Reviewed document: 19 A New Method of Internal Type-2 Fuzzy based (Dr.P.Murugeswari) 27-5-2020.docx
Processing date: 23.6.2020 9:52 CEST

A total of 264 sentences were analysed. As a result 102 sentences (38.6%) were found in other documents.

These sentences are highlighted in the text by using different color shades according to the amount of similarity. The darker the highlighting color, the more words were found in another document. You may click the highlighted sentences in order to get further details about found reference documents. Learn more about this report and how to evaluate it.

The following graphic shows the distribution of found sentences within the checked document. The colored parts of the overview bar indicate those parts of the document in which sentences were found in other documents. The left boundary of the bar corresponds to the beginning of the document and the right boundary to the end of the document accordingly. By clicking into the overview bar you are directed to the corresponding position in the document.

Reference documents

The following list contains titles and addresses of documents in which similar sentences were found. With a click on the number of found sentences (\(\times\) Sentences) the corresponding sentences are highlighted in the document as well as in the navigation bar by a colored border and you are directed to the first position of the corresponding sentences in the document. Another click on \(\times\) Sentences resets the highlighting.

21 Sentences were found in a text with the title: \(\text{electronics-08-00997-v2.pdf - 4d206ca7564d4dfdf85a5942fa6151472}\), located at: https://doi.org/article/4d206ca7564d4dfdf85a5942fa6151472

14 Sentences were found in a text with the title: \(\text{Evolutionary-Fuzzy-Integral-Based Convolutional Neural ...https://www.mdpi.com › 2079-9292 › 8 › 9 › 997 › html}\), located at: https://www.mdpi.com/2079-9292/8/9/997/html

13 Sentences were found in a text with the title: \(\text{10.1.1.834.2198.pdf - download}\), located at: http://ctiteeex.rid.psu.edu/viewdoc/download?doi=10.1.1.834.2198&rep=rep1&type=pdf

12 Sentences were found in a text with the title: \(\text{th930929382.pdf}\), located at: http://files.mathsite.com/docs/thesis/th930929382.pdf

10 Sentences were found in a text with the title: \(\text{Evolutionary-Fuzzy-Integral-Based Convolutional Neural ...2}\), located at: https://www.mdpi.com/2079-9292/8/9/997/pdf

7 Sentences were found in a text with the title: \(\text{Color Textured Image Segmentation Using ICICM Interval ...https://engj.org › index.php › ej › article › download › 222 › 199}\), located at: https://engj.org/index.php/ej/article/download/222/199

7 Sentences were found in a text with the title: \(\text{Fuzzy Logic Theory - an overview | ScienceDirect Topics}\), located at: https://www.sciencedirect.com/topics/computer-science/fuzzy-logic-theory

6 Sentences were found in a text with the title: \(\text{UIO_Presentation_Erdal_opt.pdf}\), located at: http://heim.fli.uio.no/~jimtoer/robin/UIO_Presentation_Erdal_opt.pdf

5 Sentences were found in a text with the title: \(\text{146514893.pdf}\), located at: https://core.ac.uk/download/pdf/146514893.pdf

4 Sentences were found in a text with the title: \(\text{EUS-TS9 - ss_6_1.pdf}\), located at: http://www.eurasip.org/Proceedings/Eusipco/1996/paper/ss_6_1.pdf

4 Sentences were found in a text with the title: \(\text{EUS-TS9 - download}\), located at: http://ctiteeex.rid.psu.edu/viewdoc/download?doi=10.1.1.455.8928&rep=rep1&type=pdf

4 Sentences were found in a text with the title: \(\text{ResNet, AlexNet, VGG, Inception : 各種卷積神經架構的理解 - ...}\), located at: https://www.ctolib.com/amp/topics-128115.html

4 Sentences were found in a text with the title: \(\text{Format guide for AIRCC}\), located at: https://zenodo.org/record/113648

4 Sentences were found in a text with the title: \(\text{A Survey of the Recent Architectures of Deep Convolutional ...https://arxiv.org › abs › 1901.0603227}\), located at: https://arxiv.org/abs/1901.0603227

4 Sentences were found in a text with the title: \(\text{ResNet, AlexNet, VGGNet, Inception: Understanding various ...}\), located at: https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

3 Sentences were found in a text with the title: \(\text{A Survey of the Recent Architectures of Deep Convolutional ...https://www.researchgate.net › publication › 330511306_A_Survey_of_the_Recent...}\), located at: https://www.researchgate.net/publication/330511306_A_Survey_of_the_Recent_Architectures_of_Deep_Convolutional_Neural_Networks

3 Sentences were found in a text with the title: \(\text{Accepted in Artificial Intelligence Review (Springer ...27}\), located at: https://arxiv.org/pdf/1901.06032

3 Sentences were found in a text with the title: \(\text{Object Classification - ConvNet - ImageNet dataset - Keras ...}\), located at: https://pyblog.vc/object-classification-convnet-imagenet-dataset-keras-tensorflow/

3 Sentences were found in a text with the title: \(\text{IJRNET-V31658.pdf}\), located at: https://www.irjet.net/archives/V3/16/IJRNET-V31658.pdf

3 Sentences were found in a text with the title: \(\text{Color Textured Image Segmentation Using ICICM Interval ...https://engj.org › index.php › ej › article › download › 222 › 199}\), located at: https://engj.org/index.php/ej/article/download/222/199

3 Sentences were found in a text with the title: \(\text{ResNet, AlexNet, VGG, Inception : 各種卷積神經架構的理解}\), located at: http://www.sohu.com/a/211091005_642762
A New Method of Internal Type-2 Fuzzy based CNN for Image classification

Dr.P.Murugeswari

Professor, Department of Computer Science and Engineering.
Karpagam College of Engineering, Coimbatore, Tamilnadu, India
pmurugeswark7@gmail.com (mail)

Abstract

Last two decades, neural network and fuzzy logic have been successfully implemented in intelligent systems. The fuzzy neural network system means the combination of fuzzy logic and neural network concepts, which includes the advantages of fuzzy logic and neural network. This FNN is applied in many scientific and engineering areas. Wherever there is an uncertainty associated with data fuzzy logic play a vital role. The fuzzy set can represent and handles uncertainty information effectively. The main objective of the FNN system is to achieve a high level of accuracy by including the fuzzy logic in either neural network structure, activation function or learning algorithms. In computer vision and intelligent system, Convolutional Neural Network has more popular architectures and their performance is excellent in many applications. In this paper fuzzy based CNN image classification methods are analysed, also interval type-2 fuzzy based CNN is proposed. From the experiment it is identified that the proposed method performance is well.

Keywords: CNN, FCNN, Interval Type-2 Fuzzy logic, Image Classification

INTRODUCTION

Image classification is the task of classifying a given image into one among the pre-defined categories. Traditional image classification involves two modules such as feature extraction and classification. Feature extraction involves extracting a better level of data from raw pixel values which will capture the excellence among the categories involved. This feature extraction is completed in an unsupervised manner wherein the classes of the image don't have anything to do with information extracted from pixels. Some of the normal and widely used features are GIST, HOG, SIFT and LBP etc. After the feature is extracted, a classification module is trained with the pictures and their associated labels. A few samples of this module are SVM, Logistic Regression, Random Forest, decision trees etc.

Among the different types of neural network architecture, Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), Artificial Neural Networks (ANN) and Convolutional Neural Network (CNN) etc., CNNs are the most popular architecture. The CNN is suitable to image database, it works amazingly on computer vision tasks like image classification [1,26], object detection [24], image recognition [39], etc. In CNN there is no separate feature extractor, it is built in, which includes, feature extraction and classification modules integrated system and it learns to extract, by discriminating representations from the images and classify them based on supervised data.

The CNN is used in a number of tasks which have a great performance in different applications. CNN has been presenting an operative class of models for better understanding of contents present in an image, therefore resulting in better image recognition, segmentation, detection and retrieval. CNN architectures are efficiently and effectively used in many pattern and image recognition applications [18], for example, gesture recognition [39,46], face recognition [5,33], object classification [38,16] and generating scene descriptions [29].

Zadeh[42] introduced the concept of fuzzy logic(type-1 fuzzy) for solving control system related problems. Later, researchers have contributed many interesting applications in the field of computer vision. Conceptually, Type-2 Fuzzy set (T2FS) were introduced by Zadeh in 1975[43], further it is developed by Jerry M.Mendel. In Type-1 Fuzzy Set (T1FS) the degree of membership is specified by a crisp number belonging to the interval [0, 1]. In T2FS the degree of membership itself fuzzy and it is represented by secondary membership function. If the secondary membership function is at its maximum of 1 at every point which is called as Interval Type-2 Fuzzy Set (IT2FS) [10]. The T2FS include a third dimension and footprint of uncertainty as shown in Figure 1, which gives the extra degree of freedom to handle uncertainties. This extra degree of fuzziness helps more efficient way of handling uncertainty. Figure 2, shows the secondary Membership Functions (MFs) (third dimension) of the T1FS (Figure 2a), the IT2FS (Figure 2b) and the general T2FS (Figure 2c) as induced by the same input p as shown in Figure 1.

Figure 1. An example of the three types of fuzzy sets. The same input p is applied to each fuzzy set: (a) T1FS (b) IT2FS. (c) T2FS.

Figure 2. A view of the secondary membership functions (third dimensions) induced by an input p for (a) T1FS (b) IT2FS; (c) T2FS

Generally, FCM type-1 has become the most well-known algorithm in cluster analysis. Many researchers have shown that there are limitations in the ability of T1 FSs to model and minimize the effect of uncertainties, because its membership grades are crisp. The T2FS is characterized by Membership Functions (MFs) that are themselves fuzzy. The IT2FS [6], a special case of T2FS, are currently most widely used because of their reduced computational cost. An IT2FS is bounded with two T1FSs above and below, which are called Upper MF (UMF) and Lower MF (LMF) respectively and area between UMF and LMF is Footprint of Uncertainty (FOU).

Now T2FS prove to model various uncertainties but it increases the computational complexity, because of its additional dimension of secondary grades for each primary membership. Example applications are Type-2 Fuzzy Clustering [13], Gaussian Noise Filter, Classification of coded video streams, Medical applications and Color image segmentation.

LITERATURE SURVEY:
The CNN is a special type of Neural Networks, which has shown exemplary performance on several competitions related to Computer Vision and Image Processing. Some of the exciting application areas of CNN include Image Classification and Segmentation [22], Object Detection [23], Video Processing [34], Natural Language Processing [38,39], and Speech Recognition [19,45]. The powerful learning ability of deep CNN is primarily due to the use of multiple feature extraction stages that can automatically learn representations from the data. The availability of a large amount of data and improvement in the hardware technology has accelerated the research in CNNs, and recently interesting deep CNN architectures have been reported. Several inspiring ideas to bring advancements in CNNs have been explored [5], such as the use of different activation and loss functions, parameter optimization, regularization, and architectural innovations.

Researchers have been combined neural network and fuzzy logic and implemented successfully in intelligent systems. The Fuzzy Neural Network (FNN) system means the combination of fuzzy logic and neural network concepts, which includes the advantages of fuzzy logic and neural network. This FNN is applied in many scientific and engineering areas text sentiment analysis [27], object classification with small training database [23], to extract high level emotion features from text [30], audio and visual modalities to understand the emotion understanding of movie [35], real world objects and image classification [31,34], to predict the traffic flow [11], electric load forecasting [36] and to recognize handwritten digits [40]. Keller et. al [15] proposed fuzzy system is a hierarchical deep neural network that derives information from both fuzzy and neural representations, Stanton R. Price [32] et al proposed introducing the fuzzy layers for deep learning, fuzzy approaches taken to deep learning have been through the application of various fusion strategies at the decision level to aggregate outputs from state-of-the-art pre-trained models, e.g., AlexNet, VGG16, GoogLeNet, Inception-v3, ResNet-18, etc.

**SUPPORTING THEORY**

**Type-1 Fuzzy Logic**

**Definition:** The T1FS, \( A \), which is in terms of a single variable, \( x \in X \), may be represented as:

\[
A = \{ x \in X : \mu_A(x) \} \quad (1)
\]

The fuzzy set, \( A \), can also be defined as:

\[
A = \{ x \in X : \mu_A(x) \} \quad (2)
\]

where \( \mu_A(x) \) denotes union over all admissible \( x \).

Fuzzy sets are extension of classical sets. Unlike classical sets, fuzzy sets permit partial membership. The fuzzy set \( A \), defined in the domain \( X \), is described by a membership function \( \mu_A \) which maps \( x \) to the real interval \([0, 1]\). For each \( x \in X \), \( \mu_A(x) \) yields the degree of membership of \( x \) to the fuzzy set \( A \) (Figure 3).

**Figure 3. Gaussian Type-1 fuzzy membership function**

Fuzzy sets are key components of fuzzy rules. Fuzzy rules allow a processing strategy to be expressed in the form of approximate reasoning. As an example, let us consider the following rule:

If \( x_1, A1 \text{AND} x_2, A2 \text{AND} x_3, A3 \) then \( y, B \) \( (3) \)

where, \( A1, A2, A3 \) and \( B \) are fuzzy sets associated to the quantities of \( x_1, x_2, x_3 \) and \( y \), respectively. Through observing such a rule, three antecedent clauses can be noticed, which define conditions and a consequent clause which defines the corresponding action.

**Type-2 Fuzzy Logic**

**Definition:** The T2FS, \( A \), may be represented as [22]

\[
A = \{ x \in X : \mu_A, u \} \quad (4)
\]

where \( \mu_A, u \) is the T2 fuzzy membership function in which \( 0 \leq \mu_A, u \leq 1 \). A can also be defined as (Mendel and Robert 2002):
where $J_x$ denotes union over all admissible $x$ and $u$.

$J_x$ is called as primary membership of $x$. Additionally, there is a secondary membership value corresponding to each primary membership value that defines the possibility for primary memberships (Liang and Mendel 2000 a)]. Whereas the secondary membership functions can take values in the interval of $[0, 1]$ in generalized T2FLSs (Figure 4), they are uniform functions that only take on values of 1 in interval T2FLSs. Since the general T2FLSs are computationally very demanding, the use of interval T2FLSs is more commonly seen in the literature, due to the fact that the computations are more manageable.

**Figure 4. Gaussian Type-2 fuzzy membership function**

The implementation of T2FLS involves the operations of fuzzification, inference and output processing. The output processing consists of type-reduction and defuzzification. The type-reduction methods are extended versions of T1 defuzzification methods. The type-reduction captures more information about rule uncertainties than the defuzzified value (a crisp number), however, it is computationally intensive, and except for IT2FSs it uses a simple type-reduction computation procedure.

**Figure 5. T1FLS block diagram**

**Figure 6. T2FLS block diagram**

Following are the basic blocks of a T2FLS[43,44]:

- **Fuzzifier**: The fuzzifier maps crisp inputs into T2FSs which activate the inference engine. **Rule base**: The rules in T2FLS remains the same as in T1FLS, but antecedents and consequents are represented by IT2FSs.

- **Inference**: Inference block assigns fuzzy inputs to fuzzy outputs using the rules in the rule base and the operators such as union and intersection. In T2FSs, join and meet operators, which are new concepts in fuzzy logic theory, are used instead of union and intersection operators. These two new operators are used in secondary membership functions.

- **Type-reduction**: The T2 fuzzy outputs of the inference engine are transformed into T1FSs that are called the type-reduced sets. There are two common methods for the type-reduction operation in the IT2FLSs: One is the Karnik-Mendel iteration algorithm and the other is Wu-Mendel uncertainty bounds method. These two methods are based on the calculation of the centroid.

- **Defuzzification**: The outputs of the type-reduction block are given to defuzzification block. The type-reduced sets are determined by their left end point and right end point and the defuzzified value is calculated from the average of these points.

**3.3 Interval Type-2 Fuzzy Logic**

**Definition**: When all $\mu_{Ax,u}$ are equal to 1, then $A$ is an IT2FLS. The special case of Equation (6) might be defined for the IT2FLSs[17]:

$$A = x \in \mathbb{E} \times J_{x} \times \mathbb{R} \times \mathbb{R}, J_{x} \subseteq [0, 1]$$

IT2FSs a special case of T2FSs, are currently used for their reduced computational cost.

**Figure 7. Two-dimensional representations of T2 Fuzzy Membership Function**

**3.4 Convolutional Neural Network**

Conventional CNNs[25] consist of two parts, as shown in Figure 8. The first part contains the steps of convolution and pooling which are used for feature extraction; The second part is a classifier which uses fully connected layers. Conventional CNNs still use feature extraction plus a classifier; however, lately, CNNs have replaced fully connected layers with average pooling. This can reduce the large number of required parameters and lower the degree of overfitting as well.
**Convolution layer** is used to extract the features which act as filters for images, the general conventional filters are sobel and Gabor. The Equation (7) is used to calculating the match feature points in two dimensional matrix.

\[
Y_{IJ} = i0w_{ij} = 0h_{ij} + 1(i-1)(j-1)*k_{ij} \quad (7)
\]

Where Y_{IJ} is an output matrix, and kw and kh are the width and height of a convolution kernel, respectively. In general, kw=kh represents a square convolution kernel, is the xi input matrix, and kij is the weight of the convolution kernel which needs to be updated during training. In order to maintain a constant size after convolution, zero-padding is used at the edge of the input matrix.

**Pooling**

After the convolution operation, the extracted features can theoretically be classified directly. However, this requires a huge number of parametric operations, which makes the training process difficult and prone to overfitting. Pooling is one way to reduce the dimensions.

The pooling process applies a mask operation on the input matrix with a sliding window. During the process, the moving step is equal to the width of the convolution kernel. Each element is calculated only one time. Therefore, an N X N mask can lower the input feature matrix \( \frac{1}{N} \) times to achieve the effect of reducing the dimensions.

There are two pooling operations: Max pooling and average pooling. Max pooling takes the maximum value in a region \( R_{ij} \) and ignores other values. Average pooling calculates the average value in a region. The max pooling function is expressed as

\[
ak_{ij} = \max_{p,q \in R_{ij}} p_{pq} \quad (8)
\]

and the average pooling function is shown by the following equation:

\[
ak_{ij} = \frac{1}{|R_{ij}|} \sum_{p,q \in R_{ij}} p_{pq} \quad (9)
\]

Where \( ak_{ij} \) is the output a activation of the \( k^{th} \) feature map at \( i,j \), and \( p_{pq} \) is the input activation at \( p,q \) within \( R_{ij} \) and \( R_{ij} \) is the size of the pooling region. To retain the most important features, max pooling is the pooling method which is used during the convolution process. In contrast, global average pooling is usually applied in the output layer, because when classifying a target, average pooling contains physical meaning in the calculation process, therefore, a larger target can get a higher output value in the calculation.

**Activation Function**

The purpose of the activation function is to get nonlinear outputs from linearly combined networks. The disadvantage of sigmoid is that when the network gets deeper, gradient disappearance problems occur during back propagation. Therefore, rectified linear unit (ReLU) is used herein as the activation function. The definition of ReLU is illustrated as follows:

\[
f(x) = \begin{cases} 
0, & \text{if } x < 0 \\
 x, & \text{if } x \geq 0 
\end{cases} \quad (10)
\]

where if input x is smaller than 0, then the output is 0. If input x is bigger than 0, the output is x.

**3.5 FNN structure**

Generally FNN have five layers: 1) an input layer, 2) a membership layer to fuzzify input, 3) a rules layer that computes products of the second layers outputs, 4) a normalization layer to defuzzify the values, and 5) fully connected layer to produce the output. Inputs and outputs to this system are crisp, and the fuzzy logic takes place in the hidden layers of the network.

For a multi-input FNN system, every input unit is graded for its degree of membership to each fuzzy set. Therefore, an input to the system generates a fuzzy value instead of a crisp value. Fuzzy rules can then be generated to perform fuzzy inference. For instance, given inputs \( x_i = 1,2,...n \) outputs \( y_j = 1,2,...m \) and the \( l^{th} \) fuzzy rule \( R_l \) is

\[
R_l: \text{ if } x_1 \text{ is } A_{1l} \text{ and } x_n \text{ is } A_{nl} \text{ then } y_1 \text{ is } w_{1l} \text{ and ... and } y_n \text{ is } w_{nl} \quad (11)
\]

where \( A_{il} \) is the fuzzy set with \( i^{th} \) input and \( l^{th} \) fuzzy rule. After doing fuzzy inference, the output must be defuzzified to the crisp values. For a traditional defuzzification

\[
y_j = \mu_{A_{1l}x_1}A_{1l} + \mu_{A_{nl}x_n}A_{nl} \quad (12)
\]

where \( \mu_{A_{il}x_i} \) is the membership function chosen by the fuzzy rule.

When the input number is too large, the combination of fuzzy inference becomes too complicated for traditional FNN to compute. With help of MISO FNN it is possible to split the input unit to reduce computation and merge the output with every inference result. The input unit is divided into k sets. That is \( x_{pi} = 1,2,...,k \). The \( i^{th} \) fuzzy rule is

\[
R_l: \text{ if } x_{1p} \text{ is } A_{1l} \text{ and } ... x_{np} \text{ is } A_{nl} \text{ then } y_1 \text{ is } w_{1p} \text{ and...and } y_n \text{ is } w_{np} \quad (13)
\]

But, the input units must be split with caution, that the data should be meaningful in the same set. This approach can reduce computation effectively when the amount of input is too large.
4. COMPARISON OF FCNN ARCHITECTURES

Generally CNN architecture consists of two phases; there are feature extraction and classification. The FCNN is the combination of CNN and fuzzy logic; therefore the fuzzy logic may include either feature extraction phase or classification phase. Dependence up the application the researchers have proposed various FCNN architectures including fuzzy logic in feature extraction phase or classification phase. Here the two FCNN architectures have been compared for image classification. In these two architectures fuzzy logic is included in classification phase.

**Figure 9. The structure of a convolutional neural network (CNN)**

**FCNN Model 1:**

Normally, convolutional layers learn to extract features from the input data and fully connected layers sum up the feature information from the output of convolutional layer. But in fuzzy neural network, pixels in feature map are crisp values rather than fuzzy values.

**Figure 10. The architecture of convolutional fuzzy neural network proposed by [23]**

Min-Jie Hsu et. al [23] have approached integrated a convolutional neural network with a fuzzy neural network, where the FNN summarizes the feature information from every fuzzy maps. Fuzzy maps are the maps graded with the fuzzy sets in membership function. Each feature map will come up with M fuzzy maps, where M is the number of fuzzy sets in the membership function. In this approach, there are three fuzzy sets $M = 3$, i.e., “Negative, Zero, and Positive,” in our membership function; the number of final convolutional feature maps $k = 80$ and each map is a $3 \times 3$ image. That is, this architecture have $k \times M = 240$ fuzzy maps. When, there are too many input units, which result in huge computations.

$$N = ex^{-m}22n^2, Z = ex^{-22n^2}, P = ex^{-m}22n^2$$ (14)

Therefore, here the fuzzy neural network is utilized as semi-connected layers to sum up the feature information. i.e, each of the inputs of the FNN becomes a feature map instead of forming whole feature maps.

**CFNN** has the same architecture as traditional CNN through Layer I to Layer V and this approach does not require any activation function in layer V to obtain a symmetric Gaussian distribution before the fuzzifier layer. Then, the apply dropout [20] after Layer V to prevent over fitting.

**FCNN Model 2:**

Kseniya P. Korshunova[16 ] have proposed CFNN architecture which includes the fuzzy layer, which situated in between convolutional network and classifier. CFNN architecture model has four types of layers. There are convolutional layer, pooling layer, Self-Organization or Fuzzy Layer, and fully-connected layer. The full Convolutional Fuzzy Neural Network architecture has stack of three parts: i) a convolutional network (convolutional and pooling Layers); ii) The Self-Organization Layer (The Fuzzy Layer); iii) a classifier (some fully-connected layers).

**Figure 11. The architecture of convolutional fuzzy neural network proposed by [16]**

In this model the convolutional network takes input images and forms some abstract high-level properties of it by series of convolutional and pooling layers interchange. The Fuzzy Layer performs a preliminary input data distribution into a predetermined number of clusters. The outputs of the Fuzzy Layer neurons represent the values of the membership functions for the fuzzy clusters of input data. The data point’s cluster is decided depends upon their membership grade. These values go to the input of a classifier. Its output is the full CFNN output (the class scores).

Let $L$ be the number of neurons of the fuzzy layer (the number of clusters). The neurons of the fuzzy layer activation functions are radial basis functions, which is in the form of a Gaussian function modeling the membership of the input vector $x$ to each of the $L$ clusters.

$$fx = 1\sigma^2\pi e^{(x-m)^2/\sigma^2}$$ (15)

parameter $m$ is the centroid of a cluster, parameter $\sigma$ is blurring of cluster boundaries level (both are real values).

If the vector $x=x_1,x_2,\ldots,x_j,\ldots,x_n$ is fed to the input of the network, the fuzzy layer formed a vector consisting of the degrees of belonging $x$ to the specific cluster centers: $[\mu_1x, \mu_2x, \ldots, \mu_Lx]$. The components $\mu_kx$ are calculated using Equation (16) to satisfy the normalization condition using Equation (17) for each training sample vector $x_k$, $k=1,\ldots,K$ where $K$ is the number of vectors in training set.

$$\mu_lx(k)=\sum_{s=1}^{K-1} 1_{x_lx(k)}$$ (16)

$$l=1L\mu_lx(k)=1$$ (17)

The outputs of neurons of the fuzzy layer are used as inputs of the classifier.
The new IT2FCNN architecture integrates the features from CNN and the FNN. A new architecture integrates the Interval type-2 Fuzzy Rectifying Unit (IT2FRU) [3] activation function in convolution for features extraction in CNN and interval type-2 fuzzy based classification in fuzzy layer. This method combines the advantages of both network architectures and interval type-2 fuzzy logic. IT2FCNN architecture has four types of layers: i) convolutional layer with IT2FRU ii) pooling Layer iii) fuzzy layer; iv) a fuzzy classifier

Classification using Fuzzy layer and IT2FCM

Feature Extraction – CNN with IT2FRU

Input Image

Pooling

Convolution using IT2FRU

Fuzzy Layer - Interval Type-2 fuzzy membership function

Fuzzy Classification using IT2FCM

Output

Figure 12. Block diagram of the proposed method IT2FCNN

The convolutional neural network takes an input images and perform series of convolutional and pooling layers. The fuzzy layer performs clustering using interval type-2 fuzzy clustering algorithm. The outputs of the fuzzy layer neurons represent the values of the membership functions for the fuzzy clusters of input data. The data point’s cluster is decided depends upon their membership grade. These values go to the input of a classifier. Its output is the full IT2FCNN output is the class scores for the image.

Let $C$ be the number of neurons of the fuzzy layer (the number of clusters). The neurons of the fuzzy layer activation functions are IT2FRU modeling the membership of the input vector $x$ to each of the $L$ clusters.

IT2FRU employ the following equalities; $Z = 0$ to guarantee that $\sigma = 0 \Rightarrow \phi_0 = 0$. Additionally, the height of the LMFs is employed as $m_2 = \alpha$, $m_1 = m_3 = 1 - \alpha$ as suggested in [20]. The resulting IT2-FM ($\phi_\sigma(\sigma)$) for $\sigma \in [0, 1]$ can be formulated

$$
\phi_0 = P(\sigma \kappa) (18)
$$

where $\kappa$ is define as

$$
\kappa = \frac{12\alpha + \sigma - 1 + \alpha \sigma + 1}{1 + \alpha} (19)
$$

Similarly for the input interval $\sigma \in [-1, 0]$ the IT2FM can be derived as

$$
\phi_0 = N(\sigma \kappa) (20)
$$

The activation unit can be formulated by arranging the Equation (18) and Equation (20) as following:

$$
\phi = P(\sigma \kappa), \text{if } \sigma > 0
$$

$$
\phi = N(\sigma \kappa), \text{if } \sigma \leq 0 (21)
$$

the parameter $P$ controls the slope of the function in positive quadrant, while the parameter $N$ controls the slope of the function in negative quadrant. The resulting output of the IT2-FRU could be a linear or nonlinear activation depending on selection of the parameters. IT2FRU has three learnable parameters $P$, $N$ and $\alpha$.

The vector $x = x_1, x_2, \ldots, x_j, \ldots, x_n$ is fed to the input of the network, the fuzzy layer formed a vector consisting of the degrees of belonging $x$ to the specific cluster centers: $[v_1, v_2, \ldots, v_j]$. The components $(u_jx_i, u_j(x_i))$ are calculated using Equation (22) to satisfy the normalization condition using Equation (23) for each training sample vector $x_k, k = 1, \ldots, K$ where $K$ is the number of vectors in training set. The outputs of neurons of the fuzzy layer are used as inputs of the classifier.
The Interval Type-2 Fuzzy Membership becomes

\[
\mu_{ixi} = k = 1 C \delta j i k + \alpha \delta j id i k - 1 , \quad \text{if } 1 k = 1 C \delta j i k + \alpha \delta j id i k - 1 , \\
\delta 2 m 1 - 1 , \quad \text{otherwise}
\]

Updating cluster centers

\[
v_j = v_L + v_R 2
\]

Type reduction and hard-partitioning can be obtained as follows:

\[
u_{jxi} = u_j R x_i + u_j L x_i 2, j = 1, \ldots, 1 C
\]

where \( u_j x_i = u_j x_i, \) if \( x_i \) uses \( u_j x_i \) for \( v_j R u_j x_i, \) otherwise

\[
u_j L x_i = l = 1 M u_j l x_i M
\]

where \( u_j x_i = u_j x_i, \) if \( x_i \) uses \( u_j x_i \) for \( v_j L u_j x_i, \) otherwise

The work of the IT2FCNN is divided into three stages: the input pattern (image) comes through a series of transformations, as a result a vector of high-level characteristics is formed; further, fuzzy layer performs a preliminary distribution of the input data into fuzzy clusters; the last fully connected layers perform the classification, assigning the result class label to each group of clusters.

EXPERIMENTAL RESULT:

6.1 Data set and Experimental setup

The various datasets are available for apply the neural networks. The most popular datasets are CIFAR-10, Caltech101 and ImageNet (sample images are shown in the Figure 13,14 and15 respectively). The CIFAR-10 data set consist of 60,000 images in 10 classes, with 6,000 images per class. The caltech101 data set consists of 101 classes with 40 to 800 images per class. The ImageNet dataset has more than 14 million images, with 20,000 categories. The experiment have been implemented in Windows 7 64 bit operating system and the main memory and storage capacity of the computer systems is 8 GB RAM and 1TB respectively. The type of the CPU used is Intel(R) Core(TM) i5-4590 CPU @3.30GHz, and the configuration of the graphics card is NVIDIA GeForce GT 705. The software used in this experiment are Python 3.6 and Matlab. The PyCharm IDE is used and network is built using the Keras libraries on PyCharm experimental platform.

6.2 Training the architecture

The training of IT2FCNN is foremost step, which includes three independent steps of the three components of the net. First the model is trained using the abstract properties of the input image by back propagation model. The second part of the model, fuzzy layer is tuned using the competitive learning scheme, which means choosing the parameters of the membership function for setting the cluster centers. Various fuzzy clustering algorithms are available; here IT2FCM is used for clustering. Finally, the classifier is trained using the weights tuning in the fully connected layers. When the training part is completed now the IT2FCNN becomes ready for implementation, now the image pixel array is fed to the CFNN and the output of network is input image p class scores, image is assigned the class max score value class.

Figure 13. Sample images CIFAR 10 data set

Figure 14. Sample images from caltech 101 dataset
6.3 The comparative Analysis:

In this paper the AlexNet, ZFNet, GoogleNet, VGGNet16, ResNet50 pretrained on CIFAR, ImageNet and Caltech101 datasets have been chosen for experiment. The CFNN model have fine-tuned the AlexNet, ZFNet, GoogleNet, VGGNet16, ResNet50 to classify the images. Here 2, 5 and 7 epochs have been taken for training the models. In the fuzzy layer IT2FCM clustering has been used to cluster the set of data in several times with the different number of clusters. When the fuzzy partition coefficient is maximized, that number of clusters has chosen for experiment. Adam is the stochastic optimization method which is used to classifier training (to tune weight) for the fully connected layer.

| Model    | Fine tuning epochs | Dog Vs Cat | Lion Vs Tiger | Horse Vs Donkey |
|----------|-------------------|-----------|--------------|-----------------|
|          |                   | Regular   | FCNN Model 1 | FCNN Model 2    |
|          |                   |           | IT2FCNN      | FCNN Model 1    |
|          |                   |           |              | FCNN Model 2    |
|          |                   |           |              | IT2FCNN         |
|          |                   |           |              |                 |
| AlexNet  | 3                  | 40        | 54           | 58              |
|          | 5                  | 51        | 60           | 61              |
|          | 7                  | 54        | 65           | 68              |
| ZFNet    | 3                  | 41        | 53           | 56              |
|          | 5                  | 53        | 62           | 61              |
|          | 7                  | 54        | 64           | 67              |
| GoogleNet| 3                  | 42        | 56           | 57              |
|          | 5                  | 54        | 61           | 64              |
|          | 7                  | 57        | 68           | 70              |
| VGGNet16 | 3                  | 44        | 58           | 57              |
|          | 5                  | 53        | 62           | 64              |
|          | 7                  | 55        | 70           | 72              |
| ResNet50 | 3                  | 43        | 56           | 56              |
|          | 5                  | 54        | 61           | 62              |
|          | 7                  | 56        | 69           | 71              |

Table 1. Performance comparative analysis with various fine tuning epochs (3, 5, and 7 respectively)
Figure 16. Performance comparative analysis for Dog Vs Cat with fine tuning epochs 3, 5 and 7 - (a), (b), (c) respectively

(a)  
(b)  
(c)

Figure 17. Performance comparative analysis for Lion Vs Tiger with fine tuning epochs 3, 5 and 7 - (a), (b) & (c) respectively

(a)  
(b)  
(c)

Figure 18. Performance comparative analysis for Horse Vs Doney with fine tuning epochs 3, 5 and 7 - (a), (b) & (c) respectively

(a)  
(b)  
(c)

Table 1 shows the model performance compared with existing CNN and fuzzy based CNN architectures. The Figure [16-18] clearly shows that the fuzzy based CNN architecture increase the performance accuracy compared with traditional CNN architecture. The experiment clearly shows that when including the fuzzy layer in the CNN which gives the high quality of accuracy compared to corresponding regular CNN.

6.4 The comparison of RMSE, MSE and MAE

The Percentage of Error (%Error), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) are the performance criteria for image classification; the corresponding calculation method is defined as follows:

\[
\%\text{Err} = \frac{|y_i - \hat{y}_i|}{y_i} \times 100 \quad (23)
\]

\[
\text{MSE} = \frac{1}{N_\text{i}} \sum \limits_{i=1}^{N_\text{i}} (x_i - \hat{y}_i)^2 \quad (24)
\]

\[
\text{RMSE} = \sqrt{\frac{1}{N_\text{i}} \sum \limits_{i=1}^{N_\text{i}} (x_i - \hat{y}_i)^2} \quad (25)
\]

\[
\text{MAPE} = \frac{100}{N} \sum \limits_{i=1}^{N} \left| \frac{x_i - \hat{x}_i}{x_i} \right| \quad (26)
\]

Table 2 shows the comparison of FCNN models with IT2FCNN based MSE and RMSE. The results shows that proposed to produce the good result.

| Models   | MSE    | RMSE   | MAPE |
|----------|--------|--------|------|
| AlexNet  | FCNN Model 1 .00245 .052 4.4 |
|          | FCNN Model 2 .00183 .043 3.2 |
|          | IT2FCNN .00123 .035 2.4 |
|          | FCNN Model 1 .00254 .054 4.2 |
| ZFNet    | FCNN Model 2 .00143 .045 3.1 |
|          | IT2FCNN .00134 .037 2.1 |
|          | FCNN Model 1 .00249 .053 4.2 |
| GoogleNet| FCNN Model 2 .00197 .041 3.2 |
|          | IT2FCNN .00123 .036 2.0 |
Table 2. Comparison of FCNN models with IT2FCNN based on MSE and RMSE

| Model         | MSE  | RMSE | IT2FCNN   |
|---------------|------|------|----------|
| VGGNet16 FCNN Model 1 | .00244 | .056 | 4.2      |
| VGGNet16 FCNN Model 2 | .00158 | .047 | 3.0      |
| IT2FCNN       | .00198 | .039 | 2.2      |
| FCNN Model 1  | .00268 | .058 | 4.1      |
| FCNN Model 2  | .00139 | .046 | 3.1      |
| IT2FCNN       | .00132 | .037 | 2.1      |

CONCLUSION

The experimental results say that fuzzy-neural networks represent a powerful and suitable alternative to conventional classification methods. The combination of fuzzy logic with neural networks applications are more efficient decision marking systems. In the proposed method CNN is used to extract the features and integrating the interval type-2 fuzzy to classify the images, which increase the accuracy in the experiment. Moreover, our experiment results show that it is possible to enhance testing accuracy by observing the distribution of pixels in feature maps and adjusting the membership function. This method gives the better solution and it has more advantages than the other existing methods. Although the results are more optimistic, image classification based on Interval type-2 fuzzy logic still requires more future research.

REFERENCES

A. Howard, "Some improvements on deep convolutional neural network based image classification", ICLR, 2014.

Arun D. Kulkarni and Charles D. Cavanaugh, Fuzzy Neural Network Models for Classification, Applied Intelligence 12, 207–215 ,2000,

Aykut Beke and Tufan Kumbasar, Interval Type-2 Fuzzy Systems as Deep Neural Network Activation Functions, Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019),2019

Borisov V. V, Kruglov V. V., Fedulov A. S., Fuzzy models and nets. Moscow, Goryachaya liniya – Telekom Publ., 2012. 284 p. (in Russian)

Cheng-Jan Lin , Chun-Hui Lin , Chi-Chia Sun and Shyh-Hau Wang ,Evolutionary-Fuzzy-Integral-Based Convolutional Neural Networks for Facial Image Classification

Cheul Hwang and Frank Chung-Hoon Rhee, Uncertain Fuzzy Clustering: Interval Type-2 Fuzzy Approach to C-Means, IEEE Transactions on Fuzzy Systems,Vol.15,No.1, February 2007

D.T. Mane; U.V. Kulkarni, A novel fuzzy convolutional neural network for recognition of handwritten Marathi numerals; International Journal of High Performance Computing and Networking; Vol.15 No.3-4,2019

E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research", IEEE Computational Intelligence Magazine, vol. 9, no. 2, pp. 48-57, 2014.

Fuler R. Neural fuzzy systems. – Abo: Publishing House Abo Akademi University, 1995.

Jerry M. Mendel,Robert I. John, and Feilong Liu, Interval Type-2 Fuzzy Logic Systems Made Simple, IEEE Transaction on Fuzzy System, Vol. 14, NO. 6, December 2006

Jiayan An , (Member, Ieee), Li Fu , Meng Hu , Weihong Chen, And Jiawei Zhan, A Novel Fuzzy-Based Convolutional Neural Network Method to Traffic Flow Prediction With Uncertain traffic Accident Information, [IEEE Access], Vol 7,2019.

K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR, 2015.

Karnik, N. N. and Mendel, J. M. Applications of type-2 fuzzy logic systems to forecasting of time-series, Information Sciences, Vol. 120, pp. 89-111, 1999.

Karnik, N. N., Mendel, J. M. and Liang Q. Type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst., Vol. 7, pp. 643–658, 1999.
Wei Wang, Jianxun Gang, Application of Convolutional Neural Network in Natural Language Processing, International Conference on Information Systems and Computer Aided Education (ICISCAE), 2018

Wei Wang, Yujing Yang, Xin Wang, Weizheng Wang, Ji Li, “Development of convolutional neural network and its application in image classification: a survey,” Opt. Eng. 58(4), 2019

Xiao Yan Wu, A hand gesture recognition algorithm based on DC-CNN, Multimedia Tools and Applications volume 79, pages9193–9205, 2020

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard, et al., Backpropagation applied to handwritten zip code recognition, Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.

Y. LeCun, L. Bottou and J. HuangFu, Learning methods for generic object recognition with invariance to pose and lighting, CVPR, 2004.

Zadeh, L. A. “Fuzzy sets”, Information and Control, Vol. 8, pp. 338–353, 1965.

Zadeh, L. A. “The concept of a linguistic variable and its application to approximate reasoning”, Part 1, Inform. Sci., Vol. 8, pp. 199–249, 1975.

Zeng, J. and Liu, Z.Q. Type-2 fuzzy sets for handling uncertainty in pattern recognition, In Proceeding of FUZZ-IEEE, pp. 6597-6602, Vancouver, Canada, 2006.

Zhengwei Huang, Min Dong, qirong Mao and Yongzhao Zhan, Speech Recognition using CNN, IEEE/ACM Transactions on Audio Speech and Language Processing, vol. 22, no. 10, pp. 1533-1545, 2014

Zixian Zeng, Qingge Gong, Jun Zhang, CNN Model Design of Gesture Recognition Based on Tensorflow Framework, 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019