*Journal of Computer Science*

Original Research Paper

# Thermal Face Authentication with Convolutional Neural Network

**Mohamed Sayed and Faris Baker**

*Faculty of Computer Studies, Arab Open University, Kuwait*

*Article history*
Received: 25-10-2018
Revised: 10-11-2018
Accepted: 08-12-2018

Corresponding Author:
Mohamed Sayed
Faculty of Computer Studies,
Arab Open University, Kuwait
Email: msayed@aou.edu.kw

**Abstract:** Matching thermal face images as a method of biometric authentication has gained increasing interest because of its advantage of tracking a target object at night and in total darkness. Therefore, for security purposes, it has become highly favourable and has extensive applications, for instance, in video surveillance at night. The aim of this study is to present a simple and efficient deep learning model, which accurately predicts person identification. A pre-trained Convolutional Neural Network (CNN) is employed to extract the features of the multiple convolution layers of the low resolutions' thermal infrared images. To run the program and evaluate the performance, we use a sample of 1500 resized thermal images, each with resolution 181×161 pixels. The sample comprises of images that were captured within different time-lapse and with diverse emotions, poses and lighting conditions. The proposed approach is effective compared to the state-of-the-art thermal face recognition algorithms and achieves impressive accuracy of 99.6% with less processing and training times.

**Keywords:** Deep Learning, Convolutional Neural Networks, Image Processing, Face Recognition, Thermal Images

## Introduction

Recognitions with a thermal camera is a challenge that have been recently improved by adopting deep learning methods using Convolutional Neural Networks (CNNs). Thermal camera forms a picture by capturing various heat levels emitted from objects. One of the applications of visualizing and identifying objects is face recognition. There are many useful applications for face recognition; an important one is an application for security at night where an intruder needs to be identified in the absence of light. Furthermore, it can be used as an additional tool to normal cameras for identification. Ensemble method of using both thermal and normal cameras adds more certainty weight to the proof of identity. In this study, faces are identified in different circumstances such as full light, partial light and dim light as well as in different emotional status, all based on the heat emitted from them rather than the light intensities reflected from them; the technology also allows us to measure the accuracy of the recognition using CNNs.

Image processing (Nixon and Aguado, 2002) is a technique that is used to change over a picture into an enhanced computerized shape. It plays out a few operations together with a specific end goal to get an improved picture or to concentrate some valuable data from it. It is also a sort of flag regulation which information is a picture similar to a video edge or photo and might yield a picture or attributes related with that picture. Typically, image-processing framework incorporates pictures as two-dimensional signs while effectively applying strategies of set flag handling to them. It is among quickly developing advances today together with its applications in different parts of a business. Advanced processing methods help in controlling the computerized pictures by utilizing personal computers. The three general stages that a wide range of information needs to experience while utilizing computerized system are pre-handling, improvement and show and data extraction.

CNNs are a special kind of multi-layer neural networks trained with a back-propagation algorithm that extracts important features, supplemented with filters that prevent overfitting and eliminate noise modelling. While a fully connected layer has a high cost of parameters and high risks of overfitting, the convolutional layer is more suitable for 2D spatial information because it captures the spatial characteristics

and uses fewer parameters (Hadji and Wildes, 2018). The performance of the network is largely based on the value of the weights calculated between the neural connections in the layers (Krizhevsky *et al*., 2012) and (Krizhevsky *et al*., 2017). There are many convolutional neural architectures that evolved over previous years and serve as feature extractors, widely known in the literature are: Krizhevsky *et al*. (2012), (LeCun *et al*., 1998), (Szegedy *et al*., 2015), (Simonyan and Zisserman, 2014), (Szegedy *et al*., 2017) and (Huang *et al*., 2017). Some CNN architectures in the literatures that accumulated over years of experimentation and competition includes LeNet (LeCun *et al*., 1998), AlexNet (Krizhevsky *et al*., 2012), GoogleNet (Szegedy *et al*., 2015), ResNet (He *et al*., 2016) and DensNet (Huang *et al*., 2017). In this experiment, we use Visual Geometry Group (VGG), a CNN model proposed by Simonyan and Zisserman (2014) at the University of Oxford to calculate the accuracy. This architecture is implemented because of its simplicity and accuracy over other similar applications such as Modified National Institute of Standards and Technology (MNIST) database.

This study discovers a CNN algorithm that is beneficial for the research on image recognition for various applications such as medical diagnosis (Lahiri *et al*., 2012), security monitoring, thermal vision for autonomous vehicles, agriculture and biometrics; see for instance (Vadivambal and Jayas, 2011) and (Khanal *et al*., 2017). Furthermore, the study might help researchers to uncover and explore areas of emotions and security monitoring that many researchers were not able to explore, using a state-of-the-art technology. Most significantly, the weights the CNN calculates for one application can be used for other similar applications where enough data is not available. The results of the proposed study demonstrate that CNNs can extract and recognize a person's identity. Moreover, CNNs can achieve better performance and accuracy compared with other approaches. The proposed method is known in the literature as a transfer learning approach and has been successful in many applications (Hoo-Chang *et al*., 2016), (Ng *et al*., 2015), (Wang and Deng, 2018) and (Tajbakhsh *et al*., 2017). Thus, a new theory and discovery on recognition, vision, as well as emotions may be arrived at. The study duration starting with pre-study preparations and data acquisition until the end was about 2 years.

## Materials and conventional neural network

In this experiment, MATLAB (MathWorks, 2015) tools are used to capture the data into matrices and resize them into best possible resolutions that cover maximum facial data. We further use the sklearn.preproceeing package from scikit-learn library (Pedregosa *et al*., 2011), (Kramer, 2016) to standardize the data and centre the mean into a unit variance. Moreover, the Keras Library (Chollet, n.d.), (Chollet, 2015) is used as a front end to configure the model, compile it, then train the model on the data with the TensorFlow (Abadi *et al*., 2016) as a back end engine for the numerical computations that can be deployed with Central Processing Units (CPUs), Graphics Processing Units (GPUs), or Tensor Processing Units (TPUs). Then, we adopt the neural predictive model for thermal image classification.

### *Conventional Neural Network Models*

A feature reduction is necessary to extract beneficial and informative features for classification purposes. Furthermore, the dimensionality of features extracted from input thermal images usually plays an important role in conventional classification accuracy and deserves more attention from the literature (Sayed, 2018a), (Sayed, 2018b). It is worth mentioning that each feature is masking a few pixels of the input image with a two-dimensional array of values and matches common characteristics of the images. A trainable multistage CNN might include tens or hundreds of concealed layers at each of which it learns to identify different features of the input images called feature maps. The feature learning has multiple stages each having one convolutional layer and one pooling layer. The convolution layer places the input images through a set of convolutional operations each of which activates certain features from the input images. It is occasionally common to insert a pooling layer in between two successive convolutional layers. The pooling layer simplifies the output by performing nonlinear down sampling, reducing the number of needed parameters (and computations as well) for training the network. Moreover, pooling layers control the network over-fitting. It is worth noting here that stride might be used instead of max pooling in order to reduce layer size in network architecture. Then, the connected convolutional layers are trailed by the 2×2 max-pooling (pooling images) layers that are next converted to one-dimensional vectors. We call this conversion the flattening stage. Thus, after learning features in many layers and flattening, the architecture of the CNN shifts to single or multiple fully connected layers, which compute the categories of the images by the end of the process. These layers, which combine all the features learned by the previous layers across the images to identify the larger patterns, are similar to hidden layers in regular neural networks. More specifically, after the network is trained, the last hidden layer outputs are used as thermal image characterizations to construct the face classification. In Fig. 1, we demonstrate the feature reduction and image classification where, in the next section, we refine the figure with proper steps and analyse the design cycle of the deep convolutional network for thermal facial recognition algorithm.

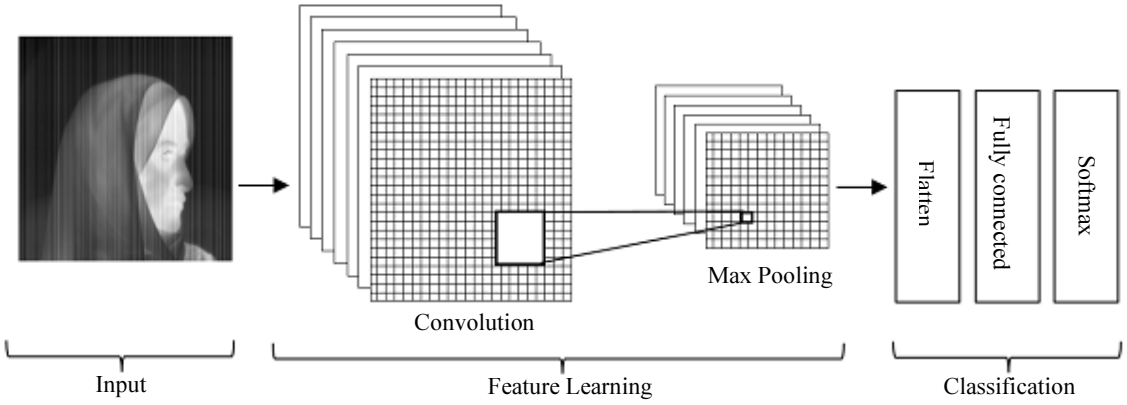


**Fig. 1:** The proposed deep convolutional network architecture for thermal face recognition

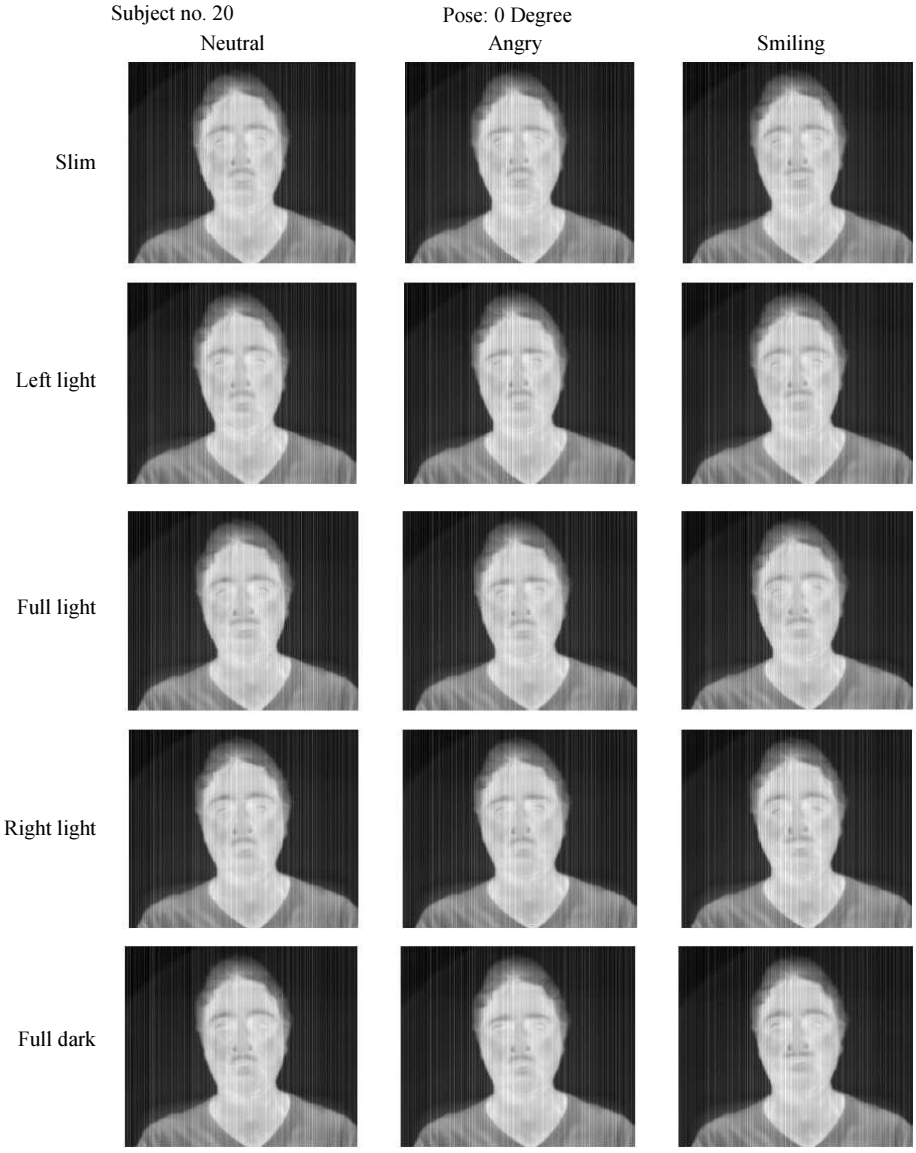


**Fig. 2:** Images of one participant in test data

■■

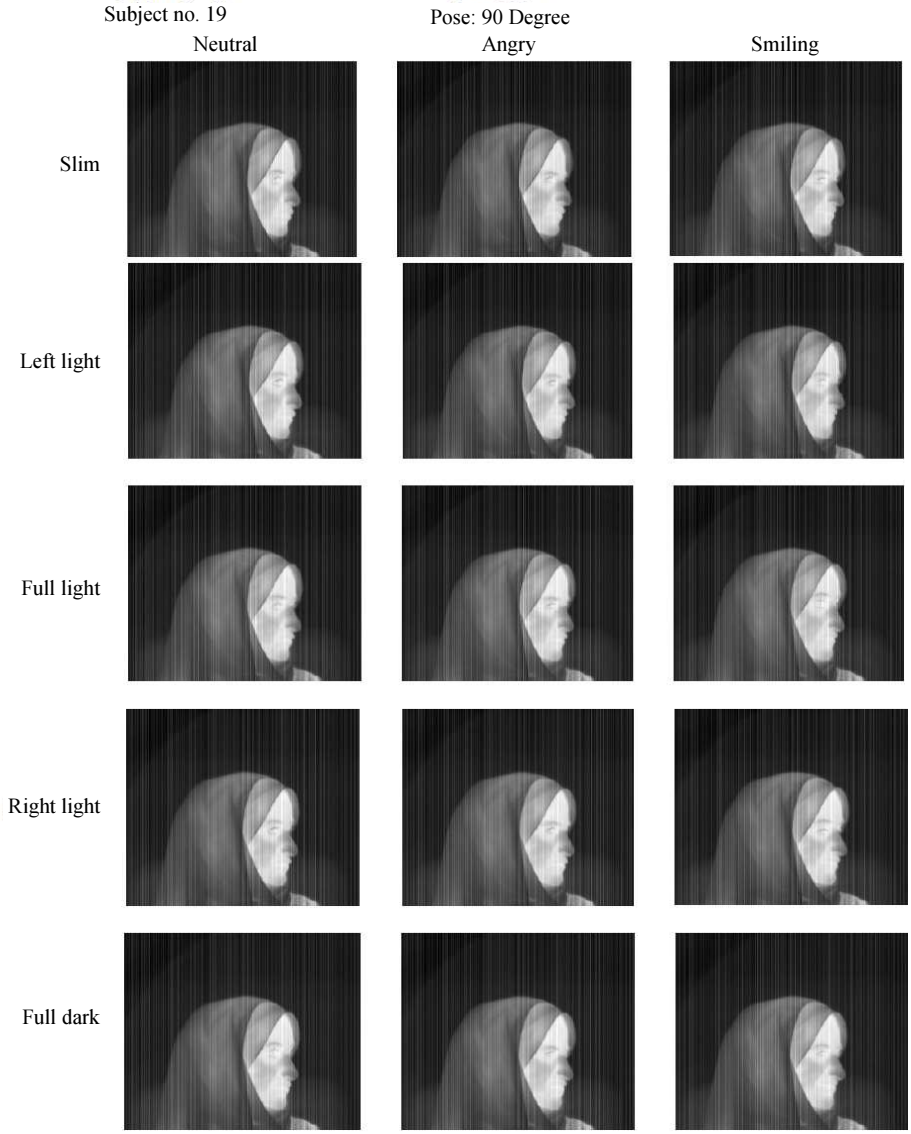


**Fig. 3:** Images of other participant in test data

*Dataset*

The Arab Open University (AOU) dataset consists of 1500 thermal pictures of 20 subjects taken in five positions (-90, -45, 0, 45, 90°C) with three emotions for each subject (angry, neutral, smiling), under five lightening conditions (slim light, left light on, full light on, right light on, full darkness). Two examples are illustrated in Fig. 2 and 3. Here, we should note that the given pictures show for different lightning conditions with the same subject and position, while the actual face temperatures matrices are not exactly the same, although through the pictures cannot be identified. The AOU dataset (Zaeri *et al*., 2015) includes males and females from a diverse ethnic background using Infrared Camera ETIP 7320.

The dataset is divided into two groups, the first group is an 80% of the data (1200 of type thermal temperatures) chosen randomly and used to build the neural predictive model by extracting the most important features incorporated in the calculated weights. All the data in the first group are labelled, in other words, the subject for each single data is known for the model. The second group comprises 20% of the data (300 of type thermal temperatures), unlabelled and is used only to test the proposed model.

## Proposed Approach and Architecture

The data is captured with the ETIP thermal camera. The data consists of faces of all the subjects in five different positions, four lightning conditions and three

emotions. Then, the data is cropped according to facial landmarks selected uniformly for each sitting position to maximize exposure of the facial area temperatures of the subjects. Furthermore, all cropped images are resized into matrices of sizes 181×161 pixels. This size is the best selection in our opinion because it sufficiently covers facial temperatures for all positions.

After acquiring the data and transforming it into the best possible format, a pre-processing stage starts by reshaping each single thermal matrix into a 1D vector, rescaling its values then reshaping it again back into its original 2D matrix form before handing it to the convolution stage. We normalise (or standardize) the data, mentioned also earlier in the introduction section, to rescale the data into having a mean of 0 and standard deviation of 1. This will maximize the available distinctive features. Next, we divide the data into two groups: Training data and testing data using random seed selection. The training data was 80% of the original data and the testing data was 20%. The training data is labelled and used to build the predictive model. The testing data is unlabelled and used to check if the model can predict the labels correctly and at what accuracy rate.

Let $x_i$ be the vector representation of a thermal 2D matrix for a subject, $y_i$ be a natural number representing the subject number and m be the number of training. The training set can be represented as:

$$\left\{\left(x_i, y_i\right) : i = 1,2,...,m\right\}. \quad (1)$$

Now, if:

$$X = \left\{x_i : i = 1,2,...,m\right\} \text{ and } Y = \left\{y_i : i = 1,2,...,m\right\}. \quad (2)$$

then:

$$X\_train = \left(m, X\right) \text{ and } Y\_train = \left(m, Y\right) \quad (3)$$

are the input tensor and the labelled tenser for the model, respectively. Keras (Chollet, n.d.), (Chollet, 2015) model requires tensors as input and acts as a front end for the TensorFlow (Abadi *et al*., 2016) backend. Tensor means n-dimensional array and flows between layers. For example, a 2D tensor takes the form of samples (features). The input shape is the only tensor that must be defined for the model, because the following layers will perform automatic shape inference. In Keras, when the input shape is defined, the number of samples is omitted, Fig, 4, because it is declared at a later stage when the specific data on the model is fitted. We are only building the architecture. Therefore, we only define the input shape in terms of number of rows, number of columns and number of channels. In our case, the number of channels is 1, but for other applications such as coloured pictures, this can be a value of 3 because there are 3 channels for an RGB picture (Red, Green, Blue) see Fig. 4. Afterwards, all inputs to other layers in the model are automatically inferred and will be calculated based on the number of (processing) units of each layer and filter type used. Each type of layer works in a specific way. For example, Dense layers have an output shape based on "units", while convolutional layers have an output shape based on "filters".

The predictive model that learns from the training data and extracts features consists of 12 sequential layers stacked together, as illustrated in Fig. 4. First, the model needs to know the shape of the input data as explained previously; hence, we pass the input shape to the first convolutional layer. The convolution layer acts as a filter with a kernel of 3×3 matrix that convolves (circulates) around the original input to 3×3 filters to produce a feature map (LeCun *et al*., 2010). Later, a pooling layer is used to reduce the spatial size and capture the most important features or portions. The pooling layer convolves with several representations to reduce the number of parameters and computations, as well as to prevent overfitting of the features. In other words, the pooling layer summarizes the matrix. The type of pooling determines how the elements of each sample are reduced to one element. In this model, the maximum element value of the chunk (the subsample) is selected. An example of max pooling operation with 2×2 filter and a stride of two is illustrated in Fig. 5, see (CNN, n.d.).

```
1:  model = Sequential()
2:  model.add(Conv2D(32, (3, 3), activation = 'relu', input_shape = (img_rows, img_cols,1)))
3:  model.add(Conv2D(32, (3, 3), activation = 'relu'))
4:  model.add(MaxPooling2D(pool_size = (2, 2)))
5:   model.add(Dropout(0.25))
6:  model.add(Conv2D(64, (3, 3), activation='relu'))
7:  model.add(Conv2D(64, (3, 3), activation='relu'))
8:  model.add(MaxPooling2D(pool_size = (2, 2)))
9:  model.add(Dropout(0.25))
10: model.add(Flatten())
11: model.add(Dense(256, activation = 'relu'))
12: model.add(Dropout(0.5))
13: model.add(Dense(20, activation = 'softmax'))
```

**Fig. 4:** Keras sequential model with 12 stacked layers

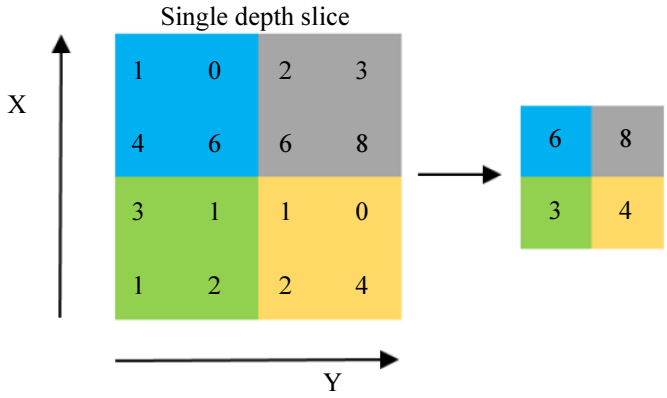


**Fig. 5:** Max pooling with 2×2 filter and stride = 2

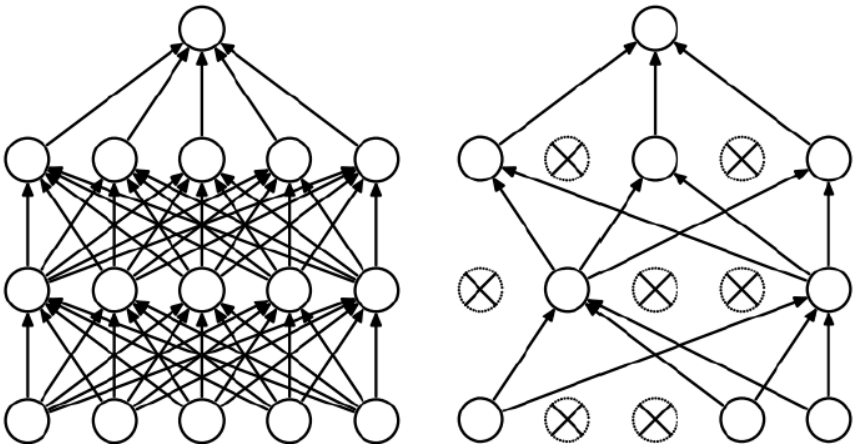

**Fig. 6:** Neural Network before and after applying dropout

In other pooling layers' architectures, the minimum value or average can be selected as well. After the pooling layer, a dropout layer is utilized specifically to prevent overfitting and to improve regularization, that is by omitting a random portion of processing units from the neural network during training (Hinton *et al*., 2012), (Srivastava *et al*., 2014). An example a network before and after applying the dropout is presented in Fig. 6 (Srivastava *et al*., 2014).

Subsequently, the process of convolution, pooling and dropout is repeated twice in the proposed model to acquire better features and minimise the noise before a flattened layer is introduced and makes it ready for two dense layers with a dropout in between. These two dense layers optimise the weights because they are fully connected, bearing in mind that convolutional layers and dense layers need the activation function to trigger the nodes and calculate the weights. There are several types of activation functions, in the model; we use Rectified Linear Unit (ReLU) (Krizhevsky *et al*., 2012) for the hidden layers and Softmax function (Goodfellow *et al*., 2016) for output classification layer. The learning method used in this model is based on Stochastic Gradient Descent (SGD) optimiser, which updates the weights proportionally to the partial derivative of the cost function (of weights). It is worth mentioning that the cost function, loss function, objective function and error function are sometimes used in place of each other in the literature of a deep learning community. To minimize the loss function, we need to find the direction in which the function decreases the quickest. In each iteration of training, the weights are updated following the formula:

$$Updated\,Weight = Current\,Weight - Learning\,Rate \times Gradient, \quad (4)$$

where, the learning rate is a scaler that specifies the size of the step that needs to be taken. When the gradient value becomes zero, this means that the minimum point has been reached; the weights are optimised at this point and the value of loss function is at a minimum. However, the gradient can be very close to zero but never reaches zero. In the latter case, the problem is named the vanishing gradient problem where it will consume many repetitions

without converging to zero and consequently give poor predictions. The vanishing gradient problem makes it difficult to know which direction the parameters (weights) should move to improve the loss function. In order to rectify this problem, we use ReLU activation function to rectify the gradient to zero when it becomes negative and hence optimise the weights, see (Kandpal, 2017).

The SGD optimizer has two hyper-parameters that need configurations, one is the Learning Rate and the other is the Momentum. They are named hyper-parameters because they differ from parameters in the model. The model parameters are learned from the algorithm such as the weights, but the hyper-parameters need to be defined specifically in the model. The Learning Rate is the size of the step that SGD needs to take to minimize the loss function. If it is a low value, then SGD is taking a tiny step each time to reach the minimum and consequently, it is more time consuming but giving a more accurate prediction. This Learning Rate can also be configured adaptively and this is defined as the decay rate. The decay rate diminishes the learning rate in every few epochs according to the decay rate. The Momentum, on the other hand, accelerates the SGD towards the relevant direction in its search for minimum loss function (Smith and Le, 2018).

As we mentioned earlier, weights calculations occur through passes (forward and backward) over the neural network. Each pass that covers the entire journey starting from the beginning of the forward direction and ending at the end of the backward direction is named an "epoch". The number of epochs is the number of passes over the completely training data and not a batch of data. For instance, if we divide our 1200 training data into 4 batches, then it will take 4 iterations to complete 1 epoch. When fitting the model architecture for our specific training data in our example where we have 1200 thermal pictures, we need to define the number of epochs that in our opinion will converge the result and optimize it. Consequently, at each epoch the SGD optimizer tries to adjust the weights so that the loss function is minimized. To consume less computer memory and to run faster on the network, nevertheless, we divided the training data into batches rather than pass the entire batch at the price of reduced accuracy.

After building the model, we experiment it on the testing data to discover the labels for each unlabelled thermal matrix used in the testing. Therefore, we calculate the accuracy as metric for the model by comparing the predictive results with the actual results.

## Results and Discussion

There are well known CNN architectures mentioned earlier in the introduction for image recognition, e.g., LeNet-5 (LeCun *et al*., 1998), AlexNet (Krizhevsky *et al*., 2012), GoogLeNet (Szegedy *et al*., 2015), VGGNet (Simonyan and Zisserman, 2014), Inception (Szegedy *et al*., 2017), ResNet (He *et al*., 2016) and DenseNet (Huang *et al*., 2017). Most of them commonly consist of convolutional layers, pooling layers, dropouts and activation functions. As an experiment, we selected the model, which has been previously proven effective in a similar application in digital recognition of hand writing where the data poses similar characteristics, such as being two-dimensional. In other words, it was a conceptual learning without transferring the weights of the application. At the end and from the results for the proposed thermal facial recognition, the model showed high accuracy.

Neural Networks learn from data via Statistical Gradient Descent (SGD) optimization, which is the process of adjusting the weights, in order to minimize the loss function (the error between the predictive results and the actual result) (LeCun *et al*., 1998). The model starts with an input layer and an initialized weight for each node (processing unit), then the weights are adjusted through several epochs (passes) over the entire training data (1200 thermal pictures), in an iterative process. At the beginning, the number of epochs, the number of nodes and the activation functions for the nodes are all defined by the model. At the end of each epoch the loss function is minimized, the weight values are adjusted and the accuracy is maximized. This iterative process continues until all epochs are completed. For the VGG model explained in the previous section, we obtained the results that are graphically represented in Fig. 7-9. It is noticed that the loss function continues and enhances until the end of the process. Moreover, after each epoch, we determine the training accuracy numbers to prove that they are improving in the training dataset and the process is moving in the right direction. Additionally, the proposed method is efficient in terms of its time cost, which is another metric to evaluate the performance of the network.

The proposed approach is mostly built on feature extraction and classification components. Accordingly, each of these two components has been separately evaluated. The overall evaluation is further carried out and the result is compared with different approaches. Usually, the major challenge of infrared facial recognition methods is to find reliable representations of thermal images and highly efficient feature extractors. In addition, most of these methods focus on the achievable classification accuracy using different deep machine learning algorithms. Computing efficiency in these methods are always significant issues. Although substantial progress on face feature extraction has been achieved, existing traditional methods can only detect thermal images with classification accuracies less than 99.1%. Based upon the experimental results, the proposed CNN model would lead to the best results with an average accuracy reaching up to 99.6%. In fact,

this achieved accuracy that confirms the superiority of our method is the research definitive objective. It should be emphasized that the proposed model is suitable for complex degree of variation in poses (up to 90°C), lighting (full darkness to full light on), facial expressions and head positions. By tuning the CNN algorithm, we can get more out of it; for example, an improved performance can be anticipated once the network is larger. Consequently, by adding an optional fully connected layer, the average recognition rate might exceed 99.9%. In addition, the results might show that the model has 100% accuracy for the moderate degree of variation in poses (up to 20°C) and lighting (dark homogenous background). Table 1 shows the comparison between average accuracy of the proposed CNN architecture and some state-of-the-art facial recognition approaches (Peng *et al*., 2016), (Wu *et al*., 2016).

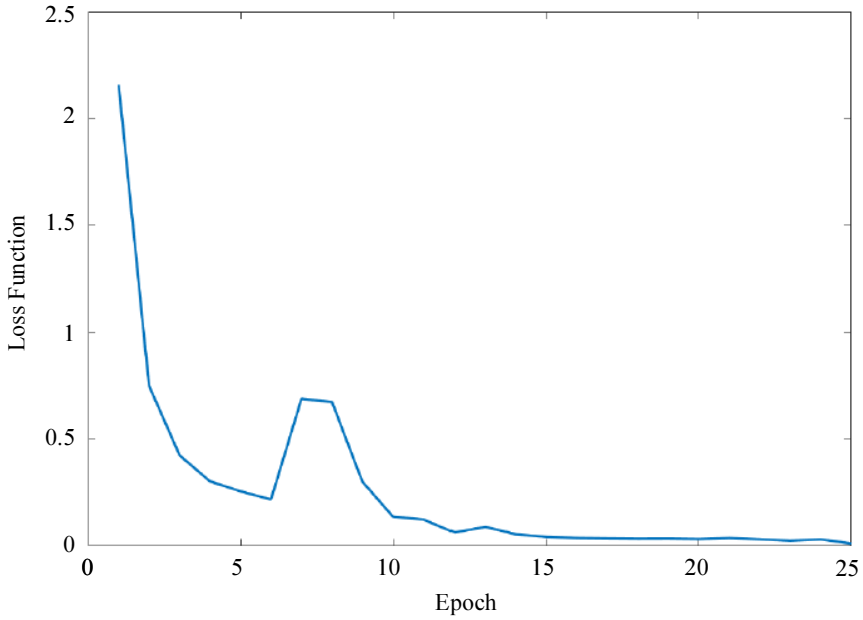


**Fig. 7:** Loss function for the training dataset

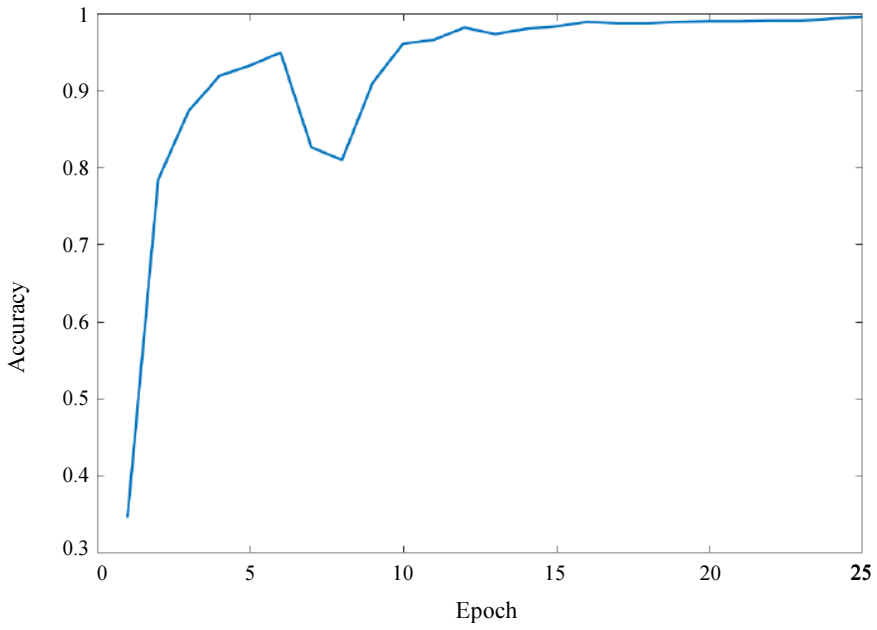


**Fig. 8:** Accuracy for the training dataset

■■

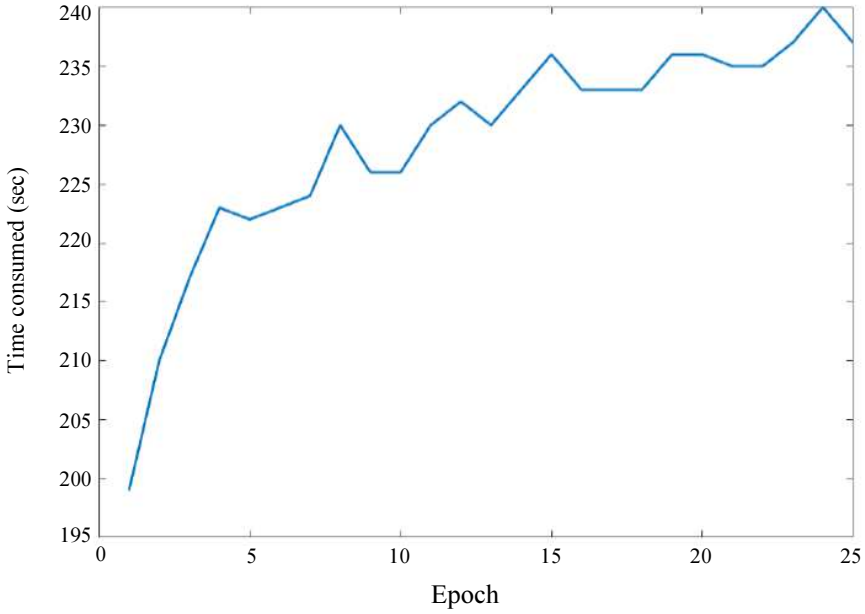


**Fig. 9:** Time consumed for training dataset in seconds

**Table 1:** Average accuracy of various methods with full training sets

| | LBP + PCA | LBP Histogram | ZMUDWT | ZMHK | GoogLeNet softmax0 | GoogLeNet softmax1 | GoogLeNet softmax2 | Moments invariants | CNN | NIRFaceNet | Proposed method |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Average accuracy (%) | 85.4 | 87.3 | 92.9 | 98.3 | 97.3 | 97.0 | 96.7 | 97.5 | 98.0 | 99.1 | 99.6 |

## Conclusion

The main aspect of face tracking and image classification is correctly identifying the object in a given image. Before the image can be used, a few steps have to be considered. This is called image pre-processing and it represents an important step in facial recognitions. Many conditions might affect the accuracy of any facial recognition algorithm such as light intensity, rotation, resolution and tilt and aging. It is worth noting that the proposed technique has taken all of these into considerations; therefore, the results obtained after running the program showed outstanding image classification performance. Furthermore, the used deep learning model has given a higher level of accuracy in such problem when compared with other techniques such as Viola Jones face detection algorithms, or local binary patterns such as AdaBoost algorithm for face detection.

Our experiment has been utilized only for identifying subjects within static thermal radiation; however, its application can likewise be investigated for emotion recognition as well as identifying subjects from thermal video clips inspired by similar applications in visible light for video classification (Karpathy *et al*., 2014) and emotion recognition (Ng *et al*., 2015). Furthermore, the results indicated that a deep learning architecture in one area of application could be utilized for other similar areas; the differences are in the values of the data for each application. Similarly, weights calculated in one comparable application that has a large number of labelled data can be used for another application where labelled data is limited. In addition, weights can be exploited to reduce processing time for very large dataset processing. This process has been experimented in the literature and defined as transfer learning (Oquab *et al*., 2014). Overall, however, the current thermal image method for face recognition still needs more enhancement in terms of pre-processing operations before it can be employed in security systems and biometric identifications.

Finally, our most recent aim is to improve the recognition accuracy and computational complexity by implementing the method on different benchmarks. We will also develop a novel and effective CNN model and utilize it in multimodal biometric systems.

## Acknowledgement

This work is based upon a project supported by Arab Open University in Kuwait. The authors would like to acknowledge the support from the University Rector, Administrative Staff and the plentiful participants in the data acquisition. We would also like to show our gratitude to Ajit Jaokar, Faculty Member at Oxford University and Jeremy Howard, Faculty Member at University of San Francisco, who provided insight and expertise.

## Author's Contributions

**Mohamed Sayed:** Identifying and proposing the method, analysing the results, writing part of the manuscript, providing proofreading and critical review.

**Faris Baker:** Implementing the proposed method, identifying the dataset, providing literature review, writing part of the manuscript.

## Ethics

There is no ethical issue involved in this article, as it is original contribution of the authors.

## References

Abadi, M., P. Barham, J. Chen, Z. Chen and A. Davis *et al*., 2016. TensorFlow: A system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, (SDI' 16), Google Brain.

Chollet, F., 2015. Keras. https://github.com/fchollet/keras

Chollet, F., (n.d.). Keras: Deep learning library for theano and tensorflow. https://keras.io/

CNN, (n.d.). Convolutional Neural Network. https://en.wikipedia.org/wiki/Convolutional_neural_network

Goodfellow, I., Y. Bengio and A. Courville, 2016. Deep Learning. 1st Edn., MIT Press, Cambridge, ISBN-10: 0262337371, pp: 800.

Hadji, I. and R. Wildes, 2018. What do we understand about convolutional networks? arXiv preprint arXiv:1803.08834.

He, K., X. Zhang, S. Ren and J. Sun, 2016. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 27-30, IEEE Xplore Press, Las Vegas, NV, USA, pp: 770-778. DOI: 10.1109/CVPR.2016.90

Hinton, G., N. Srivastava, A. Krizh, I. Sutskever and R. Salakhutdinov, 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Hoo-Chang, S., H. Roth, M. Gao, L. Lu and Z. Xu *et al*., 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging, 35: 1285-1298. DOI: 10.1109/TMI.2016.2528162

Huang, G., Z. Liu, L. Van Der Maaten and K. Weinberger, 2017. Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jul. 21-26, IEEE Xplore Press, Honolulu, HI, USA, pp: 2261-2269. DOI: 10.1109/CVPR.2017.243

Kandpal, A., 2017. Medium.com: https://codeburst.io/machine-learning-day-1-60bd231d0660

Karpathy, A., G. Toderici, S. Shetty, T. Leung and S. Rahul *et al*., 2014. Large-scale video classification with convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 23-28, IEEE Xplore Press, Columbus, OH, USA, pp: 1725-1732. DOI: 10.1109/CVPR.2014.223

Khanal, S., J. Fulton and S. Shearer, 2017. An overview of current and potential applications of thermal remote sensing in precision agriculture. Comput. Electron. Agric., 139: 22-32. DOI: 10.1016/j.compag.2017.05.001

Kramer, O., 2016. Scikit-learn. Mach. Learn. Evolut. Strategies, 20: 45-53. DOI: 10.1007/978-3-319-33383-0_5

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inform. Process. Syst., 25: 1097-1105. DOI: 10.1145/3065386

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM, 60: 84-90. DOI: 10.1145/3065386

Lahiri, B., S. Bagavathiappan, T. Jayakumar and J. Philip, 2012. Medical applications of infrared thermography: A review. Infrared Phys. Technol., 55: 221-235. DOI: 10.1016/j.infrared.2012.03.007

LeCun, Y., L. Bottou, Y. Bengio and P. Haffner, 1998. Gradient-based learning applied to document recognition. Proc. IEEE, 86: 2278-2324. DOI: 10.1109/5.726791

LeCun, Y., K. Kavukcuoglu and C. Farabet, 2010. Convolutional networks and applications in vision. Proceedings of IEEE International Symposium on Circuits and Systems, May 30-Jun. 2, IEEE Xplore Press, Paris, France, pp: 253-256. DOI: 10.1109/ISCAS.2010.5537907

MathWorks, 2015. Image processing toolbox. MathWorks. https://www.mathworks.com

Ng, H.W., V. Dung Nguyen, V. Vonikakis and S. Winkler, 2015. Deep learning for emotion recognition on small datasets using transfer learning. Proceedings of the ACM on International Conference on Multimodal Interaction, Nov. 09-13, ACM, Seattle, Washington, USA, pp: 443-449. DOI: 10.1145/2818346.2830593

Nixon, M.S. and A.S. Aguado, 2002. Feature Extraction and Image Processing. 1st Edn., Newnes, Oxford, ISBN-10: 0750650788, pp: 350.

Oquab, M., L. Bottou, I. Laptev and J. Sivic, 2014. Learning and transferring mid-level image representations using convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 23-28, IEEE Xplore Press, Columbus, OH, USA., pp: 1717-1724. DOI: 10.1109/CVPR.2014.222

Pedregosa, F., G. Varoquaux, A. Gramf, V. Michel and B. Thirion *et al*., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res., 12: 2825-2830.

Peng, M., C. Wang, T. Chen and G. Liu, 2016. NIRFaceNet: A convolutional neural network for near-infrared face identification. Information, 61: 1-14. DOI: 10.3390/info7040061

Sayed, M., 2018a. Biometric gait recognition based on machine learning. J. Comput. Sci., 14: 1064-1073. DOI: 10.3844/jcssp.2018.1064.1073

Sayed, M., 2018b. Performance of convolutional neural networks for human identification by gait recognition. J. Artif. Intell., 11: 30-38. DOI: 10.3923/jai.2018.30.38

Simonyan, K. and A. Zisserman, 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Smith, S. and Q. Le, 2018. A Bayesian perspective on generalization and stochastic gradient descent. ICLR.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Machine Learn. Res., 15: 1929-1958.

Szegedy, C., S. Ioffe, V. Vanhoucke and A. Alemi, 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. Proceedings of the 31th AAAI Conference on Artificial Intelligence, (CAI' 17), pp: 12-12.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet and S. Reed *et al*., 2015. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 7-12, IEEE Xplore Press, Boston, MA, USA, pp: 1-9. DOI: 10.1109/CVPR.2015.7298594

Tajbakhsh, N., J. Shin, S. Gurudu, R. Todd Hurst and C. Kendall *et al*., 2017. Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Trans. Med. Imag., 35: 1299-1312. DOI: 10.1109/TMI.2016.2535302

Vadivambal, R. and D. Jayas, 2011. Applications of thermal imaging in agriculture and food industry-a review. Food Bioprocess Technol., 4: 186-199. DOI: 10.1007/s11947-010-0333-5

Wang, M. and W. Deng, 2018. Deep visual domain adaptation: A survey. Neurocomputing, 312: 135-153. DOI: 10.1016/j.neucom.2018.05.083

Wu, Z., M. Peng and T. Chen, 2016. Thermal face recognition using convolutional neural network. Proceedings of the International Conference on Optoelectronics and Image Processing, Jun. 10-12, IEEE Xplore Press, Warsaw, Poland, pp: 6-9. DOI: 10.1109/OPTIP.2016.7528489

Zaeri, N., F. Baker and R. Dip, 2015. Thermal face recognition using moments invariants. Int. J. Signal Process. Syst., 3: 94-99. DOI: 10.12720/ijsps.3.2.94-99