Infrared Face Recognition Using Neural Networks and HOG-SVM

CHERIFI Dalila
Institute of Electrical and Electronic Engineering, University of Boumerdes, ALGERIA

KADDARI Redouane
Institute of Electrical and Electronic Engineering, University of Boumerdes, ALGERIA

ZAIR Hamza
Institute of Electrical and Electronic Engineering, University of Boumerdes, ALGERIA

NAIT-ALI Amine
LiSSi, EA 3956, Paris-Est Creteil (UPEC), FRANCE

***Abstract*— With the rapid development of computer vision applications, infrared face recognition is receiving much attention because of miniscule sensitivity to the face variations due to illumination changes. The infrared image is merely a gray level image but the recognition is interestingly good enough to look at. In this article we present an infrared face recognition system where we used two methods and two databases to train and test our system. We used the Histogram of Gradient (HOG) algorithm along with the Support Vector Machine (SVM) classifier as a first method. In the second method, we implemented the Backpropagation algorithm. Although the results of the HOG-SVM were very promising and we were able to get a better results compared with a previous work, neural network gave us a perfect results by using the same number of training examples.**

Index Terms: *Face Recognition, HOG-SVM , Back Propagation Algorithm, Biometric.*

I. INTRODUCTION

As digital systems became more complex and reliable, computer vision gained access to this advancement. One of the most influenced fields in computer vision is face recognition where the biometric systems got much interest in a wide variety of applications, mostly security. In many visible light face recognition systems, authentication and surveillance systems are not good enough for uncontrolled environments [1]. This is why we look for more efficient systems. Another problem with visible face recognition systems is that they limited by the visible light so they are of no use in the dark. For these reasons, an innovative solution is to look beyond the visible spectrum. Infrared (IR) systems. Many ideas were proposed to overcome the above problems and come up with powerful algorithms for recognizing the face just like the visible light systems but with better performance as it can recognize the face even in the dark, which makes it a plus for the system. Infrared image is robust to environmental changes and no need for visible light. All this seems too good and free but it is not. Because of a basic logical a principle that is if there are no photons, there is no image to be seen. A photon must be emitted in order for the light sensor such as the camera to detect it. In other words, unlike in visible light where the light is mostly available as part of another system such as lightning or perhaps the sunlight, infrared systems must emit their own infrared light if the body does not emit any. Hence, this infrared radiation must be considered when discussing the cost of the camera though the modern cameras have their built in infrared light sensor.

Many techniques are used in the field of infrared face recognition such as: template matching, thermal contours, feature extraction for metrics matching, wavelet analysis for face matching, symmetry waveforms with bar code encoding of face waveforms; the use of facial minutiae [2,3],other techniques used are inspired from their visible counterparts techniques can be divided into holistic and feature based techniques; recently, a great interest in the research community has focused on the fusion of different modalities. [4,5,6,7].

In this work, we implemented an infrared face recognition system using neural networks and HOG-SVM algorithms in order to increase the accuracy and overcome the limitations of the visible light systems.

II. RECOGNITION TECHNIQUES

A. Backpropagation algorithm

The Backpropagation algorithm is a neural network algorithm. Its basic idea is to forward propagate then it calculates the error starting from the output to the input. The concept of a neural network is a motivation from the fascinating functionality of the human brain. It has been always crucial to finding the correct combination of weights for the system to be reliable enough in predicting the output. In this part, we discuss a popular machine learning algorithm capable of handling large learning problems "the Backpropagation algorithm". The Backpropagation algorithm received much attention over the years and it has been one of the most used algorithms for neural networks [8]. The Backpropagation algorithm looks for the minimum of the error function in weight space using the method of gradient descent. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. Since this method requires computation of the gradient of the error function at each iteration step, we must guarantee the continuity and differentiability of the error function [8]. There exist many activation functions which are widely used in different algorithms like the sigmoid function that we used in the implementation of the Backpropagation algorithm.

The learning problem consists of finding the optimal combination of weights so that the hypothesis function $h_{\Theta}(x)$ predicts the output as accurate as possible. The performance of the hypothesis is analysed by comparing the predicted output with the obtained one as described below.

978-1-7281-3578-6/19/$31.00 ©2019 IEEE

Algorithm

This is a general description of the method to assign each output to its class correctly.

For a training set {($x^{(1)}, y^{(1)}$), ($x^{(2)}, y^{(2)}$), ..., ($x^{(m)}, y^{(m)}$),} we introduce$\Delta_{ij}^{(l)}$.

As an initialization we set $\Delta_{ij}^{(l)}$*=0 for all (i,j,l)* [9]

After that, iterating through all the inputs using the iteration index "i" we perform the following:

- *Set* $a^{(1)}=x^{(1)}$
- *Perform the forward propagation to compute* $a^{(l)}$ *for l= 1,2, ... L.*
  by computing and $Z^{(i)}$ *'s .*
- *Calculating the error that we want to minimize using a cost function based on the network's output. i.e. Use* $y^{(i)}$ *to compute* $\delta^{(L)}$
  $$\delta^{(L)} = a^{(l)} - y^{(i)}$$
- *Back propagating the error in the network then find its derivative with respect to each weight. i.e. Compute* $\delta^{(L-1)}, \delta^{(L-1)}, ...\delta^{(2)}$
  $$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)}\delta^{(l+1)}$$
- *Compute the derivatives* $D_{ij}^{(l)}$
  $$D_{ij}^{(l)} = \frac{1}{m}\Delta_{ij}^{(l)} + \lambda\theta_{ij}^{(l)},\ if\ j \neq 0$$

To speed up the calculations, we vectorize by using
$$\Delta^{l} := \Delta^{l} + \delta^{(l+1)}[a^{(l)}]^{T}$$

B. The Proposed Method "HOG + SVM"

In this method, we combine the HOG algorithm which is used for the features extraction with the SVM algorithm to classify the subjects. The calculation of the HOG feature descriptor can be achieved by the following steps:

1) The calculation of Histogram of Oriented Gradient

The calculation of the HOG feature descriptor can be achieved by the following steps:

a) Pre-processing

The HOG feature descriptor is used on the [320 x 240 pixels] images of the database. Typically, the image can be of any size. The only constraint is that the sub images (the patches) should have a fixed aspect ratio even though patches of different sizes are analyzed at different image locations.

b) Calculation of the Image Gradient

After pre-processing the image, the horizontal and the vertical gradients are calculated then the result comes in a form of histogram of gradients.

c) Calculate Histogram of Gradients in 8×8 cells

The image is divided into 8×8 cells. Each cell is accompanied with a histogram of gradients. At each pixel there exist two values (the magnitude and the direction) which gives a total of 128 values for each cell. The reason for this is that a histogram over 8×8 cell image is less sensitive to noise than that of an individual gradient .

d) Block Normalization

The purpose of normalizing the histogram is that the effect of light variation is eliminated. If the pixels are divided by any number say *k* (*k* is larger than 1), the image will become more dark. Keep in mind that it is always better to normalize over a bigger sized block.

e) Calculation of the HOG feature vector

The calculation of the feature vector is performed by concatenating the smaller patch vectors to build an overall vector.

2) Support Vector Machine (SVM) training

SVM is one of the legacy classifiers that are still used on datasets. SVM is one of the supervised machine learning algorithms which can be used for both classification and regression even though it is mostly used in classification problems, it is also called the large margin classifier. The data is plotted in an *N-dimensional* space with *N* being the number of features; the value of each point refers to the value of a particular feature. After this, data points are normally grouped in clusters so that the SVM algorithm classifies them according to the plane (for *N* being more that 2) that differentiates the two classes in case of binary classification. It is not always trivial to separate the classes when they are not linearly separable or when we have more than two classes for these reasons we introduce what is called kernels [10]. First, some landmarks are chosen, then new set of features are computed. This can be done by introducing the similarity function [10].

III. EXPERIMENTS AND RESULTS

We propose in this section the implementation of our two methods. In the first method, we used HOG for features extraction and SVM for classification and for the second method, we used the Backpropagation Algorithm.

A. Data Bases

In this work, we have used two different databases, one in the thermal spectrum and the other one in the NIR spectrum. The first database used is Terravic facial IR (Figure 1), The dataset set consists of 20 classes, we have used 18 classes as two classes were corrupted, for each class we have used 200 greyscale images of size 320 by 240 pixels with different variations (front, left, right, indoor/outdoor, glasses, hat), the total number of images is 3600 [11].

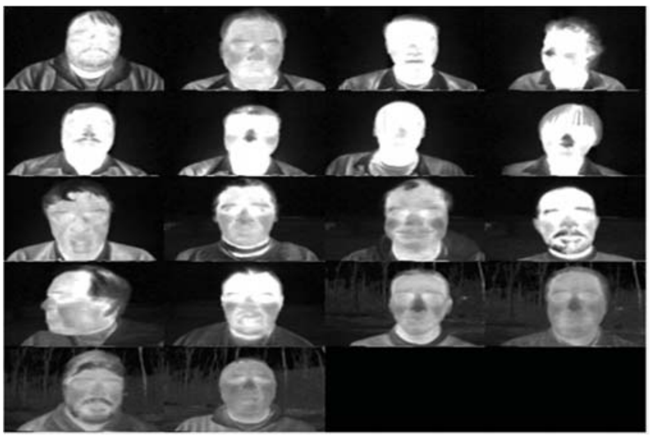

Fig.1: Samples of Terravic Facial IR Database

The second database used is CBSR NIR Faces (Figure 2), it consists of 197 classes, each class represents one face, the image size is 480 by 640 pixels, we limit ourselves to 30 classes, each class has 20 images, the variation in this database is changes in pose, expression and glasses [11].

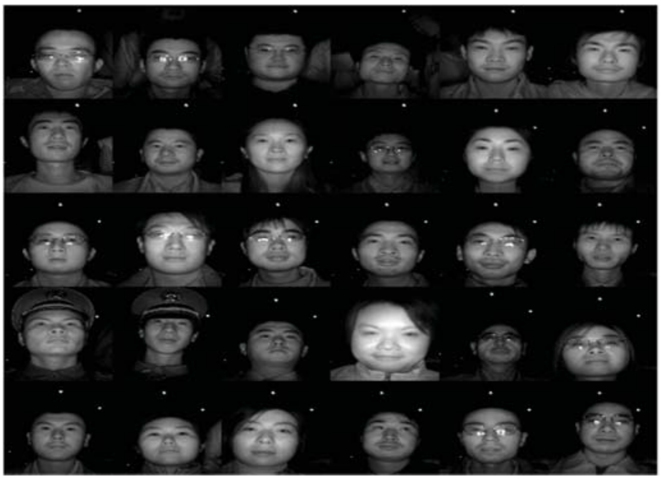

Fig 2 : Samples of CBSR NIR Database

B. *Implemented Methods*

1) *HOG-SVM Method*

First, we load our databases then we split each database into a training set to learn how to discriminate between the face and a test set so we can test our algorithm against, we used different sizes of both training and test sets. We extract the HOG features (Figure 3) of each face and we store it as a row vector, and we give each image the class number that belongs to. The first training set (Terravic Facial IR) is a matrix of 3600 by 40716 and the second training (CBSR NIR Faces) set is 600 by 40716. The rows represent total number of images in our training set, the columns represents the number of pixels of an image after extracting the HOG features. Then we pass the training set and its labels to SVM classifier, once the training is done, we extract the features of our test set and also we give each image the class number. Finally, we pass our classifier and we compare it with the true class in order to see if the predicted class is true or not.

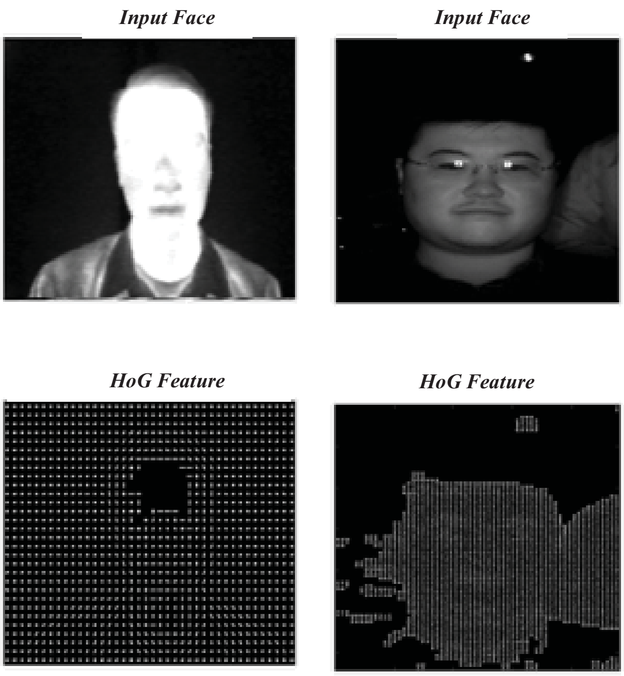

Fig. 3: Samples of extracting HOG feature

2) *Backpropagation based method*

a) *Model representation*

Our neural network has 3 layers: an input layer, a hidden layer and an output layer, as shown in Figure 4. The input layer has a size of the number of pixels for each database, so it has a size 76800 for the first dataset and a size 307200 for the second dataset (not counting the extra bias unit which always outputs +1). The hidden layer has 100 unit (without the extra bias unit), and the output layer has the size of the number of classes that we want to predict (18 for the first database and 30 for the second). Since we have used one hidden layer, we end up with two weight matrices $\Theta^{(1)}$ and $\Theta^{(2)}$.

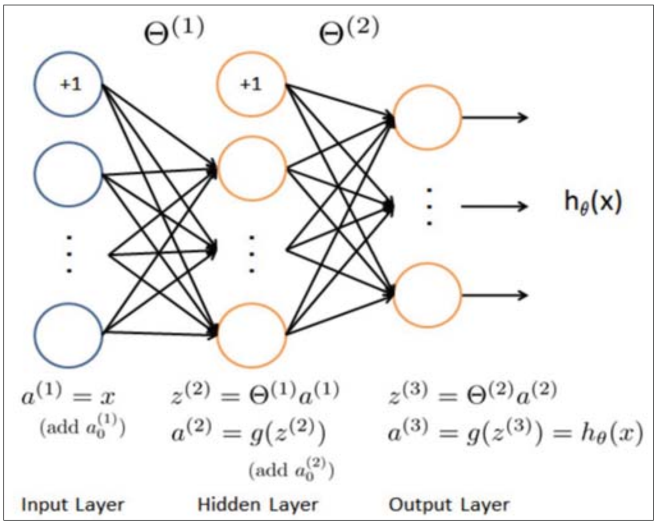

Fig. 4: Neural network model

b) *Database preprocessing*

Before start building our model, we have to prepare our training set and test set, let's take the first database as an example, it has 3600 image where each image is 320 by 240 pixel. This pixels are unrolled into a 76800-dimensional row vector, this give us 3600 by 76800 matrix **X** where every row is a training example of a face image. Vanishing gradient on sigmoid is a common problem in neural networks and it is caused when our input matrix has large values, which will make all the outputs after multiplying with the weight matrix and applying the sigmoid activation function either 1 or 0, if that is the case, the local gradient on sigmoid will be in both cases 0 ("vanish") as shown in Figure 5, so we will end up with dead neurons that do not learn from the network and this will let the cost function decreasing very slowly, so it is very important to work in a safer zone called an active region of the sigmoid.

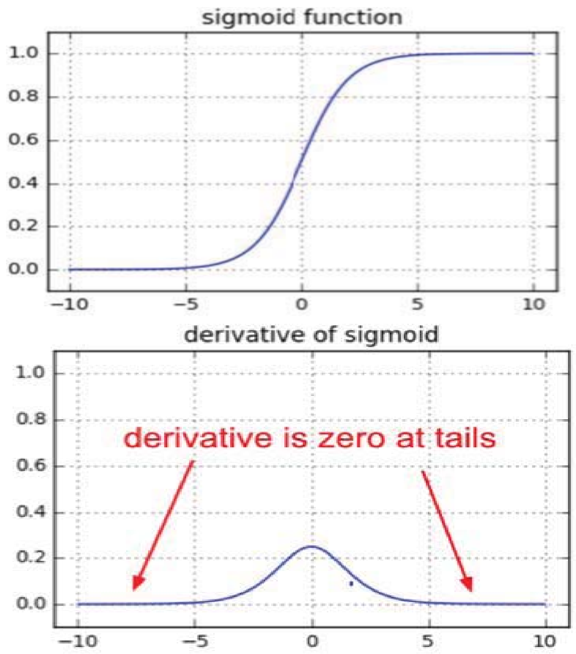

Fig. 5: Sigmoid and its derivation

To deal with this problem we have used the mean normalization, it means for every single pixel we compute its mean and standard deviation values over the training set and we subtract and divide that out.

The second part of the training set is a 3600-dimensional column vector $\boldsymbol{y}$ that contains labels for the training set. We split these two matrices into a training and test sets (Xtraining, Xtest, ytraining, ytest). The same thing is done to the other database.

c) Weight initialization

We have to pick some initial values for the weight parameters theta, if we initialize them all to zero, all the neurons output the same thing, if we initialize them with a big numbers we will end up with the vanishing gradient problem (Figure 6), it is important to randomly initialize the parameters with small numbers for symmetry breaking.

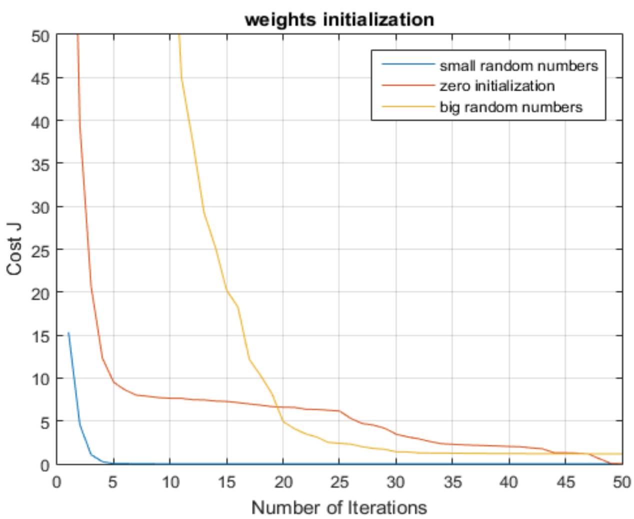

Fig. 6: Weights initialization

d) Number of neurons

There is no general rule for choosing the number of neurons, so we picked up a sample from our database and we pass it to the neural network to see how the recognition rate changes when we increase the number of neurons, it turned out that when we increase them the recognition rate increases as shown in Figure 7. In order to be sure when we have used the whole database, we have chosen the number of neurons to be 100.

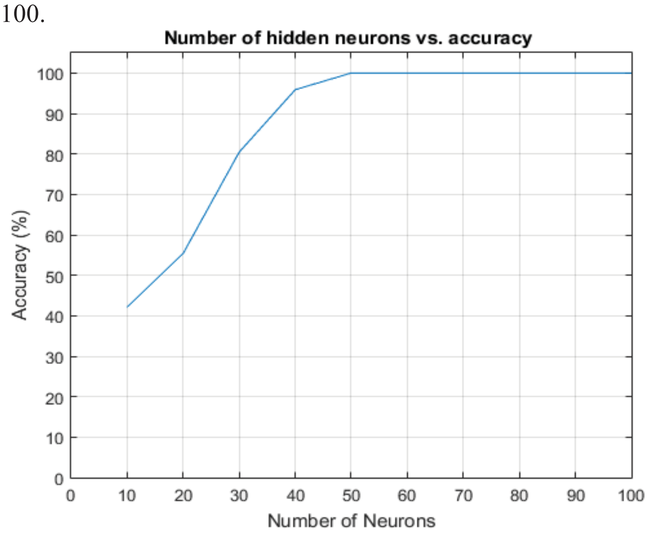

Fig. 7: Number of neurons accuracy

e) Regularization

Another important parameter that we should take care of is lambda, if it is too small we may end up under fitting our data, on the other hand, if it is too large we may over fit our data, after we tried many values for lambda, we get a value of ($\lambda = 2$).

f) Training the neural networks

First, we used 20% of the database as a training set and the remaining as a test set, we pass the training set to our model to get the weights theta, and then we use these weights to predict the output of test set using test function and we get the accuracy, after that we increased the size of the training set and see how well it is doing on the test set.

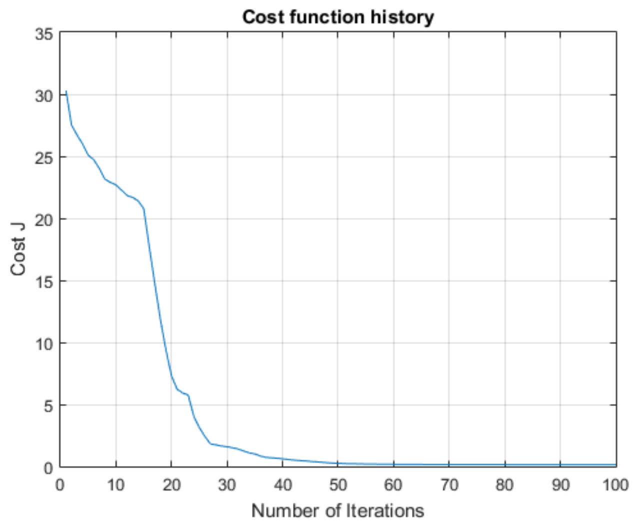

Fig. 8: Cost function history

After training and implementing the algorithms and training the system we record the classification precision in the following two tables:

TABLE 1: Results from Terravic facial IR Database

Method	Training set %	Test set %	Accuracy %
Back-Propagation	20	80	88.77
	40	60	100
	60	40	100
	80	20	100
HOG- SVM	20	80	72.63
	40	60	83.62
	60	40	94.78
	80	20	98.43

TABLE 2: Results from CBSR NIR Database

Method	Training set %	Test set %	Accuracy %
Back Propagation	20	80	86.87
	40	60	97.50
	60	40	100
	80	20	100
HOG- SVM	20	80	84.37
	40	60	88.33
	60	40	91.25
	80	20	95.83

Table 1 and 2 shows the recognition rate considering the Terravic facial IR and CBSR NIR databases, respectively. Two remarks can be noticed, the first one: the more data we used in the training set, the higher accuracy was obtained, this means that if we use a large database we will get a higher accuracy, the second one: the Backpropagation method is better than HOG-SVM because even without feature extraction, it could classify with 100% accuracy when we used only 40% as a training set of the first database and 60% of the second database whereas the best accuracy for the HOG-SVM method is 98.43% for the first database and 95.83% for the second.

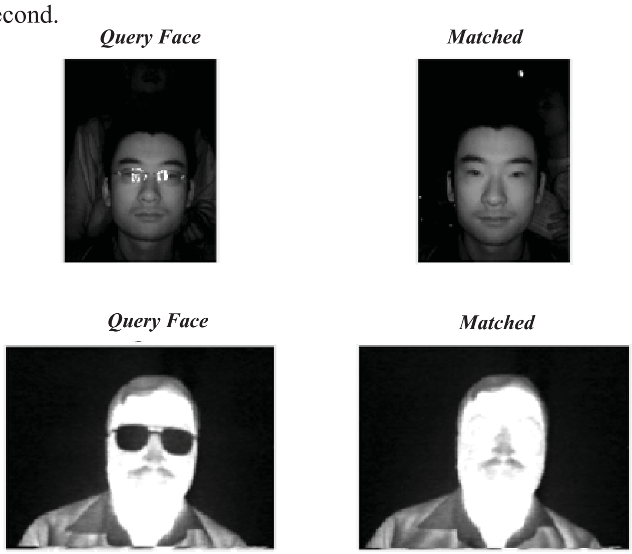

Fig. 9: Samples of matched faces from the two databases

In Gaber and al. [12], they applied PCA and LDA on the same database. Using this methods the best recognition rate they could achieve was 94.12% (See Table 3).

TABLE 3: Results from using Terravic Facial IR [12]

Dimensionality reduction method	Training set %	Test set %	Accuracy %
Principle Component Analysis (PCA)	80	20	94.12
Linear Discrimenant Analysis (LDA)	80	20	94.12

Comparing the results we obtained with a related work [12]. While we achieved 98.43% using HOG- SVM and 100% using Backpropagation method.

IV. CONCLUSION

Infrared face recognition is a very interesting research area but there are a few number of researches about it compared with the visible face recognition. In this paper, we have presented two different methods for infrared face recognition using two different databases. Implementing Backpropagation using only one hidden layer we were able to classify all the faces of the thermal and the NIR databases correctly without extracting any features. HOG-SVM method is also able to classify with a high accuracy and it is better than related work in [12]. Compared with the traditional recognition methods for the infrared face recognition, which mainly take the following steps: face data are pre-processed and the features are then extracted, a dimensionality reduction is used to reduce the size of the data and then a classifier is trained to predict the output of the test set, Backpropagation could classify our test set with 100% accuracy using only a few data. The results of the two algorithms were acceptable, but, but we have to reduce the running-time of the training step.

REFERENCES

[1] http://www.flir.com/cores/display/?id=51886
[2] Prokoski F., *"History, current status, and future of infrared identification",* Proceedings of The IEEE Workshop on Computer Vision Beyond the Visible Spectrum (CVBVS'00), Hilton Head, SC, June, 2000
[3] Kong S., Heo J., Abidi B., Paik J., Abidi M., *"Recent advances in visual and infrared face recognition: a review"*, Computer Vision & Image Understanding, vol. 97, pp. 103-135, 2005.
[4] Chen X., Flynn P., Bowyer K., *"IR and visible light face recognition"*, Computer Vision & Image Understanding, vol. 99, pp. 332-358, 2005.
[5] Moulay Akhloufi, Abdel Hakim Bendada and Jean-Christophe Batsale, *"State of the art in infrared face recognition"*, Quantitative Infrared Thermography Journal (QIRT), vol. 5, no 1, pp. 3-26, 2008.
[6] Reza Shoja Ghiass, Ognjen Arandjelovic, Hakim Bendada, Xavier Maldague.*"Infrared face recognition: a comprehensive review of methodologies and databases"*. Pattern Recognition. January 2014. DOI: 10.1016/j.patcog.2014.03.015.
[7] Krishna Rao Kakkirala and Srinivasa Rao Chalamala and Santosh Kumar Jami, *"Thermal Infrared Face Recognition: A Review"*, 2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim), pp 55-60, 2017.
[8] Raul Rojas *"Neural Networks - A Systematic Introduction"*. Springer-Verlag, Berlin, New-York, 1996.
[9] http://www.livescience.com/50260-infrared-radiation.html
[10] Mrinal Kanti Bhowmik, Kankan Saha, Sharmistha Majumder, Goutam Majumder, Ashim Saha, Aniruddha Nath Sarma, Debotosh Bhattacharjee, Dipak Kumar Basu and Mita Nasipuri. *"Thermal Infrared Face Recognition – A Biometric Identification Technique for Robust Security system"*. Chapter book in: Reviews, Refinements and New Ideas in Face Recognition book, Published: July 27th 2011, ISBN: 978-953-307-368-2, Jadavpur University India.
[11] http://vcipl-okstate.org/pbvs/bench/
[12] Tarek Gaber, Alaa Tharwat, Abdelhameed Ibrahim, Václav Snáel, Aboul Ella Hassanien. *"Human Thermal Face Recognition Based on Random Linear Oracle (RLO) Ensembles"*, International Conference on Intelligent Networking and Collaborative Systems; Taipei, Taiwan, 2-4 Sept. 2015.