A new Rule to Constrain Convolution Neural Network Architecture in Face Recognition System

Qusay AL-Khalidi, Motlak H.J, Hilal Al-Libawy
Department of Electrical Engineering, College of Engineering, University of Babylon, Babylon, Iraq
qusaisami280@gmail.com

Abstract. Face recognition (FR) system is an essential part of a biometric security system, which runs faster than other security methods and done remotely. One of the most important techniques that used in FR is the convolutional neural network (CNN). Traditionally, the choice of the CNN architecture is achieved by experimental trails. In this paper, a new approach is proposed to build a mathematical model that helps to select a proper architecture. This model is built from the experimental results by applying different architectures on the well-known dataset (Vggface2). By changing the class number, image number and convolutional layer number, where the accuracy of each case is recorded. Finally, the proposed model is evaluated on the sets of datasets (Essex, FEI, Caltech), where the accuracies of (99.13, 98.51, 97.78) respectively, are achieved. The evaluation results proved that the proposed model is an efficient for many types of small and middle scale dataset.

Keywords: Face Identification System, Deep Learning Technique, Convolutional Neural Networks

1. INTRODUCTION

Face Recognition (FR) is a computer vision (CV) area which attracted more attention for many years. It has a very large number of ranging from biometric security, to the automatic tagging of photos for friends, author and more. The reason belongs to the possibilities, many companies and research centres are working on it. However, this problem is also a big problem. It did not happen last time you get good results. Actually, this issue is often fragmented to various sub-issues for making it easier to work. However, face detection is mainly in a picture, followed by actual facial recognition. Some other duties may also be implemented between periods, like the Formulate faces, or draw more features from them. Across the years, numerous approaches and algorithms have been introduced. In the early 1990s, the researches on FR gained popularity after introducing the historical method of Eigenface (M. Turk et al,1991). The most used methods are the deep learning (DL), in particular Convolutional Neural Networks (CNNs). Those approaches presently give high quality results. During the past years, the CNNs had a major effect on boosting the performance of state-of-the-art is applied in a wide range of tasks of visual classification like object identification (A. Krizhevsky et al,2012&P. Sermanet et al,2013& K. He et al,2015& H. J. Mottak et al,2018), face verification (Y. Taigman et al,2014&Y. Sun et al,2014&Y. Sun etal,2015) . The layered learning architecture, in combination with pooling and convolution that carefully obtain features from local to global, renders the strong ability of visual
representations of CNNs; in addition to their existing significant positions in tasks of large-scale visual identification. Encountering data of increasingly higher complexity, CNN has been continuously enhanced with smaller strides (K. Simonyan et al, 2014), deeper structures (Y. Jia et al, 2015) and new non-linear activations (D. Warde-Farley et al, 2013& V. Nair et al, 2010).

Therefore, their efficiencies were significantly enhanced, for instance, the precision on the faces labeled challenge in the wild (LFW) features was enhanced from 97% (Y. Taigman et al, 2014) to 99% (Y. Sun et al, 2014 & O. M. Parkhi et al, 2015 & F. Schroff et al, 2015) This enhancement basically gets advantage from which CNN are capable of learning strong face inserting from training data with many subjects. For the sake of achieving the best precision, the training data-set scale for CNN keeps consistent increase. Some large-scale data-sets of faces were released like CelebFaces+ (Y. Taigman et al, 2014), CASIA-WebFace (D. Yi et al, 2014), UMDFace (A. Bansal et al, 2017), VGG face dataset (V. Nair et al, 2010) VGGFace2 data-set (Q. Cao et al, 2018), MS-Celeb-1M (Y. Guo et al, 2016) However, those large-scale data-sets usually include considerable noisy signals particularly in the case where they are automatically obtained through Search engines for images or from movies (Y. Taigman et al, 2014).

While benefit from the strong learning ability, the time factor was also required, where this recognition should be in a short time as possible. Controlling the outside conditions is very difficult problem in the task of recognizing faces. Actually, there were numerous methods during the past years, which did not work. Regardless of the difference between images of the same person, as expression, lighting conditions or hair, the situation is hard to specify the details that make the face recognizable. The implementation of VGG2 (Q. Cao et al, 2018) is selected because it is getting high precision results to be very close to the technical situation quality description. This study arranged as the following manner. Section (1) is a review of facial recognition history, which includes the current state of art, section (2) is a related work, section (3) presents the theoretical background of CNN, the base of the proposed system, the results and discussions are in section (4), evaluation is in section (5), section (6) is a comparison with the resent researches and section (7) is the obtained conclusion.

2. RELATED WORK

Approaches of CNNs have been recently drawing a great deal of interest in the face recognition area. It achieved promising results on face recognition. In the present section, a brief review of those CNNs has been presented. In 2012 (A. Krizhevsky et al, 2012) ImageNet has been used to train a large, deep CNN for the classification of the 1.2 million images of high-resolution, in ImageNet LSVRC2010 contest to 1000 various classes, the network includes 8 weighed layers; the first 5 are convolutional and the rest is fully connected . In (K. Simonyan et al, 2014) it works on improving on (Y. Taigman et al, 2014) increasing depth by adding more convolutional layers with the use of an architecture with quite small (3x3) convolutional filters. Researchers in the AI group in Facebook company have trained an 8-layer convolutional neural network, which is referred to as Deep Face (Y. Taigman et al, 2014). The first 3 layers are conventional “convolution-pooling-convolution” layers. The following 3 are connected locally, which are followed by two fully connected layers. Pooling layers increase the robustness of the learned features against local transforms, however, they yield missing details of local texture. The importance of the pooling layers can be noticed for object identification, due to the fact that image objects aren’t well aligned. None-the-less, facial images are well aligned prior to CNN training. It has been presumed that one layer of pooling is a sufficient balance between the preservation of texture details and robustness of the local transformation.

In (Y. Sun et al, 2014), a face representation based on CNNs, known as a Deep Hidden Identity feature (DeepID), has been presented. In contrast to the Deep Face whose characteristics are learned with a single big CNN, Deep ID is learned via the training of a set of small CNNs, which are utilized for the fusion of the network. The input of a single CNN is facial image crops/patches and features that are learned with each CNN are merged for forming powerful characteristics. Both RGB and Gray crops that have been obtained around facial points have been utilized for the training of the Deep ID. Every one of the networks includes four convolution layers, two fully connected layers, and three max pooling layers.
layers. Deep ID utilizes information of identification only for supervising the training of the CNN. Compared to it, Deep ID-2[7] an extension of the Deep ID, utilizes each of the verification and identification information for training the CNN, which aims at the maximization of inter-class difference at the same time as minimizing intra-class variations. For the sake of additionally improving the efficiency of Deep ID and Deep ID-2.

The work (Y. Taigman et al, 2014) has proposed one more pipeline for face recognition, which is known as the WebFace, it also learns face representation with the use of a CNN. It collects a data-base containing about 500000 images and 10000 subjects and makes this data-base available. Taking motivation from the very deep architectures of (K. Simonyan et al, 2014& Y. Jia, 2015), WebFace performs the training of a considerably deeper CNN than the ones ((Y. Taigman et al. 2014 & Y. Sun ET al.2015& Y. Sun et al, 2014) utilized for face identification. Particularly, WebFace performs a training on a 17-layer CNN that includes ten convolution layers, two fully connected layers, and five pooling layers. It has to be noted that utilizing very small convolution filters (3x3), avoiding a considerable amount of texture data diminish along a very deep architecture, is highly important for learning a powerful feature. Moreover, Web-Face stacks two 3x3 convolution layers (with no pooling in between) and that has the same effectiveness as a 5x5 convolution layer.

3. THEORETICAL BACKGROUND

FACE IDENTIFICATIN SYSTEM

The face recognition system consists of two parts; the first is facial verification. It is a process of comparing two images and verifying whether the two images belong to the same person or not. The second part: is the face identification, which is covered in our search, which means knowing the identity of the person, one image is compared with the total images’ set of the stored in the system from the trained images. When the correspondence or similar is done in any image, this means the identity of the person to whom the item is to be classified [1]. In this paper, we introduce a new approach in the field of face identification, where the experience of an infinite number of convolutional layers depends on changing the variables until an appropriate mathematical model is reached at the values of top accuracy and Figure 1 shows the methods used in the architectural structure.

Figure 1 the proposed system Block diagram by CNN
4. DEEP LEARNING TECHNIQUE

A sub-field of the machine learning (ML) methods is the DL, the main feature of DL method is the ability to make abstractions, it has the ability of building complicated models from uncomplicated ones, such as, its ability of learning concepts in images like cats, humans, and cars via joining sets of fundamental aspects like corners or edges. Such approach is implemented via successive layers, which elevate the complexity regarding learning concepts. Each one of the output layers is passed as an input to the next layer that it utilizes for learning the features of higher-level (have extra complexity), (X. SERRA, 2017 & A. Al-Waisy et al, 2018) Furthermore, this could be extremely labour intensive and must be repeated for all tasks. Such task becomes more complicated when handling data of high-dimensional (such as images). The second option, automatic learning about features in the data, can overcome both the above-mentioned drawbacks and can well be achieved by Deep Learning (Y. Guo et al, 2016) The deep learning technique is used in neural network such as the convolutional neural network (CNN) where many convolutional layers are cascaded.

5. CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNNs, can be defined as a special type of NNs, their aim is to process the data which has recognized, grid-like topologies, the examples are the time-series data, which could be viewed as a 1-Dimensional grid that take the sample at regular time intervals, and the image data that could be considered as 2-Dimensional pixel grid. Traditional networks were extremely effective in real-world applications. The term “convolutional neural network” refer to the fact that the network employ a mathematical operation (convolution), which is thought of as certain type of linear operations. CNNs are NNs, which utilize convolution instead of the general matrix multiplication in no less than one of their layers.

The first layer’s input, which is the image itself, could be thought of as 3-D volume: width, height, channels. Each layer stack uses different kernel configuration; all layers in it are connected to all the layers in previous stack (Y. Sun et al, 2014).

Unique features of CNN are the sharing of weights and the local connection, through that it can learn from the input images local features. in general, CNN structure consists of three main layers: convolution, maxpooling, and fully connected.

The major parameters, which must be considered in CNN layers, are:

Kernel Size: The height and the width regarding the neurons’ receptive field in that stack. In general the sides are the same size - square filters. Stride: The processing of input parts is achieved on each pixel. Since such parts may be to overlap, the spaces from centre to centre are indicated by the stride. In a way that, stride of 1 mean which each one of the neurons process same region as their neighbour apart from one column. As the stride gets larger, the output width and the layer’s height will be smaller.

Padding: With regard to certain conditions, a whole receptive field cannot be processed via the neurons in the layer’s borders. This could occur because of the stride. As an example for that, Figure 2.5 presents 4 × 4 images that are processed through 3 × 3 kernels with stride of two. Since the difference between the receptive fields is two pixels, previous column of the receptive field of 2nd neuron “falls” outside the image and will not be handled. For the purpose of handling such a problem, a “border” could be added around an image of 0’s. In such manner, the neurons will definitely process a receptive field. Padding, if applied, will generally be 1 or 2 (M. Thoma, 2017).
6. LAYER TYPE

CONVOLUTION LAYER

It does receipt few input feature-maps and convolute those maps by using n fixed size learnable kernels for producing n output of feature-maps, as can be seen in eq. (1):

\[ N^{cl} = F^{cl} \otimes X^{cl-1} \]  

(1)

Where \( N^{cl} \) indicates to number feature-maps, \( F^{cl} \) indicate filter weights in convolutional layer \( cl \), \( \otimes \) indicates the convolution operative. Fig. (2.3) displays one filter convoluted with one input feature-map. (M. Thoma, 2017)

7. MAX POOLING LAYERS

They are also referred to as the pooling layers, their major task is reducing the spatial size regarding convolution layers production feature-maps, pooling is majorly utilized with stride of \( s \in \mathbb{N} \geq 2 \), that is applied for the reduction of data to \( (1s2) \)th. Maxpooling can be defined as the most important pooling operation type. Pooling is utilized for 3 reasons: for getting local translational invariance getting invariance in contradiction of insignificant local changes and, majorly for reduction of data. (F. Schroff et al, 2015)
8. FULLY-CONNECTED LAYERS

This type is considered to be as a conventional MLP that is defined previously in which all the layer neurons are connected to each neuron in the next layer, fully connected layers are applied for classification tasks. Dropout approach can be utilized over this layer for preventing the issue of overfitting (F. Schroff et al, 2015).

9. LOCALLY-CONNECTED LAYERS

This type of layers is similar to the convolution layers, however the difference between them is the first one it doesn’t use the shared weight technique. Such technique is justified in the usual CL while the related features are usually independent of their image location. Yet, where this could not be valid, there are certain condition, for example, all faces in images is centred to the same position, it will make sense to look for various features at eye zone instead of the mouth. This can be implemented via providing each neuron with own set of weights, in the same way as the regular ANNs, yet they only process them receptive field. Such layers are usually utilized following certain Pooling and Convolutional because of 2×2 reasons. The first, for the purpose of the features from, p.e, mouth and eyes, to be different, they should be fairly abstract one. Major structures, like corners or edges are related in the two cases. As previously defined, such level of abstraction can be implemented via utilizing successive neuron layers that build “complex” features with the use of the simpler ones (M. Thoma, 2017).

10. TRAINING THE CNN

In back-propagation algorithms, there are 2 steps, forward and backward steps. In the two steps, fully connected layer(s) in the CNNs act as the same as back-propagation algorithms of the MLPs. In the rest layers regarding CNNs, the case will be more complex, backward error propagation and the forward signal propagation, next are some of the distinctive rules for each one of the layers (D. Yi et al, 2014).

11. FORWARD PROPAGATION OF CONVOLUTION LAYER

A convolution operation is applied via each convolutional layer between all its kernels and inputs. Let convolution layer cl with a filter of weights f cl that has size SxS, and set of input feature-maps Fcl-1 that is either the feature-maps output of the earlier layer or input image. Every one of the feature-maps m in Fl-1 has the equal size as the NxN, following the operation of convolution the output feature-map size will be (N – S + 1) × (N – S +1). The output neuron regarding the convolution process \( z_{h_{l,j}}^{cl} \) is estimated with eq. (2):

\[
z_{h_{l,j}}^{cl} = \sum_{m \in F^{cl-1}} \sum_{A=0}^{S-1} \sum_{B=0}^{S-1} x_{m,(i+A),(j+B)}^{cl-1} \cdot f_{m,h,AB}^{cl}
\]  

In which h represent the feature-map in the output feature-maps set \( F^{cl} \), i, j \( \in (0, N - s +1) \), l can be defined as the index regarding the presentation layer, and \( x_{m,(i+A),(j+B)}^{cl-1} \) is the respective failed from m feature-map that is output from cl-1 layer.

The output pass through activation such as ReLU in eq.( 3):

\[
(z) = \max(z, 0)
\]  

for producing \( \gamma_{m,l,j}^{cl} \) that represent neuron in m feature-map in cl convolution layer (D. Yi, 2016 & Z. Zhang, 2016).
12. BACKWARD PROPAGATION OF CONVOLUTION LAYER

The backward propagation for the convolution layer will follow the same rules as defined in section (2), the main difference is the convolutional kernel share weights for the while layer. The impact regarding the kernel weights on Loss function that is utilized for updating weights will be calculated through applying equation (4):

\[
\frac{\partial L}{\partial f_{m,h,A,B}^{cl}} = \sum_{i=0}^{N-S} \sum_{j=0}^{N-S} \frac{\partial L}{\partial y_{h,i,j}^{cl}} g'(x_{h,i,j}^{cl}) x_{m,((i+A),(j+B))}^{cl-1}
\]

(4)

As the expression \( \frac{\partial L}{\partial y_{h,i,j}^{cl}} \) is suggested to come from cl+1 layer, the final phase is propagating error into cl-1 layer that will be calculated via equation (5):

\[
\frac{\partial L}{\partial y_{m,h,i,j}^{cl}} = \sum_{m=0}^{S-1} \sum_{A=0}^{S-1} \frac{\partial L}{\partial z_{m,(i+A),(j+B)}^{cl}} \frac{f_{m,h,A,B}^{cl}}{f_{m,h,A,B}^{cl-1}}
\]

(5)

The result appears to be similar to convolutional operation and may be considered as error convolution with flip kernel (Z. Zhang, 2016 & J.-C. Chen et al, 2016).

13. EXPERIMENTAL RESULTS

This section consists of the trail results of the proposed approach. The MATLAB software package is used on a laptop with the GPU capability of the device (GTX1050, RAM 16G, CORI7, GPU 6GB) for all experiments. There are many face recognition datasets such as VGGF2, Essex, and Caltech. In this work, VGGF2 is used for training the CNN. The strategy of this work is divided into two stages represented by:

First stage before start training and testing, pre-processing on the VGGF2 dataset because the size of the images is not equal. The images are resized to 200 * 200 as the first step and then reduced the size to 131 * 131. Split the data into five parts, in each part, we use five cases of the number of classes and trained each case on the convolutional layers of (1, 3, 5, 7, 9). And so repeat the same way for all other parts of the data which then record the highest value of the accuracy we obtained from the experiments so that at the end we have five readings for each (layers number, total number of images, cl layers) Since the training process in the classes and the selection of weights in the filters are random for the same parameters in the processes of one experiments did not rely on a single training process, therefore we relied on more than five times for each experiment and take the situation of high precision so that we conducted more than 625 process tests as whole. 125 experiences are selected in five class cases each one has 25 experiments.

Table I presents our results of the appropriate number of convolution layers for dataset size and number of classes in the face recognition system respectively. After the training experiments on vggface2, the data were divided into 80% training and 20% validation. The first five rows represent the highest accuracy results obtained corresponding to the number of convolution layers when the number of classes is constant (5) while the number of images is variable for each class for five cases (100, 200, 300, 400, 500). And when we move to the second five rows, the number of class changes to (10) and the number of images for each Class (200, 400, 600, 800, 1000), and in the same way recorded the highest results of accuracy of the number of corresponding classes and so was working with the other 15 cases as shown in the Table1.

Table I the top 25 accuracies values recorded from 125 experiments
Table 2 is an abbreviation of all experimental cases where we take the highest value for the accuracy of the classes number of (5, 10, 15, 20, 25). In these results, we obtain three equations first: the number of classes with the number of layers as shown in Fig.4 and second: the size of the dataset with the number of convolution layers as shown in Fig.5 Third, the number of classes and data size with the number of convolution layers at the same time as shown in Fig.6

Table 2 The top 5 accuracies values recorded from 125 experiments.

| Exp. No. | No. of subject | No. of images | No. of CLs | Accuracy % |
|----------|----------------|---------------|------------|------------|
| 1        | 5              | 100           | 3          | 90         |
| 2        | 5              | 200           | 3          | 87.5       |
| 3        | 5              | 300           | 5          | 88.33      |
| 4        | 5              | 400           | 7          | 87         |
| 5        | 5              | 500           | 5          | 88         |
| 6        | 10             | 200           | 5          | 88         |
| 7        | 10             | 400           | 3          | 89         |
| 8        | 10             | 600           | 3          | 87         |
| 9        | 10             | 800           | 5          | 90         |
| 10       | 10             | 1000          | 5          | 87         |
| 11       | 15             | 300           | 5          | 87         |
| 12       | 15             | 600           | 7          | 82         |
| 13       | 15             | 900           | 7          | 80         |
| 14       | 15             | 1200          | 7          | 85.5       |
| 15       | 15             | 1500          | 7          | 87.67      |
| 16       | 20             | 400           | 5          | 82         |
| 17       | 20             | 800           | 7          | 86         |
| 18       | 20             | 1200          | 7          | 88         |
| 19       | 20             | 1600          | 7          | 88         |
| 20       | 20             | 2000          | 5          | 87         |
| 21       | 25             | 500           | 5          | 82         |
| 22       | 25             | 1000          | 5          | 85         |
| 23       | 25             | 1500          | 7          | 87         |
| 24       | 25             | 2000          | 9          | 84.75      |
| 25       | 25             | 2500          | 9          | 87.5       |

The second stage of this work is the extraction of the mathematical model for different cases.
The mathematical model is:

\[ y = 0.000755 \frac{1}{g18_{\partial 6}} - 0.0459 \frac{1}{g18_{\partial 0}} + 1.04 \frac{1}{g18_{\partial 6}} - 1.23 \]  \quad (6)

The extracted model is:

\[ y = -1.3e^{-10}x^3 + 1.4e^{-7}x^2 + 0.0029x + 2.7 \]  \quad (7)

Based on the curve fitting, the polynomial model is obtained:
\[ N = 3.105 - 0.04516 \times + 0.002903Y \]  
(8)

Where, \( N \) is the quantity of convolution layers, \( x \) is the quantity of classes, and \( y \) is the quantity of all images using in experiment (training & validation).

14. EVALUATION

We evaluated our work on three sets of data (Essex (F. Ahmad et al, 2013), Caltech [28], FEI (V. do Amaral et al, 2016)). After taking the dataset size and the quantity of classes for each type of data to give us the number of layers required for each case through the mathematical model derived from the experiments as explained previously and we obtained high accuracy results as shown in the table 3. All the details regarding the accuracy and the dataset size and the quantity of layers used in the architecture of the algorithm. Although there are many differences between the data in terms of external conditions, but the experiments proved that we can apply this approach on many data that are characterized as medium or small size as shown in table 3:

| Dataset type | Images size | # of subjects | Details of dataset | # Convolution layers | Test accuracy% | losses | Elapsed time |
|--------------|-------------|---------------|--------------------|----------------------|----------------|--------|--------------|
| Essex        | 5177        | 305           | expression & background | 4 CL                  | 99.13          | 0.05   | 00:02:32     |
| FEI          | 2800        | 200           | Pose & expression   | 2 CL                  | 98.51          | 0.06   | 00:05:24     |
| Caltech      | 452         | 27            | Pose & expression & background & illumination | 3 CL                  | 97.78          | 0.07   | 00:09:8      |

Finally, for the purpose of verifying our work, we have applied ten experiments on Essex Dataset using a number of convolution layers (1 to 10) to prove that the appropriate number of layers for Essex (number of images of five thousand with the number of 300 Classes) is four according to Eq.(8) mentioned above which is the best layer because it gave us the highest accuracy among the ten tests as shown in Fig. 7.

Figure 7 Chart of Test accuracy with convolution layers’ number from (1-10) layers of Essex data set.
In the same way, we tested the FEI by conducting ten experiments on ten layers (1 to 10) and found that the second layer is the one that gives the best accuracy that proved from the Eq.(8) and as shown in the fig.8

![Figure 8 Chart of Test accuracy with convolution layers’ number from (1-10) layers of FEI data set.](image1)

For Caltech dataset, we repeated the same process of ten cases of experiments on each layer (1 to 10) and proved that the third layer is the best with highest accuracy that exact what Said from the Eq.(8) and as shown in the fig.9

![Figure 9 Chart of Test accuracy with convolution layers’ number from (1-10) of Caltech data set.](image2)

15. COMPARISON

The evaluation results were compared with NIRFaceNet (M. Peng et al, 2016) result. Our dataset different from (NBN) dataset but in comparison from fig. (10) It is clearly the (Essex, FEI, and CALTECH) datasets have more challenge than CASIA NIR dataset in term of background, expression, pose, and colour. From another side our dataset is larger than CASIA NIR is or near in size table (10) shows all comparison details.

Figure 10. Samples of CASIA NIR data set

![Figure 10. Samples of CASIA NIR dataset](image3)
16. CONCLUSION

A new approach to select the best parameters of CNN architecture is suggested, designed, implemented and tested in this work. This approach is implemented by using Matlab software and it’s tested comprehensively using the well-known dataset (vggface2 dataset). Then it is evaluated with three groups of data set (Essex, FEI, and Caltech). The results are revealing very good network performance, especially for many types of small and middle scale dataset. This good performance is shown as an accurate metric of the classification results (99.21, 98.51 and 97.78) of the three datasets (Essex, FEI, Caltech) respectively. The authors are still developing this approach using different types and scales of convolutional neural network.

References

[1] Ahmad, F., Najam, A. & Ahmed, Z. 2013. Image-based face detection and recognition: state of the art, arXiv preprint arXiv:1302.6379.
[2] Ahonen, T., Hadid, A., Pietikinen, M. 2006. Face description with local binary patterns: Application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell,28:2037–2041.
[3] Al-Waisy, A. S., Qahwaji, R., Ipson, S., Al-Fahdawi, S. & Nagem, T.A. 2018. A Multi-Biometric Iris Recognition System Based On A Deep Learning Approach, Pattern Analysis and Applications, 21(3):783-802.
[4] Bansal, A., Nanduri, A., Castillo, C. D., Ranjan, R. & Chellappa, R. 2017. Umdfaces: An annotated face dataset for training deep networks, IEEE International Joint Conference on Biometrics (IJCB), 464-473.
[5] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. & Zisserman, A. 2018. Vggface2: a dataset for recognising faces across pose and age, IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 67-74.
[6] Chen, J.-C., Patel, V. M. & Chellappa, R. 2016. Unconstrained face verification using deep cnn features, IEEE winter conference on applications of computer vision (WACV), IEEE, 1-9.
[7] do Amaral, V., Figaro-Garcia, C., Gattas, G. J. F. & Thomaz, C. E. 2016. Normalização espacial de imagens frontais de face em ambientes controlados e não-controlados, FaSCI-Tech,
13

[1] Turk, M., Pentland, A. 1991. Eigenfaces for recognition, Journal of cognitive neuroscience, 3(1):71-86.

[8] Farokhi, S., Shamsuddin, S., Sheikh, M., Flusser, U., Khansari, J., Jafari-Khouzani, M. 2014. Near infrared face recognition by combining Zernike moments and undecimated discrete wavelet transform, Digit. Signal Process. 31:13-27.

[9] Farokhi, S., Sheikh, U., Flusser, J.&Yang, B. 2015. Near infrared, face recognition using Zernike moments and Hermit kernels, Inf. Sci. 316: 234-245.

[10] Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A.& Bengio, Y. 2013. Maxout networks, arXiv preprint arXiv:1302.4389.

[11] Guo, Y., Zhang, L., Hu, Y., He, X. & Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition, European Conference on Computer Vision, Springer, 87-102.

[12] He, K., Zhang, X., Ren, S.& Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. in Proceedings of the IEEE international conference on computer vision, 1026-1034.

http://www.vision.caltech.edu/html-files/archive.html.

[13] Jia, Y. 2015. Going deeper with convolutions, in CVPR.

[14] Krizhevsky, A., Sutskever, I.& Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks, in Advances in neural information processing systems, 1097-1105.

[15] Li, S., Chu, Z., Liao, R.& Zhang, S. 2007. Illumination invariant face recognition using near-infrared images. IEEE Trans. Pattern Anal. Mach. Intell, 29:627–639.

[16] Motlak, H. J., Hussein, E. A. R.& Hassony, A. S. 2018. Face Recognition System Based-On Imperialist Competitive Algorithm And Neural Network (ICANN). ARPN Journal of Engineering and Applied Sciences. 13( 22).

[17] Nair, V., Hinton, G. E. 2010. Rectified linear units improve restricted boltzman machines, in Proceedings of the 27th international conference on machine learning (ICML-10), 807-814.

[18] Parkhi, O. M., Vedaldi, A.& Zisserman, A. 2015. Deep face recognition in bmvc, 1(3): 6.

[19] Peng, M., Wang, C., Chen, T. & Liu, G. 2016. Nirfacenet: A convolutional neural network for near-infrared face identification. Information, 7(4).

[20] Schroff, F., Kalenichenko, D.& Philbin, J. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering-1A_089.

[21] SERRA, X. 2017. Face recognition using Deep Learning. Polytechnic University of Catalonia.

[22] Sermanet, P., Eigen, D., Zhang, X., Mathieu, Fergus, M. R.& LeCun, Y. 2013. Overfeat Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.

[23] Simonyan, K., Zisserman, A.2014. Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:1409.1556.

[24] Sun, Y., Chen, Y., Wang, X.& Tang, X. 2014. Deep learning face representation by joint identification-verification,in Advances in neural information processing systems. 1988-1996.

[25] Sun, Y., Wang, X.& Tang, X. 2015. Deeply learned face representations are sparse, selective, and robust, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2892-2900.

[26] Sun, Y., Wang, X.& Tang, X. 2014. Deep learning face representation from predicting 10,000 classes, IEEE conference on computer vision and pattern recognition, 1891-1898.

[27] Thoma, M. 2017. Analysis and optimization of convolutional neural network architectures, ArXiv preprint arXiv:1707.09725.

[28] Vojt, J. 2016. Deep neural networks and their implementation.

[29] Wu, J. 2017. Convolutional Neural Networks, Published online at https://cs. nju. edu. Cn/wujx/teaching/15 CNN.

[30] Yi, D., Lei, Z., Liao, S& Li, S. Z. 2014. Learning face representation from scratch, ArXiv preprint arXiv:1411.7923.
Zhang, Z. 2016. Derivation of Backpropagation in Convolutional Neural Network (CNN). University of Tennessee, Knoxville, TN.