International Journal of Engineering Trends and Technology Volume 71 Issue 7, 117-123, July 2023
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I7P212

Original Article

Soft Computing Particle Swarm Optimization based Approach for Classification of Handwritten Characters using Deep Learning Model

B. Meena[1], K. Venkata Rao[2], Suresh Chittineni[3]

[1]Department of Computer Science & Systems Engineering, Andhra University College of Engineering (A), Andhra University and Associate Professor, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.
[2]Department of Computer Science & Systems Engineering, Andhra University College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India
[3]Department CSE, GITAM (Deemed to be University), Department of Computer Science and Engineering, Visakhapatnam, Andhra Pradesh, India

[1]Corresponding Author : meenabhagvathula100@gmail.com

Received: 03 January 2023 Revised: 18 June 2023 Accepted: 21 June 2023 Published: 21 July 2023

Abstract *- Applications of Deep Learning have proved successful in a number of fields, including Pattern Recognition, Automated Manufacturing and Translation. Deep Learning needs to have its parameters set up correctly in order to provide results of a high calibre. The number of neurons and hidden layers have a significant impact on the performance of a Deep Learning Network. However, manual parameter setup makes configuring important settings simpler for users. However, this technique is tedious. In the present work, it is shown that Particle Swarm Optimization (PSO) is used to optimize parameter values and configure the network. It is allowed for the fine-tuning of the Deep Learning Model by utilizing minimal computational resources. Several machine learning techniques are evaluated, including Decision Trees, K-Nearest Neighbour, and Support Vector Machines (SVM) for image classification. SURF descriptor image features are extracted during the feature extraction process. This work aims to clarify the correct classifier selection procedure and emphasize the importance of picking the appropriate classifier parameters using optimization methods. It is demonstrated in the basic experiment done in our work that PSO gives an excellent method for altering the appropriate Deep Learning algorithm with a number of hidden layers and the number of neurons in each layer compared with other machine learning classifiers like SVM, DT, and KNN. When compared to other classification algorithms, the findings indicate that the methodology has an overall precision of 98 - 100 % for image categorization. Our work demonstrates that PSO actually generates results with a great deal greater precision.*

Keywords *- Image Classification, Deep Learning, Parameter Optimization, PSO, Support Vector Machine.*

1. Introduction

The paper has six sections. Section 1 describes various methods of classification algorithms used. Section 2 gives the pieces of evidence from the literature. Section 3 is the methodology used in the proposed system. Section 4 describes the datasets used in the experiment. Section 5 shows the evaluation results. Section 6 consists of the conclusion part.

1.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO), which employs a straightforward approach to searching for the optimal solution in the problem space, is one of the bio-inspired algorithms. It does not depend on the gradient or any other differential form of the objective, in contrast to other optimization techniques, and only needs the objective function. There are not a lot of hyperparameters, either. A generalized model for neuronal biology and human cognition was developed using a method for processing information called a neural network. Instead of utilizing Back Propagation (BP), Particle Swarm Optimization (PSO) methods train the neural network to solve character recognition issues that are considered one of the significant applications in the classification field. The suggested approach is discovered by altering the NN weights to learn the NN and solve the handwritten character recognition problem. This is done by computing the fitness value, which is a threshold value. In order to determine whether the NN learning approach is more effective in resolving the letter recognition problem. A Neural Network (NN) learns to solve problems rather than having to be programmed to do so. A NN may gain the solution techniques during training. Particle Swarm Optimization (PSO), which is straightforward in concept, has few parameters to alter, and is simple to use, is one of the significant learning techniques. PSO is used in many different contexts. PSO is generally an excellent fit for all the application fields related to the other evolutionary approaches.

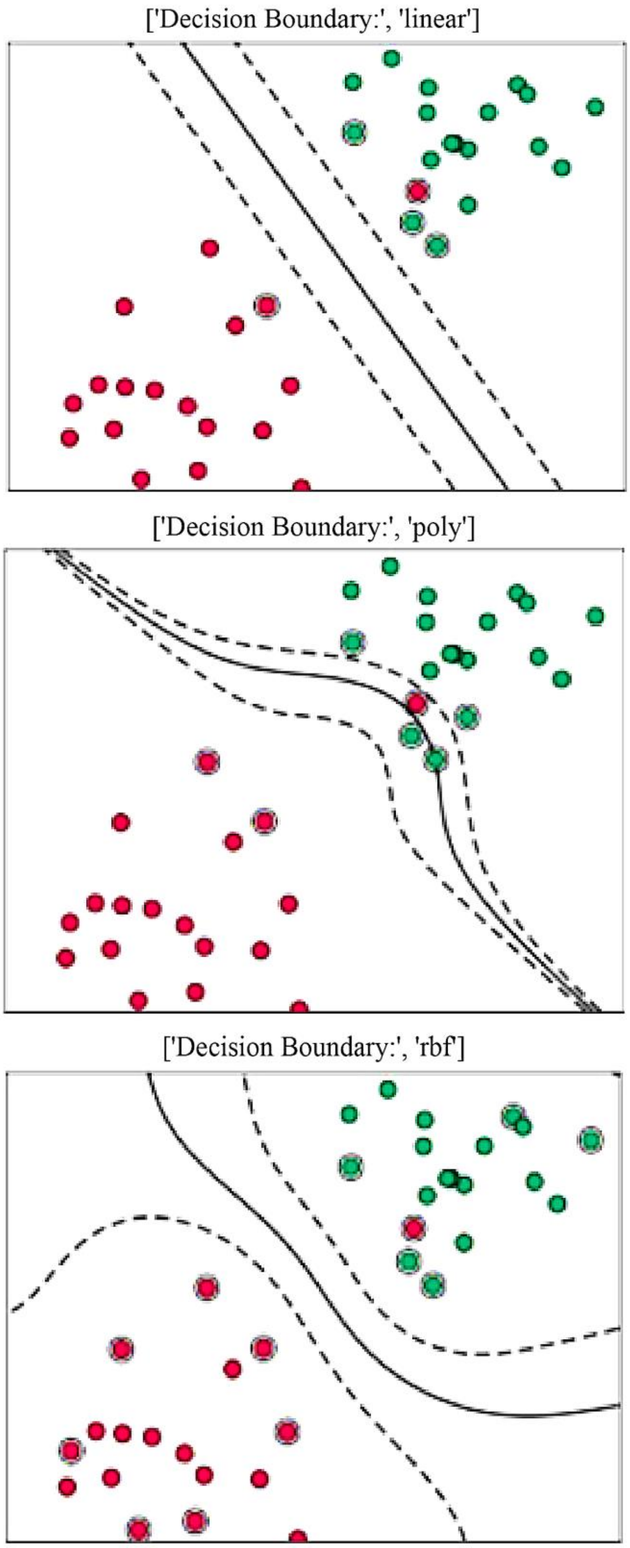

Fig. 1 Kinds of kernels

1.2. Support Vector Machine

Machine learning has a wide range of uses, and image classification is one of them. Classification or regression issues can be addressed using the "Support Vector Machine" (SVM) supervised machine learning method. SVM is also used in various problems like Face Recognition, Writing Recognition, Classifications for Images and Category Creation for Text and Hypertext.

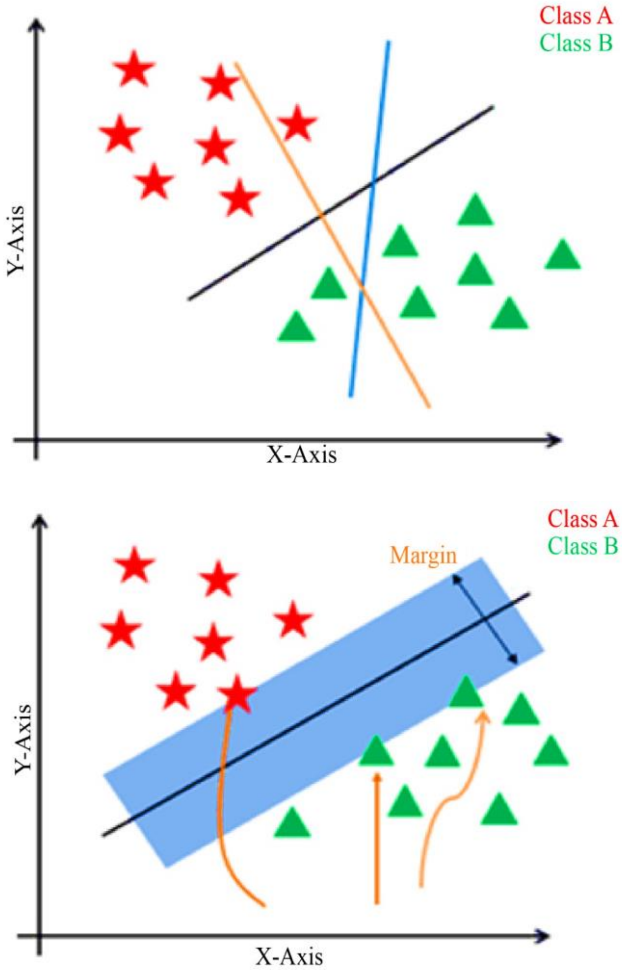

Fig. 2 Hyperplanes in SVM

In this system, SVM is used to categorize written character images. In the SVM technique, each data point is represented by a point in n-dimensional space with n features. The value of each attribute represents the value of a particular coordinate. After that, classification is performed by locating the hyper-plane that successfully separates the classes. The two important key values in SVM are Gamma and C. where Gamma - quantifies the extent to which a single training example's influence can lead to biased results. C - reduces the cost of errors in computations, small c — lowers the misclassification low, and Large C — Highers the misclassification; kernel: The kernel of an SVM algorithm is a collection of mathematical operations. There are three different kinds of kernels: linear, RBF, and polynomial, as shown in the figure.

Support vectors are the data points nearest to the hyperplane. By computing margins, these points will better define the dividing line. These ideas are more crucial to the classifier's design. A decision plane called a hyperplane distinguishes between a collection of objects with various class memberships. The distance between the two lines on the nearest class points is known as a margin. This is calculated as the angle between the line and the nearest points or support vectors. A wider gap between the classes is seen as a good gap; a smaller gap is regarded as a bad gap.

The main goal is to separate the provided dataset as effectively as feasible. The margin is the separation between the two nearest points. The goal is to choose a hyperplane in the given dataset with the largest margin between support vectors. SVM performs the different stages to get the maximum marginal hyperplane, such as creating hyperplanes and choosing appropriate hyperplanes.

1.3. Tree Classifier

A tree classifier is one the best classifier to find out a class for a given dataset. It works from the Root node by comparing the values of the root with the real dataset value and then switching to the next. The algorithm continues by cross-checking the value with sub-nodes and immediate notes. This process is repeated until the algorithm reaches the end leaf nodes. This process is simplified and has a smaller number of steps.

1.4. K-Nearest Neighbours

The K-Nearest-Neighbours (KNN) nonparametric classification method does not make any presumptions about the simple dataset. According to a predefined distance metric, the k-NN algorithm involves voting among a datapoint's k nearest neighbours. The distance metric is chosen while keeping in mind the application and the nature of the problem. It can be defined especially for the intended application or chosen from any well-known metrics, including Euclidean, Manhattan, Cosine, and Mahala Nobis. Here, the Euclidean distance metric is defined as follows.

$$D^{nxm}\,(x^n, y^m)\; D^{nxm}\,(x^n, y^m) = \left[\sum_{i=0}^{imgh*imgw-1} (x_i^n - y_i^m) * (x_i^n - y_i^m)\right] \quad (1)$$

Where (x^n, y^m) are trainset and test set with n and m sizes, respectively. The kNN classifier is easy and effective to classify, but it affects lower performance and the proper k value to be chosen.

Wang's researcher gave a new method to choose the proper k value without effects; it was recommended using probability, but the complexity is $O(n^2)$. This approach worked correctly and made k less dependent due to its time complexity $O(n^2)$. Hence it cannot be used in dynamic conditions for huge repositories due to its performance.

2. Related Work

Kennedy J. and Eberhart R. developed the idea of nonlinear function optimization through the use of particle swarm optimization in 1998. The development of numerous paradigms was in detail, and one of the paradigms was put into practice [Kennedy, J., and Spears, W. M.. (1998)].

In 1999, Eberhart R.C. and Hu X. developed a new technique for analyzing human tremors using PSO, which is utilized to generate an NN that separates tremor-affected subjects from normal subjects. [Kennedy, J., and Spears, W. M.. (1998)].

Shi Y. conducted a survey of PSO research and development in 2004. He divided it into five categories: applications, hybrid PSO algorithms, topology, parameters, and algorithms. There are undoubtedly other PSO research projects that were excluded due to a lack of space. [Kennedy, J., and Spears, W. M.. (1998,1998a).].

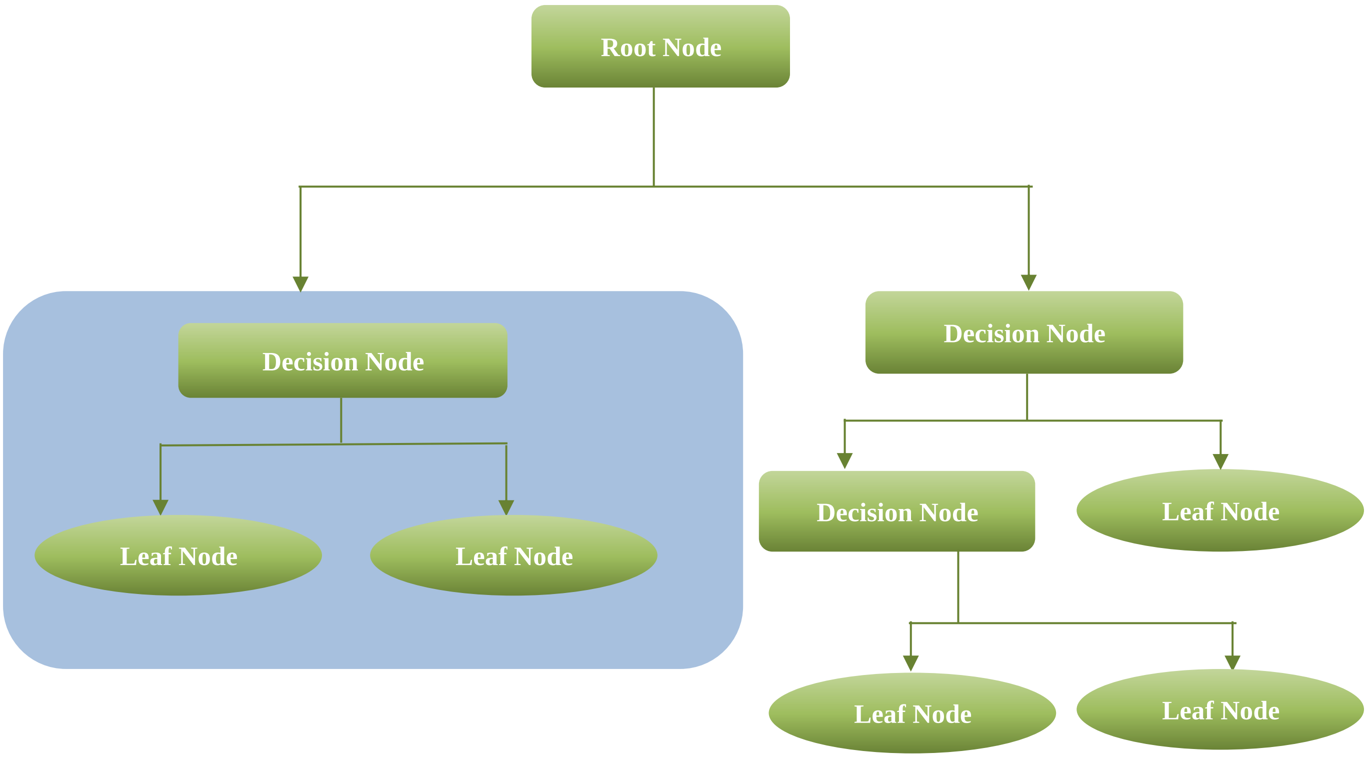

Fig. 3 Decision tree structure

3. Methodology

This paper proposes a method that trains with PSO. The technique of converting raw data into manageable numerical features while retaining the original information is known as feature extraction—network weights. Initially, features are extracted. It renders superior outcomes when compared to utilizing machine learning on the raw data directly. Features can be extracted manually or automatically. Manual feature extraction requires both the application of a technique to extract those features as well as the identification and description of the qualities that are pertinent to a particular situation.

Making decisions on whether features might be advantageous typically benefits from having a solid grasp of the context or domain. After years of research, engineers and scientists have created feature extraction methods for images, signals, and text. The early layers of Deep Networks have largely replaced feature extraction with the introduction of Deep Learning [Goodfellow, Y. Bengio, and A. Courville(2016)], mostly for images. The Histogram of Oriented Gradients (HOG), Speeded-up Robust Features (SURF), and Local Binary Pattern (LBP) features are techniques for extracting image features. The SURF method is efficient and trustworthy for local, similarity-invariant representation and imagery comparison; real-time applications like tracking and object detection are made possible by the SURF technique's quick computation of operators utilizing box filters.

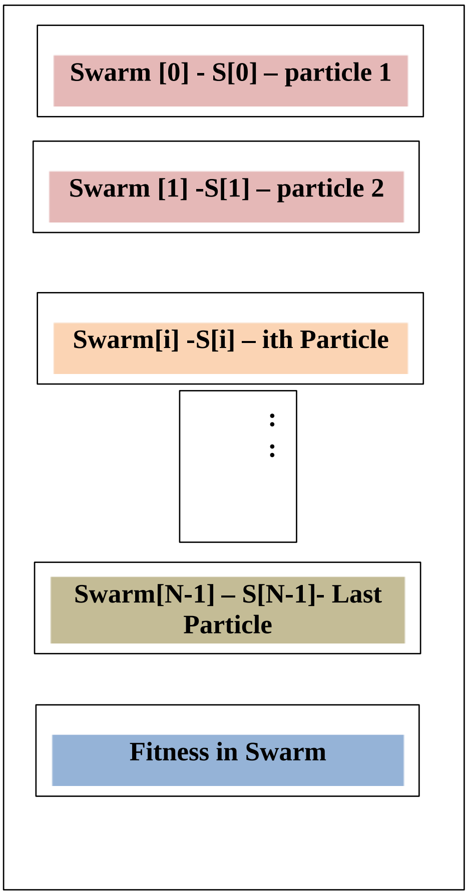

Fig. 4 Data structure of swarm particle

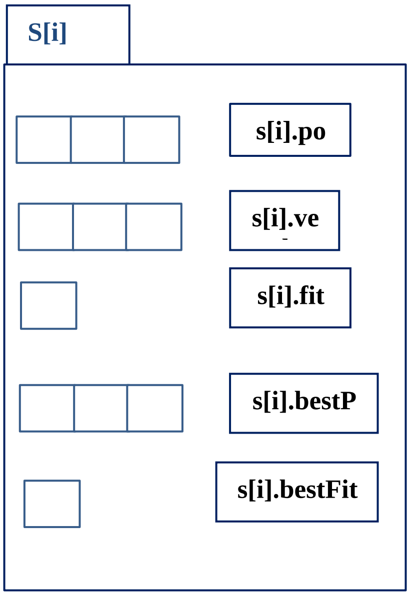

Fig. 5 Structure of ith particle -s[i]

H. Bay and Zurich 2009 presented a study on the SURF framework from the publication co-authored by A.ESS, Van GoolS. Our proposed system extracts features using the SURF method. A sample image has 63 features, and the network model is shown below to classify one image category. Our system is a hybrid of a shallow neural network and a PSO. The neural network itself generates the initial structure or body of the system. However, PSO is crucial for weighting the neurons throughout training, illustrating the actual power of evolutionary algorithms in this case. Data structures to store particle swarm optimization are shown in the figure 4.

3.1 . PSO Algorithm

3.1.1. Parameters of the Problem

The following are the parameters of the problem d is the number of dimensions; minx is the lower bound value maxx is the upper bound value. The Hyperparameters of the algorithm are N is the number of particles, maxiterations are maximum iterations, weights are inertia, c1 is the cognition of the particle, and c2 is the social influence of the swarm. Each swarm is a particle in the algorithm; it is represented as s[i].

3.1.2. Algorithm

Step 1: Initialize the population of particles of size N Xi (i=1, 2, …, n) randomly

Step 2: choose hyperparameter values such as weight, c1 and c2

Step 3: For i 1 to maxitertions:
For i = 1 to N:
Repeat 3.1 to 3.4

Step 3.1. Calculate the new velocity of the ith particle as
s[i].vel = weight*s[i].vel + product of (r1,c1,diff(s[i].bestPos,s[i].pos) + product of(r2,c2,(best_pos_swarm - s[i].pos)

Step 3.2. If velocity is not within the range of min and max, then clip it by the following operations
minx=minimum(s[i].vel, minx)
maxx=maximum(s[i].vel,maxx)

Step 3.3. Calculate and update ith particle position with ith velocity. s[i].pos += s[i].vel

Step 3.4. Update the new best value and new best of swarm using the below conditions
compare if s[i].fit < s[i].bestFit then :
    set s[i].bestFit <- s[i].fit
    set s[i].bestPos <- s[i].pos
compare if s[i].fit < best_swarm
    set best_swarm <- swarm[i].fit
    set best_pos_swarm <- swarm[i].pos
  End-for
End -for

Step 4. Return the best swarm particle.

3.2. Proposed Model

In the proposed system, the data is loaded then the SURF feature extraction process is applied. The network model is trained with data, and network weights are trained with a particle swarm optimization algorithm. The result of the PSO model, SVM model, Tree Model and KNN Model are finally evaluated for comparison. The figure below illustrates the proposed model. The model is verified for 20 iterations of experiments on six different classes.

The image features are extracted using a speed-up and robust feature method during the process. 63 features are obtained in the input image. The model is depicted in the experiments & results section. The labels are then manually given using code. The resultant file is an input to the model for classification to the Deep Network Model using PSO.

The figure 6 is the workflow of the system having the evaluation of machine learning models with PSO-NN.

4. Datasets

The dataset consists of four major classes of Achchulu, gunintalu, hallulu and oththulu, downloaded from ieee-dataport.org. Each class further has subclasses with a total of 20000 images. Six classes are chosen for our experiment. The experiment is done in a single Nvidia GPU Process.

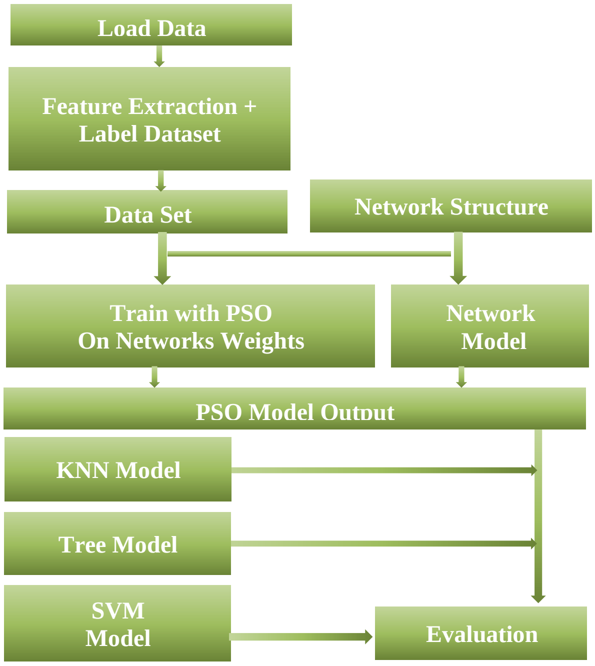

Fig. 6 Work flow of the system

5. Experiments & Results

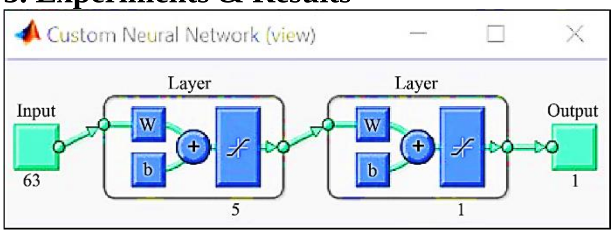

Fig. 7 Network Model with 63 features for an input image

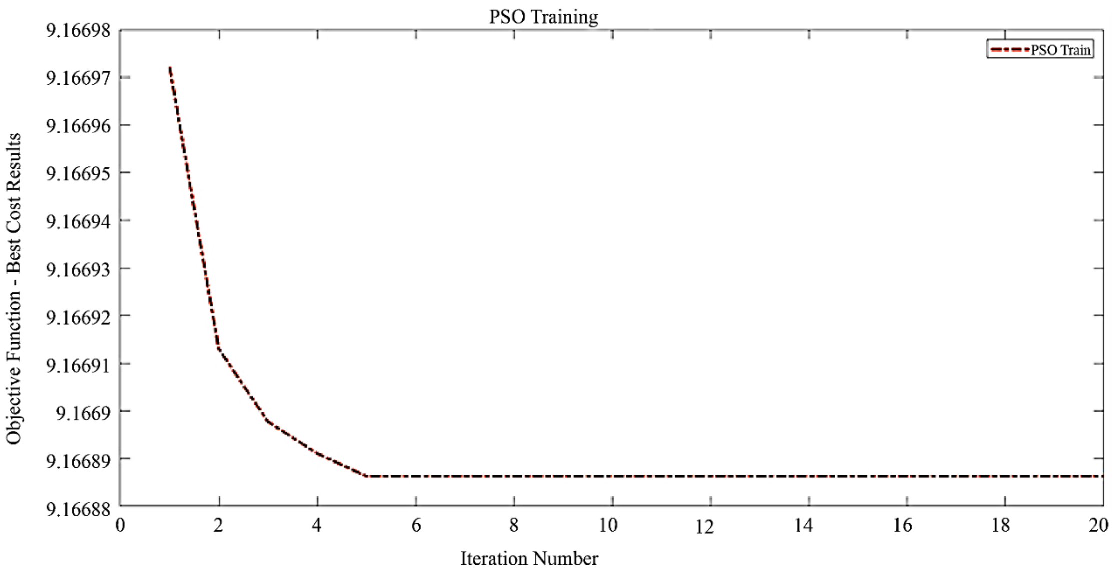

Fig. 8 Graph showing the result of best cost with iteration

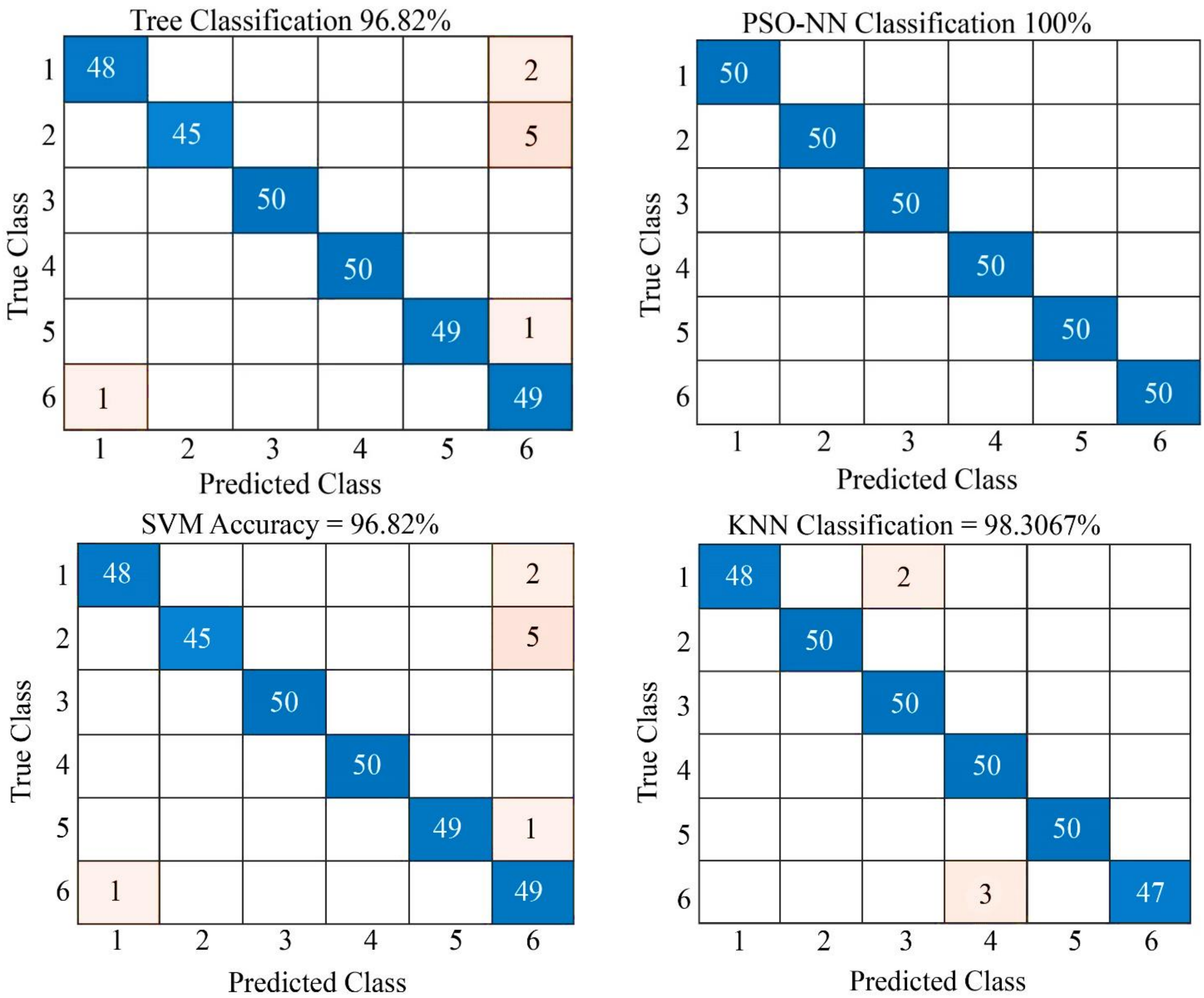

Fig. 9 Result of models accuracy

6. Conclusion

The Character Recognition in Telugu problem has been taken into account in this work. The image features are extracted using the SURF features selection approach. In order to increase the recognition rate, PSO is employed. PSO has been utilized to give an ideal set of weights to these features. Throughout the work, a dataset that comprises 6 classes of characters and 16 samples in each class has been taken into consideration. In comparison to existing classification algorithms, our suggested PSO-based technique also provides a higher level of classification accuracy. The PSO parallel versions are simple to construct on GPUs.The work can be extended using PSO to adjust other Deep Learning parameters in the future, including the activation functions and the number of epochs with the massive dataset on the Amazon Rekognition API.

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, Cambridge, MA: the MIT Press, 2016. [Publisher Link]

[2] Berry, Michael, “Machine Learning and Understanding for Intelligent Extreme Scale Scientific Computing And Discovery,” *Advanced Scientific Computing Research (ASCR) Division of the Office of Science, U.S. Department of Energy*, Workshop Report, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yuri Malitsky et al., “Tuning Parameters of Large Neighborhood Search for the Machine Reassignment Problem,” *International Conference on AI And OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, pp. 176–192, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Yasser Ganjisaffar et al., “Distributed Tuning of Machine Learning Algorithms Using Mapreduce Clusters,” *Proceedings of the Third Workshop on Large Scale Data Mining: Theory and Applications*, pp. 1-8, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[5] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization,” *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995. [CrossRef] [Google Scholar] [Publisher Link]

[6] C. W. De Silva, Mechatronic Systems: Devices, Design, Control, Operation and Monitoring, Boca Raton: CRC Press, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jan Karwowski, Michał Okulewicz, and Jarosław Legierski, "Application of Particle Swarm Optimization Algorithm to Neural Network Training Process in the Localization of the Mobile Terminal," *Engineering Applications of Neural Networks*, pp. 122–131, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yana Mazwin Mohmad Hassim, and Rozaida Ghazali, "Solving a Classification Task Using Functional Link Neural Networks with Modified Artificial Bee Colony," *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 189–193, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[9] Habib Shah, and Rozaida Ghazali, "Prediction of Earthquake Magnitude by an Improved ABC-MLP," *2011 Developments in E-Systems Engineering*, pp. 312–317, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[10] L. Qiongshuai, and W. Shiqing, "A Hybrid Model of Neural Network and Classification in Wine," *2011 3rd International Conference on Computer Research and Development*, vol. 3, pp. 58–61, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[11] Beatriz A. Garro et al., "Artificial Neural Network Synthesis By Means of Artificial Bee Colony (ABC) Algorithm," *2011 IEEE Congress of Evolutionary Computation (CEC)*, New Orleans, LA, pp. 331-338, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[12] K. Bovis et al., "Identification of Masses in Digital Mammograms with MLP And RBF Nets," *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, pp. 342-347, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ankit Sharma, and Dipti R Chaudhary, "Character Recognition Using Neural Network," *International Journal of Engineering Trends and Technology*, vol. 4, no. 4, pp. 662-667, 2013. [Google Scholar] [Publisher Link]

[14] SeyedAli Mirjalili et al., "Training Feedforward Neural Networks Using Hybrid Particle Swarm Optimization and Gravitational Search Algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125–11137. [CrossRef] [Google Scholar] [Publisher Link]

[15] Seyed Mojtaba Hosseini Bamakan et al., "Parameters Optimization for Nonparallel Support Vector Machine by Particle Swarm Optimization," *Procedia Computer Science*, vol. 91, pp.482–491, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Adam Blum, *Neural Networks in C++: An Object-Oriented Framework for Building Connectionist System*s, 1st Edition. New York: Wiley, 1992. [Google Scholar] [Publisher Link]

[17] Z. Boger, and H. Guterman, "Knowledge Extraction from Artificial Neural Network Models," *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 4, pp. 3030–3035, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[18] K. Swingler, Applying Neural Networks: A Practical Guide, Pap/Dsk Edition. San Francisco: Morgan Kaufmann, 1996. [Google Scholar]

[19] M. Gnaneswari, and T. Chaitanya Kumar, "Application of Deep Convolutional Neural Networks to Telugu Scripts for Optical Character Recognition," *International Journal of Computer Trends and Technology*, vol. 71, no. 1, pp. 50-55, 2023. [CrossRef] [Publisher Link]

[20] Gordon S. Linoff, and Michael J. A. Berry, *Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management*, 3 Edition. Indianapolis, IN: Wiley, 2011. [Google Scholar] [Publisher Link]

[21] Anisha Arora et al., *Deep Learning with H2O*, 2015. [Google Scholar] [Publisher Link]

[22] Carmelo J. A. Bastos-Filho et al., "Multi-Ring Particle Swarm Optimization," *Evolutionary Computation*, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[23] Varick L. Erickson et al., "Occupancy Modeling and Prediction for Building Energy Management," ACM Transactions on Sensor Networks, vol. 10, no. 3, pp. 1-28, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] Yuhui Shi, and Russell C. Eberhart, "Parameter Selection in Particle Swarm Optimization," *Proceedings of the 1998 Annual Conference on Evolutionary Computation*, pp. 591-600, 1998. [Google Scholar] [Publisher Link]

[25] Y. Shi, and R.C. Eberhart, "Empirical Study of Particle Swarm Optimization," *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, pp. 1945-1950, 1999. [CrossRef] [Google Scholar] [Publisher Link]

[26] J. Kennedy, and W.M, "Spears Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator," *1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pp. 78-83, 1998. [CrossRef] [Google Scholar] [Publisher Link]