International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-3S, February 2019

Uncertainty-Handling of Multi-Object Categorization in Scenes using Fuzzy Swarm Intelligence-based Deep-Belief Network

S. Kumaravel, S. Veni

Abstract--- In current digital era, most people are using multimedia not only for entertainment, but also for various commercial usages. As the demand for multimedia increased vigorously, the researchers have started focusing on machine learning systems, in image processing. Traditional models outfit the process of image recognition, classification, etc., due to their excessive size and computation time. To overcome the aforementioned issues in conventional classification models, deep learning greatly influences the researchers in the field of image processing, with vast amount of qualities. This paper aims at developing an optimized fuzzy swarm intelligence-based deep-belief network (FSIDBN), which handles the issue of multiple object categorization in scenes. The uncertainty in handling is one of the prominent issues in image processing, when there is vagueness in determining, a greater number of objects. At the same time, conventional deep-belief network itself holds some of the disadvantages, when it assigns weights randomly. The existing issues are overcome, by representing the input images in fuzzy domain using degree of membership, to which each object belongs, and assigning optimal weights on each layer of stacked Restricted Boltzmann Machines using Fuzzy Swarm intelligence. The simulation result proves that, FSIDBN has achieved a higher degree of accuracy in the categorization of multiple distinct objects, compared to FDBN and DBN models.

Keywords--- Multi-Object Categorization, Fuzzy Deep Belief Network, Classification, Membership and Restricted Boltzmann Machine.

I. INTRODUCTION

In the field of object categorization, conventional approaches mainly used appearance of features, for recognizing classes of objects in real world images. This type of recognition considers color, edge, responses and shapes to identify variance in objects, with a limited constraint. In view of clutter, the presence of variation and noise in illumination and pose, may disambiguate the appearance of objects by, the coherent composition of objects which was commonly exhibited by real world scenes. Image classification has become one of the key pilot use-cases, for demonstrating machine learning. The Regular Neural Networks (NN) is not capable of dealing with images. If each pixel is connected to one neuron, there will be thousands of neurons, which will be computationally expensive. Consequently, the researchers found that, adding more layers to a neural network vastly improved its performance. Such neural networks with several hidden layers are common today, including in image classification.

Deep learning handles images in different ways, but still it follows the general concept of NN. Nowadays, approaches related to deep learning have been extensively studied for image classification and image processing tasks such as perceiving the underlying knowledge from images. Deep neural networks (DNN) utilize their deep layer-wise design, to emulate latent features from data, and thus pick up the possibility to appropriately classify patterns.

Deep Belief Network is a kind of deep neural network, which comprises multiple layers of graphical model, having both directed and undirected edges. It comprises multiple layers of hidden units, where every layer is connected to every other, but units are not. The two significant caveats of Deep Belief Networks are:

- Belief Network
- Restricted Boltzmann machine

Belief Network

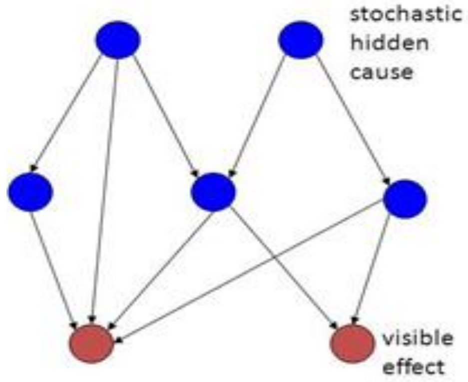

Figure 1: General Structure of Belief Network

It consists of stochastic binary unit layers, where each connected layer has some weight. The binary units have a state value of 0 or 1. With the aid of bias and weights of previous layer units, the probability of becoming 1 is identified. In general, belief net is considered as directed acyclic graph which comprises stochastic variables. It helps in solving two issues, namely inferring states of the unobserved variables *and* adjusting interaction among variables, to enhance the network, to produce more likely output data.

Manuscript published on 28 February 2019.
* Correspondence Author (s)
S. Kumaravel, Research Scholar, PhD (PT) in Computer Science, Karpagam Academy of Higher Education, Coimbatore.
Dr.S. Veni, Prof., & HOD, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore.

The conventional **Restricted Boltzmann machine** consists of:

- A single layer of **visible units**
- A single layer of **hidden units** and
- A bias unit

Every visible unit is linked to every hidden unit. Every visible unit and every hidden unit are linked to the bias unit. In RBM, no visible-visible units and hidden-hidden units are connected.

Every layer of a Deep Belief (Multi-layer) network is a Restricted Boltzmann Machine (RBM). Such RBMs are heaped, one upon another, for building the DBN [13].

The task of Image classification is to categorize images into, one of many already defined classes, which is a vital issue in computer vision. Other tasks like segmentation, localization and detection [1], are dependent on this vital issue. While for humans, this kind of task can be treated as secondary, this is a difficult task in the field of automated system. Some of the difficulties faced by automated systems consist of viewpoint-dependent object variability and the high in-class variability of consuming several object types

Currently, deep learning paradigm has established its ability to extract features [5] and performing transformation. In addition, it **has** overcome challenges in classification and pattern analysis [8,9,11], by utilizing nonlinear information processing, in each of its layers.

II. RELATED WORK

Rohit Patiyal *et al.* [2] proposed a model for Acoustic Scene Classification, using deep neural networks (DNN). In this work, various feature extraction methods, along with, several classifications algorithms were implemented. From the simulation result, it was inferred that, DNN performed better than other methods, with same feature set. This consequently, led to the conclusion that, using MFCC features with DNN, produced better results.

Robert *et al.* [3] in their work, analysed whether, aerial images belonged to residential or nonresidential domain, using deep learning model. Usage of this method, considerably reduced man power and eradicated the assumptions.

Zou *et al.* [4] developed a feature reconstruction problem by proposing deep learning method using Deep Belief Network (DBN). In their work, when the reconstruction error exceeded the predefined threshold values, the features learned by DBN were eliminated.

Yanfei *et al.* [5] transformed the random scaling of images to the specified scale, and the reconstructed image was fed as input, to train Convolution Neural Network (CNN); the CNN model, thus, was able to extract features, by robustly handling the variation of scale issue; also, its performance was enhanced through a multi-perspective fusion used for scene classification.

Roy *et al.* [6] aimed at investigating DNN-based, better noisy image classification system. At first, the autoencoder (AE)-based denoising techniques were considered, to reconstruct native image from the input noisy image. Then, convolutional neural network (CNN) was employed, to classify the reconstructed image, as CNN was a prominent DNN method with the ability to preserve better representation of the internal structure of the image data.

Singh *et al.* [7] proposed an efficient classification model for multi-class object images, subject to Gaussian noise. They applied wavelet transform-based image denoising techniques by employing the Neigh Shrink thresholding over the wavelet coefficients, to eliminate wavelet coefficients causing noise in the image, and picking up only useful ones.

Ajeet Ram *et al.* [8] in their work, revealed the function of deep learning techniques in the field of object detection, using CNN. They also highlight, the services available for object detection, using deep learning frameworks.

Ya-Fang Shih *et al.* [9] reported three main problems, while integrating part-based signs in CNN, for recognition of objects. The first problem was that a majority of part-based models depended on some predefined object parts. But, recognition of optimal object parts varied from category to category. The second problem was that getting part-level annotation as training data required The third problem was that designing spatial relationship among object parts in CNN, regularly encompassed more labour. an in-depth search of part templates, over multiple network streams.

Shusen Zhoua *et al.* [10] introduced fuzzy deep belief network (FDBN), for classification of sentiment, using two step semi-supervised learning strategy. Using the obtained training dataset, conventional DBN was trained using semi-supervised learning. After learning the deep architecture, fuzzy membership function for each class was designed. It mapped each review of trained DBN into the out space of DBN; the scattering of all training dataset was considered as prior knowledge and was coded by a series of fuzzy membership functions. The empirical validation on five different sentiment-based classification was illustrated with the efficacy of FDBN and AFD methods.

Ping Yang *et al.* [18] proposed an evolutionary fuzzy deep belief network with incremental rules (EFDBNI) for sentimental classification. In their work, they evaluated this scheme using empirical dataset, and from the simulation result, it was observed that EFDBNI produced significant improvement over existing approaches.

Yue Deng *et al.* [12] introduced the concept of fuzzy deep learning to overcome the issues in fixed representation. In their work, they used bulk of fuzzy system as a hierarchical DNN. After gaining knowledge from these two perspectives, a fused model was designed for classification.

III. METHODOLOGY

The functionality of RBF architecture, is first briefly explained (equations (1) to (7)), and then the Training of DBN is explained.

3.1. Restricted Boltzmann Machines (RBM) [13] is a double layer bipartite graph . It consists of visible units $\{0,1\}\ D\ v \in$ and hidden units $\{0,1\}\ P\ h \in$, where every visible unit is connected to, each of the hidden units by a weight matrix, as shown in **Fig 2 (a) and (b),** while the units do not connect with each other, within the same layer.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

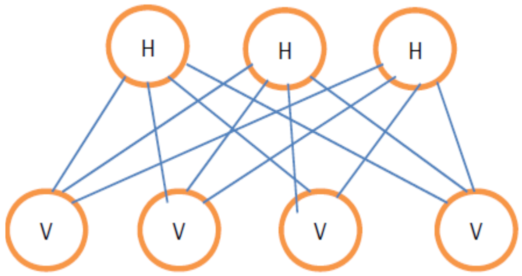

Figure 2(a): Simple RBM layers

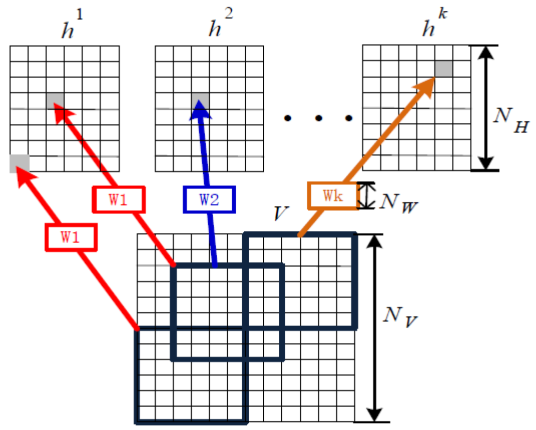

Figure 2(b): Pixel representation of layers

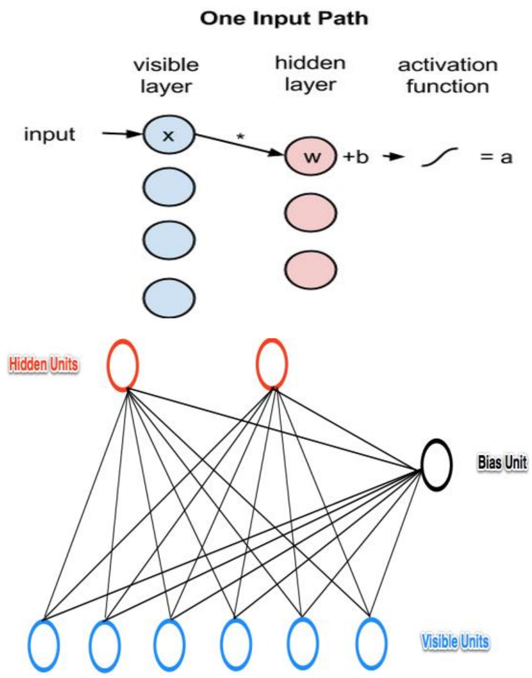

Figure 3: Function of one input path of RBM

Figure 3 depicts, how a value of pixel **x**, passes into the network, with two layers. In the first node of the hidden layer, the value of **x** is multiplied by a weight **w,** assigned to the link between visible layer and hidden layer. Then, the bias value **b** is added to it. The resultant value is passed into the activation function, which generates the output to the node. It is denoted in the equation (1)

activation f ((weight w * input x) + bias b) = output a (1)

with several inputs combined at a single hidden unit, whose activation energy is $a_i = \sum_j w_{ij}$

The typical category of RBM has value of binary for both hidden and visible units, and comprises a matrix of weights W= $w_{i,j}$ (size $m \times n$) related with the link among hidden unit h_j and visible unit v_i, bias weights a_i for the visible units and b_j for the hidden units. Specified these, the energy of a configuration (v,h) is well-defined as

$$E(v,h,\theta) = \sum_{i,j} v_i w_j h_j - \sum_i a_i v_j - \sum_i b_j h_j \quad (2)$$

where i and j are the sequence numbers of the visible and hidden units; the variable θ is a set of model parameters θ = {w, b_j, a_i}, where b_j and a_i denotes the hidden and visible unit biases, respectively and w denotes the connection between the pairs of visible and hidden units.

The probability of the visible variables in an RBM with the parameter set θ in accordance with a joint energy E(v,h,θ) is defined as:

$$p(v,h) = \frac{1}{z(\theta)} \sum_h \exp\left(-E(v,h,\theta)\right) \quad (3)$$

where I and j denote the vectors of the visible and hidden variables, respectively; exp(x) represents the power of a constant e ; and z(θ) is a normalization constant defined by

$$z(\theta) = \sum_v \sum_h \exp\left(-E(v,h,\theta)\right) \quad (4)$$

where E(v,h, θ) is a free energy function of the RBM

Inferring the distribution of these hidden variables is easy, since there is no connection between the hidden variables.

$$P(h_j|v) = \sigma\left(c_j + \sum_i v_i w_{ij}\right) \quad (5)$$

Where, $\sigma(c) = 1/(1+e^{-x})$ is a sigmoid function

The extreme possibility learning procedure can train the network, by solely switching among, each hidden unit and each visible unit, in parallel:

$$\frac{\partial \log P(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_\infty \quad (6)$$

To increase the learning capability of RBM, contrastive divergence (CD) algorithm is applied, where, updation of all the hidden units in parallel (beginning with visible units) is done first, and then, visible units reconstructed from the hidden units, followed by, updating of the hidden units again. The rule for learning is:

$$\partial w_{ij} = \langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_1 \quad (7)$$

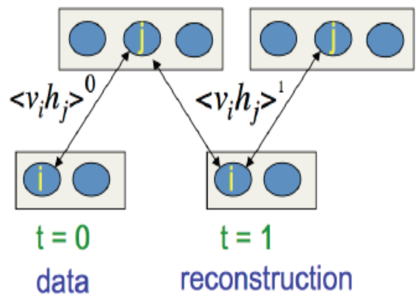

Figure 4: reconstruction of DBN

The deep belief network consists of a stack of RBM, which is represented in the form of multi-layer belief networks. Each of the layers is connected, to build DBN [13].

3.2. Training a DBN

The initial process of DBN during training phase is to learn about the visible unit features, using Contrastive Divergence (CD) algorithm [14].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Next, in the second hidden layer, activation of previously trained features as visible units is considered and the characteristics of the features have to be learned. *Lastly*, the entire DBN is trained, when the learning process reaches the final hidden layer.

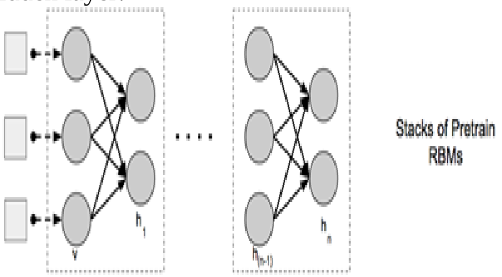

Figure 5: stacked pretrained RBM

The greedy layer wise pre-training, followed by fine-tuning, involves two stages during the training of DBN. During pre-training stage, the first layer RBM is trained using the given input x, then the hidden units' values are treated as training data for the next layer RBM and so on. The same process is repeated, layer by layer, till it reaches the final layers, which results in deep model. This kind of learning process produces an approximate inference, and it requires only one bottom-up pass to get the values of the top-level variables of hidden layer.

3.3. Uncertainty handling Methodology in FSIDBN

In this work, a fuzzy swarm intelligent-based deep belief network (FSIDBN) is established, for the process of multiple object categorization in scenes. To establish FSIDBN, the methodology consists of 3 different processes:

1. Conversion of input images to the fuzzy representation
2. Applying fuzzy restricted Boltzmann machine as pretrained network in deep belief network
3. Optimal weights assigned between the hidden nodes, using fuzzy particle swarm optimization.

The overall framework of the proposed architecture, is shown in **Figure 6.**

Image Representation in Fuzzy Domain

Step 1: Data set collected from PASCAL VOC Challenge 2007 Image Dataset [16] and for training individual objects 101_object Categories [15].

Step 2: Apply normalization process on the crisp value of images

$$\text{Normalized fuzzy range} = \left[\frac{x - minimum\_value}{maximum\_value - minimum\_value}\right] \quad (8)$$

Step 3: Assign membership function within the interval value [0, 1] for their corresponding linguistic term such as {L, M, H}

$$\mu(px_i) = \left\{\frac{px_i - \min(P)}{\max(P) - \min(P)}\right. \quad (9)$$

Where $\mu(P_i)$ is the fuzzy value of a single pixel px_i, P is a set of pixel value of input image**, min** is the minimum intensity value of the whole image pixels, **max** is the maximum intensity value of the image. Here, the uncertainty is greatly handled by, converting the image pixel into fuzzy values, for more accurate categorization of objects, by denoting them in terms of degree of membership

$$\mu'(P_{i,j}) = \begin{cases} 5 * \mu(i,j)^{\wedge}3 & \mu_{i,j} \geq 0 \&\& \leq 0.5 \\ 1 - 5 * (1 - \mu(i,j)^3) & \mu_{i,j} \geq 0.5 \&\& \leq 1 \\ 0 & otherwise \end{cases} \quad (10)$$

The given input image is enhanced by applying fuzzy rule as mentioned in the equation(10) and the entire image P is resultant as $\mu'(P)$.

Stacked RBMs, to form the DBN (refer to Figure 6).

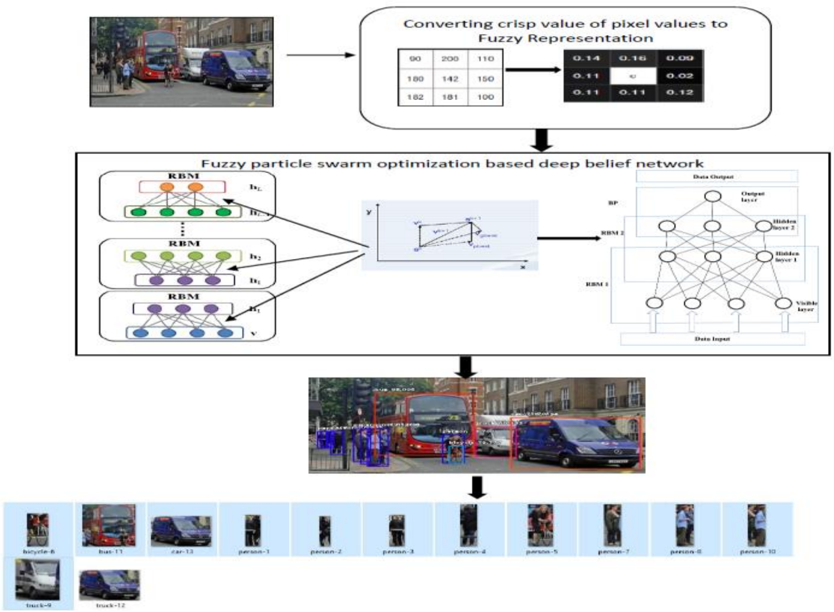

Figure 6: Uncertainty handling Framework in FSIDBN

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Fuzzy-PSO Algorithm-based weight assignment

Fuzzy particle swarm optimization (FPSO) adapts the theory of fuzzy set to handle vagueness, in the determination of optimal swarm, based on the problem space. Pang et al [17] introduced FPSO which used relation of fuzzy among variables. It was used for redefining particles velocity and position. In this work, the position of the particle was utilized, to choose the optimal weight for the hidden units, during the reassignment of weights between visible and hidden layers, to decrease the mean square error of observed and expected output of the fuzzy based deep belief network, in multi-object categorization. The fuzzy relation for set of objects in an image IM={ob_1, ob_2, ob_3… ob_n}, where ob_1, ob_2, ob_3… ob_n is the set of objects present in an input image. The objects with same type are clustered, by defining *cluster centers* for each category of objects as, FZ={fz_1,fz_2,….fz_n} which can be denoted as

$$IM=\begin{bmatrix}\mu_{11} & \cdots & \mu_{1c}\\ \vdots & \ddots & \vdots\\ \mu_{n1} & \cdots & \mu_{nc}\end{bmatrix} \qquad (11)$$

Here, μ_{ij} is the membership function of the ith object with the jth cluster object with limitations; **n** is the number of objects; **c** is the number of clusters or categories

$$\mu_{ij} \in [0,1] \qquad \forall_i = 1,2 \ldots n\,, \quad \forall_j = 1,2 \ldots c \qquad (12)$$

$$\sum_{j=1}^{c}\mu_{ij} = 1 \qquad \forall_i = 1,2 \ldots n \qquad (13)$$

In Fuzzy C-Means algorithm, the fuzzy matrix identifies, the position of individual particle. The velocity of a piece of particle is measured using a matrix, where **n** denotes the Rows and **c** denotes the columns, and the elements lie in range (-1,1).

The equations (14) and (15) used for updating the positions and velocities of the particles based on the matrix.

$$Vel\,(k+1) = wt * Vel(t) + const_1\, rnd_1 * (pb(k) - pos(k) + \qquad (14)$$

$$pos(K+1) = pos(K) + Vel\,(k+1) \qquad (15)$$

where, **vel** and **pos** are *velocity* and *position* of the particle correspondingly, **wt** denotes weight of *inertia*, **$const_1$** and **$cosnt_2$** are *coefficients of acceleration* which are positive constants, used to control the **pb** and **gb** during process of search, **rnd_1** and **rnd_2** are *random values* in the interval [0, 1]. While updating, the velocity of each particle involved, during identification of objects in a given scene, two best positions are considered for searching process, namely, **personal best position (pb)** which is the best position, a specific particle has visited so far and **global best position(gb)** which is the best position the swarm has visited, since the startup time of search process.

After the position matrix is updated, there is a possibility of constraint violation as given in equations (12 & 13). To Overcome this issue, normalization is applied on the position matrix by first considering a set of negative elements in the matrix and converting it to zero value. If a whole row is converted to zero, then reevaluation is performed, by assigning random values, whose range lies between [0-1]. Thus, the position matrix endures the following transformation, as shown in equation (16)

$$Xnormal = \begin{bmatrix}\mu_{11}/\sum_{j=1}^{c}\mu_{1j} & \cdots & \mu_{1c}/\sum_{j=1}^{c}\mu_{1j}\\ \vdots & \ddots & \vdots\\ \mu_{n1}/\sum_{j=1}^{c}\mu_{nj} & \cdots & \mu_{nc}/\sum_{j=1}^{c}\mu_{nj}\end{bmatrix} \qquad (16)$$

This method uses the subsequent calculation as fitness function, for assessing the solutions

$$f(X) = \frac{K}{J_m} \qquad (17)$$

Where the numerator, *K*, is a constant and the denominator, J_m is the objective function in Fuzzy C-means clustering algorithm. If J_m, is smaller, then the clustering effect produces better result, and the individual fitness value *f (X) must be higher*. The process continues, till the iteration is reached or gb value has no improvements, within the iteration period. **Algorithm 1** shows Fuzzy PSO for clustering similar objects in an input scene and assign weights. Fuzzy C-Means algorithm is integrated with Fuzzy Swarm Intelligence algorithm, to form a fusion clustering algorithm called ***fused FPSO***, which retains the benefits of both algorithms (FCM and PSO), and is **given below:**

Algorithm 1: Fused FPSO

1. Initialize Fuzzy PSO parameters (size of population, number of iterations, position, n * c matrix and velocity of each particle).
2. Define Deep Belief Network topology based on it number of particles to be assigned on weights are initialized.
3. Each path of the particle relays to a DBN connection with associated threshold or weight. Now dimension of the particle can be expressed as follows
4. Swarm Particles: SPi = (spi1, spi2, ..., spD) i = 1,2,. . .,n
5. Dim = (IU*HU) +(HU*OU) +HU+OU
6. Where IU, HU and OU represent the number of units in the input layer, hidden layer and output layer of DBN respectively and Dim is Dimension.
7. Initialization of swarm particle parameters **pos, vel, pb** and **gb**.
8. Cluster centers for each particle, is given by : $Z_j \; \frac{\sum_{i=1}^{n}\mu_{ij}^{m}\, obj_i}{\sum_{i=1}^{n}\mu_{ij}^{m}}$

   Where μ_{ij} is membership value of i^{th} object which belongs to j^{th} category of a scene. The objects are denoted as Obj_i m > 1 is a controller which controls the fuzziness of the group.
9. Compute each particle's fitness value using equation (17).
10. If current fitness value is lesser than the Pb (fitness value), the current position is replaced to Pb.
11. The best particle weight chosen is stored for the concern hidden unit of DBN.
12. Update velocity and position of each swarm particle using eqns., (14) and (15)
13. If velocity or position of particle is beyond the boundary, the velocity or position should be reset randomly
14. If termination criteria is not met then go to step 8

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The training phase of the FSIDBN

After *enhancing* input images using fuzzy domain and assigning weights between visible and hidden layers, the proposed fuzzy deep belief network uses labeled images and unlabeled images by keying them into layers, one by one, starting from the layer h_0.

The architecture of fuzzy deep belief is constructed, based on layer by layer approach, traversing from the bottom to the top, whilst the parameter wt^i is trained by the calculated output in the i-1th hidden layer.

Algorithm: Procedure of FSIDBN

Input: image data: IM, object labels: OB^L

No. of training data TR; No. of test data TS;

No. of layers NL; No. of epochs EPQ;

No. of hidden layers in FDBN is $HL^1,..,HL^{NL}$

No. of units in every hidden layer $HU^1...HU^N$;

Wt = {wt_1 , . . . , wt_n}; biases a, b;

Learning rate $\wp$, momentum τ

Output: FSIDBN architecture with parameter space W_t

1. Unsupervised learning using greedy step wise algorithm

for p = 1; p ≤ NL− 1

do

for s = 1; s ≤ EPQ do

for r = 1; r ≤ TR + TS

do

Compute the non-linear positive and negative phase

$$positive\ phase : probability(h_t^k = 1|h^{k-1}) = sigm(a_t^k + \sum_s wt_{st}^k h_s^{k-1})$$

$$Negative\ phase : probability(h_s^{k-1} = 1|h^k) = sigm($$

t

End

End

End

2. DBN as supervised learning paradigm, using equation as follows

$arg_{wt} \min f(HL^n(IM^L), OB^L)$

3. Calculate the parameters

χ = maximum |dt(im^i)|, i=1,…TS+TR

ρ = ξ * χ,

Where ξ denoted degree of separation and its values is ≥ 2, χ is separate point in fuzzy membership domain value representation and ρ is distance

Update weights by calling fuzzy PSO algorithm 1

$$wt_{st}^k = FuzzyPSO\,(HU^{1..n}) = f(X)$$

Testing process

Unsupervised learning using greedy-layer-wise on test images

Using Fuzzy PSO refine proposed FDBN

4. Categorize the objects in scene IM based on trained FSIDBN using equation

$$\check{U} = \arg_{\check{U}} \max HL^{NL}(im)$$

The above algorithm has specified the complete procedure of the current work, which encompasses functions of each layer, starting from input layer to the output layer, with stacked pretrained fuzzy Restricted Boltzmann, *and* each of the weights assigned between the hidden layers is reassigned using fuzzy-PSO during refine process, to improve the optimal solution of categorization of multiple objects, in a given scene.

IV. RESULTS and DISCUSSIONS

The performance of the FSIDBN is simulated using Python 3.6. The dataset used for training process are collected from Caltech 101 [15]. For Multi-object categorization the scenes with multiple objects are collected from PASCAL VOC 2007[16] which offers consistent image data sets for object class recognition. The performance comparison of FSIDBN is done with two approaches, namely, Deep Belief Network and Fuzzy Deep Belief Network. To assess the performance, the metrics, Accuracy, Precision and Recall, are been used.

$$\textbf{Accuracy} = \frac{Total\ number\ of\ correcly\ categorized\ objects}{Total\ number\ of\ objects\ in\ a\ scene}$$

Precision of object categorization technique is determined by finding, the ratio of the number of correctly identified objects with number of identified objects.

$$\text{Precision} = \frac{Total\ number\ of\ correcly\ categorized\ objects}{Total\ number\ of\ identified\ objects\ in\ a\ scene}$$

Recall of an object categorization technique is calculated using, number of correctly identified objects with number of actual correctly recognized objects.

$$\text{Recall} = \frac{Total\ number\ of\ correcly\ categorized\ objects}{Total\ number\ of\ correct\ objects\ in\ a\ scene}$$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figure 7: Simulation result of multiple object categorization using FSIDBN

Figure 7 shows the output of FSIDBN for multiple object detection in sample image 1, which consists of many objects like bus, car, person, bicycle and motorcycle. The accuracy of recognizing each object is done precisely, by handling uncertainty in determining objects in real time images. It is achieved by integrating fuzzy-PSO with deep belief network.

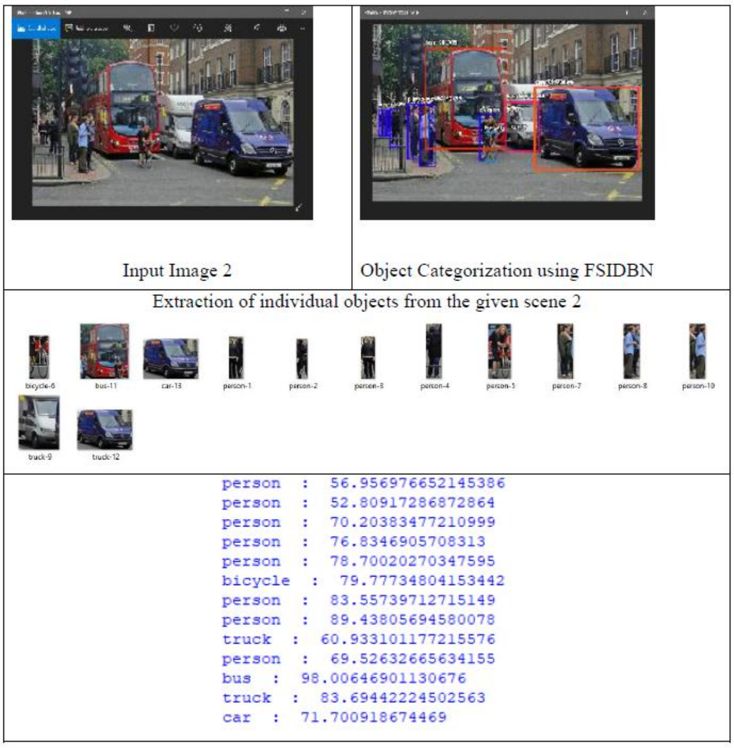

Figure 8: Simulation result of multiple object categorization using FSIDBN

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figure 8 shows the output of FSIDBN for multiple object detection in sample image 2, which comprises bus, car, person, bicycle and truck. The accuracy of FSIDBN is greatly improved, when compared to other techniques, by enhancing input image to fuzzy domain, to overcome vagueness in categorization and the representation of degree of membership on each object, of the selected scene.

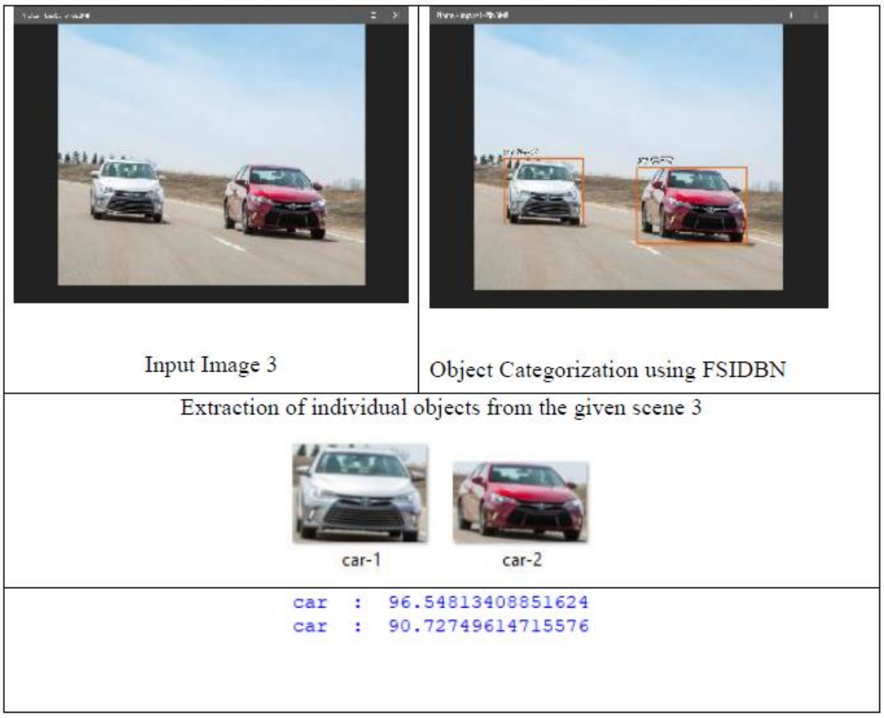

Figure 9: Simulation result of multiple object categorization using FSIDBN

Figure 9 shows the output of FSIDBN for multiple object detection in sample image 3, which consists of same category of two objects namely car. This work fine-tunes the categorization of objects, by overcoming issues in existing fuzzy deep belief network, with the ability of swarm intelligence.

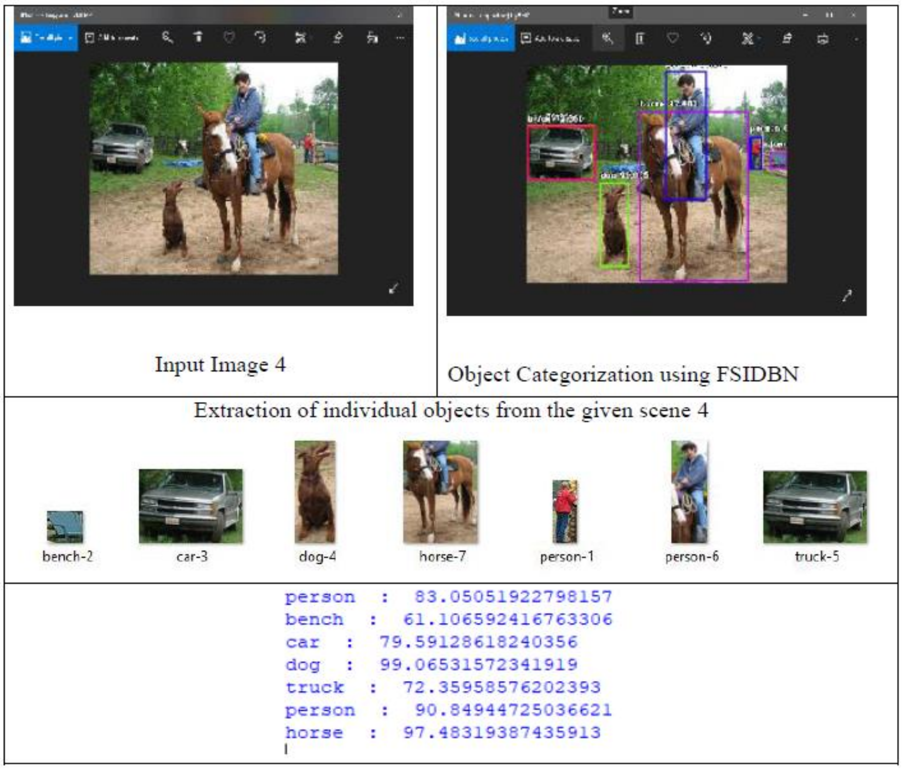

Figure 10: Simulation result of multiple object categorization using FSIDBN

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figure 10 shows the output of FSIDBN for multiple object detection in sample image 4, which consists of many objects like bench, dog, truck, car, person and horse. While the existing approaches fail to optimize the weights assigned to stacked fuzzy RBM, this work FSIDBN achieves optimization in categorizing multiple objects, by using swarm intelligence for assigning weights, in the stacked fuzzy RBM.

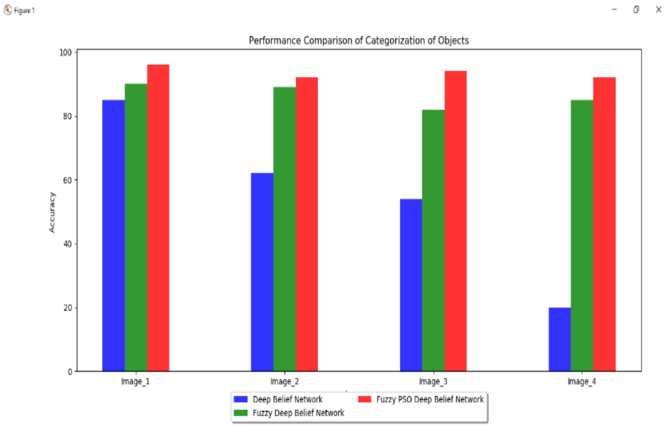

Figure 11: Performance comparison of three different methods based on Accuracy of multiple object categorization

Figure 11 depicts the performance comparison of conventional deep belief network, fuzzy deep belief network and swarm intelligent-based deep belief network, measured using accuracy of object categorization, in four different scenes. From the result it is observed that the proposed FSIDBN achieves better accuracy, because of its ability to handle vagueness, by representing the objects in terms of degree of membership towards the recognized objects.

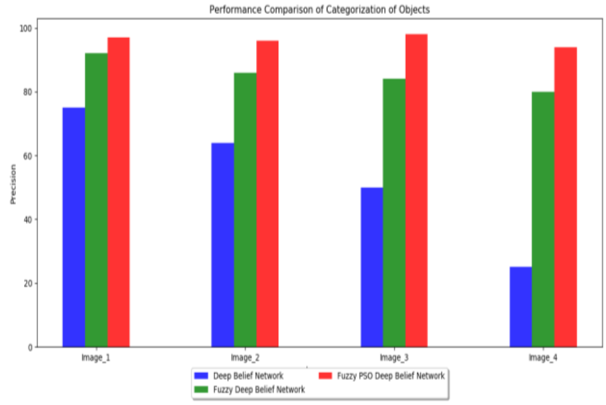

Figure 12: Performance comparison of three different methods based on precision of multiple object categorization

Figure 12 depicts the performance comparison of conventional deep belief network, fuzzy deep belief network and swarm intelligent based deep belief network measured using precision of object categorization, in four different scenes. From the result it is observed that the proposed FSIDBN gains the knowledge of precision using fuzzy swarm intelligence, which reassigns the weights of the hidden layers in an optimized way, to fine-tune the object categorization more precisely.

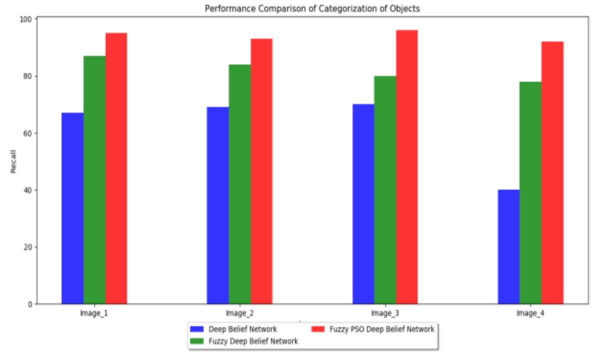

Figure 13: Performance comparison of three different methods based on recall of multiple object categorization

Figure 13: it is observed that the FSIDBN holds high recall value and it outperforms the existing two approaches namely DBN and FDBN, because they fail to handle the problem of uncertainty when there is a vagueness, in determining multiple objects in scene. For analysis, four different scenes are used for simulation.

V. CONCLUSION

This work has used Fuzzy Swarm intelligence for optimizing the performance of existing fuzzy deep belief network, in multi-object categorization of scenes. In real time object categorization process, the presence of vagueness and uncertainty is inevitable. To overcome this issue, this work has introduced fuzzy swarm intelligence, to update the assignment of weights between hidden layers of stacked fuzzy RBM, with its characteristics of global and local search strategy. The optimal weights are determined by two important factors namely, velocity and position of each particle in swarm. From the simulation results, it is evident that the FSIDBN greatly influences the process of multiple object categorization in deep learning system, in a more optimistic manner.

REFERENCES

1. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L., Large-scale video classification with convolutional neural networks. In Computer Vision and Pattern Recognition CVPR 2014, IEEE Conference on, pp 1725–1732, 2014.
2. Rohit Patiyal, Padmanabhan Rajan, Acoustic Scene Classification using Deep Learning, Detection and Classification of Acoustic Scenes and Events, 3 September '16, Budapest, Hungary, 2016.
3. Robert F. Chew, Safaa Amer, Kasey Jones, Jennifer Unangst, James Cajka, Justine Allpress and Mark Bruhn, Residential scene classification for gridded population sampling in developing countries using deep convolutional neural networks on satellite imagery, International Journal of Health Geographics, 17:12, 2018.
4. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning-based feature selection for remote sensing scene classification. IEEE Geosci. Remote. Sens. Lett. , 12, pp 2321–2325, 2015.
5. Yanfei Liu, Yanfei Zhong, Feng Fei, Qiqi Zhu and Qianqing Qin, Scene Classification Based on a Deep Random-Scale Stretched Convolutional Neural Network, Remote Sensing, pp1-13, 2018.
6. Roy, S. S., Ahmed, M., & Akhand, M. A. H., Noisy image classification using hybrid deep learning methods. Journal of Information and Communication Technology, 17 (2), 233–269, 2018.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

7. Singh, A. K., Shukla, V. P., Biradar, S. R., & Tiwari 1, S., Multiclass noisy image classification based on optimal threshold and neighboring window denoising. International Journal of Computer Engineering Science (IJCES), 4(3), pp 1-11, 2014.
8. Ajeet Ram Pathaka, Manjusha Pandeya, Siddharth Rautaraya, Application of Deep Learning for Object Detection, International Conference on Computational Intelligence and Data Science (ICCIDS '18), Procedia Computer Science 132, Science Direct, pp 1706–1717, 2018.
9. Ya-Fang Shih, Yang-Ming Yeh, Yen-Yu Lin, Ming-Fang Weng, Yi-Chang Lu and Yung-Yu Chuang, "Deep Co-Occurrence Feature Learning for Visual Object Recognition." In Proc. Conf. Computer Vision and Pattern Recognition, CVPR '17), pp 1-10, 1063-6919, 2017.
10. Shusen Zhoua, QingcaiChenb, XiaolongWangb. Fuzzy deep belief networks for semi-supervised sentiment classification., Neurocomputing, Vol 131, pp 312-322, 2014.
11. Kumaravel.S and Veni.S, Enhanced Object Categorization Model (EOCM) with Combined Efficiencies of SVM and NN, Int. J Pure and Applied Mathematics, IJPAM, ISSN 1314-3395, Issue 119, pp 18-21, 2018.
12. Yue Deng ; Zhiquan Ren ; Youyong Kong ; Feng Bao ; Qionghai Dai. A Hierarchical Fused Fuzzy Deep Neural Network for Data Classification., IEEE Transactions on Fuzzy Systems, Volume: 25 Issue: 4, pp 1006-1012, 2017.
13. G. E. Hinton, S. Osindero, Y. Teh, "A fast learning algorithm for deep belief nets.," Neural Comput., vol. 18, pp. 1527–1554, 2006.
14. A. Kae, K. Sohn, H. Lee, E.Learned-Miller, "Augmenting CRFs with Boltzmann Machine Shape Priors for Image Labeling.," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
15. http://www.vision.caltech.edu/Image_Datasets/Caltech101/
16. http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html
17. Pang, W., Wang, K., Zhou, C., & Dong, L. (2004). Fuzzy discrete particle swarm optimization for solving traveling salesman problem, Proceedings of the fourth international Conf. on Computer and Information Technology (CIT '04), 2004.
18. Ping Yang, Dan Wang, Xiao-Lin Du, Meng Wang. Evolutionary DBN for the Customers' Sentiment Classification with Incremental Rules. ICDM 2018: Advances in Data Mining. Applications and Theoretical Aspects, pp 119-134, 2018.