c_w-1} R(x, y)}, \tag{8.3}$$

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

8.6 Results

Table 8.3 and Fig. 8.2 display the results for training the networks using the manual, automated and joint training methods, providing EER, FMR and ZFMR as well as DET (Detection Error Trade-off) plots, respectively. The baseline result for each network is given using only manually annotated labels ("CNN-Manual"). We observe that the baseline is already quite different for the three networks, i.e. U-Net is superior to RefineNet, while SegNet clearly is worst among the three.

Next we look into results for training networks with the automatically generated labels adding just 40 pcs of corresponding manual labels (automated training). The overall impression is that (i) this approach can improve the results significantly in many cases and (ii) results again differ depending on the network considered. As it can be seen in the table, recognition performance gets considerably improved

Table 8.3 Networks performance, trained with manual, automated and joint labels

Network	U-net			RefineNet			SegNet		
Measures	EER (%)	FMR (%)	ZFMR (%)	EER (%)	FMR (%)	ZFMR (%)	EER (%)	FMR (%)	ZFMR (%)
CNN-Manual	0.877	1.851	5.185	2.735	5.833	11.851	2.917	6.759	12.638
CNN-MC	0.322	0.601	0.925	0.280	0.370	1.574	1.433	2.453	5.648
CNN-GF	0.793	2.731	3.796	2.133	5.046	8.750	1.204	2.685	5.555
CNN-RLT	2.091	11.620	24.861	1.101	2.824	3.750	1.279	3.009	7.592
CNN-Manual-MC	0.924	3.055	3.379	0.933	2.361	17.870	1.531	2.500	3.240
CNN-Manual-GF	0.648	0.925	2.916	2.039	3.564	5.416	2.595	4.675	8.750
CNN-Manual-RLT	1.433	8.518	17.500	1.703	4.027	5.878	1.433	2.361	3.194
CNN-MC-GF-RLT	1.713	15.046	23.750	0.877	2.037	3.055	0.929	1.805	4.027

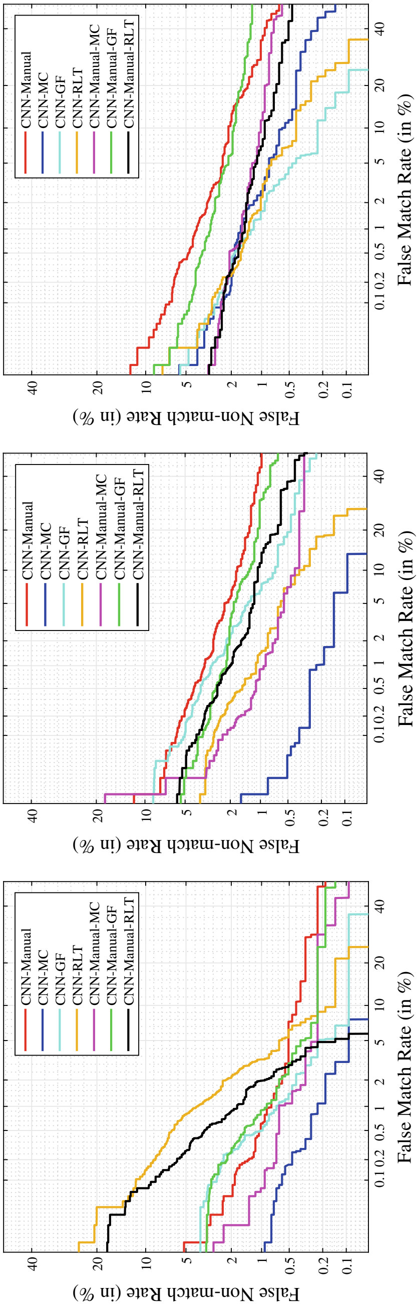

Fig. 8.2 DET curves for the: U-net (left), RefineNet (middle), and SegNet (right)

Table 8.4 Classical algorithms performance

Method	Maximum curvature			Gabor filter			Repeated line tracking		
Measures	EER (%)	FMR (%)	ZFMR (%)	EER (%)	FMR (%)	ZFMR (%)	EER (%)	FMR (%)	ZFMR (%)
Scores	0.4155	0.555	1.296	1.111	2.453	4.120	2.175	5.879	9.351

when training networks with labels generated by the MC algorithm for all networks (compared to training networks with manual labels). As well, labels generated by the GF algorithm improve the networks' performance (especially SegNet's), but in a fewer degree. When trained with the labels generated by the RLT algorithm, SegNet and especially RefineNet recognition results are also clearly improved, while U-net results are significantly deteriorated. Obviously, the different network architectures react very differently when trained with labels of different origin. It is also interesting to directly consider the recognition accuracy of the vein patterns generated by the classical algorithms (recognition results shown in Table 8.4): The very clear ranking is that MC is best, GF is ranked second while RLT exhibits the worst recognition performance. This ranking is not consistently reflected by the vein patterns generated by the networks when trained with these classical algorithms' labels. For the U-net, the ranking is MC, GF, RLT (thus reflecting the "original" ranking), while for the RefineNet the ranking is MC, RLT, GF, and the SegNet's ranking is GF, RLT and MC. Training networks jointly with manual labels and equivalent number of labels generated by the classical algorithms (joint approach) again result in different performance behaviours. As compared to training with manual labels only, results are improved in all cases for SegNet, while for RefineNet and U-net we observe both result improvements as well as degradations, respectively. As compared to the automated training method, we observe both result improvements and degradations for all CNNs. There is a tendency that for those automatically generated labels, which perform well when used just with 40 pcs of manual label in training, we typically do not observe improvements when used jointly with equivalent number of manual labels. Considering training the networks jointly with MC, GF and RLT labels at once (joint method), we get results never improving the best result obtained when training with a single technique (results shown only in Table 8.3) for U-net. However, using such a training configuration, we can improve the performance of SegNet and RefineNet even further than the best result obtained when training with a single technique. Thus, this training configuration can be recommended only for this network.

In order to assess the recognition performance of the vein patterns generated by the different network training approaches presented in Table 8.3, we compare the corresponding recognition performance to that of the vein patterns as generated with classical algorithms directly in terms of DET as presented in Figs. 8.3, 8.4 and 8.5. We conduct this comparison for different CNNs and the automated label generation techniques separately to facilitate a clear comparison. For MC labels (left graphs), we observe that CNN-generated vein patterns considerably enhance the recognition

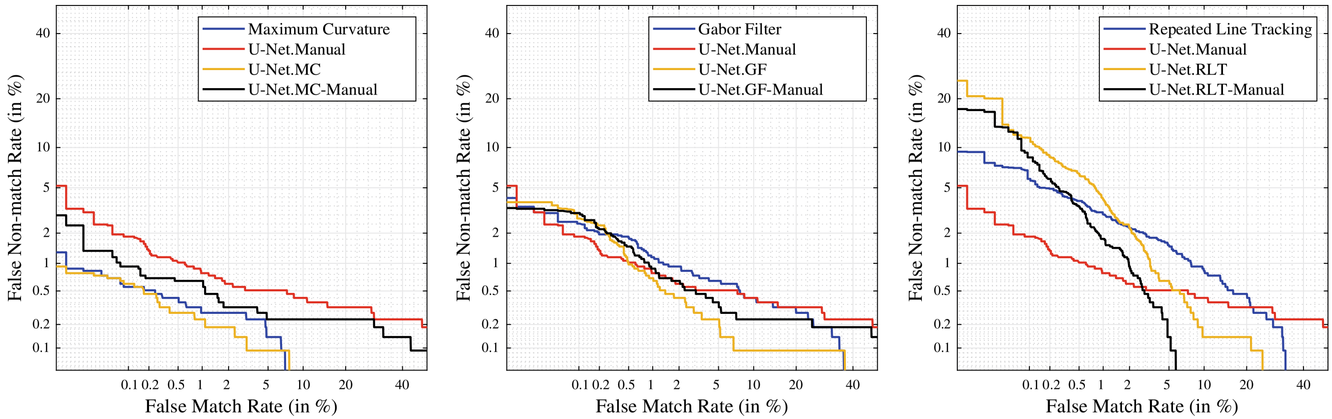

Fig. 8.3 DET curves for the U-net trained with labels generated by: MC (left), GF (middle) and RLT (right)

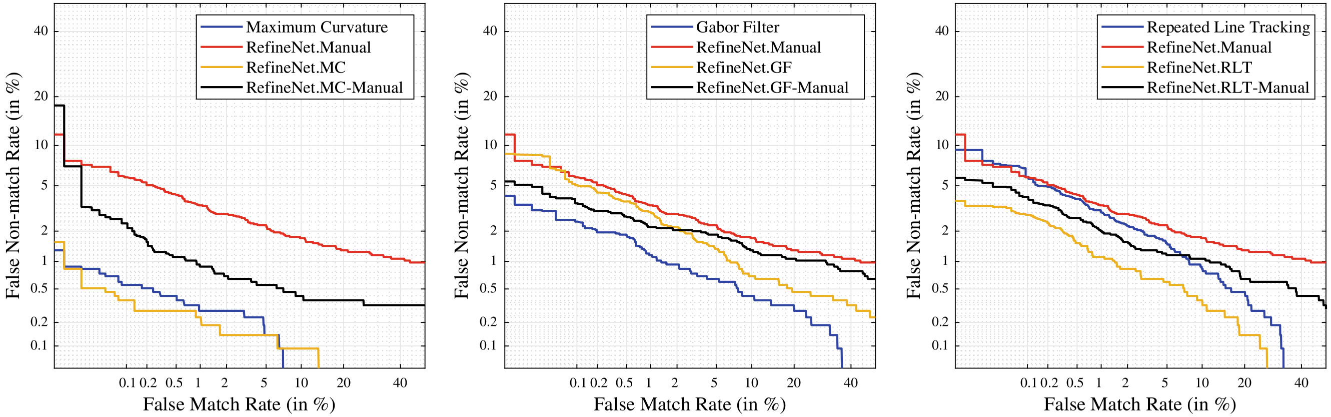

Fig. 8.4 DET curves for the RefineNet trained with labels generated by: MC (left), GF (middle) and RLT (right)

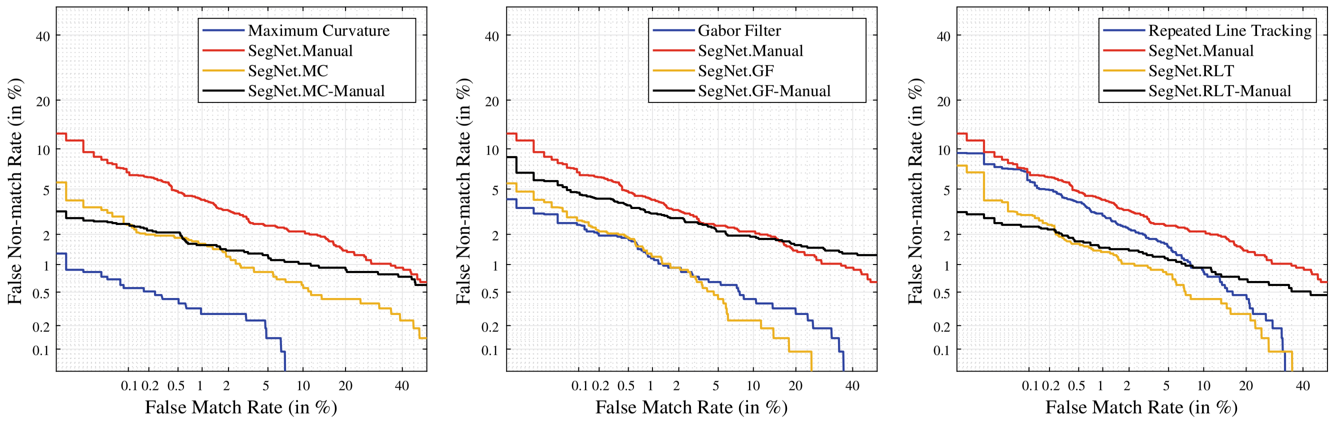

Fig. 8.5 DET curves for the SegNet trained with labels generated by: MC (left), GF (middle) and RLT (right)

Table 8.5 Networks performance, trained with fused labels

Network	U-net			RefineNet			SegNet		
Measures	EER(%)	FMR(%)	ZFMR(%)	EER(%)	FMR(%)	ZFMR(%)	EER(%)	FMR(%)	ZFMR(%)
CNN-Manual&MC	0.462	0.787	1.388	2.128	3.564	4.444	1.573	2.500	3.564
CNN-Manual&GF	0.602	1.342	1.990	2.544	3.796	5.138	0.686	1.388	2.916
CNN-Manual&RLT	3.235	21.620	36.666	1.437	2.870	5.185	2.035	8.611	27.175
CNN-MC&GF&RLT	5.639	45.601	69.722	1.666	5.462	9.120	1.624	5.879	13.703

performance of RefineNet and U-net, respectively, as compared to the classically generated patterns. The most interesting results here are obtained by RefineNet, which clearly outperforms the best classical algorithms results (obtained by MC algorithm) in all terms.

For GF labels (middle graphs), we see improvements using automated and also joint training method for the U-net, while for the SegNet and RefineNet no specific improvement is visible. Finally, for the RLT labels (right graphs), all the CNN-based vein patterns outperform the classical ones, whether using automated or joint training method.

As a further training configuration, we trained the networks considering several training label fusion scenarios. Table 8.5 demonstrates the results for this experiment. Comparing the results to the corresponding results obtained using the joint training method, interestingly we can see that training the networks with the labels generated by fusing all types of automatically generated labels (last line of the table) not only doesn't improve the networks' performance but also undermines them severely (especially in case of U-net). Furthermore, training the networks with labels which are result of fusion between the manual and automatically generated labels by MC or GF algorithms improves the performance of RefineNet, U-net and rather SegNet. We also observe that while training the networks with the labels generated by fusing manual and automatically generated labels by RLT algorithm improves the results obtained by RefineNet, yet this training label configuration degrades the corresponding results for U-net and SegNet.

8.7 Discussion

When analysing our results, the poor performance of the networks trained with manual labels is surprising. Thus, the first issue to be discussed is the quality/accuracy of our manual labels (see Fig. 8.6a for an example). Human annotators have been instructed to only annotate vein pixels without any ambiguity in order to avoid false-positive annotations. When looking at the example, 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 (i.e. the outputs of CNNs trained with the manual labels, 8.6e, i and m in the example figures) 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 and their corresponding outputs of CNNs trained with these labels (8.6f, j and n in the figures) exhibit much more fine-grained vasculature details, reflected in much better recognition accuracy. RLT vein patterns, on the other hand, tend to over-segment and obviously also contain many false-positive vein pixels (e.g. at the border between finger texture and background, 8.6d in the figures). Consequently, this also applies to outputs of CNNs trained with RLT labels (see 8.6h, l and p in the figures).

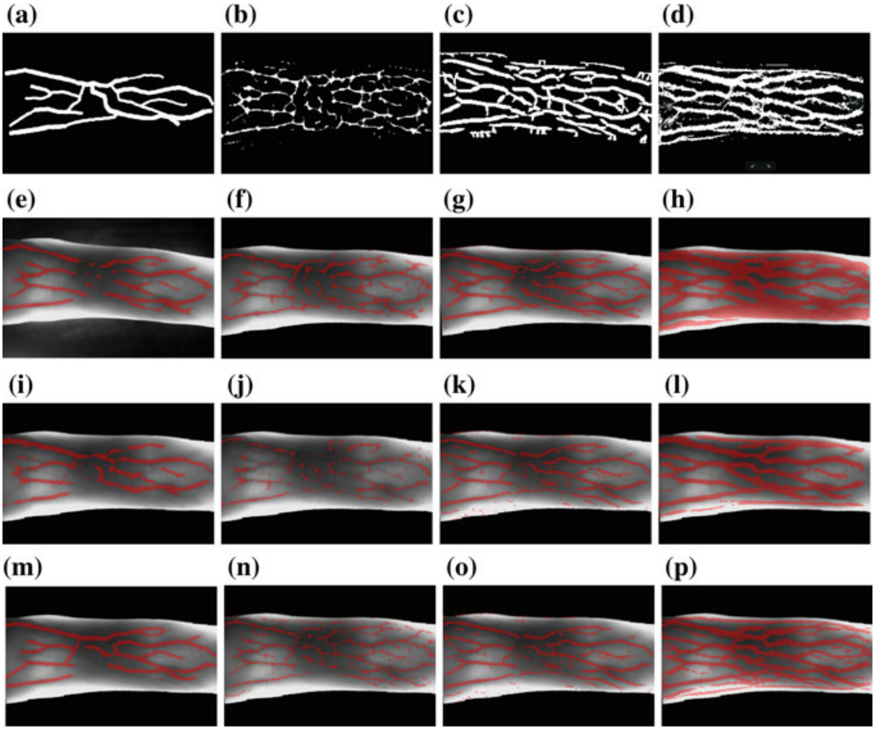

Fig. 8.6 A sample of manual (**a**), MC (**b**), GF (**c**), RLT (**d**) labels, and thier corresponding segmentation results when used to train: U-net (**e**, **f**, **g**, **h**), SegNet (**i**, **j**, **k**, **l**) and RefineNet (**m**, **n**, **o**, **p**), respectively

We have observed that in many configurations, utilising automatically generated labels is beneficial for the recognition performance of the vein patterns generated by CNN-based segmentation (i.e. training U-net and RefineNet with labels automatically generated by MC algorithm). However, there is significant interplay of the nature of the used labels (in particular, their accuracy with respect to the relation between false-positive and false-negative vein pixels) and the network architecture. Interestingly, it is not the case that training with the vein patterns generated by the classical algorithm exhibiting the best recognition performance (i.e. MC) does lead to the best performing CNN segmentations for all networks. As observed before, the ranking among the schemes is not maintained after network training, which indicates a distinct capability of the networks to cope with false positives (highly present in RLT labels for example) and false negatives (obviously highly present in manual labels).

For example, while the performance of SegNet and especially RefineNet improve when trained with labels generated by the RLT algorithm, the U-net's performance clearly degrades when trained with identical labels. In general, the RefineNet exhibits comparably better segmentation capabilities, especially when trained with MC labels.

Basically, this network has a multi-path refinement architecture, which exploits the information available along the down-sampling process to enable high-resolution prediction, emphasising on preservation of edges and boundaries. Consequently, introducing further vein pixels to the network by automatically generated or fused labels seems to improve the actual network vein pattern extraction capabilities significantly (compared to the other two networks).

The network architecture of the U-net has been proven to excel in many biomedical applications, and when trained with precise vein labels, it is able to deal well with the ambiguous boundary issue between vein and non-vein regions in finger vein images. This capability is mainly due to the large number of feature channels built in the network architecture, which allow for propagating key context information to higher resolution layers. However, due to the high sensitivity of the network, imprecise labels can equally degrade the network's performance seriously. A simple comparison of the network's performance when trained with labels generated by the MC algorithm (more precise labels, some false negatives) with when trained with labels generated by the RLT algorithm (less precise labels with more false positives) underpins this behaviour, as also reflected in Fig. 8.6.

The SegNet network is unable to extract vein patterns from the NIR finger images well when trained with manual labels. However, the network's performance consistently improves by introducing further vein pixel labels using automatically generated or fused labels. This network shows good ability to exclude the incorrectly labelled vein pixels (i.e. false positives) during the training process, as imprecise labels (i.e. those generated by RLT algorithm) do not degrade the network's performance significantly. 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.

Considering the applied training approaches (fusion versus joint), as the results show, in majority of cases (i.e. training networks jointly with labels automatically generated by all classical algorithms), the latter approach (joint) performs superior to the fusion technique, and results more in improvement of networks' performance rather than degradation. However, the extent of such improvement/degradation again is subject to the interplay of the nature of the used labels and the network architecture. Conflicting results obtained by Unet and RefineNet when trained jointly with manual and automatically generated labels by MC or GF algorithms indicates this fact clearly. Therefore, we can conclude that selection of the proper network training approach is highly subject to these two key factors (nature of the used labels and the network architecture).

8.8 Conclusion

In this work, we proposed a new model for finger vein recognition using fully convolutional neural networks (FCN), focusing on direct extraction of actual finger vein patterns from the finger images by segmentation, and using them as the actual finger vein features for the recognition process. In this context, we trained three different FCN architectures, utilising different combinations of manual and automatically generated labels, and evaluated the respective recognition performance of the generated vein patterns in each case. We showed that automatically generated labels (whether used solely or fused with manual labels) can improve the network's performance in terms of achieved recognition accuracy. It also turned out that these improvements are highly dependent on the interplay between properties of the used labels and the network architecture. In any case, we have demonstrated that utilising automatically generated labels to train the networks eliminates the need for manual labels, whose generation is an extremely cumbersome, difficult and error-prone process.

In future works, we will change the way how to employ and combine additionally available label data. In particular, we will assess the strategy to pre-train with manual labels (as they do not contain false-positive vein pixels) and refine networks with automatically generated ones (as these do contain more fine-grained vascular details). Also, an evaluation of cross-database (using training data from a different vein sensor) and 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 also more detailed and 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. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
2. Ce L, Jenny Y, Antonio T (2011) Sift flow: dense correspondence across scenes and its applications. IEEE Trans Pattern Anal Mach Intell 33(5):978–994
3. Cheng-Bo Y, Hua-Feng Q, Yan-Zhe C, Xiao-Qian H (2009) Finger-vein image recognition combining modified hausdorff distance with minutiae feature matching. Interdiscip Sci Comput Life Sci 1(4):280–289
4. Dario M, Davide M, Raffaele C, Jim W, Anil J Fvc (2004) Third fingerprint verification competition. In: Lecture notes in biometric authentication. Springer, pp 1–7
5. Das R, Piciucco E, Maiorana E, Campisi P (2018) Convolutional neural network for finger-vein-based biometric identification. IEEE Tran Inf Forensics Secur 1–1
6. Dasgupta A, Singh S (2017) 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). IEEE, pp 248–251
7. Eui-Chul L, Hyeon-Chang L, Kang-Ryoung P (2009) Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. Int J Imaging Syst Technol 19(3):179–186
8. Eui-Chul L, Hyunwoo J, Daeyeoul K (2011) New finger biometric method using near infrared imaging. Sensors 11(3):2319–2333
9. Havaei M, Davy A, Warde-Farley D, Biard A, C-Courville A, Bengio Y, Pal C, Marc-Jodoin P, Larochelle H (2017) Brain tumor segmentation with deep neural networks. Med Image Anal 35: 18–31

10. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385
11. Huafeng Q, El-Yacoubi M (2017) Deep representation-based feature extraction and recovering for finger-vein verification. IEEE Trans Inf Forensics Secur 12(8):1816–1829
12. Huafeng Q, Lan Q, Chengbo Y (2011) Region growth-based feature extraction method for finger-vein recognition. Opt Eng 50(5):1–9
13. Hong H-G, Lee M-B, Park K-R (2017) Convolutional neural network-based finger-vein recognition using nir image sensors. Sensors 17(6):1–21
14. Jabbar SI, Day CR, Nicholas H, Chadwick EK (2016) Using convolutional neural network for edge detection in musculoskeletal ultrasound images. In: Proceedings of international joint conference on neural networks (IJCNN). IEEE, pp 4619–4626
15. Jian-Da W, Chiung-Tsiung L (2011) Finger-vein pattern identification using principal component analysis and the neural network technique. Expert Syst Appl 38(5):5423–5427
16. Jinfeng Y, Yihua S (2014) Towards finger-vein image restoration and enhancement for finger-vein recognition. Inf Sci 268:33–52
17. Joon-Hwan C, Wonseok S, Taejeong K, Seung-Rae L, Hee-Chan K (2009) Finger vein extraction using gradient normalization and principal curvature. In: Proceeding of SPIE, image processing: machine vision applications II, vol 7251. International Society for Optics and Photonics, pp 1–9
18. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th international conference on neural information processing systems, vol 1 of NIPS 12, pp 1097–1105
19. Kumar A, Zhou Y (2012) Human identification using finger images. IEEE Trans Image Process 21(4):2228–2244
20. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4): 541–551
21. Lin G, Anton M, Chunhua S, Reid I (2017) Refinenet: multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 5168–5177
22. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
23. Lu Y, Gongping Y, Yilong Y, Lizhen Z (2014) A survey of finger vein recognition. In: Zhenan S, Shiguang S, Haifeng S, Jie Z, Yunhong W, Weiqi Y (eds) Lecture notes in Chinese conference on biometric recognition. Springer International Publishing, pp 234–243
24. Naoto M, Akio N, Takafumi M (2004) Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. Mach Vis Appl 15(4):194–203
25. Miura N, Nagasaka A, Miyatake T (2007) Extraction of finger-vein patterns using maximum curvature points in image profiles. IEICE Trans Inf Syst 90(8):1185–1194
26. Neumann C, Tnnies K-D, Pohle-Frhlich R (2018) 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. INSTICC, SciTePress, pp 331–338
27. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
28. Piciucco E, Maiorana E, Kauba C, Uhl A, Campisi P (2016) Cancelable biometrics for finger vein recognition. In: Proceedings of the 1st workshop on sensing, processing and learning for intelligent machines (SPLINE 2016), Aalborg, Denmark, pp 1–6
29. Radzi F, Khalid S-I, Gong F, Mustafa N, Chiew-Wong Y, Mat-ibrahim M (2016) User identification system based on finger-vein patterns using convolutional neural network. ARPN J Eng Appl Sci 11(5):3316–3319
30. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Lecture notes in international conference on medical image computing and computer-assisted intervention. Springer, pp 234–241

31. Roy-Abhijit G, Conjeti S, Navab N, Wachinger C (2018) Fast MRI whole brain segmentation with fully convolutional neural networks. In: Bildverarbeitung für die Medizin 2018. Springer, pp 42–42
32. Sadanandan-Sajith K, Ranefall P, Le-Guyader S, Whlby C (2017) Automated training of deep convolutional neural networks for cell segmentation. Sci Rep (Nature Publisher Group) 7:1–1
33. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. arXiv:1503.03832
34. Song W, Kim T, Chan-Kim H, Hwan-Choi J, Joong-Kong H, Rae-Lee S (2011) A finger-vein verification system using mean curvature. Pattern Recogn Lett 32(11):1541–1547
35. Syafeeza-Ahmad R, Mohamed-Khalil H, Rabia B (2016) Finger-vein biometric identification using convolutional neural network. Turkish J Electr Eng Comput Sci 24(3):1863–1878
36. Ton BT, Veldhuis RNJ (2013) A high quality finger vascular pattern dataset collected using a custom designed capturing device. In: Lecture notes in 2013 international conference on biometrics (ICB), pp 1–5
37. Warfield S, Zou K, Wells W (2004) Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. IEEE Trans Med Imag 23(7):903–921
38. Wu X, He R, Sun Z, Tan T (2018) A light CNN for deep face representation with noisy labels. IEEE Trans Inf Forensics Secur 13(11): 2884–2896
39. Li X, Huang D, Wang Y (2016) Comparative study of deep learning methods on dorsal hand vein recognition. In: Lecture notes in Chinese conference on biometric recognition. Springer, pp 296–306
40. Xie C, Kumar A (2018) Finger vein identification using convolutional neural network and supervised discrete hashing. Pattern Recogn Lett