INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

EVALUATION OF PERFORMANCE OF PSO-BASED KERNEL SUPPORT VECTOR MACHINE IN OFFLINE DIGIT RECOGNITION

Olusayo D. Fenwa[1], Modupe O. Alade[2], Oluyinka T. Adedeji[3]

[1,2,3]Computer Science Dept. LAUTECH Ogbomoso, Oyo State, Nigeria
fenwadeborah@yahoo.com, moalade@lautech.edu.ng, otadedeji@lautech.edu.ng

ABTRACT- *Offline handwritten character recognition is one of the most challenging and ongoing areas of research in the field of pattern recognition. Selection of optimal features is one major problem in offline character recognition system, hence in this paper, Particle Swarm Optimization (PSO) is proposed for selecting optimal features. Support Vector Machines is a set of related supervised leaning methods which can be used for both classification and regression. In this paper, Particle Swarm Optimization is integrated with KSVM (i.e. an improved SVM) to enhancement the performance of the classifier in terms of recognition accuracy and recognition time. Experiments were conducted on KSVM and PSO-Based KSVM classifiers using the WEKA machine learning tool on the MNIST offline handwritten digit dataset. The best recognition accuracy amidst three kernels was determined by experimenting on Polynomial Function, Gaussian Radial Basis Function and Dot kernel. However, experiments were designed to allow for comparison within the dataset in a cross validation and results indicate that PSO- Based KSVM performed better than KSVM with a maximal accuracy of 98.57%.*

Keywords - *Support Vector Machine, Feature Extraction, Feature Selection, Particle Swarm Optimization.*

1. INTRODUCTION

The development of handwriting recognition systems began in the 1950s when there were human operators whose job was to convert data from various documents into electronic format, making the process quite long and often affected by errors. Automatic text recognition aims at limiting these errors by using image pre-processing techniques that bring increased speed and precision to the entire recognition process. However, current OCR devices in use can offer good accuracy and high speed but they are still far away compared to the performance reached by the human being ([11]). Optical character recognition is a field of study than can encompass many different solving techniques. Neural networks ([4]), support vector machines and statistical classifiers seem to be the preferred solutions to the problem due to their proven accuracy in classifying new data.

There are several published studies that compare the paradigm of neural networks against the support vector machines. The main difference between the two paradigms lies in how the decision boundaries between classes are defined. While the neural network algorithms seek to minimize the error between the desired output and the generated output by the network, the training of an SVM seeks to maximize the margins between the borders of both classes. SVM approach has some advantages compared to others classifiers. They are robust, accurate and very effective even in cases where the number of training samples is small. SVM technique also shows greater ability to generalize and greater likelihood of generating good classifiers. Hence, in this paper, KSVM (i.e. an improved SVM) classifier is adopted in offline digit recognition. Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition system ([9]). No matter how sophisticated the classifiers and learning algorithms, poor feature extraction will always lead to poor system performance [8].

Many feature extraction techniques have been proposed in literature to improve overall recognition rate; however, most of the techniques used only one property of the handwritten character. This paper focuses on using a feature extraction technique that combined two characteristics (contour pixels and zoning) of the handwritten digit to create a global feature vector. Hence, in this work, a hybrid feature extraction algorithm was developed to alleviate the problem of poor feature extraction algorithm of offline digit recognition system. The hybrid feature extraction algorithm was developed using Geometrical and Statistical features. Integration of Geometrical and Statistical features was used to highlight different character properties, since these types of features are considered to be complementary. Contour pixel of the image containing the digit to be recognized together with certain statistical details of the digit area as proposed by [11] are adopted in this work. Thirteen features from two categories (Geometrical features and Statistical features) were utilized in this paper.

However, feature selection refers to the problem of dimensionality reduction of data, which initially consists of large number of features. The objective is to choose optimal subsets of the original features which still contain the information essential for the classification task while reducing the computational burden imposed by using many features. In this work, Particle Swarm Optimization is proposed for feature selection.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

2. RELATED WORK

[2] Considers that the selection of valuable features is crucial in character recognition so he introduces a new set of features called "Uniform Differential Normalized Coordinates" (UDNC), which are shown to improve the recognition rate using simple classification algorithms with a simple neural network on a reduced database. [11] proposed an Optical Character Recognition System using Support Vector Machine. He used a segmentation algorithm to reduce the time of image classification.

The problem of continuous handwriting segmentation into individual characters was addressed using an analytic approach where word recognition is based on the individual classification of characters by [13]. The input to the proposed segmentation method is a handwritten text image consisting of known words and the method consists of three stages: creation segmentation variants using genetic algorithm, refinements by applying a sequence of subtle segment boundary displacements and selection of most typical character prototypes. Their experiments show that application of each stage improves the word recognition accuracy. [12] proposed a recognition model for English handwritten (lowercase, uppercase and letter) character recognition using Freeman chain code (FCC) as the representation technique of an image character. Support vector machine (SVM) was chosen for the classification step and experiments conducted using NIST databases. Results gave a relatively high accuracy for the problem of English handwritten recognition.

3. RESEARCH METHODOLOGY

In this paper, *four phases* of development of the proposed digit recognition system which include; Data Acquisition; Image Processing which include feature extraction and feature selection; Classification using KSVM and PSO-Based KSVM; and Testing were presented as shown in Figure 3.1.

3.1 Data Acquisition

The Offline MNIST Digits Dataset used in this work is a part of the Modified National Institute of Standards and Technology (MNIST) character recognition database ([6]). The MNIST database was constructed from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits. The MNIST training set is composed of 30,000 patterns from Special Database3 and 30,000 patterns from Special Database1. 60,000 pattern training set containing examples from approximately 250 writers were extracted as well as 10,000 test images. The original black and white (bi-level) images from MNIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. 1000 images were sampled without replacement from the 10,000 test images ensuring an

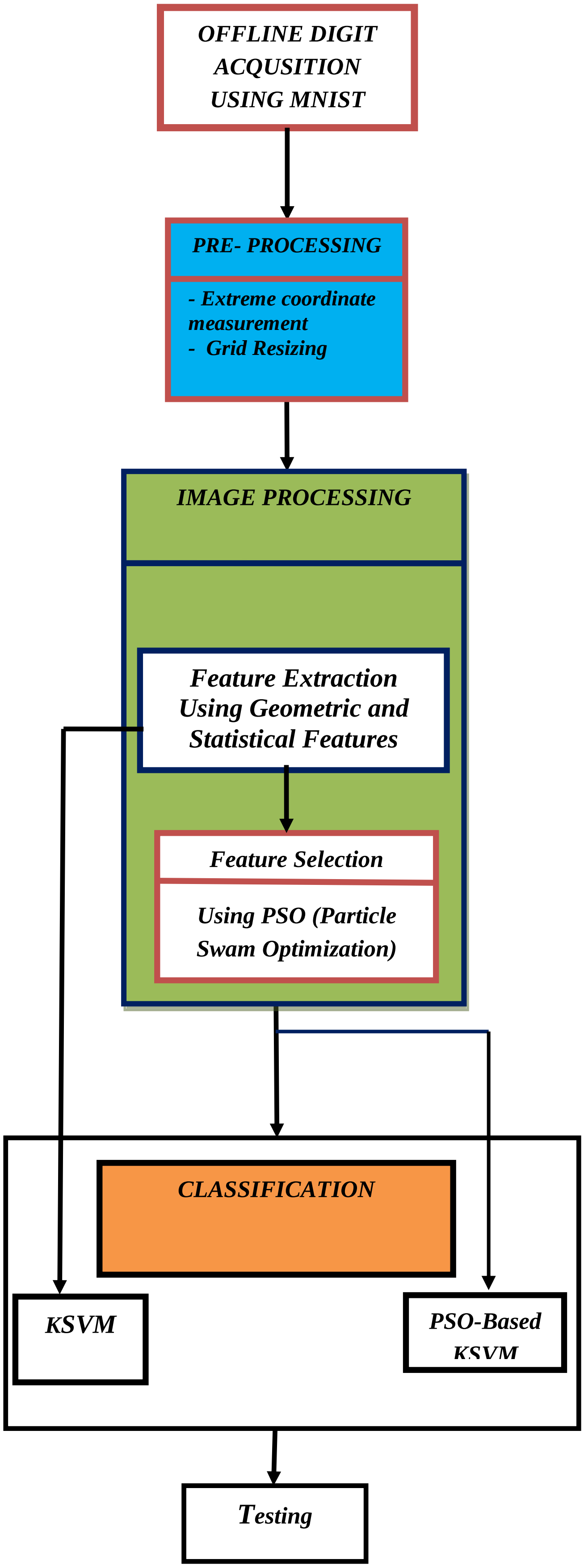

Figure 3.1: THE BLOCK DIAGRAM OF THE PSO-BASED SVM DIGIT RECOGNITION SYSTEM

equal class distribution. Only 1000 images were extracted and the extraction was done using the WEKA filter called "resample" which produces a random subsample of a dataset with or without replacement.

3.2 Image Processing

This phase is carried out in *two stages* which are: feature extraction and feature selection.

3.2.1 Feature Extraction: This stage is carried out in two steps:

Step 1: The Contour pixel of the image containing the digit to be recognized is used as the geometric feature of the digits.

Step 2: Statistical details of the digit area as proposed by [11] are adopted in this work. These statistical details are added to the discrete cosine transformation components to define its features as listed below:

- ➢ number of black pixels from a matrix (the so called "on"" pixels);
- ➢ mean of the horizontal positions of all the "on" pixels relative to the centre of the image and to its width;
- ➢ mean of the vertical positions of all the "on" pixels relative to the centre of the image and to its height;
- ➢ mean of horizontal distances between "on" pixels;
- ➢ mean of vertical distances between "on" pixels;
- ➢ mean product between vertical and horizontal distances of "on" pixels;
- ➢ mean product between the square of horizontal and vertical distances between all ''on'' pixel;
- ➢ mean product between the square of vertical and horizontal distances between all ''on'' pixels;
- ➢ mean number of margins met by scanning the image from left to right;
- ➢ sum of vertical positions of the margins met by scanning the image from left to right;
- ➢ mean number of margins met by scanning the image from bottom to top;
- ➢ sum of horizontal positions of the margins met by scanning the image from top to bottom.

''One geometric feature (contour pixel'') and *''twelve statistic features''* (making a total of thirteen features) were used in this work. The last operation in this module is the normalization of the results obtained up until now so as they correspond to the format accepted by the support vector machine module.

3.2.2 Feature Selection

Feature selection refers to the problem of dimensionality reduction of data, which initially consists of large number of features. The objective is to choose optimal subsets of the original features which still contain the information essential for the classification task while reducing the computational burden imposed by using many features.

In this work, Particle Swarm Optimization is proposed for feature selection.

Particle Swarm Optimization (PSO)

The PSO method is a member of wide category of Swarm Intelligence methods for solving the optimization problems. It is a population based search algorithm where each individual is referred to as particle and represents a candidate solution. Each single candidate solution is "an individual bird of the flock", that is, a particle in the search space. Each particle makes use of its individual memory and knowledge to find the best solution. All the particles have fitness values, which are evaluated by fitness function to be optimized and have velocities which direct the movement of the particles. The particles move through the problem space by following a current of optimum particles. The initial swarm is generally created in such a way that the population of the particles is distributed randomly over the search space. At every iteration, each particle is updated by following two "best" values, called ***pbest*** and ***gbest***. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness value). This fitness value is called ***pbest***. When a particle takes the whole population as its topological neighbour, the best value is a global best value and is called ***gbest***. The detailed algorithm as proposed by [5] is given as follows:

Step 1: Set the constants k_{max}, c_1, c_2, r_1, r_2, w.
Randomly initialize particle positions $x_0(i)$ for i = 1, 2,………,p.
Randomly initialize particle velocities $v_0(i)$ for i = 1, 2,……....,p.
Step 2: Set k = 1.
Step 3: Evaluate function value f_k using design space coordinates $x_k(i)$
If $f_k \geq f_{pbest}$, then pbest(i) = $x_k(i)$
If $f_k \geq f_{gbest}$, then gbest = $x_k(i)$
Step 4: Update particle velocity using the following equation

$$v_{k+1}(i) = w*(v_k(i)) + c_1r_1*(pbest_k(i) - x_k(i)) + c_2r_2(gbest_k - x_k(i)) \qquad (1)$$

Update particle position vector using the following equation

$$x_k + 1(i) = x_k(i) + v_k + 1(i) \qquad (2)$$

Step 5: Increment i. If i > p, then increment k and set i = 1.
Step 6: Repeat steps 3 to 5 until k_{max} is reached.
The notations used in this algorithm are:
k_{max} = maximum iteration number
w = inertia weight factor
c_1, c_2 = cognitive and social acceleration factors
r_1, r_2 = random numbers in the range (0, 1).

In this paper, each of the thirteen features are represented by a chromosome (string of bits) with *''13 genes (bits)''* corresponding to the number of features. An initial random population of 20 chromosomes is formed to initiate the genetic optimization. The initial coding for each particle is randomly generated. The order of position of the features in each string is

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

contour pixel, number of black pixels from a matrix (the so called "on"" pixels); mean of the horizontal positions of all the "on" pixels relative to the centre of the image and to its width; mean of the vertical positions of all the "on" pixels relative to the centre of the image and to its height; mean of horizontal distances between "on" pixels; mean of vertical distances between "on" pixels; mean product between vertical and horizontal distances of "on" pixels; mean product between the square of horizontal and vertical distances between all "on" pixels; mean product between the square of vertical and horizontal distances between all "on" pixels; mean number of margins met by scanning the image from left to right; sum of vertical positions of the margins met by scanning the image from left to right; mean number of margins met by scanning the image from bottom to top; sum of horizontal positions of the margins met by scanning the image from top to bottom respectively. A suitable fitness function is estimated for each individual. The fittest individuals are selected and the crossover and the mutation operations are performed to generate the new population. This process continues for a particular number of generations and finally the fittest chromosome is calculated based on the fitness function. The features with a bit value "1" are accepted and the features with the bit value of "0" are rejected. The fitness function used in this work is given by

$$\text{Fitness } (\alpha * \gamma) + \beta * \left[\, |c| - |r| / |c| \,\right] \qquad (3)$$

where γ = classification accuracy
c = total number of features
r = length of the chromosome (number of '1's)
$\alpha \in [0, 1]$ and $\beta = 1 - \alpha$

This formula shows that the classification accuracy and the feature subset length have different significance for feature selection. A high value of α assures that the best position is at least a rough set reduction. The goodness of each position is evaluated by this fitness function. The criteria are to maximize the fitness values. An optimal solution is obtained at the end of the maximum iteration. This value is binary coded with ''thirteen bits''. The bit value of "1" represents a selected feature whereas the bit value of "0" represents a rejected feature. Thus an optimal set of features are selected from the PSO technique. Out of the **''thirteen features''** extracted, **''eight optimal set of features''** are selected from the PSO algorithm.

3.3 Classification Module

3.3.1 Support Vector Machines

The standard SVM is a linear classifier which is composed of a set of given support vectors **z** and a set of weights **w**. The computation for the output of a given SVM with N support vectors $z_1, z_2, \ldots, z_N$ and weights $w_1, w_2, \ldots, w_N$ is then given by:

$$F(x) = \sum_{i=1}^{N} w_i (z_i, x) + b \qquad (4)$$

A decision function is then applied to transform this output in a binary decision. Usually, sign(.) is used, so that outputs greater than zero are taken as a class and outputs lesser than zero are taken as the other. In simple words, given a set of training examples, each marked as belonging to one of two categories, an SVM classification training algorithm tries to build a decision model capable of predicting whether a new example falls into one category or the other. If the examples are represented as points in space, a linear SVM model can be interpreted as a division of this space so that the examples belonging to separate categories are divided as shown in figures 3.2 and 3.3 by a clear gap that is as wide as possible. New examples are then predicted to belong to a category based on which side of the gap they fall on.

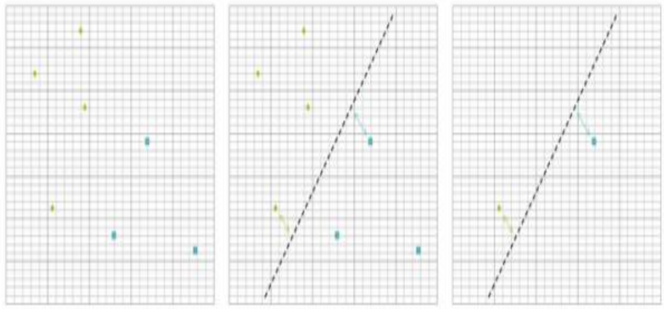

Figure 3.2: Linear separation in feature space ([10])

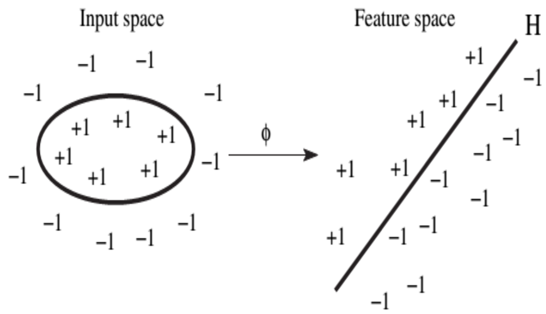

Figure 3.3: Linear separation in feature space

3.3.2 Kernel Support Machine

As detailed in 3.3.1, the original SVM optimal hyperplane algorithm is a linear classifier. However, many years later, some researchers suggested a way to create non-linear classifiers by applying the *kernel trick* to those maximum-margin hyperplanes. The result was a "boom" in the research of kernel machine, which became one of the most powerful and popular classification methods to date because the Kernel trick is a very powerful tool able to provide a bridge from linearity to non-linearity to algorithms which solely depend on the dot product between two vectors. It comes from the fact that, if we first map our input data into a higher-dimensional space, a linear algorithm operating in this space will behave non-linearly in the original input space. The *"trick"* resides in the fact that this mapping does not ever need to be computed: all we have to do is replace all dot products by a suitable kernel function. The kernel function denotes an inner product in feature space, and is usually denoted as:

$$k(x, y) = \langle \varphi(x), \varphi(y) \rangle \qquad (5)$$

This is a Kernel function in which φ denotes a possibly non-linear mapping function

Using a Kernel function, the algorithm can then be carried into a higher-dimension space without explicitly mapping the input points into this space. This is highly desirable, as sometimes our higher-dimensional feature space could even be infinite-dimensional and thus infeasible to compute. Since the original formulation of SVMs mainly consists of dot products, it is straightforward to apply the Kernel trick. Even if the resulting classifier is still a hyperplane in the high-dimensional feature space, it may be non-linear in the original input space.

A good separation of the features space is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (functional margin, δ) as shown in Figure 3.4. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

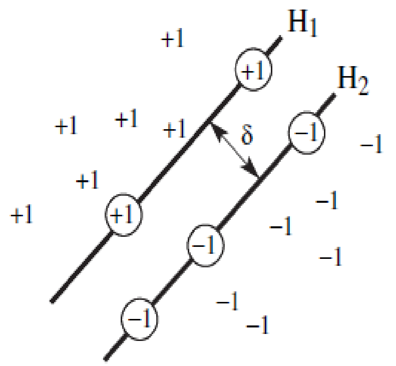

Figure 3.4: Maximum separation hyperplane

For any particular set of two-class objects, SVM finds the unique hyperplane having the maximum margin (Figure 3.4). The hyperplane H_1 defines the border with class +1 objects, whereas the hyperplane H_2 defines the border with class -1 objects. Two objects from class +1 define the hyperplane H_1, and three objects from class -1 define the hyperplane H_2. These objects, represented inside circles in Figure 3.4, are called *support vectors*. A special characteristic of SVM is that the solution to a classification problem is represented by the support vectors that determine the maximum margin hyperplane. SVM can also be used to separate classes that cannot be separated with a linear classifier knowing well that digit recognition vectors are non-linear in nature ([3]). In such cases, the coordinates of the objects are mapped from the input space into a feature space using non-linear functions called feature functions *Ø(x)* e.g. two-dimension to three-dimension. The feature space (Figure 3.3) is a high-dimensional space in which the two classes can be separated with a linear classifier (kernel trick). The nonlinear feature functions *Ø(x)* used are called Kernels $K(x_i,x_j)$.

Kernels have the advantage of operating in the input space, where the solution of the classification problem is a weighted sum of kernel functions evaluated at the support vectors. Commonly used kernels are Linear or Dot kernel (Equation 6), Polynomial, Radial Basis Function (Equation 7), Gaussian Radial Basis function and Sigmoid kernel (Equation 8).

$$k(x_i , x_j) = x_i . x_j \quad (6)$$

$$k(x_i , x_j) = exp\,(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \quad (7)$$

$$k(x_i , x_j) = tanh\,(kx_i . x_j + b) \quad (8)$$

where x_i and x_j are the corresponding coordinates of the two-dimensional pattern x, σ controls the shape of the separating hyperplane with σ > 0, b is the bias or threshold and k is the Lagrangian multiplier with k ≥ 0

The question of "which SVM kernel gives the best recognition accuracy amidst numerous kernels?" can only be solved by experimenting the various available kernels to ascertain the one which has the best recognition accuracy for a given problem ([3]). Hence, Polynomial function, Gaussian Radial Basis and Dot kernel of the SVM kernels were adopted in this paper. Character recognition classifications usually need more than two classes for classification (multiclass). Two common methods to build such binary classifiers are those where each classifier is trained to distinguish: (i) one of the labels against to all the rest of labels (known as one-versus-all) or (ii) every pair of classes (known as one-versus-one). Classification of new instances for one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class ([7]). The classification of one-versus-one case is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally, the class with more votes determines the instance classification.

4. EXPERIMENTAL SETUP

Experiments were carried out on the Waikato Environment for Knowledge Analysis (Weka) ([1]) tool (version 3.6.6) using the MNIST offline handwritten digits. The Support Vector Machine kernels used in the experiment were three: Polynomial function, Gaussian Radial Basis and Dot kernel. The experiment was performed on 1000 images on offline database. Comparative analysis od the two classifiers was carried out on offline MNIST dataset in which a ten fold cross validation procedure was repeated ten times for the dataset. The system configurations used in this work is Kernel Support Vector Machine on the WEKA LIBSVM used one after the other with all other parameters left at default.

5. RESULTS AND DISCUSSION

This paper has carried our comparison of percentage of accuracy of classifiers in single digit recognition using offline MNIST digit images, test of significance was done at 95% confidence (Two tail). Tables 5.1, 5.2 and 5.3 show the classification accuracies across the two classifiers (KSVM and PSO-Based KSVM) of different kernels; Polynomial function, Gaussian Radial Basis and Dot kernel using offline MNIST dataset.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Table 5.1: Effects of different Kernels on Recognition Accuracy

Classifiers	Polynomial Function	Gaussian Radial Basis	Dot Kernel
KSVM	97.88	96.23	97.95
PSO-Based KSVM	98.35	97.91	98.57

Experimental results in table 5.1 indicates that Dot kernel, outperformed other kernels and PSO-Based KSVM (with Dot kernel) gives better accuracy than KSVM (with Dot kernel) due to its ability to remove irrelevant features in order to enhance recognition accuracy.

Table 5.2: Evaluation of KSVM and PSO-Based KSVM on Training Time (in Seconds)

Classifiers	Polynomial Function	Gaussian Radial Basis	Dot Kernel
KSVM	9.85	12.40	9.46
PSO-Based KSVM	8.95	9.25	6.58

From table 5.2, the training time of PSO-Based KSVM is smaller when compared with the KSVM classifier due to its ability to achieve dimensionality reduction and removal of irrelevant features of the images. Also, complex and large sized input sets require a large topology network with more number of iterations (Epochs). The epochs is directly proportional to the training time, this implies that the larger the image size, the more the training time.

Table 5.3: Effect of Variation of Image Size on Recognition Accuracy

	KSVM			PSO-Based KSVM		
Image Sizes	*Polynomial Function*	*Gaussian Function*	*Dot Kernel*	*Polynomial Function*	*Gaussian Function*	*Dot Kernel*
8 by 8	97.88	96.23	97.95	98.35	97.91	98.57
20 by 20	96.43	95.31	96.54	97.45	96.89	97.22

Table 5.3 shows the results of variation in image size on recognition accuracy. The higher the image size the lower the percentage of recognition, although, the rate of change was small. The more the dimensional input vector (Digit matrix size), the less the recognition performance due to introduction of more noise as the image size increases.

6. CONCLUSION AND FUTURE WORK

This paper explores the need for optimization algorithms to enhance the performance of the classifiers. In this work, PSO is used as the optimization algorithm and it is used along with the KSVM classifier. Experimental results suggest better improvement in the classification accuracy for the PSO-Based KSVM over the other classifier. However, an increase in the convergence rate is also achieved by the PSO-Based KSVM classifier which is highly essential for real-time applications. Therefore an optimization technique is highly essential irrespective of the classifiers under consideration.

However, the application of PSO optimization algorithm for performance improvement of the Kernel Support Vector Machine classifier has been explored in the context of offline digit image classification. Two classifiers, Support Vector Machines (SVM) and Kernel Support Vectot Machine have been compared for handwritten digit recognition using MNIST offline datset. Experimental results indicate that KSVM,using a Dot kernel, outperformed kernels.

Our future work will be tailored towards hybridization of other classifiers to further enhance the performance of the system. The work will also be extended by using different optimization algorithms to estimate the performance of the classifiers. However, different set of features will be used to improve the classification accuracy and experiments will be carried out on a different set of database in order to generalize the technique. We also intend to investigate the performance of these classifiers in recognition of other characters (alphabets, punctuations and other symbols). Irrespective of the modifications and the systems used, this paper has been able to present the significance of optimization algorithm for accurate and quick image classification systems.

REFERENCES

[1] Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., and Scuse, D. (2013): "WEKA Manual for Version 3-6-6", The University of Waikato.

[2] Agell C., Neural Networks for Pen Characters Recognition. (online) http://www.ics.uci.edu/~dramanan/teaching/ics273a_winter08/projects/cagell_AgellProject.pdf, 2008.

[3] Ovidiu, I. (2007): "Applications of Support Vector Machines in Chemistry". In: Reviews in Computational Chemistry, 23:291-400.

[4] Sandu V., Leon F., Recognition of Hadwritten Digits Using Multilayer Perceptrons. Bul. Inst. Polit. Ia%i,s. Automatic Control and Computer Science, LV (LIX), 4,103−114, 2009.

[5] Jude, D. H, Vijila, C. K. and Anitha, J (2010): ''Performance Improved Pso Based Modified Counter propagation Neural Network For Abnormal Mr Brain Image Classification'',

International Journal Advanced Soft Computing Application, 2(1): 65-84.

[6] Le Cun, Y. (1998): "The MNIST Database of Handwritten Digits".

[7] Thomé, A. C. G. (2012): “SVM Classifiers – Concepts and Applications to Character Recognition”, Advances in Character Recognition, (2): 25-50.

[8] Marc, P., Alexandre, L., and Christian, G. (2001): ‘’Character Recognition Experiment using Unipen Data’’, Parizeau and al., Proceeding of ICDAR, Seattle: 10-13.

[9] Pradeep, J., Srinivasan, E. and Himavathi, S. (2011): ‘’Diagonal Based Feature Extraction for Handwritten Alphabets Recognition using Neural Network’’, International Journal of Computer Science and Information Technology (IJCS11), 3(1): 27-37.

[10] Cesar de Souza (2012): ‘’Hand writting Recognition Revisited: Kernel Support Vector Machine’’ Engineer Xerox Research Centre, Europe.

[11] Eugen-Dumitru and Florin Leon (2012): ‘’Optical Character Recognition System using Support Vector Machine’’ “Gheorghe Asachi” Technical University of Iasi, Faculty of Automatic Control and Computer Engineering

[12] Nasien, D., Haron,H. and Yuhaniz, S. (2010): " Support Vector Machine (SVM) for English Handwritten Character Recognition", Second International Conference on Computer Engineering and Applications, 249-252

[13] Sas, J. and Markowska-Kaczmar, U. (2012): "Similarity-based training set acquisition for continuous handwriting recognition", Information Sciences, 191, 226-244.