ISSN 2464-4617 (print)
ISSN 2464-4625 (DVD)
Computer Science Research Notes
CSRN 2901
WSCG Proceedings Part II

CNN Approaches for Dorsal Hand Vein Based Identification

Szidónia Lefkovits
University of Medicine, Pharmacy, Sciences and Technology
Department of Computer Science
Tg. Mureș, Romania
szidonia.lefkovits@science.upm.ro

László Lefkovits
Sapientia Hungarian University of Transylvania
Department of Electrical Engineering
Tg. Mureș, Romania

László Szilágyi
Sapientia Hungarian University of Transylvania
Department of Electrical Engineering
Tg. Mureș, Romania

ABSTRACT

In this paper we present a dorsal hand vein recognition method based on convolutional neural networks (CNN). We implemented and compared two CNNs trained from end-to-end to the most important state-of-the-art deep learning architectures (AlexNet, VGG, ResNet and SqueezeNet). We applied the transfer learning and fine-tuning techniques for the purpose of dorsal hand vein-based identification. The experiments carried out studied the accuracy and training behaviour of these network architectures. The system was trained and evaluated on the best-known database in this field, the NCUT, which contains low resolution, low contrast images. Therefore, different pre-processing steps were required, leading us to investigate the influence of a series of image quality enhancement methods such as Gaussian smoothing, inhomogeneity correction, contrast limited adaptive histogram equalization, ordinal image encoding, and coarse vein segmentation based on geometrical considerations. The results show high recognition accuracy for almost every such CNN-based setup.

Keywords

Biometric identification, dorsal hand vein recognition, CNN architectures, transfer learning.

1 INTRODUCTION

The increasing number of smart devices, cloud computing and home automation have made people more conscious of security and privacy. They have realised the importance of protecting their personal data stored in LAN or on servers in WAN networks.

The credentials in access control systems must be a unique ID that a user knows or owns. Based on this the user is associated and authenticated and it is given access to the requested resource.

Traditional identification and authentication methods are gradually losing ground because they can be easily hacked using a man-in-the-middle attack. The usual credentials such as tokens, passwords or ID-cards are not sufficiently reliable, since they can be relatively easily intercepted or falsified. Tokens, passwords and other important data may be caught or lost, and reused afterwards serving as fake documents. Finally, knowledge-based information may be easily forgotten. Biometrics is one of the most secure and reliable means of personal identification or authentication methods. Biometrics makes use of unique, unforgeable characteristics based on body measurements or comportment that are difficult to copy or steal.

The most important features used in biometric identification are the face, fiducial points, fingerprints, veins, palm and dorsal hand veins, finger veins, the iris, the retina, ears, the voice and the DNA.

In recent years, biometrics based on patterns of the vascular system have gained increasing attention. Analysis of finger vein, palm vein and dorsal hand vein patterns are the most common vascular structure biometrics. Their advantages are uniqueness, stability, contactless acquisition and unforgeability. They are also typical of the living.

In this paper, a dorsal hand vein identification system is presented which exploits the strength of convolutional neural network architectures.

The rest of the paper is organized as follows: after a short review of dorsal hand vein detection and authentication systems in the literature (Section 2), the database used in the experimental setup is presented (Section 3). Section 4 summarises the fine-tuned and retrained CNN architectures, followed by the detailed description of the approach proposed with two CNNs trained end-to-end (Section 5).

Finally, our experiments and results are presented in Section 6, followed by the conclusion and discussion (Section 7).

2 RELATED WORK

The blood circulation system of every creature is already formed in embryo state. The exact path and form of the veins, from the medical point of view, is not known in detail, and neither is the reason of the uniqueness of the vessel network. It is only known that the probability of finding two individuals with the same pattern of blood circulation system is quite low.

Technically, dorsal hand vein detection is the easiest compared to finger vein, palm vein or other vein pattern acquisitions. The veins are under the skin and carry blood from the organs towards the heart. They are bluish in colour and can be detected by light reflecting the temperature difference between the warm blood flow and the surrounding cells. The acquisition of dorsal hand images is usually done by Near Infrared (NIR) or Far Infrared (FIR) cameras. The acquisition protocol is not standardized and differs from one database to the other. The resolution of the images obtained is usually very low, with low contrast and a restricted region of grayscale intensities.

Vein-based approaches can be classified into two different types: shape-based methods and texture-based methods.

Shape-based methods use vascular structure information, extracting line or curve features and measuring different types of distances: the Hausdorff distance or the Line Segment Hausdorff distance or angles [1]. Here the geometric representation is obtained based on an approximation of segments by short vectors, bifurcation, endpoints and crossing points. These features are obtained after the vessel is detected and the skeleton of the vein structure is obtained. The segments of the skeleton can be approximated by line detection using Hough transform. In this case the veins are extracted by line tracking methods or local curvature extraction [2]. The disadvantage of these methods is the requirement for an aligned and registered skeleton structure.

The registration and alignment of two binary skeleton images is usually done by considering local invariant feature points such as SIFT (Scale Invariant Feature Transform) [3], SURF (Speed-Up Robust Features) [4] and Hessian-Laplace interest points [5].

Another approach capable of registering two vein skeleton images using a 3D rigid transformation that maximised the skeleton overlap was described in our previous article [6].

The texture-based methods can be divided into two categories: global texture methods and local texture methods. The global methods, also called holistic methods, extract the features from the whole region of interest, characterising the texture and vein structure globally. For this purpose, Principal Component Analysis (PCA) was used by Wang et al. in [7], the Fisher Linear Discriminant was applied Liu et. al in [8] and Independent Component Analysis (ICA) by Yuksel et al. [9]. These methods rely on eigenvalue and eigenvector decomposition. The results in these cases were severely influenced by the variation of position, viewpoint, contrast, illumination changes, occlusion, distortion and misalignment of images. However, these methods are a good starting point in combination with local texture methods.

The most important local texture-based approaches are different variants of locally binary patterns (LBP) or circular locally binary patterns (CLBP) [10] or Gabor texture descriptors [11]. Another type of local information extraction is based on local key point matching, for example SIFT (Scale Invariant Feature Transform) [12], Difference of Gaussian [13], Oriented Gradient Maps [14], Centroid-based Circular Key-point Grid (CCKG) [12].

The results obtained by a single key-point extraction method are not good enough; therefore, Huang et. al [15] proposed a combination of LBP for local texture, binary coding for local shape extraction and graph matching for global shape characterisation.

The most recent research field of convolutional neural networks is not widespread in the case of vein-based identification. Hong et al. proposed a reduced complexity four-layered CNN for finger vein recognition [16] and Wan et al. [17] proposed a CNN based on VGG-19 [18] for dorsal hand vein recognition.

3 THE DATABASE

The proposed approach is evaluated on the NCUT (North China University of Technology) Part A [8] dorsal hand vein dataset. It is the largest publicly available database used for automatic dorsal hand vein image recognition systems. The NCUT database contains 2040 near infrared images of dorsal hands of 102 individuals, and both hands have 10 samples each. Due to the low-cost acquisition equipment, this database contains low-quality images with a high noise level. All images were acquired by the same dorsal hand scanner, resulting in roughly aligned images. There are only small changes in translation, rotation and illumination between the images. We can observe more significant changes in viewpoint because of the hand rotating around the handle of the scanner, considered to be the *Ox* axis of the images. The NIR intensity is stable, but there is a circular variation in illumination from the centre to the margins because of a light source illuminating form

 ISBN 978-80-86943-38-1

above. Thus, some veins near the margins are difficult to distinguish, and vein pixel intensities are sometimes similar to skin pixel intensities. The acquisition equipment through the NIR camera determined the low contrast with a resolution of 640 ×480 pixels on an 8-bit greyscale image. Moreover, the dorsal hand occupies only about half of the available area and there are only 80 integer intensities, which restricts the range of contrast into the [101, 180] interval.

4 TRANSFER LEARNING

Transfer learning is used to fine-tune the weights of a given CNN architecture to be suitable for new input data. It has been shown that architectures based on transfer learning obtain better results in accuracy and performance than retraining the same network structure from the beginning.

In our experiments we created two networks built and trained from scratch, and fine-tuned four other well-known CNN architectures (AlexNet, VGG-16, ResNet and SqueezeNet) with pretrained weights.

4.1 Alex Net

AlexNet [19] was the winning architecture of the ImageNet [20] Challenge in 2012. It was the first architecture implementing parallelism in training the network. The original architecture has an input image size of 224×224 and 3-channel RGB image. It consists of 5 convolutional layers, 3 overlapping max-pool layers and two fully-connected layers at the end. The activation functions in the conv-layers was the ReLU. The first conv-layer has a depth of 96 with a kernel size of 11×11 and a stride of 4. The output of this layer is a 55×55×96 response, which is reduced to 27×27×96 by an overlapping max-pool layer of 3×3 kernels and stride=2. The second conv-layer has a kernel size of 5×5, a depth of 256, a stride of 1 and a zero-padding of 2 pixels in each direction. LRN (Local Response Normalization) was applied to the responses of the previous convolutional layers, i.e., batch-norm was not known at that time. The output is resized to half in width and height to 13×13×256 with the same type of max-pool as before. The following 3 convolutions have a kernel size of 3×3 with a padding of 1, astride of 1 and a depth of 384, 384 and 256, respectively. These 3 layers maintain the previous size of 13×13×256, which is reduced to half 6×6×256=9216 afterwards. These responses are fed into 2 FC layers, each having 4096 activations. The final decision of predicting 1000 class layers is made by a softmax function.

In our approach we refined the trained weights for ImageNet and replaced the final FC layer to be suitable for our problem of dorsal hand vein identification by changing the output layer to 204 instead of 1000.

4.2 VGG

VGG was introduced by Simonyan and Zisserman [18] and was the winning architecture of the ImageNet competition in 2014. It is a much deeper model compared to AlexNet, containing 16 (VGG-16) and 19 (VGG-19) layers, respectively. Their idea was to stack multiple convolutional layers, with each having a kernel size of 3×3 with stride 1. This kernel size reduced the number of parameters in the layer. It has been shown that two stacked 3×3 kernel layers have the same receptive fields as a single 5×5 one. Likewise, a stack of 3 conv-layers has the same receptive field as a single 7×7 kernel conv-layer. The VGG-16 has 13 conv-layers, 5 max-pooling layers and 3 fully connected layers at the end. The other novelty here was the doubling of filter depth after reducing the input image size to half in both height and width. This led to a parameter reduction of only 1/2 instead of 1/4. The input image size of 224×224×3 was reduced after two conv-layers with a depth of 64 (conv1_1, conv1_2) to 1/2 size using a max-pool layer with stride 2. , Two other conv-layers were applied on the 1/2 image size, with double depth 128 and a max-pooling resize of the output to 1/4. Three conv-layers were stacked on the 1/4 size, each having a depth of 256. On the 1/8 and 1/16 sizes, 3-3 stacked convolutional layers followed, each having a depth of 512. The output of the last layer, in this way, was 7×7×512, and was fed into three FC layers of sizes 4096, 4096 and 1000. The last two FC layers were followed by dropout with a probability of 0.5, like in AlexNet. This architecture has about 138 million parameters and was extremely hard to train. The training was done in several stages, gradually adding the interior layers.

In our approach we refined the initially trained weights for ImageNet and replaced the final FC layer in order to obtain 204 class scores.

4.3 ResNet

The main reason for introducing ResNet [21] was the observation of lower training performances, although the number of layers increased. Thus, CNNs with a high number of convolutional layers are more difficult to be optimized. Moreover, if the number of layers is above a certain limit between 20 and 30, the CNN will also have a higher training and test error. The convergence of such deep architectures is not reached. The authors of paper [21] solve the problem of mapping an input layer to an output layer by computing only the difference that must be added (subtracted in case negative values) to the input values. So, instead of determining *Out(x)* it determines the residual *R(x)*, where *R(x)=Out(x)-x*. The shortcut connection, also called bypass, is the identity mapping (*x*) which adds the input to the residual and obtains the output. The residual is

ISBN 978-80-86943-38-1

determined by multiple (2 or 3) consecutive conv-layers. In this way they are able introduce very large networks with 18, 34, 50, 101 and 152 layers. Every conv-layer is batch normalised and fed into ReLU activation.

In our experiments we have used the ResNet50 architecture. The input layer is the dimension used in ImageNet. The first layer is 7×7×64 with stride 2. The next dimension is 112×112 because of resizing by max-pooling. On this size, 3 consecutive conv-layers are used to determine the residual (1×1×64, 3×3×64, 1×1×256), and these are repeated 3 times. The next feature map dimension is 56×56. Here the residual is computed using another 3 conv-layers (1×1×128, 3×3×128, 1×1×512), and these are repeated 4 times. The next input feature dimension is 28×28, and here the residual block consists of these three 1×1×256, 3×3×256, 1×1×1024 conv-layers. These layers are replicated 6 times. On the final dimension of 14×14, the 3 layers are included in the residual block (1×1×512, 3×3×512,1×1×2048). The last layer is a fully-connected layer of 7×7×2048, mapping to 1000 class scores determined by softmax.

In our experiments we have fine-tuned the pretrained weights of this architecture using the NCUT dataset, added data augmentation and adapted the weights and FC layers according to the dorsal hand identification problem.

4.4 Squeeze Net

The SqueeezeNet architecture introduced in [22] is a CNN with a drastically reduced number of parameters (of 1.2 million only), but with accuracy results in ImageNet object detection completion similar to AlexNet. Because of the lower number of parameters, it requires less memory in parallel training, and the model obtained can be easily loaded onto embedded systems or microchip-based intelligent systems with lower hardware resources. The main ideas of this network are the so-called fire modules along with the conservation of the activation maps by downsampling their spatial dimension later and fewer times in the network. The fire module consists of 3 components: a squeeze layer (of kernel 1×1) to reduce the depth of the input filter-map (reducing the number of parameters) and two expansion layers of kernel 1×1 and 3×3 to transform the reduced layer back to its original size. $s_{1\times1}$ squeeze-layers have the role of reducing its input size of $w\times h\times d$ to only $w\times h\times s_{1\times1}$. The squeeze layers are followed by ReLU activation and two consecutive expansion layers: the 1×1 conv-based returns to a higher dimensional space $w\times h\times e_{1\times1}$ and the 3×3 conv-based reduces the number of weights compared to the previous layer. After the expansion the output layer is of size $w\times h\times (e_{1\times1}+e_{3\times3})$ and is followed by a ReLU activation. The SqeezeNet architecture is made up of a conv-layer, max-pool for downsampling, three fire modules, max-pool again, four fire modules, max-pool, a final conv-layer and average pooling at the end, meaning a total of 11 layers based on convolutions. The model trains faster not only because of the fire modules, but also due to using sparse weight matrixes and computation not with float numbers but 6- or 8-bit quantized integers [23]. The authors present three variants of the SqueeezeNet: the vanilla architecture, the SqueezeNet with bypass connections and bypass with convolutions.

In our application we used the vanilla SqueeezeNet without bypasses retrained the original network and adapted the last FC layer to the purpose of dorsal hand vein identification. Thus, instead of 1000 class scores we had to obtain only 204.

5 OUR APPROACH

The goal of this research is the evaluation of two novel CNN network architectures proposed and trained for dorsal hand vein recognition. Their identification accuracy and training and testing performances are compared to well-known CNN architectures such as AlexNet, VGG-16, ResNet-50 and SqueezeNet, summarised above.

The first network proposed and trained from scratch is a 6-layered CNN made up of 4 convolutional and 2 fully connected layers. The input image was redimensioned and centre-cropped to the same dimension as required for the other types of networks, i.e., to 224×224×3. The original image is only grayscale, but the depth of 1 was converted by replication to a depth of 3.

The NCUT dataset has only 10 different image samples of both hands for every subject. This number of training images is very small for proper end-to-end training of convolutional neural networks

In order to enlarge the training dataset available, we have applied classic data augmentation methods to generate, instead of 10, thousand similar images during the training process. There are several types of augmentation techniques. We have used centre-crop based on a binary mask to consider only the hand, followed by a resize to the standard input size of all networks. We applied random rotation of the images by ±10°. Finally, the images were normalized with Z-score normalisation using the mean and standard deviation of the image set.

In this study, our goal was to create a simple CNN model which is easily trained and has only a few parameters, but obtains reasonable identification accuracy compared to other state-of-the-art pretrained CNNs with initial weights fine-tuned for the ImageNet Challenge, these are much harder to be trained.

 ISBN 978-80-86943-38-1

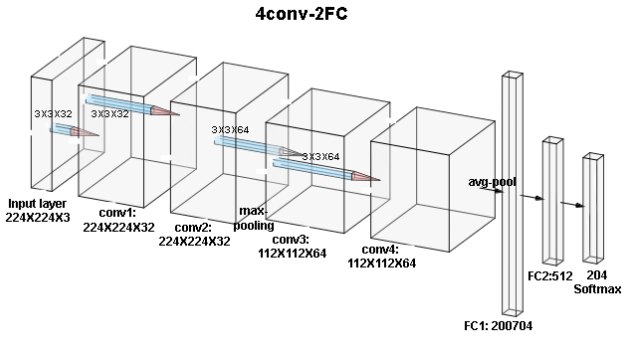


Figure 1: 4conv-2FC Architecture Trained from Scratch

	Layer type	No. of filters	Size of input feature map	Size of output feature map	Size of kernel	No. of stride	No. of padding
Dimension 224×224	Conv1_1	32	224×224×3	224×224×32	3×3	1×1	1×1
	Conv1_2	32	224×224×32	224×224×32	3×3	1×1	1×1
	Batch norm 4×32						
	Max-pool 1		224×224×32	112×112×32		2×2	0×0
	Dropout p=0.25						
Dimension 112×112	Conv2_1	64	112×112×32	112×112×64	3×3	1×1	1×1
	Conv2_2	64	112×112×64	112×112×64	3×3	1×1	1×1
	Batch norm 4×64						
	Avg-pool 2		112×112×64	56×56×64		2×2	0×0
	Dropout p=0.25						
	FC1		56×56×64	512			
	Dropout p=0.25						
	FC2		512	204			

Table 1: 4conv-2FC architecture

To this end, we have first built a CNN with 4 convolutional layers, called 4conv-2FC (Figure 1). The convolutional filter in each case was a 3×3 convolution. The 3×3 filter is a reasonable dimension to extract a patch of the image. If the filter is larger, the number of weights between two layers will be higher, implying the growth of the final number of parameters. If the input has a depth of D and the filter $f \times f \times d_f$, the number of weights between the particular input and the filters output will be $(w_f \times h_f \times d_f + 1) \times D$. The 1 constant represents the bias term. In the literature [18], it has been shown that a 5×5 kernel response can be obtained as a stack of two conv-layers with a kernel of 3×3. It is obvious that $2\times(3\times3\times d+1)\times D$ is 40% less than $(5\times5\times d_f+1)\times D$. Every convolutional layer output is fed into a ReLU activation function. The rectifying linear unit $ReLU(x)=max(0,x)$ is preferred instead of other activations because the its derivate is easier to compute. The hyperbolic tangent and sigmoid activations are rarely used in CNNs because they quickly saturate, and their derivate in that case is 0.

Every conv-layer is normalized using batch normalization. Batch normalization (BN) learns the mean and standard deviation of a given layer and preserves them even if the input weights are changing. It makes the learning of subsequent layers easier and has a regularisation effect as well. For every depth (d), the BN learns the γ (scale) and (β) shift parameters of the normalized outputs of a given layer. For the normalisation of the values, it computes the mean and standard deviation of the layer; therefore, the number of parameters in this case is $4 \times d$. We use a dropout with a probability of 0.25 and 0.5 after the second and fourth conv-layer. The convolutional layers maintain the same feature map size in width and height by using a zero-padding of p=1 and stride s=1 for every 3×3 convolutional kernel. Only the depth of the feature maps changes. The output width (w_O) and height (h_O) can be calculated based on the input feature maps size $w_I \times h_I$, the filter size $w_f \times h_f$ and the padding p and stride s, according to the formula: $w_O=((w_I-f_I)+2*p)/s+1$.

ISBN 978-80-86943-38-1

The depths of the feature maps gradually increase by a factor of 2 when their width and height is reduced by 2. This operation reduces the number of activations between layers by a factor of only 2 instead of a factor of 4. The smaller size of the feature maps is obtained by max-pooling layers of 2×2 with a stride of 2. This means that the feature map is reduced to half its size by keeping the maximum value in every block of 2×2 pixels. The last pooling layer in our network is global average pooling in blocks of 2×2, with the role of minimising the overfitting before the first dense layer. The convolutional layers are followed by 2 fully connected layers. The last fully connected layer is fed into a softmax function. The softmax converts the class scores obtained into probability distributions, representing the probability of the input image to be considered in class *i*, where *i* is the number of subjects in the database ×2 (left/right land), in our case 2×102=204.

Our first CNN architecture 4conv-2FC (Figure 1, Table 1) considers two dimensions of the original image: the full size (224×244 feature map) and the half-size feature map (112×112). For each dimension, two convolutional layers are considered. The first layer has an input of 224×244×3 and produces an output of 224×244×32 followed by another similar layer. These layers are obtained using 3×3×32 filters with s=1 and p=1. After these two layers, the feature map is reduced to half its size, and two conv-layers are computed again. The third has an input of 112×112×32 and an output of 112×112×64 fed into the forth, obtaining an output of 112×112×64. Before the dense layers, the output is again reduced to 56×56×64 (=200704). Thus, the first fully connected layer creates a mapping between 200704 and 512. The last FC layer has a number of outputs equal to the number of classes, i.e., 204. The parameters of this CNN architecture are summarized in Table 1.

Our second CNN architecture 6conv-2FC (Figure 2, Table 2) considers the dimensions of the original image: the full size, the half-size feature map (112×112) and the 1/4 size one (56×56). For each dimension, two convolutional layers are added. The first four conv-layers are similar to the first approach, but there is another pair of conv-layers at the 56×56 size. The fifth layer has an input of 56×56×64 and obtains an output of 56×56×128, while the sixth layer is a replica of this, obtaining once more an output of 56×56×128.

Before the dense layers, the output is again reduced to 28×28×128 (100352). Thus, the first fully connected layer makes a mapping between 100352 and 1000. The last FC layer must have 204 responses, considering the number of classes. The parameters of this CNN architecture are summarised in Table 2.

6 RESULTS AND EXPERIMENTS

In this section we describe our experiments related to dorsal hand vein recognition with different convolutional network architectures. We compare the effect of several preprocessing steps on the recognition performance of the trained networks.

The databased used in our approach is the NCUT database described in detail in Section 3.

The block diagram of our system is depicted in Figure 3. The flowchart of our approach has the database as the input, followed by a ROI detection, based on a binary mask, different preprocessing steps, and Z-score normalization with a mean of 0 and a standard deviation of 1. The main part of the system is the implemented and trained CNN architecture used for dorsal hand vein identification purposes.

The hardware used for carrying out our experiments was a Windows 10 64bit operating system with 32GB of memory and an NVIDIA Geforce GTX Graphics Card with 11 GB of memory using the Python and Pytorch framework.

In our experiments we considered different types of settings. In the first settings we compared our proposed two different CNN architectures trained from scratch. These two architectures considered the same original dataset. We have split the dataset randomly into 6 samples per person for training and 4 for testing and out of these 2 for validation. In both cases, we considered the same number of training epochs (400) and the same hyperparameters (optimization of Stochastic Gradient Descent with a learning rate of 0.01).

The number of parameters in these two cases are approximately the same 103M (4conv-2FC) or 100M (6conv-2FC). The training and the validation curves in Figure 4 show a better dropdown of the training curve and a better stabilization of the validation curves for the CNN architecture with 6 conv-layers. The accuracy of the networks is 92% and 95%. (Figure 9 and Table 3).

In the next stage, we compared the training and testing performance versus the number of epochs, the learning rate and the optimization criterion only on 6conv-2FC.

A total number of iterations of about 400-500 was enough in the case of the two networks trained from scratch, and only 200-250 iterations run were enough for the pretrained and fine-tuned networks.

A low learning rate of 0.001 led to a very slow convergence and a higher learning rate of 0.1 showed an oscillating behaviour in the training loss.

 ISBN 978-80-86943-38-1

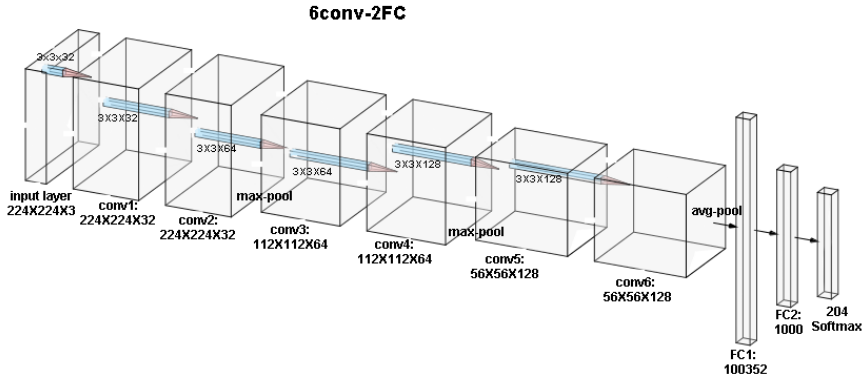


Figure 2: 6conv-2FC Architecture Trained from Scratch

	Layer type	No. of filters	Size of input feature map	Size of output feature map	Size of kernel	No. of stride	No. of padding
Dimension 224×224	Conv1_1	32	224×224×3	224×224×32	3×3	1×1	1×1
	Conv1_2	32	224×224×32	224×224×32	3×3	1×1	1×1
	Batch norm 4×32						
	Max-pool 1		224×224×32	112×112×32		2×2	0×0
	Dropout p=0.25						
Dimension 112×112	Conv2_1	64	112×112×32	112×112×64	3×3	1×1	1×1
	Conv2_2	64	112×112×64	112×112×64	3×3	1×1	1×1
	Batch norm 4×64						
	Max-pool 2		112×112×64	56×56×64		2×2	0×0
	Dropout p=0.25						
Dimension 56×56	Conv3_1	128	56×56×64	56×56×128	3×3	1×1	1×1
	Conv3_2	128	56×56×128	56×56×128	3×3	1×1	1×1
	Batch norm 4×128						
	Avg-pool 3		56×56×128	28×28×128		2×2	0×0
	Dropout p=0.25						
	FC1		28×28×128	1000			
	Dropout p=0.5						
	FC2		1000	204			

Table 3: 6conv-2FC architecture

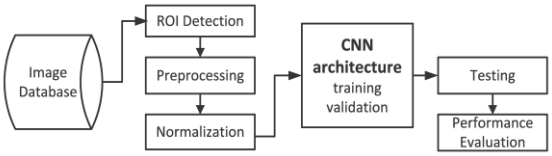


Figure 3: The Proposed System

Thus, the most adequate value for the learning rate in our case was 0.01. We also compared two different weight update methods, simple SGD with lr=0.01 as well as SGD with momentum (lr=0.01, β=0.9) and RMSProp updates. RMSProp update led to an oscillating behaviour, and simple SGD worked slightly better when training the two new networks; at the same time, SGD with momentum worked well in transfer learning.

We also proposed to compare different types of known networks and fine-tuned them by modifying their weights and final decision layer tailored to the problem of dorsal hand vein identification. Here we compared the training and test performances of the state-of-the-art networks described in Section 4.

Figure 5 compares the training behaviour. Here we can observe that the ResNet-50 network obtains the best training loss (with 50 conv+FC-layers), our 6conv-2FC (6+2 layers) and 4conv-2FC (4+2 layers) network is slightly worse, but their performance is similar to VGG-16 (13+3 layers) in this case.

ISBN 978-80-86943-38-1

The training loss of these four networks drops from the early epochs. AlexNet and SqueezeNet training losses drop in a relatively late epoch compared to the other four networks mentioned above.

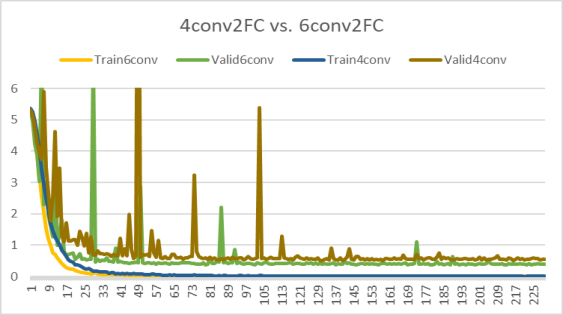


Figure 4: Training and Validation Losses for 4conv-2FC and 6conv-2FC

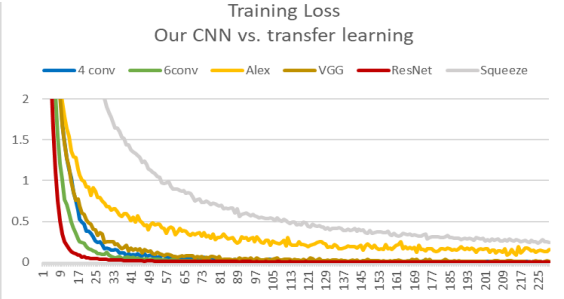


Figure 5: Training Loss in Transfer Learning

The 3×3 filters computed in the case of 6conv-2FC are shown in Figure 6. In fact, the kernel filters represent the most common 3×3 image patches in a given layer.

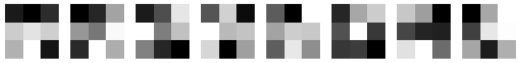
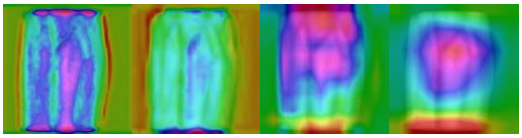


Figure 6: 3×3 conv-filters

The layer-wise heatmap obtained from GradCam [24] visualizations shows (Figure 7) the most activated regions of the dorsal hand layer by layer, from layer 1 to 4.



a) Layer1 b) Layer2 c) Layer3 d) Layer4

Figure 7: Heatmap of Layers 1-4

The activation is gradually concentrating on the upper part of the hand, where the main longitudinal veins fork into diagonal veins going into the fingers. This is the most representative part of the hand vessel system from the perspective of identification.

Our next experiment analysed the effect of the various preprocessing steps applied to the original image before the training of different CNN architectures. In our previous paper [6] we proposed a geometry-based vein extraction algorithm.

The process started with some image preprocessing steps followed by segmentation, which allowed extracting the skeleton of the detected veins. These preprocessing steps are image inhomogeneity correction, contrast limited adaptive histogram equalization and codification of the veins with an ordinal measure.

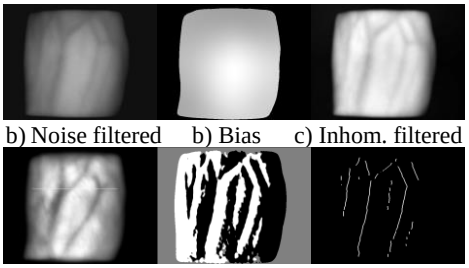
The preprocessing steps mentioned (Figure 3) are applied using a binary mask representing the ROI of the hand. Its boundary is determined by the contour of the black background and the non-black foreground. The next step (a) is the noise filtering (Figure 8a) of the masked image. Here we used a 2D Gaussian kernel of 3×3 with $\mu=0$ and $\sigma=0.5$. This eliminated the blurriness of the images. In addition, we noticed that the illumination of the images is not uniform. The intensity of the central part is much higher than at the margins. An inhomogeneity correction filter can reduce the variation in illumination. The variation in intensities can be modelled using a bias image (step b) (Figure 8b), which is a smooth multiplicative field over the filtered image. Thus, we solved this artefact by adapting the N3 inhomogeneity correction (step c) algorithm [25] to the given images (Figure 8c). Another important step (d) is contrast limited adaptive histogram equalization (CLAHE) (Figure 8d), which ensures the image quality and contrast enhancement required for better segmentation or identification. The parameters used in this case are a block size of 7 and 127 for the contrast-limiting threshold.

Finally, the images are segmented based on an ordinal image encoding (step e) procedure by comparing the successive pixel intensities along a horizontal line. Pixels with decreasing intensities along the line are coded black (0), pixels with increasing intensities are coded white (255), and pixels with almost constant intensities are coded grey (128) (Figure 8e). This encoding creates images that are a sort of relief-like representation of the veins. The threshold (T) which determines increasing, decreasing or constant behaviour is a parameter imposed by the width of the veins detected. A value of 0.5 for this threshold and a minimum region around the minimum point larger than 4 pixels is considered to be a vein centre. By applying this segmentation procedure (step f) we obtain coarse vein segmentations (Figure 8f).

The presented CNN networks were trained and tested on the original images, on the inhomogeneity-corrected and CLAHE images, on the coded and on the segmented images as well. The results for each type of image are summarized in Table 3 and Figure

ISBN 978-80-86943-38-1

9. We can observe that the best identification results are obtained on preprocessed images with steps a, b, c and d (column InhCLAHE).



b) Noise filtered b) Bias c) Inhom. filtered

d) CLAHE e) Coded f) Segmented

Figure 8: Preprocessed Images

The accuracy is 2-3% better than that for the normalized original images without applying inhomogeneity correction and CLAHE (column Orig.). The codification based on the ordinal measure did not bring any advantages to the training of the system or to testing accuracy. It is obvious from the results that the segmented black and white images with very few white (information) pixels and a lot of background (black) ones make the final identification performance much worse. This behaviour is because of the many background pixels contributing to a vanishing gradient most of the time.

Comparing the recognition accuracies of CNNs fine-tuned for the NCUT dataset and using the pretrained weights via transfer learning, we can observe that the ResNet-50 network has almost perfect 99% results for the test set, followed by VGG-16's 97-98%. AlexNet has similar results to our 6conv-2FC network, 95-96% and 94-95%, respectively. The worst recognition results were obtained by the SqueezeNet network. We can see that by combining any two networks as a decision ensemble and joining their softmax probability responses, 99-100% accuracy can be obtained.

CNN	Orig	InhCLAHE	Coded	Segment
4conv2FC	0.9195	0.9265	0.9142	0.8456
6conv2FC	0.9485	0.9583	0.9412	0.8775
alexnet	0.9650	0.9608	0.9510	0.9167
vgg	0.9722	0.9804	0.9706	0.8946
resnet	0.9951	0.9951	0.9951	0.9804
squeezenet	0.9069	0.9118	0.8824	0.7598

Table 3. Identification Accuracy on the Test Set

Our approach and results can be compared to the most recent state-of-the-art papers using CNN architectures. Paper [17] presents a variation of RCNN with self-growth strategy. They report a 95% recognition rate on the training and 91.25% on the test set. However, the database used in that case is unknown, i.e. not publicly available. Thus, the accuracy results are not referring to the same dataset.

The results of the other article [26] presenting dorsal hand vein recognition by CNNs is much more comparable to our experiments. They work with the same NCUT database and create a system with five parallel SqueezeNets. The accuracy obtained by majority voting of these parallel networks is 99.52%. This architecture which needs a five-times as much training and 5 times as many parameters, as our networks. Yet we demonstrated that by combining the outputs of any two of our trained networks a result of 99-100% can be easily achieved.

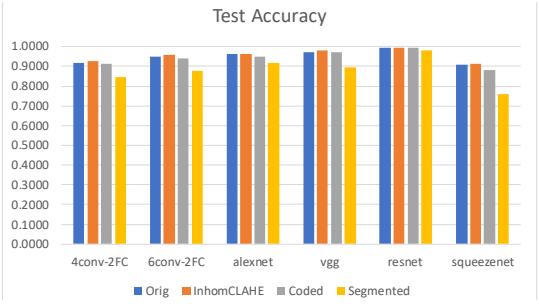


Figure 9: Identification Accuracy on Test Set

7 CONCLUSION AND DISCUSSION

In this paper, two different CNN architectures are proposed for dorsal hand vein-based identification. These CNNs are compared to existing state-of-the-art deep learning approaches. The training process based on the fine-tuning of the final layers and the parameter transfer on the rest of the pretrained weights lead to better identification performances, especially in the case of the VGG and ResNet architectures, compared to the end-to-end trained proposed CNNs (4conv-2FC and 6conv-2FC). The number of layers and the number of parameters for our CNNs is much smaller than in the other two better architectures. Therefore, they are more adequate in systems with lower hardware resources. However, the experiments show superior performance of deep learning training compared to other dorsal hand vein identification systems based on feature extraction.[12, 14, 15] In addition, we have found that different kinds of preprocessing steps, especially inhomogeneity correction of the images, gains a 2-3% increase in accuracy.

To generalise the identification process, a more complex database with much more subjects and image samples would be required for real-life validation and everyday use. Therefore, in the future we propose to enlarge the training dataset using several augmentation methods, and extend the NCUT database. Moreover, we intend to validate the proposed architecture not only for dorsal hand veins, but for palm veins and finger vein recognition also.

ISBN 978-80-86943-38-1

8 ACKNOWLEDGMENTS

The work of L. Lefkovits was partially supported by a grant from the Sapientia KPI, the research carried out was facilitated by Domus Hungarica Scholarship. L. Szilágyi is János Bolyai Fellow of the Hungarian Academy of Sciences.

9 REFERENCES

[1] M. M. Pal and R. W. Jasutkar, "Implementation of hand vein structure authentication based system," *Int. Conf. on Communication Systems and Network Technologies*, pp. 114–118, 2012.

[2] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods.", *British Machine Vision Conf.*, vol. 2, no. 4, 2011

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Jour.of Computer Vision*, vol. 60: no.2, pp. 91–110, 2004.

[4] M. Pan and W. Kang, "Palm vein recognition based on three local invariant feature extraction algorithms," *Chinese Conference on Biometric Recognition*. Springer, pp.116–124, 2011.

[5] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *Computer Vision ECCV 2002*, pp. 128–142, 2002.

[6] S. Lefkovits, S. Emerich, and L. Szilágyi, "Biometric system based on registration of dorsal hand vein configurations," *Pacific-Rim Symposium on Image and Video Technology*. Springer, pp. 17–29, 2017.

[7] L. Wang, G. Leedham, and D. S.-Y. Cho, "Minutiae feature analysis for infrared hand vein pattern biometrics," *Pattern recognition*, vol. 41, no. 3, pp. 920–929, 2008.

[8] Y. Wang, K. Li, and J. Cui, "Hand-dorsa vein recognition based on partition local binary pattern," in *Int. Conf. on Signal Processing (ICSP), 10th* . IEEE, pp. 1671–1674, 2010.

[9] A. Yuksel, L. Akarun, and B. Sankur, "Hand vein biometry based on geometry and appearance methods," *IET Computer Vision*, vol. 5, no. 6, pp. 398–406, 2011.

[10] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE PAMI* no.7, pp.971–987, 2002.

[11] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei, "Object-centric spatial pooling for image classification," *European Conference on Computer Vision*. Springer, pp. 1–15, 2012.

[12] D. Huang, R. Zhang, Y. Yin, Y. Wang, and Y. Wang, "Local feature approach to dorsal hand vein recognition by centroid-based circular key-point grid and fine-grained matching," *Image & Vision Computing*, vol. 58, pp. 266–277, 2017.

[13] Y. Wang, Y. Fan, W. Liao, K. Li, L.-K. Shark, and M. R. Varley, "Hand vein recognition based on multiple keypoints sets," in *IAPR Int. Conf. on Biometrics (ICB)*. IEEE, pp. 367–371, 2012.

[14] D. Huang, W. B. Soltana, M. Ardabilian, Y. Wang, and L. Chen, "Textured 3d face recognition using biological vision-based facial representation and optimized weighted sum fusion," in *CVPR*. IEEE, pp. 1–8, 2011.

[15] D. Huang, X. Zhu, Y. Wang, and D. Zhang, "Dorsal hand vein recognition via hierarchical combination of texture and shape clues," *Neurocomputing*, vol. 214, pp. 815–828, 2016.

[16] H. Hong, M. Lee,K. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors" *Sensors*,17:6, p. 1297, 2017.

[17] J. Wang and G. Wang, "Hand-dorsa vein recognition with structure growing guided cnn," *Optik*, vol. 149, pp. 469–477, 2017.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *preprint arXiv:1409.1556*, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks"*Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, pp. 770–778, 2016.

[22] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *preprint arXiv:1602.07360*, 2016.

[23] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *preprint arXiv:1510.00149*, 2015.

[24] R. R. Selvaraju, A. Das, R. Vedantam, et al., "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,"*CoRR*,abs/1610.02391, 2016.

[25] N. Tustison, B. Avants, P. Cook et. al, "N4itk: Improved n3 bias correction"*IEEE Trans Medical Imaging*,29:6,pp.1310–1320,2010.

[26] H. Wan, L. Chen, H. Song, and J. Yang, "Dorsal hand vein recognition based on convolutional neural networks. BIBM pp. 1215-1221, 2017.

 ISBN 978-80-86943-38-1