IJICTR
International Journal of Information & Communication Technology Research

Volume 15- Number 1 – 2023 (16 -23)

A New Fuzzy Convolutional Neural Network for Face Recognition to Classify Authorized and Unauthorized Persons

Navid Mahmoudabadi
Department of Information Technology Management
Qeshm branch,
Islamic Azad University
Qeshm,Iran
n.mahmoodabadi@gmail.com

Mohammad Ali Afshar Kazemi*
Department of Industrial Management
Central Tehran Branch,
Islamic Azad University
Tehran, Iran
dr.mafshar@gmail.com

Reza Radfar
Department of Industrial Management
Islamic Azad University,
Science and Research Branch
Tehran, Iran
radfar@gmail.com

Seyed Abdollah Amin Mousavi
Department of Information Technology Management
Islamic Azad University, Central Tehran Branch
Tehran, Iran
dr.aminmousavi@gmail.com

Received: 5 May 2022 – Revised: 25 August 2022 - Accepted: 27 November 2022

Abstract—**Deep learning methods use neural networks that try to discover patterns within the image without human intervention and to learn that. One of the most popular algorithms in this field is the convolutional neural network algorithm. This algorithm uses several layers to receive the input image and process it so that the class label can be found. These layers are mostly based on Neural Networks. This research aims to provide a model of neural-fuzzy, based on convolutional neural network algorithm. In this research, we use the positive advantages of deep learning methods and fuzzy inference systems and present a new model of their application to Classify authorized and unauthorized Persons. For this purpose, we designed new neural-fuzzy layers to pass the image through them and finally classify each image. The results of the implementation of the above model show the efficiency and success of this system.**

Keywords: Deep learning, Convolutional Neural Network, fuzzy inference systems, Face Recognition

Article type: Research Article

Publisher: ICT Research Institute

* Corresponding Author

[DOI: 10.61186/itrc.15.1.16]

I. INTRODUCTION

The human sense of sight is incredibly advanced. With the help of this sense, we can recognize objects in a fraction of a second without hesitation and just say their names. We can also recognize features such as the color or dimensions of objects and distinguish them from the background. Our eyes receive color data and our brain converts that data into meaningful basic information, such as lines of curves or shapes, and this is how we realize what we are looking at now [1].

Nerve neurons are the basis for the formation of the human sense of sight. Certain neurons are responsible for receiving raw information from the human eye. This raw information enters the visual cortex of the brain after the preprocessing stage for final analysis. In general, neurons are solely responsible for performing this operation [2]. Therefore, this can be a good motivation for researchers to try to upgrade neural networks to achieve better results in the field of system vision.

The increasing demand for data organization and analysis is mostly due to the abundance of raw data generated by users on social networks and other data sources. Not all data generated is linear, so perceptron single layer neural networks or linear classifiers, as they are widely known, cannot be used to classify these data. We do not need to use any hidden layers when dealing with linearly separable data. In other cases, a hidden layer can usually be used to solve the problem. Also, in some limited cases it may be necessary to use two hidden layers. But in this case, we need a large number of random initializations or other methods to perform general optimization [3].

Deep learning is one of the trends in artificial intelligence that uses multilayer perceptron networks. Deep learning architecture is a convenient way to extract features that can be used for specific purposes such as classification, regression, information retrieval, speech recognition, visual object recognition, object recognition, and many other areas such as drug discovery or genetics.

One of the most common methods of deep learning is the convolutional neural network method. The architecture of convolutional neural networks is designed to take advantage of the two-dimensional structure of an image or any other two-dimensional input such as a speech signal. There are generally four key ideas about using convulsive neural networks. These four ideas are local connectivity, shared weights, alliances, and the use of multiple layers. Another advantage of using these networks is that they can be taught more easily than fully connected neural networks because they have fewer parameters [4].

These days importance of security is obviously clear to everyone and many researchers or technicians try to develop systems to increase the security level. These systems can be applied to sensitive politic or military areas or even smart home etc. [5 and 6]. That was our motivation to design a model which uses deep learning concepts to take part in the domain. As it is described later, our model uses fuzzy logic to be a more flexible and reliable model.

In this research, in our designed model, the input image passes through different layers of a convolutional neural network to estimate the classification of the image, but some of these layers have adapted the concepts of fuzzy inference systems to take advantage of this logic. Therefore, the final model is a neural-fuzzy model with the ability to classify face images and thus distinguish between authorized and unauthorized people.

Rest of the paper describe literature review in which background and concepts are explained, Our proposed method and theoretical basis and explanation of its novelty are next, and finally we show implementation results and analyze them.

II. LITERATURE REVIEW

Image classification is defined as the task of classifying images into a specific category and is considered a fundamental problem in computer vision and forms the basis of other computer vision tasks such as location, detection, and segmentation. Although this task can be considered secondary to human nature, it poses a greater challenge for an automated system. Some of its complexities include perspective-based subject diversity and class diversity due to the large variety of topics [7]. Previously, a two-step approach was used to solve categorization problems. Artificial features were first extracted from the image using feature extractors and then used as input for a teachable classifier. The main drawback of this approach was the accuracy of the classification task, which strongly depended on the design of the feature extraction stage and was usually considered a difficult task [8].

Deep learning models use multiple layers in nonlinear information processing to extract and transfer features and therefore pattern analysis and classification can overcome these challenges. Among these models, convolutional neural networks [9] have become a leading architecture for most image recognition, categorization, and recognition tasks [4]. Deep convolutional neural networks, despite initial success, have received more and more attention as a result of the deep learning renaissance [10] and [11]. And even with the advent of GPUs, Better algorithms of this attention also increased [12] and [13].

The most significant development that drew particular attention to classification task using convolutional neural networks was the 2012 Large-Scale Imaging Image Network (ILSRVC) Challenge [14], which was won by Convolutional neural networks used to categorize approximately 1.2 million images per 1000 categories, breaking previous records. Since then, convolutional neural networks have dominated subsequent versions of the ILSVRC, particularly the image classification section [15].

In addition, some advances in the following aspects of convolutional neural networks have been

[DOI: 10.61186/itrc.15.1.16]

implemented in recent years. Some of these aspects are: 1- Network architecture [16-18]; 2- Nonlinear activation capabilities [19] and [20]; 3- Components of monitoring [21] and [22]; 4- Adjustment mechanisms [22]; In addition, some of the open challenges, such as the size of the model and the slowness of the calculations and the attractive exploration of hostile specimens, have led researchers to pay more attention to image categorization with convolutional neural networks [23].

In addition, several general review studies in the field of deep learning have reviewed deep learning for visual perception, recent advances in convolutional neural networks, and the classification of convolutional neural networks for computer vision tasks [24-26].

III. Convolutional Neural Network Architecture

Convolutional neural networks are forward networks. There are several versions of convolutional neural network architecture; however, they all have convolutional and pooling layers that are grouped into modules. One or more fully connected layers, like a standard feedforward neural network, are placed after these modules. The modules are usually stacked to form a deep model. The following figure shows the typical architecture of convolutional neural networks to perform a face image classification task [27] and [28].

A. *Convolutional layer*

Convolutional layers act as feature extractors and thus learn the properties of their input images. Neurons are arranged in convolutional layers in feature maps. Each neuron in a feature map has an impact field that connects to a neighborhood of neurons in the previous layer through a set of measurable weights called a filter bank. Inputs are intertwined with learning weights to compute a new feature map and then the results of convolution operations are sent via the nonlinear activation feature. All neurons in a feature map have equivalent weights; however, different feature maps in a convolutional layer have different weights so that several features can be extracted at each location [4] and [8]. The output feature map is calculated as follows:

$$Y_k = f(W_k \times x)$$

Where the input image is indicated by x; the convolutional filter associated with the km feature map is indicated by W_k; the multiplication sign in this text refers to a two-dimensional convolution operator that is used to calculate the internal multiplication of the filter model at each location in the input image, and f represents the nonlinear activation function [29]. Nonlinear activation functions allow the extraction of nonlinear properties. Hyperbolic and sigmoid tangent functions have been used in the past, but more recently, single-fusion linear units have been used, the popularity and success of which have led to a field of research focusing on the development and application of new activation functions of convolutional neural networks to improve performance [4] and [30].

B. *Pooling layer*

The purpose of pooling layers is to reduce the spatial resolution of feature maps [31]. Early on, it was common to use mean pooling layers to propagate the average of all input values of a small neighborhood from one image to the next. However, newer models use maximum pooling layers to propagate the maximum values in one field of impact to the next layer. The Maximum Pooling selects the largest element in each field of influence in such a way that:

$$Y_{kij} = MAX\left(x_{kpq}\right), ((p,q) \in \mathbb{R}_{i,j})$$

Where the output of the pooling operation corresponding to the kth of the feature map is shown with Y_{kij}, and X_{kpq} shows the element in location (p, q) in the pooling area $\mathbb{R}_i$, j that places an impact field around that position [32]. The following figure shows the difference between the maximum alliance and the mean alliance. As the image size is 4× 4, if a 2 × 2 filter is applied, the maximum pooling outputs are the maximum values in each 2× 2 region, while the average pooling outputs are the correct values of the average rotation in each region of the sample.

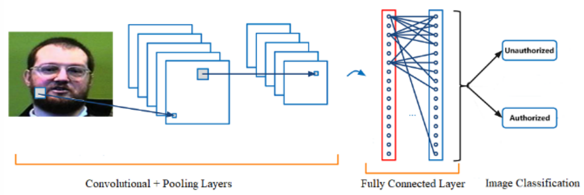


Figure 1. A view of Convolutional Neural Networks

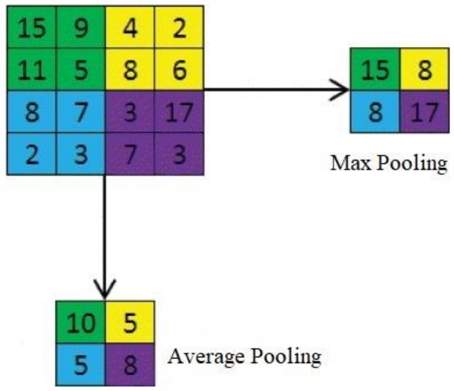


Figure 2. Two different types of Pooling

C. *Fully connected layers*

During the processing process, several cohesive and convolutional layers are superimposed on each other to extract more abstract features from the image by moving through the network. The fully connected layers that follow these layers interpret these properties and perform high-level reasoning capabilities. The standard task in categorization problems is to use the Softmax operator at the end of a convolutional neural network [12] and [15].

D. *Learning*

Convolutional neural networks generally use learning algorithms to adjust their parameters (biases and weights) to achieve the desired output in the network. The most common algorithm used for this purpose is post-publishing. The gradient post-diffusion

[DOI: 10.61186/itrc.15.1.16]

algorithm calculates an objective function (also known as cost, loss, or performance) to determine how to adjust the parameters of a network to minimize performance-related errors. A common problem in convolutional neural network training is over-adaptation, which is a poor performance in the experimental set after network training in a large or small training set. This affects the model's ability to generalize to unseen data and is an important challenge for convolutional neural networks that can be reduced by adjusting [33] and [34].

IV. Proposed Method

The proposed model developed in this research, like any other convolutional neural network, has four general layers as follows. To better convey the concept, each layer will be explained separately and then a description of the final model will be provided.

A. *Fuzzy Convolutional layer*

If in the simplest case we assume the input image to be a 7 by 7 matrix, the purpose of this layer is to combine adjacent pixels with each other in the hope that firstly it reduces the size of the image and secondly, to gather their information in a single pixel because adjacent pixels do not have much different information. For this purpose, it is necessary to slide another small matrix called the property vector, like a window, onto the input image, aggregating the information at each step, and finally reducing the size of the input matrix. For example, if we consider the feature vector 3 by 3, the image obtained from this step will have dimensions of 5 by 5 [35] and [36]. This problem is shown in the figure below.

In fact, what happens here is that the values in the feature vector play a major role in how the adjacent pixels are aggregated, how useful or unhelpful the information is, and in fact one of the steps in network learning is to update these values so that they can transmit better information, but the main question is whether the existing information is best combined and useful information is obtained by performing multiplication and addition operations. If one or more pixels for any reason have unreal values or noise, do multiplying these values and aggregating them lead to the transfer of noise from one step to the next? Suppose a pixel with the actual value of 10 is changed to 50 for some reason, such as noise. Now, do multiplying this value by the values in the property vector lead to a further increase in current noise (for example, 50 × 2 = 100)?

The numerous questions, some of which were raised above, were an incentive to find a model that could overcome the existing problems and limitations as much as possible and be successful in many cases where we face uncertainty. It is natural that wherever the issue of managing and controlling uncertainty is raised, the first and most effective approach that comes to mind is to use the concepts of fuzzy logic, because this concept is basically designed and predicted for this purpose. In a fuzzy set, unlike non-fuzzy sets, the boundary between a member's membership or non-membership is a gradual boundary. So while a member can be a member of the set, it can also be a member of another set.

In the model proposed in this research, we used a fuzzy inference system instead of a non-fuzzy property vector that merely multiplies and aggregates pixels. The goal of this system is to find the most important pixel that contains the most information. Since it is necessary to map the multi-pixel set property vector to a final numeric output value, we decided to use a Sugino fuzzy inference system. Therefore, in the new model, the numbers in the attribute vector are practically no longer coefficients and act as a degree of membership.

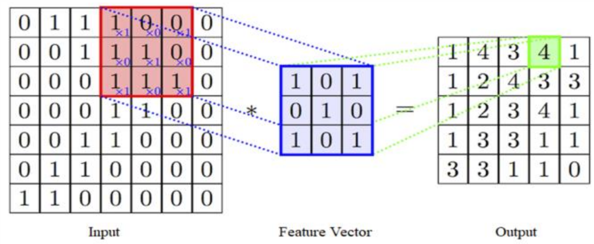


Figure 3. Operations performed in the convolutional layer

Because in the Sugino fuzzy system, the sequence of rules is defined as a non-fuzzy function, there is a great deal of flexibility in defining the function in such a way that on the one hand all the input pixels are included in the calculation and on the other hand the effect of Or, reduce some incompatible pixels (for example due to noise). In this step, we used the weighted average to calculate the final answer. In this case, each pixel has its share in the formation of output pixels without any prejudice and judgment, but incompatible pixels have no choice but to dissolve in the average of other pixels and thus have little effect during the process. Therefore, it can be hoped that after passing the image through several layers, the remaining information is only useful and original information that can play a key role in classification. Obviously, after deducting each rule, the Sugino system summarizes the result of the rules by performing a weighted average to arrive at a final number.

If we assume that the feature vector of a matrix is 4 by 4, the system rules will be as follows.

1- IF p1 is a1,1 AND p2 is a1,2 AND p3 is a1,3 AND p4 is a1,4 THEN z=avg(a1,1,a1,2,a1,3,a1,4)

2- IF p1 is a2,1 AND p2 is a2,2 AND p3 is a2,3 AND p4 is a2,4 THEN z=avg(a2,1,a2,2,a2,3,a2,4)

3- IF p1 is a3,1 AND p2 is a3,2 AND p3 is a3,3 AND p4 is a3,4 THEN z=avg(a3,1,a3,2,a3,3,a3,4)

4- IF p1 is a4,1 AND p2 is a4,2 AND p3 is a4,3 AND p4 is a4,4 THEN z=avg(a4,1,a4,2,a4,3,a4,4)

In the above rules, pis (i = 1,…, 4) are the pixels in question from the input image, ai, j (i = 1,…, 4 and j = 1,…,) are fuzzy sets that are matrices 4 in 4 attributes practically indicate the membership of each pixel to each of these fuzzy sets. Z is also a function that calculates the average pixels of the input image.

B. *Fuzzy ReLU layer*

The rectified linear unit layer cannot be considered an independent layer alone. In fact, this layer complements the tasks of the previous layer. As mentioned earlier, the purpose of this layer is to increase the nonlinearity of the image. In fact, images are all

[DOI: 10.61186/itrc.15.1.16]

non-linear naturally, a nd this non-linearity allows us to see and understand them, for example, lines, borders or color changes in images help us to recognize those images [35]. Despite the fact that the main purpose is to strengthen and increase the non-linearity of the image, what is done in practice expresses this issue accurately and completely, and in fact the function of this layer can be simply to eliminate negative pixels and convert it. The pixels are set to 0, and virtually nothing happens to the input image in this circle except for this change. Is it enough to just delete the negative pixels? Do pixels that themselves are the result of merging several pixels in the previous step, and now at this stage may have very close values due to the nature of the merger, necessarily convey a common knowledge? Is this process really reinforcing the nonlinearity of the input image?

For non-negative pixels, the rectified unit linear layer does not make any modifications or changes to this input image and directs the same input to the output. So in fact this layer does nothing for images containing non-negative pixels. The function on which this layer operates is shown below.

The idea presented in this layer is that instead of using the above diagram, fuzzy membership functions can be used. Fuzzy systems have several membership functions, such as bell, Gaussian, trapezoidal, triangular, and so on. By studying these fuzzy membership functions, we found the closest membership function to the above diagram, which was the S-shaped function. The image below is an example showing the S-shaped membership function.

The above proposed function, on the other hand, performs the main task of this layer well, which is to remove the negative pixels and replace them with 0, and in this respect it has no defects compared to the original model, but since it is a linear diagram with a slope of 45 Grade is not can also be effective in amplifying the pixels because it actually makes bold values bolder and lighter colors lighter so that the image is more non-linear and the knowledge gained from the previous step can be better. Reflect on the next step.

C. *Fuzzy Pooling layer*

This layer, as mentioned before, is responsible for selecting the main pixel that can contain the maximum knowledge. In fact, the goal of this layer is to reduce the size of the input image to include the least amount of knowledge loss. So, in this layer, in a specific neighborhood of the input image, the pixel containing the highest value goes to the next step and the other pixels are removed. But the important point here is

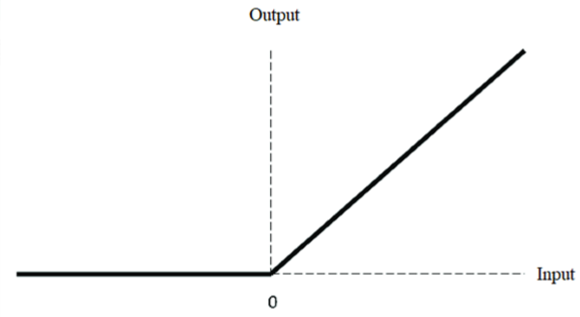


Figure 4. ReLU Function

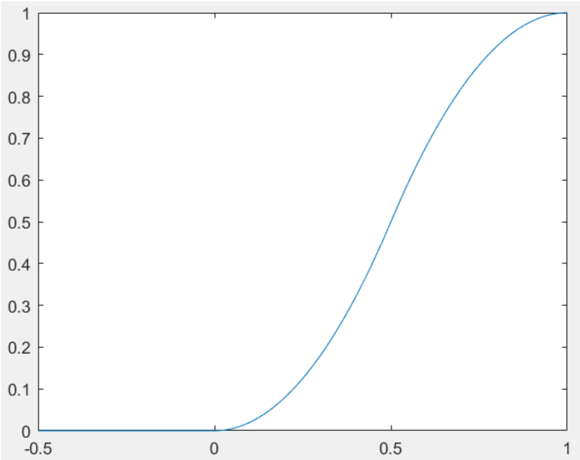


Figure 5. Fuzzy ReLU Function

does deleting other pixels mean lack of knowledge in them? Is there a scientifically significant difference between two pixels that have values of 0.018 and 0.019, for example? Due to its nature, this layer only selects only one pixel with the maximum value, and in the above example, assuming that the other pixels in the neighborhood under study have less values of 0.019; it deletes all those pixels and only the maximum pixel moves to the next stage.

Some methods recommend the mean alliance method instead of the maximum alliance method to solve this problem and make the most of the knowledge entered into this layer. In this case, instead of selecting one pixel in each neighborhood and deleting the other pixels, the average of all the pixels is selected and transferred as the output pixel to the next layer. But the important point is that the average is not necessarily a good way to do this.

Obviously, the best-case scenario could be an intermediate case that is not as rigid as the maximum pooling method and takes into account other pixels. Thus, this motivated us to find a solution in which we could hope that while all pixels played a role in determining the amount of output pixels, the pixel with the highest value would have the least effect with pixels that did not contain much knowledge. .

The proposed method for this layer is to examine how much each pixel has an impact on each neighbor's set, and to calculate the average weight of all pixels based on the effect of each pixel, or in other words

[DOI: 10.61186/itrc.15.1.16]

based on the degr ee of membership of each pixel. . In this case, each pixel plays a role in determining the final pixel as much as it does (more or less) because the value of pixels containing higher knowledge is not supposed to be the same as pixels containing lower knowledge. Therefore, first the sum of numbers in each neighborhood is calculated and then it is determined what percentage of each pixel played a role in creating this set. Then, the percentage of the role of each pixel is transferred to the range of zero to one and it is viewed as a membership. Then the same degree of membership is used as the weight in calculating the weighted average.

V. Implementation

In order to be able to implement and test the proposed layers, we implemented a hypothetical model of layers as follows:

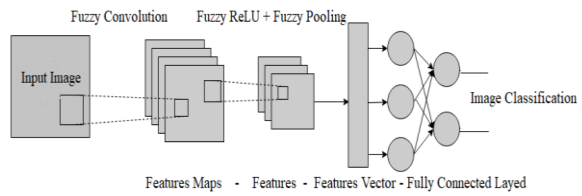


Figure 6. Fig. 1 Proposed Method

TABLE I. PROPOSED METHOD

Layer Number	Layer Name	Input Dimensions	Output Dimensions
1	Fuzzy Convolut ional	80 × 80	76 × 76
2	Fuzzy ReLU	76 × 76	76 × 76
3	Fuzzy Pooling	76 × 76	38 × 38
4	Fuzzy Convolut ional	38 × 38	34 × 34
5	Fuzzy ReLU	34 × 34	34 × 34
6	Fuzzy Pooling	34 × 34	17 × 17
7	Fully Connect ed	17 × 1	No. Of Classes

The above algorithm is implemented in a computer system with a 2.7 GHz core i7 processor and 16 GB of RAM. Also, in terms of software, the operating system of this computer is Windows 10 and the algorithm has been implemented in MATLAB 2018b software.

We also used some different databases shown in table below to test our model. As our goal is to classify authorized and unauthorized persons, we randomly divided each database into two equal parts, giving one part an authorized and the other an unauthorized label.

VI. Results and Analysis

After implementing the model, we calculated the accuracy to determine whether it is working properly or not. For this purpose, we ran the algorithm and calculated the ratio of the number of correct estimates to the total data. The following table shows the results of algorithm implementations with different databases.

TABLE II. USED DATABASES

Row	Name	No. Of Samples	No. Of Classes
1	Faces94	3059	153
2	Faces95	1440	72
3	Faces96	3016	152
4	LFW	13233	5749

TABLE III. TABLE 1 IMPLEMENTATION RESULTS

Database	LFW	Faces94	Faces95	Faces96
1	99.20	100.00	98.97	78.91
2	99.23	99.51	98.97	79.91
3	98.97	99.67	98.97	79.58
4	99.31	100.00	98.90	79.97
5	99.12	100.00	98.95	79.84
6	98.95	99.67	98.96	79.74
7	99.31	100.00	98.93	79.08
8	99.15	100.00	98.90	79.84
9	99.20	99.84	98.97	79.54
10	98.97	100.00	98.97	79.84

TABLE IV. MEAN AND STANDARD DEVIATION OF IMPLEMENTATION RESULTS

Database	Mean and standard deviation
Faces94	99.98 ± 0.19
Faces95	98.95 ± 0.03
Faces96	79.63 ± 0.36
LFW	99.04 ± 0.39

Obviously, the average alone is not a sufficient criterion for judgment, so it is necessary to calculate the standard deviation along with the average in order to have a better view of the results. The table below shows the mean and standard deviation of the proposed method on each database.

Despite the above results seem to be acceptable, we decided to compare these results to FaceNet method. FaceNet is one of the uses of face recognition based on deep learning. This method is a fairly new method, introduced by Googleresearches in 2015, using Deep Convolutional Network method [38 and 39].

Table below shows the comparison of our proposed method to two types of pretrained models were taken from CASIA-WebFace [40] and VGGFace2 [41].

In order to be able to display the accuracy of the algorithm visually, we also drew the confusion matrix in the form of an image each time we run it.

The following image also shows the error diagram (training data error and test data error) in one of the algorithm implementations with the Faces95 database. In this image, the error reduction status while running the program is quite obvious.

TABLE V. TABLE 2 COMPARISON OF OUR MODEL TO EXISTING MODELS

[DOI: 10.61186/itrc.15.1.16]

Database	Our method	VGGFace2	CASIA-WebFace
Faces94	99.98 ± 0.19	99.37	99.37
Faces95	98.95 ± 0.03	100	99.65
Faces96	79.63 ± 0.36	77.67	76.86
LFW	99.04 ± 0.39	99.05	99.05

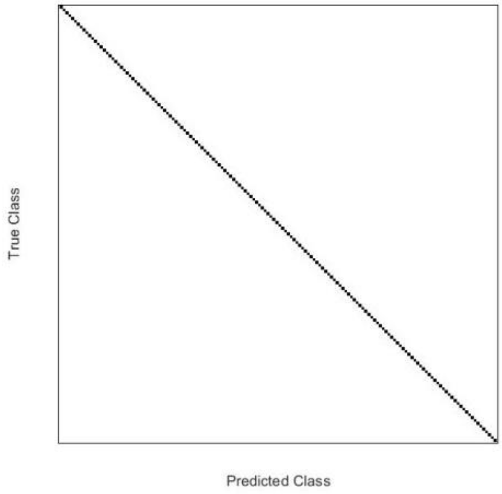


Figure 7. Confusion Matrix

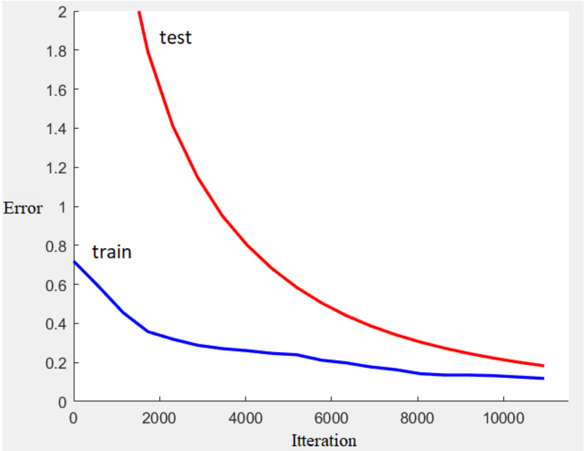


Figure 8. Train and Teas errors

Due to time and hardware constraints, the proposed model was implemented only on a few instances of the databases. The selection of these databases was based on the number of samples as well as the number of classes, and an attempt was made to select databases that were acceptable in terms of both the number of samples and the number of classes. The results show that our Fuzzy model is a very good and reliable classifier on these datasets exactly when we compare our model to FaceNet model. As explained above, using fuzzy logic in the layers gives us more strength to face mistaken values or noisy pixels.

It should be explained that fine tuning of the parameters is not discussed in this paper and can be a motivation for other researchers to work on this part.

VII. Coclusion

In this research, relying on the strengths of convolutional neural networks algorithm and fuzzy infer ence systems, a new model was presented to be able to recognize people's faces in images and classify those images. The use of fuzzy inference systems helped to make convolutional neural networks more flexible and, as a result, better transfer knowledge from one layer to another. As it is known in the algorithm of convolutional neural networks, the input image, passing through each layer, inevitably loses some of its information and enters the next layer. Therefore, it can be stated that practically the power of classification has a direct relationship with how to choose the pixels that are going to go to the next step and the pixels that are going to be deleted. Therefore, it seems that choosing the desired pixels to go to the next layer is complete; Determine a fuzzy inference system, considering all the pixels in the neighborhood. The obtained results show that this method can be a successful and acceptable method.

References

[1] Huble, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology, 215-243.

[2] Cohen, A. I. (1972). Rods and Cones. In Physiology of Photoreceptor Organs (pp. 63-110). Springer Berlin Heidelberg.

[3] Panchal, G., Amit, G., P., K. Y., & Devyani, P. (2011). Behaviour Analysis of Multilayer Perceptronswith Multiple Hidden Neurons and Hidden Layers. International Journal of Computer Theory and Engineering, 332-337.

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436–444.

[5] D. A. R. Wati and D. Abadianto, "Design of face detection and recognition system for smart home security application," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017, pp. 342-347, doi: 10.1109/ICITISEE.2017.8285524.

[6] D. A. Chowdhry, A. Hussain, M. Z. Ur Rehman, F. Ahmad, A. Ahmad and M. Pervaiz, "Smart security system for sensitive area using face recognition," 2013 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (CSUDET), 2013, pp. 11-14, doi: 10.1109/CSUDET.2013.6670976.

[7] Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. the International Joint Conference on Artificial Intelligence (pp. 1237–1242). California: AAAI Press.

[8] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. in Proceedings of the IEEE, 86(11), 2278-2324.

[9] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, 396–404.

[10] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507.

[11] Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. Advances in neural information processing systems, 2814–2822.

[12] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[13] Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing, 3(2), 1-29.

[14] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211–252.

[DOI: 10.61186/itrc.15.1.16]

[15] Simonyan, K., & Zisserman. (2014). Very deep convolutional networks for large-scale image recognition. ICLR.

[16] Zeiler, M. D., & Fergus, R. (2013). Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. arXiv:1301.3557.

[17] Gong, Y., Wang, L., Guo, R., & Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. the European Conference on Computer Vision (pp. 392–407). Berlin: Springer.

[18] Szegedy, C., Ioffe, S., & Vanhoucke, V. (2016). Inception-v4, Inception-Resnet and the impact of residual connections on learning. arXiv 1602.07261.

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv 1512.03385.

[20] Zhao, Q., & Griffin, L. D. (2016). Suppressing the unusual: Towards robust CNNs using symmetric activation functions . arXiv 1603.05145v1.

[21] Tang, Y. (2013). Deep learning using linear support vector machines. arXiv 1306.0239.

[22] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv 1207.0580.

[23] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. IEEE Conference on Computer Vision and Pattern Recognition, (pp. 1701-1708). Columbus.

[24] Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2(1), 1-127.

[25] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117.

[26] Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. Foundations and Trends in Signal Processing, 197–387.

[27] Hubel, D. H. (1959). Receptive fields of single neurones in the cat's striate cortex. Journal of Physiology, 148(1), 574–591.

[28] Hubel, D. H. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Physiology, 160(1), 106–154.

[29] Yu, D., Wang, H., Chen, P., & Wei, Z. (2014). Mixed pooling for convolutional neural networks. the 9th International Conference on Rough Sets and Knowledge Technology (pp. 364–375). Berlin: Springer.

[30] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. the 27th international conference on machine learning, (pp. 807–814).

[31] Thikshaja, U. K., & Paul, A. (2017). A Brief Review on Deep Learning and Types of Implementation for Deep Learning. In S. Karthik, A. Paul, & N. Karthikeyan, Deep Learning Innovations and Their Convergence With Big Data (pp. 20-32). IGI Global.

[32] Yu, W., Yang, K., Bai, Y., Yao, H., & Rui, Y. (2014). Visualizing and Comparing Convolutional Neural Networks. Computer Vision and Pattern Recognition.

[33] Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S., & Babu, R. V. (2016). A taxonomy of deep convolutional neural nets for computer vision. Frontiers in Robotics and AI .

[34] Buduma, N. (2017). Fundamentals of Deep Learning. O'Reilly Media, Inc.

[35] Denyer, S. (2018). Beijing bets on facial recognition in a big drive for total surveillance. washingtonpost.

[36] Wiatowski, T., & Bölcskei, H. (2018). A mathematical theory of deep convolutional neural networks for feature extraction. IEEE Transactions on Information Theory, 64(3), 1845-1866.

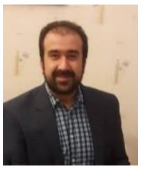
[37] Scherhag, U., Rathgeb, C., Merkle, J., Breithaupt, R., & Busch, C. (2019). Face Recognition Systems Under Morphing Attacks: A Survey. IEEE Access, 7, 23012-23026.

[38] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815-823, 2015.

[39] I. William, D. R. Ignatius Moses Setiadi, E. H. Rachmawanto, H. A. Santoso and C. A. Sari, "Face Recognition using FaceNet (Survey, Performance Test, and Comparison)," 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019, pp. 1-6, doi: 10.1109/ICIC47613.2019.8985786

[40] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning Face Representation from Scratch," 2014.

[41] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," International Conference on Automatic Face and Gesture Recognition, 2018.



Navid Mahmoudabadi received the B.Sc. degree from Islamic Azad University, in the field of Information Technology, and the M.Sc. degree from Shiraz University in the field of Artificial Intelligence. He is currently a lecturer at Zand University in Shiraz. His research interests include the area of Deep Learning, Machine Learning and Fuzzy Logic.

Mohammad-Ali Afshar Kazemi received the Ph.D. degree in Industrial Management, Operations Research. He is an Assistant Professor and a faculty member at the Islamic Azad University, Central Tehran Branch. His interests include Fuzzy Logic and Artificial Intelligence.

Reza Radfar received the Ph.D. degree in Industrial Management. He is a Professor and a faculty member at the Islamic Azad University, Science and Research Branch.

Seyed Abdollah Amin Mousavi has a Ph.D. degree in Information Systems. He is an Assistant Professor and a faculty member at the Islamic Azad University, Central Tehran Branch.

[DOI: 10.61186/itrc.15.1.16]