A CNN approach for the identification of dorsal veins of the hand

Abdelkarim Benaouda, Aymen Haouari Mustapha, Sarâh Benziane, University of Science and technology of Oran Mohamed Boudiaf

Abstract—In this paper, we proposed a dorsal hand vein recognition method based on Convolutional Neural Network (CNN). Firstly, implementations of raw images the region of interest (ROI) of dorsal hand vein images was extracted, and then contrast limited adaptive histogram equalization (CLAHE) and were used to preprocess the images. Next, the extraction of information using the Sato filter and the Otsu thresholding algorithm to create a new database containing only the processed images. Finally, CNN was applied for identification. The experimental results was has been optimized with Hyperparameter Optimization. The dorsal hand vein recognition rate reaches 99%.

Keywords—dorsal hand vein recognition; deep learning; convolutional neural network, CLAHE, SATO, OTSU, mask.

I. INTRODUCTION

Biometric technology is one of the effective techniques for authenticating and identifying people. Biometrics is the computer science term for the field of mathematical analysis of unique human characteristics such as fingerprints, hand, palm and finger veins, eyes, voice, signature, gait and DNA. Biometric solutions have experienced accelerated growth in the global security market over the past few decades, primarily due to increasing public safety requirements against terrorist activities, sophisticated crimes and electronic fraud. Biometrics is the science of identifying a person based on their behavioral and physiological characteristics.

Biometric systems fall into two categories: physical and behavioral. Physical systems are related to body shape, such as fingerprints, facial recognition, DNA, vascular patterns, eye iris, vein pattern, etc. Behavioral biometrics are related to a person's behavior, such as voice, gait, signature, etc. We focus on the venous network of the back of the hand (i.e., dorsal hand) because it is clearly visible, easy to acquire, and efficient to process. Compared to other popular biometric features, such as face or fingerprints, the dorsal hand vein has several advantages.

It is also the best variant of biometric systems that require physical contact with the machine, as it extracts the vein pattern, the hand is not in contact with the device, the hand can easily stretch, and capturing the vein pattern can be done easily. Since the system is based on three features such as live body, internal veins and non-contact type, there is no possibility of tampering, and no misuse by criminals, so it can be used in places requiring high level of security to avoid crimes and frauds.

*A. **Related Works :***

The important characteristic of hand vein patterns is stability, which means that the structure of the hand and the configuration of the hand veins remain relatively stable throughout the life of the individual. For this reason, vein identification systems are considered a promising and reliable biometric. In this section, some of the vein identification systems are presented.

Huang et al [1] proposed a method for identifying dorsal veins in the hand. A new process integrating together the holistic and local then hierarchically joint analysis with that of the surface modality, born by a reputable texture operator, that local binary patterns (LBP), binary coding (BC) and graph for decision generation by Factored Graph Matching(FGM). The results obtained are superior to those of the state of the art described so far in the work, which proves its effectiveness.

Traditional palm vein recognition algorithms use physical models, including minutiae, ridges, and texture, to extract features for matching. For example, the adaptive multispectral method [4], 3D ultrasound method [5], and adaptive contrast enhancement method [6] are applied to improve image quality. Ma et al [7] proposed a palm vein recognition scheme based on an adaptive 2D Gabor filter to improve image quality. 2D Gabor filter to optimize selection's parameter. Yazdani et al [8] presented a new method based on wavelet coefficient estimation with an autoregressive model to extract the texture feature for verification. Some new methods have also been presented to overcome the drawbacks, including image rotation, shadows, darkness, and deformation [9, 10]. However, as the database becomes larger, traditional palm vein recognition techniques are prone to have higher time complexity, which has a negative effect on practical applications.

Recently, deep learning, which is one of the most promising technologies, has disrupted traditional cognition and has also been introduced into the field of palm vein recognition. Fronitasari et al [11] presented a palm vein extraction method that is a modified version of local binary pattern (LBP) and combined it with probabilistic neural network (PNN) for matching. In addition, supervised deep hashing technology has attracted more attention to large-scale image retrieval due to

its higher accuracy, stability, and lower temporal complexity in recent years. Lu et al [12] proposed a new deep hashing approach for evolutionary image retrieval by a deep neural network to exploit linear and nonlinear relationships. Liu et al [13] proposed a supervised deep hashing (DSH) scheme for fast image retrieval combined with a CNN architecture. The superior performance of deep hashing approaches for image retrieval prompts researchers to expand the applications of deep hashing from image retrieval to biometrics.

Finger vein can be used in combination with other modalities to improve accuracy. Kang and Kang [18] presented a multimodal biometric recognition based on the fusion of finger vein and finger geometry. The recognition accuracy was significantly improved. Yang and Zhang [16] proposed a multimodal biometric approach based on the fusion of finger vein and fingerprint feature levels by CCA. The proposed SLPCCAM approach performed well for person recognition. Xi et al [17] presented an effective personalized fusion approach combining finger outline and finger vein at the score level. The CCS-based weight fusion method exploits additional classification information and improves the recognition performance.

Recently, a number of multimodal works have attempted to combine ECG and other modalities to improve performance. ECG has been combined with electroencephalogram (EEG) for human identification [4]. Fatemian et al [3] presented a decision-level fusion approach by combining ECG and phonocardiogram (PCG) to achieve improved identification performance. al [2] fused ECG with face to improve identification performance. A novel multimodal and behavioral biometric system that combines ECG and audio signal was proposed in [14]. The approach used popular statistical coefficients that were simple and computationally efficient. In [15], ECG signals were fused with a traditional fingerprint lividity detection approach to achieve better detection performance, and also with a fingerprint identification approach for higher recognition rate.

B. DORSAL HAND VERIN IMAGE COLLECTION

The hardware configuration has a crucial role in the frame grabbing of veins. Two aspects can be underlined. The real camera used to take the stereotype has only one important parameter, the answer to the near infrared radiation [16]. The space resolution and the framework rate are of less importance since, for the acquisition of a model of vein, the image is necessary and the details are easily visible, even to a lower resolution. In [Cri] the design of the lighting system is one of the most important aspects of the process of frame grabbing. A good lighting system will provide precise data contrasts between the surrounding veins and fabrics while keeping Illuminations of the errors to a minimum. For a lighting system, they used IR LED because they offer a high contrast and easily available. But with a LED table formed using IR LED, they do not give a uniform illumination. Various arrangements were needed stamps LED to modify lighting. The LED is laid out like a simple or double table in 2D or rectangular table or concentric networks. Concentric arrangement table LED gives a better distribution of the light with only one or more concentric network of the LED and lens of the camera in the center of the image can acquire good contrasts.

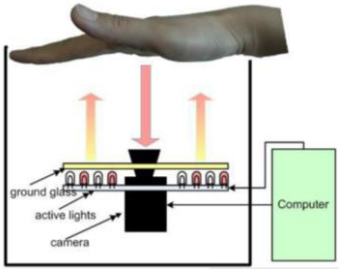


Fig 1 : Other system similar but which collects the veins of the fingers[HIT06].

II. PREPROCESSING AND EXTRACTION OF THE ROI

Preprocessing is the basis for feature extraction and matching. The quality of preprocessing has a significant impact on the recognition results. We mainly focus on the development of ROI extraction algorithms. This is indeed the main step of preprocessing, apart from other procedures such as image enhancement, image filtering, etc. The ROI is used to align different hand dorsal vein images and to segment the center for feature extraction. Most ROI extraction algorithms use the key points between the fingers to establish a coordinate system.

A. ***Aquiring the ROI image:***

In vein images, the region of interest is only the region that contains the vein pattern information. We therefore extract the region of interest (ROI) from the image.

To speed up the processing time and to standardize the collection of vein images, Fig.2 (b) was masked on the raw image taken and the boundaries of the hand surface were determined. Then, the outline of the image taken by the mask Fig.2(c) is taken and the highest point and the point at the right end Fig.2(d) .

Then from these two points we take the x-coordinate of the highest point and the y-coordinate of the point on the right to have the center point Fig.2 (e). And this point will be the center point of the square with 100 pixel to all directions (100 pixel to the top, left, right and bottom) Fig.2(f) [personal contribution].

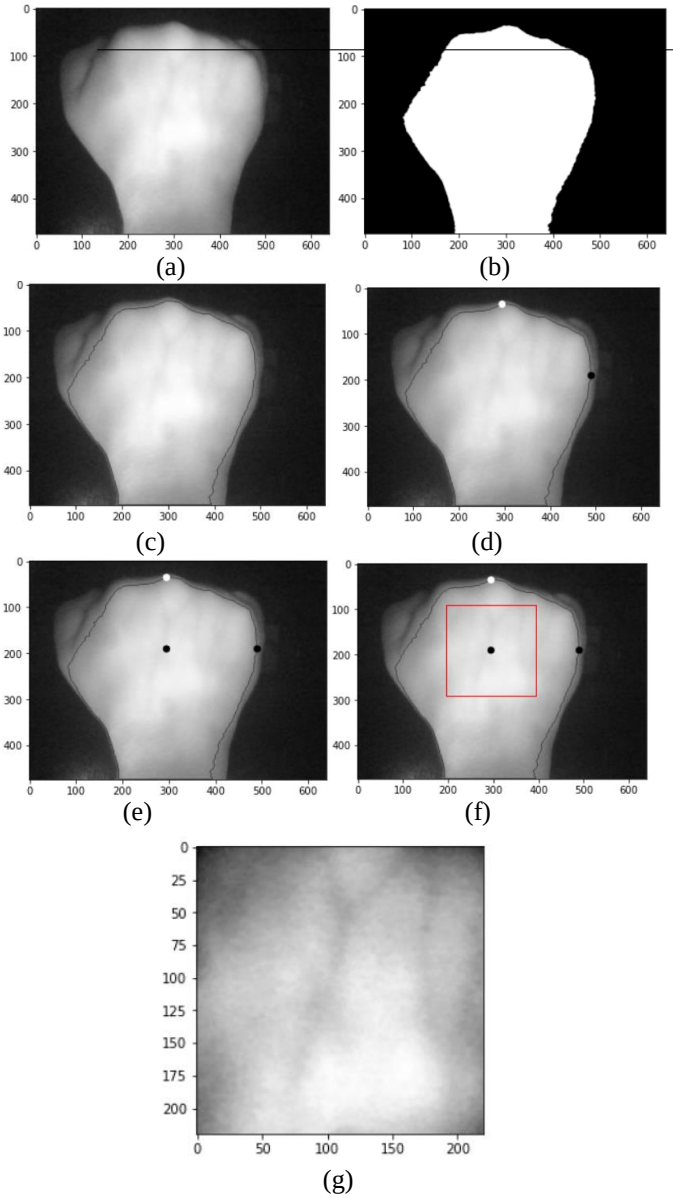


Fig.2 : (a). the original (raw) image (b).contour mask(c).the contour on the original image (d.) extraction of the highest point and the point at the right end (e.)the center point from the two points(f.)the square obtained from the center point (g.) the ROI image

*B. **Image processing***

Ordinary AHE tends to overamplify the contrast in the near-constant regions of the image, because the histogram of these regions is very concentrated. As a result, AHE can cause noise amplification in the near-constant regions. Contrast-limited AHE (CLAHE) is a variant of adaptive histogram equalization in which the contrast amplification is limited, so as to reduce this noise amplification problem [19].

In CLAHE, the contrast amplification near a given pixel value is given by the slope of the transformation function. This is proportional to the slope of the cumulative distribution function (CDF) of the neighborhood and thus to the value of the histogram at that pixel value. CLAHE limits the amplification by cutting the histogram at a predefined value before calculating the CDF. This limits the slope of the CDF and thus the transformation function. The value at which the histogram is clipped, called the clipping limit, depends on the normalization of the histogram and thus the size of the neighborhood region. Common values limit the resulting amplification to between 3 and 4.

It is advantageous not to discard the part of the histogram that exceeds the clipping limit, but to redistribute it equally among all bins in the histogram [19].

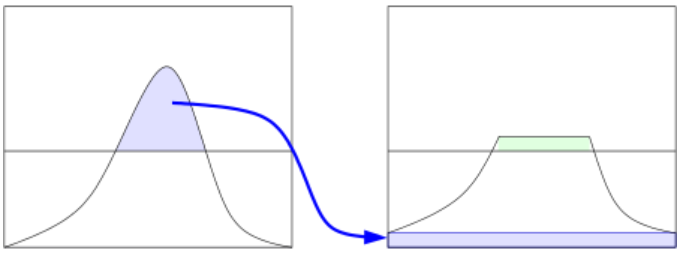


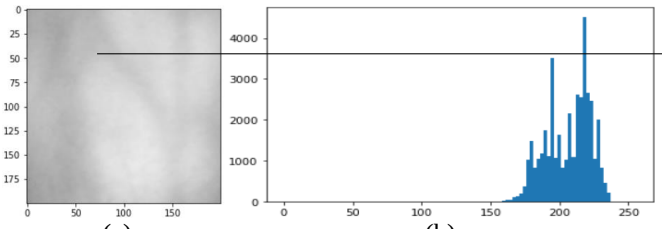
Fig.3 Adaptive equalization of the contrast limited histogram

The redistribution will cause some bins to rise above the clipping limit (green shaded region in the figure), resulting in an effective clipping limit that is higher than the prescribed limit and whose exact value depends on the image. If this is not desirable, the redistribution procedure can be repeated recursively until the excess is negligible.

After acquiring the ROI image, we convert it from the BGR format to the LAB color format (which expresses color in three values: L: for perceptual brightness, and A and B: for the four unique colors of human vision: red, green, blue and yellow).

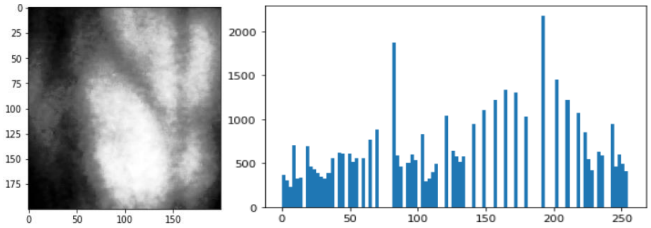
Then we separate the values and take only the L value. Then we create a CLAHE instance with the right parameters and apply it to the L value.

After that we take the equalized L-value and merge it with the A- and B-values and convert it back from LAB format to BGR format.



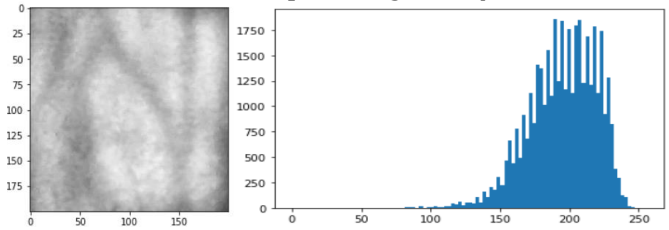
(a) (b)

Fig.4: a: original ROI image b: histogram of L



(a) (b)

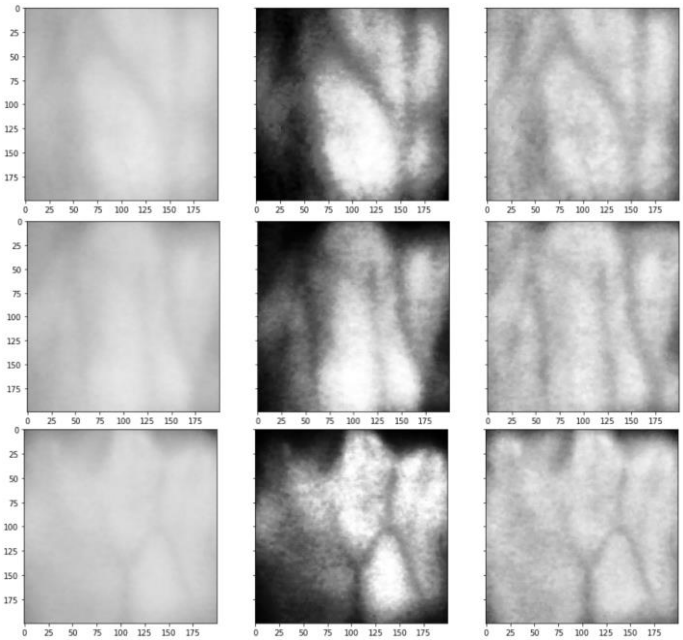
Fig.5: a: Image after merging the equalized L-value b: L-value with a simple histogram equalization



(a) (b)

Fig.6: a:image after merging the value legalized by CLAHE b: the value L with a CLAHE equalization.

C. Results :



(a) ***(b)*** ***(c)***

Fig.7: a: ROI image b: image processed by simple histogram equalization c: image processed by CLAHE

III. FEATURE EXTRACTION :

A. SATO filter :

The second derivative has typically been used for line enhancement filtering. The Gaussian convolution is combined with the second derivative in order to tune the filter response to the specific widths of lines as well as to reduce the effect of noise. In the one-dimensional (1-D) case, the response of the line filter is given by :

$$R(x;\sigma_f)=\{-\frac{d^2}{dx^2}G(x;\sigma_f)\} *I \qquad (7)$$

where * denotes the convolution, I(x) is an input profile function, and $G(x;\sigma_f)$ is the Gaussian function with the standard deviation σ_f defined as $(1/\sqrt{2\pi\sigma_f})exp\ (-\frac{x^2}{2\sigma_f^2})$.The sign of the Gaussian derivative has been inverted so that the responses have positive values for a bright line. We consider a profile having the Gaussian shape given by :

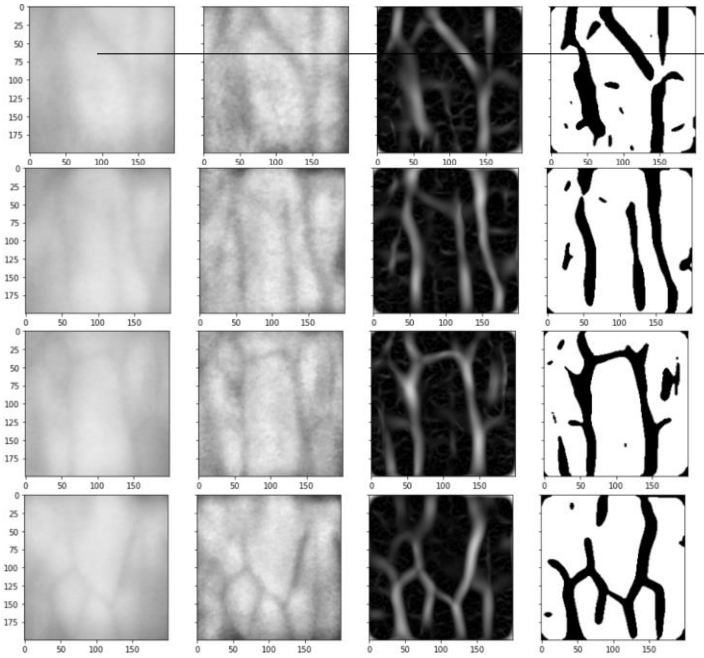
$$L(x;\sigma_x)=exp\ (-\frac{x^2}{2\sigma_f^2}) \qquad (8)$$

B. OTSU thresholding algorithm :

Otsu's thresholding method corresponds to the linear discriminant criterion which considers the image to consist of only the object (foreground) and background, and the heterogeneity and diversity of the background is ignored [20]. Otsu set the threshold to try to minimize the overlap of class distributions [20]. Given this definition, Otsu's method segments the image into two light and dark regions T0 and T1, where region T0 is a set of intensity levels from 0 to t or, in ensemble notation, T0={0,1,...,t} and region T1={t,t+1,...,l-1,l} where t is the value of the threshold, l is the maximum gray level of the image (e.g. 256). T0 and T1 can be assigned to the object and the background or vice versa (the object is not necessarily always located in the bright region). The Otsu thresholding method scans all possible threshold values and calculates the minimum value for the pixel levels on either side of the threshold. The goal is to find the threshold value with the minimum entropy for the sum of the foreground and background. Otsu's method determines the threshold value based on the statistical information of the image where, for a chosen threshold value t, the variance of clusters T0 and T1 can be calculated. The optimal threshold value is calculated by minimizing the sum of the weighted cluster variances, where the weights are the probability of the respective clusters.

C. Experimental results :

We take the enhanced ROI image and convert it to gray scale to lighten it and reduce the amount of data. Then we apply the sato filter with the default settings as they are suitable for what we need. And after that we apply a median filter of kernel size 5 to smooth the edges of the image. Finally we use the Otsu algorithm to acquire the mask [personal contribution].



(a) (b) (c) (d)

Fig.8: a: original ROI image b: ROI image enhanced with CLAHE c: application of the sato filter on the enhanced image d: mask obtained from c .

IV. CLASSIFICATION

A. *CNN Architecture :*

Set of parallel features maps are developed by sliding diff erent kernels over the input images and stacked together in a layer which is called as Convolutional layer. Using smaller dimension as compares with original image helps the parameters sharing between the feature maps. In the case of overlapping of the kernel with images, zero padding is used to adjust the dimension of the input images. Zero padding also introduce to control the dimensions of the convolutional layer. Activation function decides which neuron should be fired. The weighted sum of input values is passed through the activation layers. The neuron that receives the input with higher values has the higher probability of being fired. Diff erent types of activation function are developed for the past few years that includes linear, Tanh, Sigmoid, ReLu and softmax activation functions. In practice, it is highly recommended that selection of activation function should be based deep learning framework and the field of application. Down sampling of the data is carried out in pooling layers. It reduces the data point and overfitting of the algorithm. Also, pooling layer reduces the noises in the data and smoothing the data. Usually pooling layer is implemented after the convolution process and non-linear transformations. Data points derived from the pooling layers are stretched into single column vectors and fed into the classical deep neural networks. The architecture of a typical ConvNet is given in the Fig.9. The cost function, also known as loss function is used to measuring the performance of the architecture by utilizing the actual y_i and predicted $\hat{y}_i$.

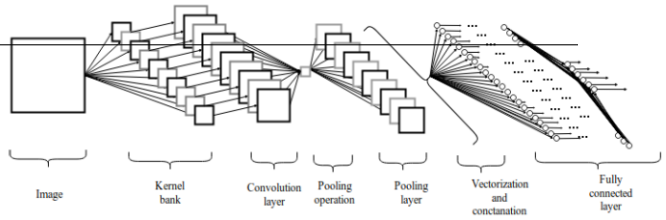


Fig.9: Architecture Convolution Neural Network

B. *Experimental results :*

The image pool is collected from a specified event as discussed in the database definition. The sampled images are divided into training and test data such that the training and test data contain 0.8% and 0.2% of the total amount of images. The training set is used for architecture training, and the test set is used for architecture validation. For computational simplicity, gray-scale images are used for network training. The input image windows are chosen from 64×64 pixel windows including the image information. However, the windows are carefully chosen to capture the rich amount of information about the data as well as to reduce the noise in the input and also to reduce the number of false positives. The image normalization procedure is performed in all training and test sets prior to array training. Batch gradient descent with the addition of the Adam optimizer is used to minimize the error. The network is trained for 40 training epochs, and in each epoch the images in the training set are randomly shuffled. In practice, it is understood that the batch size 32 turns out to be eff ective. Therefore, the training batch size is fixed to 32. Along with the batch size hyperparameter, other hyperparameters such as activation layer types and cost function are chosen manually. The ReLU activation function is used in the intermediate layers and the softmax is introduced in the classification layer of the architecture. The categorical cross-entropy loss function is used to measure the error between the actual and predicted values. However, the hyperparameters such as the number of convolutional layers, dense layers, convolutional kernel size, dense layer size, dropout, weight regularization, learning rate are determined from the Bayesian optimization algorithm. The activation and pooling layers are implemented after each convolutional layer. The Bayesian optimization algorithm is eff ectively introduced to optimize a large number of hyperparameters of the search space.

For the eff ective implementation of the optimization algorithm, it is common to perform 10N trials where N is the number of hyperparameters. In this context, 1000 trials are performed for hyperparameter tuning. In the Bayesian optimization implementation, the number of convolutional layers, pooling layers and the number of dense layers vary from 1 to 6 with the interval of 1, the number of convolutional kernels varies from 32 to 526 with the interval of 32, the units of dense layers vary from 128 to 1004 with the interval of 128, and the continuous type hyperparameters such as dropout, regularization L1 , L2 and learning rate vary from 0 to 0.2, 0 to 0.2, 0 to 0.2 and 0 to 0.2 respectively. No additional

hyperparameters are considered and the default value of the Keras deep learning framework is used. From the range of values, the optimal values are determined with respect to the cost function. The resulting values are given in Table 2.

The finale ConvNet architecture Fig. 21 is developed with four convolutional layers followed by activation and pooling layers, four dense layers. The size of the architecture is determined by Bayesian optimization, resulting in a total of 11 layers excluding the output classification layers. The size of the convolutional kernels is 3 × 3 on each convolutional layer where in the pooling layers, the fixed pixel window size of 2 × 2 is selected. The input image window is selected as 64 × 64 and is fed into the architecture for training. The output of the architecture is developed to classify the 26 categorical images of the data sets. The developed architecture has about 2000 trainable parameters and took 2 minutes for training. The elementary features of the images are acquired in the middle layers, and the combined complex features are found in the final convolutional layer. Because the ReLU activation function removes unwanted features and noise, only natural image information is likely to be transferred through the layers. From layer 1 to 7, there is a series of successive convolutional layers and an activation layer followed by pooling layers. All convolutional layers are developed by employing 3 × 3 filters, known as convolutional kernels and connected to nonlinear ReLU activation functions. The output of the convolutional layers is used to form activation layers and is then subsampled using the max pooling operation in the pooling layers. The last four layers are fully connected dense layers that use the features extracted from the previous layers to classify the images. Each layer receives information from the previous layers.

Hyper-paramètres	Hyperparameter Space	Type	Best fit
Input shape	[3,64,64]	Fixed	-
Number of Convolutional Layer	[1,2,3,4,5,6]	Discrete	[2]
Number of Convolutional kernel	[32,64,128,526]	Discrete	[32],[32], [64],[64]
Size of Convolutional kernel	(5,5)	Fixed	-
Activation unit in convolutional layer	ReLU	Fixed	-
Number of Pooling Layer	[1,2,3,4,5,6]	Discrete	[3]
Size of Pooling Layer	(2,2)	Fixed	-
Dropout	[0-0.2]	Continuous	[0]
Number of dense hidden layers	[1,2,3,4,5,6]	Discrete	[8]
Hidden layer size	[128,256,502,1004]	Discrete	[128]
Dropout	[0-0.2]	Continuous	[0.1]
Activation unit in dense layer	ReLU	Fixed	-
Weight regularization on dense layer, L1	[0-0.2]	Continuous	[0.2]
Weight regularization on dense layer, L2	[0-0.2]	Continuous	[0.2]
Learning rate	[0.01-0.00001]	Continuous	[0.00001]
Loss function	cross-entrophy	Fixed	-
Activation unit in classification layer	Softmax	Fixed	-

Table 1 Search space for hyperparameters

Couches Attributs	C-size	Act or P-size	Params	O/P shape
Conv2d	(5×5)	-	2432	(non,60,60,32)
Activation	-	ReLU	0	(non,60,60,32)
maxpooling2d	-	(2×2)	0	(non,30,30,32)
Conv2d	(5×5)	-	51264	(non,26,26,64)
Activation	-	ReLU	0	(non,26,26,64)
maxpooling2d	-	(2×2)	0	(non,13,13,64)
Flatten	-	-	0	(non,10816)
Dense	-	-	692288	(non,64)
Activation	-	ReLU	0	(13)
Dense	-	-	1690	(non,26)
Activation	-	Softmax	0	(26)

Table 2 Learning parameters

*C. **Training of the architecture :***

Convolutional layer 1 (Conv-1) is composed of 32 convolutional kernels of dimension (32 × 60 × 60). Each unit of the convolutional kernels is connected to neighboring units of the input image (5 × 5), resulting in 747,674 trainable parameters.

A weighted sum of the feature map inputs with trainable bias addition is computed and passed through sigmoidal nonlinear activation functions. Then, subsampling of the convolved images is performed in pooling layer 1 (pooling1).

The output of the pooling layer is fed into the following convolutional layers. Convolutional layer 2 (Conv-2) contains 32 convolutional kernels and is formed with the dimension of 26 × 26. Each unit calculates the weighted sum and adds the resulting bias with 44 trainable parameters.

Then they are passed through nonlinear activation functions. The feature maps are subsampled from the total units in pooling layers 2.

Therefore, the feature extraction layers have two convolutional layers with two activation layers followed by two subsampling activation and pooling layers where stable low-dimensional features are extracted and used for classification by the fully connected dense layer. These parameters are concatenated and vectorized into single-column vectors and fed into the classical multilayer perceptron, known as the fully connected dense layers.

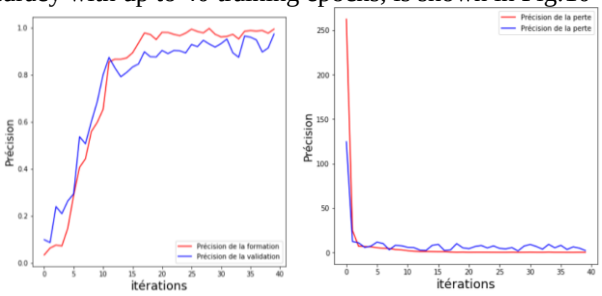
From the Bayesian optimization, we find that two layers of dense layers are formed with 64 units in each layer.

The units in each layer are dropped with 0.2% of the total units and added with L2 and L1 regularization.

The fully connected dense layers act as a classifier, and the preceding convolutional layers act as feature extractors.

In the fully connected dense layer, each of the units is connected with vectorized parameters from the pooling layer-2. Similarly, the single unit in a dense layer is fully connected with all units in the previous dense layers. In these layers, the dot product is performed with the weight and input values and added with the bias values. The weighted sum of the input is passed through the nonlinear activation function (ReLU).

The final output of the dense layers is passed through the softmax activation where the categorical cross-entropy is implemented to measure the error between the actual and predicted values. As a result, the fully connected dense layers have 692288 trainable parameters. The network is trained using batch gradient descent, a modified version of the stochastic gradient descent algorithm. The Adam optimization algorithm is implemented to accelerate the convergence of the gradient. The performance of the network, such as loss and accuracy with up to 40 training epochs, is shown in Fig.10



a:accuracy *b:loss*

Fig.10: Learning curves a :learning and learning validation b :loss and loss validation

The red curve represents the performance of the architecture on the training set, and the blue curve represents the performance of the architecture on the validation set. The early stopping algorithm is implemented to avoid overfitting. The early stop tends to stop the training procedure if the performance of the test data is not improved after the fixed number of iterations. It avoids surfing by attempting to automatically select the inflection point where performance on the test data set begins to decline while performance on the training data set continues to improve as the model begins to surfiter. Overall accuracies of 95% and 99.5% for the training and validation test images are achieved on the training architecture.

V. CONCLUSION

Biometrics is a growing technology. It is more and more used in applications related to security because of the advantages it offers in contrast to older methods.

In this thesis, we have studied a biometric identity verification system based on the dorsal vein network of the hand. After a presentation of the general context of our study, the fundamental concepts of biometric systems and methodology are studied in the first chapter (state of the art). Then, we detailed the different components of the hand dorsal vein identification system proposed in chapter 2 and 3.In chapter 4 we tuned the neural network parameters in such a way that our system successfully identified the images in the database.

In order to give more value to our work, we plan to test our system on large databases and by applying other feature extraction methods.

References

[1] 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

[2] X. Zhu and D. Huang, "Hand dorsal vein recognition based on hierarchically structured texture and geometry features", In: Proc. Chin. Conf. Biometric Recognit., pp. 157-164, 2012.

[3] H. Wan, L. Chen, H. Song, and J. Yang, "Dorsal hand vein recognition based on convolutional neural networks", In: Proc. IEEE International Conf. Bioinformatics Biomedicine, pp. 1215-1221,2017.

[4] Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Trans. Circuits Syst. Video Technol. 14(1), 4–20 (2004)

[5] Liu, J., Xue, D.-Y., Cui, J.-J., Jia, X.: Palm-dorsa vein recognition based on kernel principal component analysis and locality preserving projection methods. J. Northeastern Univ. Nat. Sci. (China) 33, 613–617 (2012)

[6] Lajevardi, S.M., Arakala, A., Davis, S., Horadam, K.J.: Hand vein authentication using biometric graph matching. IET Biom. 3, 302–313 (2014)

[7] Chen, H., Lu, G., Wang, R.: A new palm vein matching method based on ICP algorithm. In: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, Seoul, pp. 1207–1211. ACM (2009)

[8] Bhattacharyya, D., Das, P., Kim, T.H., Bandyopadhyay, S.K.: Vascular pattern analysis towards pervasive palm vein authentication. J. Univers. Comput. Sci. 15, 1081–1089 (2009)

[9] Xu, X., Yao, P.: Palm vein recognition algorithm based on HOG and improved SVM. Comput. Eng. Appl. (China) 52, 175–214 (2016)

[10] Elsayed, M.A., Hassaballah, M., Abdellatif, M.A.: Palm vein verification using Gabor filter. J. Sig. Inf. Process. 7, 49–59 (2016)

[11] B. Hartung, D. Rauschning, H. Schwender, S. Ritz-Timme, A simple approach to use hand vein patterns as a tool for identification, Forensic Sci. Int. (2020) 110115.

[12] S.-X. Zhang, H.-M. Schmidt, Clinical anatomy of the subcutaneous veins in the dorsum of the hand, Ann. Anat. AnatomischerAnz. 175 (4) (1993) 381–384.

[13] M.A. Ferrer, A. Morales, L. Ortega, Infrared hand dorsum images for identification, Electron. Lett. 45 (6) (2009) 306–308.

[14] Rahul, R.C., Cherian, M., Manu Mohan, C.M.: 'A novel Mf-ldtp approach for contactless palm vein recognition'. 2015 Int. Conf. on Computing and Network Communications (CoCoNet), Trivandrum, India, December 2015, pp. 793–798

[15] Mirmohamadsadeghi, L., Drygajlo, A.: 'Palm vein recognition with local texture patterns', IET Biometrics, 2014, 3, (4), pp. 198–206

[16] Akbar, A.F., Wirayudha, T.A.B., Sulistiyo, M.D.: 'Palm vein biometric identification system using local derivative pattern'. 2016 4th Int. Conf. on Information and Communication Technology (ICoICT), Bandung, Indonesia, May 2016, pp. 1–6

[17] Piciucco, E., Maiorana, E., Campisi, P.: 'Palm vein recognition using a high dynamic range approach', IET Biometrics, 2018, 7, (5), pp. 439–446

[19] Tome, P., Marcel, S.: 'Palm vein database and experimental framework for reproducible research'. 2015 Int. Conf. of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, October 2015, pp. 1–7

[19] Image enhancement based on contourlet transform, Asmare, M.H., Asirvadam, V.S.,Hani, A.F.M. Signal, Image and Video Processing, 20 March 2014, DOI10.1007/s11760-014-0626-7

[20] N. Otsu. A threshold selection method from gray level histograms. IEEE Trans. systems. Man. and Cybernetics, 9:62-66, 1979.