ISSN: 0976-3104
SPECIAL ISSUE (ASPM)
Kuttichira et al.

THE IIOAB JOURNAL

ARTICLE OPEN ACCESS

DIGIT RECOGNITION USING MULTIPLE FEATURE EXTRACTION

Deepthi Praveenlal Kuttichira*, V. Sowmya, K.P. Soman

Centre for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, INDIA

ABSTRACT

Digit Recognition is one of the classic problems in pattern classification. It has ten labels which are digits from 0-9 and each prototypes in the test set has to be classified under these labels. In this paper, we have used MNIST data for training and testing. MNIST database is a standard database for digit classification. A number of neural network algorithms have been used on MNIST to get high accuracy outputs. These algorithms are computationally costly. Here, we have used multiple feature extraction based on SVD and histogram to create testing and training matrix. To the feature vector formed by SVD, histogram values along x-axis and y-axis of an image is appended. These vectors are mapped to hyperplane using polynomial and Gaussian kernel. For classification open source software like GURLS and LIBSVM is used to obtain a fairly good accuracy.

Received on: 1st-April-2016
Revised on: 5th-April-2016
Accepted on: 10th -April-2016
Published on: 16th -April-2016

KEY WORDS

Support Vector Machine (SVM); Single Value Decomposition (SVD); Overall accuracy (OA); Class wise Accuracy (CA); Regularized Least Squares (RLS), digit recognition

***Corresponding author:** Email: deepthikuttichira93@gmail.com **Tel:** +40-9001010010; **Fax:** +40-9001010012

INTRODUCTION

Recognition of handwritten digit is a typical character recognition problem with ten labels, each corresponding to a digit from 0-9. Many classification methods like K-nearest neighbors, linear classifiers, SVMs, neural networks, convolutional neural networks etc. has been used on MNIST database. A comparison between these algorithms are given [1]. Our approach for solving digit recognition problem, requires less computational cost and has considerable accuracy.

In machine learning, there are two types of learning approaches called supervised and unsupervised learning [2]. Class labels are a priori knowledge for supervised learning. Whereas, in unsupervised learning class labels are not a priori knowledge. In our proposed method, we have used multiple feature extraction to create vectors for each prototypes for classification [3]. The learning method proposed here, uses SVM as a supervised learning method. In any classification, the objective is to find a curve that separates two classes of data points. If that curve is a straight line, then the data is said linearly separable. To find this curve or plane of separation, the data points are plotted in a space and then we try to find a curve that separates them into separate classes. If such a curve cannot be found in the plane on which data points were originally projected, then the data points are projected on to a higher dimensional plane and a curve or a hyperplane that separates these data points is found. For this the data points have to be mapped from lower dimensional plane to higher dimensional plane. This mapping of data points from lower dimension to higher dimension is done using kernel functions.

Several neural networks have been trained to give good accuracy for MNIST dataset [4]. Algorithm that gives the best accuracy is that of convolutional neural networks. Neural networks and most of these best performing algorithms require large dataset and heavy computation. The algorithm proposed in this paper uses a subset of MNIST dataset and is computationally less costly. The proposed algorithm can be run on an ordinary desktop PC with a fairly good accuracy of 85.5%. For classification, we have used two tools LIBSVM and GURLS. A comparative analysis of these two tools, based on the classification result given by them is presented in section 3 of this paper. The tools, LIBSVM and GURLS are two open source software that offer several libraries based on kernel function for SVM. LIBSVM was developed by C.-C. Chang and C.-J. Lin. [6]. GURLS was introduced by Tacchetti, Mallapragada et al. [8]. GURLS can be efficiently applied for multiclass problems [5]. LIBSVM has packages for various kernel methods for SVM [5]. GURLS uses Regularized Least Squares for

COMPUTER SCIENCE

THE IIOAB JOURNAL ISSN: 0976-3104

classification and regression [5]. We observed that GURLS gave better results for digit classification than LIBSVM. Section 2 of this paper describes the dataset, preprocessing techniques and algorithm used in this paper. Once the feature extraction was done we have used both GURLS and LIBSVM for classification. A comparison study of these two algorithms is presented in the section 3 of this paper.

MATERIALS AND METHODS

LIBSVM

LIBSVM provides several inbuilt kernel functions for SVM [5]. SVM maximizes the margin of line of separation between classes. Support Vectors are the data points that fall on the margin [2]. If data points can be classified to two classes, SVM can be visualized as follows. Let (a_i, b_i) be two sets of data point belonging to the respective classes where i =1, 2… n. Let a_i of class 1 be labeled as +1 and b_i of class 2 be labeled as -1.Let A be the set that contains both a_i and b_i .The hyperplane that separates these two classes is given by the equation

$$w^T A - \gamma = 0 \quad (1)$$

Where w= $(w_1, w_2, w_3, \ldots, w_n)$ and A= $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)$ .W is the weight matrix that is learned. A is the feature vector. γ is the bias term. Depending on the sign of the function the class is chosen. The decision function is given below

$$f(x) = sgn(w^T x - \gamma) \quad (2)$$

If the data points cannot be linearly separated, they have to be projected into a plane with higher dimension, where it can be partitioned linearly by a hyperplane [2]. The projection into higher dimensional plane is done using a mapping function called kernel function. A mapping function is:

$$\phi : R^n \longrightarrow R^k$$

Here k>>>n. The mapping is given as

$$K(a_i, a_j) = < \varphi(a_i).\varphi(a_j) > \quad (3)$$

Few kernels provided by LIBSVM are listed below.

- LINEAR KERNEL

$$K(a_i, a_j) = a_i^T a_j \quad (4)$$

- POLYNOMIAL KERNEL

$$K(a_i, a_j) = (\gamma a_i^T a_j + r)^d, \gamma > 0 \quad (5)$$

- RADIAL BASIS FUNCTION KERNEL

$$k(a_i, a_j) = \exp(-\gamma \| a_i - a_j \|^2), \gamma > 0 \quad (6)$$

- SIGMOID KERNEL

$$k(a_i, a_j) = \tanh(\gamma a_i^T a_j + r) \quad (7)$$

Here γ ,r and d are kernel parameters.

GURLS

GURLS is an open source software that uses RLS for classification [5].In RLS say there are N data points $A = [a_1, a_2, .., a_n]$ belonging to R^n and let $Y = [y_1, y_2, \ldots, y_n]$ where Y is the label vector such that $y_i \in \Re^p$. In RLS for a p-class problem every data point x_i will have a corresponding y_i which belongs uniquely to a particular class in p. So the goal is to learn the weight matrix W, which maps the data vectors to the unique label vector. Learning of weights is a minimization problem of the form:

$$\min_w \sum_i \| w a_i - y_i \|_2^2 \quad (8)$$

THE IIOAB JOURNAL ISSN: 0976-3104

GURLS is a very efficient multiclass classifier [5]. One of the biggest advantages of GURLS is its automatic parameter tuning. GURLS models the problem faster too. The kernel used from GURLS library for this paper is Gaussian kernel with RBF type. Its mathematical formulation is

$$K(\mathrm{a}_i, \mathrm{a}_j) = \exp(-\gamma \| \mathrm{a}_i - \mathrm{a}_j \|^2), \gamma > 0 \qquad (9)$$

MATHEMATICAL BACKGROUND

For feature extraction, we have used the concept of Singular Value Decomposition (SVD). SVD is a decomposition method in Linear Algebra which decompose a matrix into three sub matrices U, S, V. If A is a $m \times n$ matrix then U is a $m \times m$ matrix that contains the Eigen vectors of $A^T A$ and V is a $n \times n$ matrix containing Eigen vectors of AA^T. S is a $m \times n$ diagonal matrix that contains square roots of non-zero Eigen values of AA^T and $A^T A$

$$A = USV^T \qquad (10)$$

DATASET

The dataset used for our paper is a subset of MNIST database. MNIST contains 60,000 images for training and 10,000 images for testing. We have used 1,200 images for training and 200 images for testing. The testing image set for each class is set to 20. The number of training images is decided by keeping the ratio of test and train of MNIST database. MNIST dataset was created from NIST dataset in which training data was collected from US Census Bureau employees and testing data taken from American High School students. MNIST dataset contains 28 $\times$ 28 pixel grayscale images for each prototypes. **Figure– 1** shows some sample images from MNIST dataset. The images shown are rather the well written ones in the dataset.

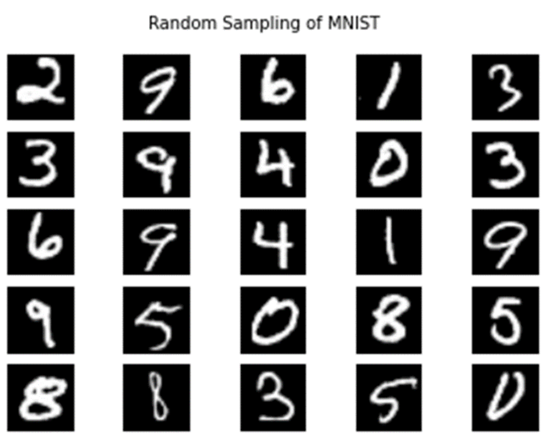

Fig: 1. Sample images of MNIST Dataset

PREPROCESSING

The prototypes for testing as well as training were cropped closely to the digits and then resized to 28 $\times$ 28 pixels. This was done to center align the digits. Blurring helps in digit recognition as specified in ref [7] [4]. The burring technique that we have used in this paper is Gaussian. Gaussian blur blurs the digits along its gradient. It helps to blur the sharp outlines and enhance digit recognition [2]. The mathematical formulation of Gaussian blur is

$$g(\mathrm{r}, \mathrm{s}) = e^{-(\mathrm{r}^2 + s^2)/(2\sigma^2)} \qquad (11)$$

The fast decay offered by this Gaussian blur helps in reducing the computational time [4]. The standard deviation σ controls the blur. A value of 0.9 is used as standard deviation. This is the same value specified in the [7]. **Figure– 2** shows Gaussian blurred corresponding to the digit 5. As decay factor increases, blurring also increases. The decay factor is chosen in such a way that, the blurring does not result in loss of structural information of the digit but does blur the fine outlines of the digit. Choice of the decay factor plays an important role in decreasing computational time [7].

ISSN: 0976-3104

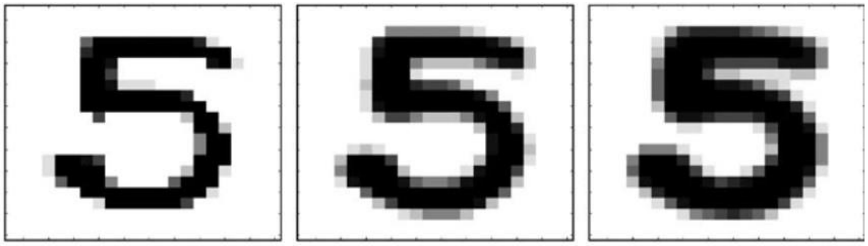

Fig: 2. Gaussian blurred image

FEATURE EXTRACTION

We have used two feature extraction techniques in this paper. For each prototype SVD is done and U*S matrix is converted to a column vector. This gives a feature vector of size 784. To this feature vector we append the structural information of the prototype. As structural information we have taken histogram of the prototype along x- axis as well as y-axis." It is the count of black pixels corresponding to each line and column of the prototype" [3]. So our final feature vector for each prototype has a length of 840 [784+28+28]. These feature vectors are extracted for all training data. Features of the same data type are labelled accordingly. The training matrix consists of these feature vectors and corresponding labels. This training matrix is given as input to LIBSVM and GURLS.

ALGORITHM

The algorithm used in this paper is briefly described below:

Step 1: The prototype images of training data are cropped and then resized to 28×28 pixels. This is done to center align the digits.
Step 2: Gaussian blur is added to the images with a decay factor of 0.9.
Step 3: SVD of the prototype and histogram along x-axis and y-axis are taken.
Step 4: The U*S matrix is made into a column vector and the histogram values are appended together to form a feature vector of size 840.
Step 5: The feature vectors along with labels form the training matrix.
Step 6: Steps I to 4 are repeated for test prototype.
Step 7: The training matrix and testing matrix are passed to GURLS and LIBSVM.
Step 8: Classification result is obtained.

RESULTS AND DISCUSSION

ACCURACY COMPARISON

Here we have discussed the results obtained from LIBSVM and GURLS. The best results was given by GURLS using Gaussian kernel. Overall accuracy and class wise accuracy given by LIBSVM and GURLS are discussed here. First, the formulas used for calculating Overall accuracy (OA) and class wise accuracy (CA) are listed below [5].

$$CA = \frac{\text{Total no : of correctly classified digits of each class}}{\text{Total no : of digits in each class}} \tag{12}$$

$$OA = \frac{\text{Total No. of correctly classified digits}}{\text{Total no. of digits}} \tag{13}$$

The confusion matrix obtained from LIBSVM using polynomial function is shown in **[Table– 1]**. From confusion matrix, we can infer the number of test data that are correctly classified as well as misclassified. The elements along the diagonal of a confusion matrix shows the count of prototypes that are correctly classified according to the respective classes. The non-diagonal elements are the misclassified prototypes. From the confusion matrix of

THE IIOAB JOURNAL ISSN: 0976-3104

[Table– 1] we can see that the digit 9 is highly misclassified as 1. This is because if the loop of digit 9 is too small, it can be easily mistaken as digit 1. Such similarities between digits that incur error rate can be easily observed using confusion matrix.

Overall accuracy and Class wise accuracies are 2 common accuracy measures for pattern classification [3]. These both measures for LIBSVM and GURLS are enlisted in [Table– 2]. Only for digit 6 and digit 9 LIBSVM shows a better accuracy. For rest all classes GURLS shows better accuracy. Digit 6 and digit 9 can be easily misplaced as 6 rotated would look like 9 and vice versa. GURLS also shows better overall accuracy than LIBSVM. GURLS parameter tuning is very efficient. The Gaussian kernel that we have used from GURLS library appears to be an apt mapping function for SVM based digit classification.

Table: 1. Confusion Matrix for Polynomial Function in LIBSVM

	Digit One	Digit Two	Digit Three	Digit Four	Digit Five	Digit Six	Digit Seven	Digit Eight	Digit Nine	Digit Zero
Digit One	15	0	2	0	0	0	0	1	0	2
Digit Two	0	14	0	3	0	1	0	1	0	1
Digit Three	1	0	14	0	1	0	1	0	0	3
Digit Four	0	0	0	12	0	1	1	0	3	3
Digit Five	0	1	3	0	13	0	0	3	0	0
Digit Six	0	1	0	0	1	19	0	0	0	0
Digit Seven	1	0	0	0	0	0	15	1	3	0
Digit Eight	1	0	1	0	0	0	1	13	1	3
Digit Nine	3	0	0	0	0	0	1	1	15	0
Digit Zero	0	0	0	0	0	0	0	0	0	20

Class wise accuracy of GURLS and LIBSVM is plotted in **Figure–3**. Except for digit 6 and digit 9 GURLS gives better results for all other digits than LIBSVM. Overall accuracy obtained by LIBSVM when using polynomial function for classification is 75% and that of GURLS is 85.5% when using Gaussian kernel. The overall accuracy plot between LIBSVM and GURLS is shown in **Figure–4**. Overall accuracy of GURLS is also higher than that of LIBSVM.

ISSN: 0976-3104

Table: 2. Class wise accuracy for Gaussian Kernel in GURLS and polynomial function in LIBSVM

	Digit One	Digit Two	Digit Three	Digit Four	Digit Five	Digit Six	Digit Seven	Digit Eight	Digit Nine	Digit Ten	OA
CA GURLS	95%	75%	90%	90%	65%	85%	90%	80%	65%	95%	85.5%
CA LIBSVM	75%	70%	70%	60%	65%	95%	75%	65%	75%	100%	75%

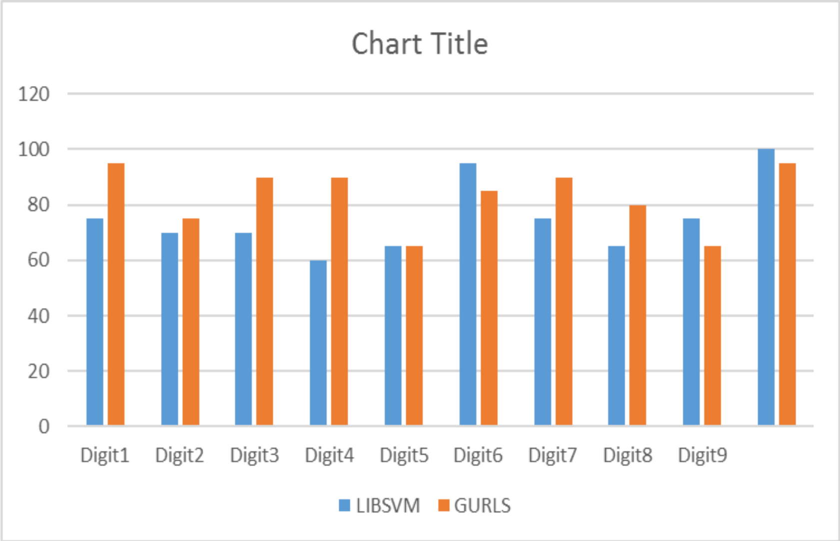

Fig: 3. Class wise accuracy of LIBSVM and GURL

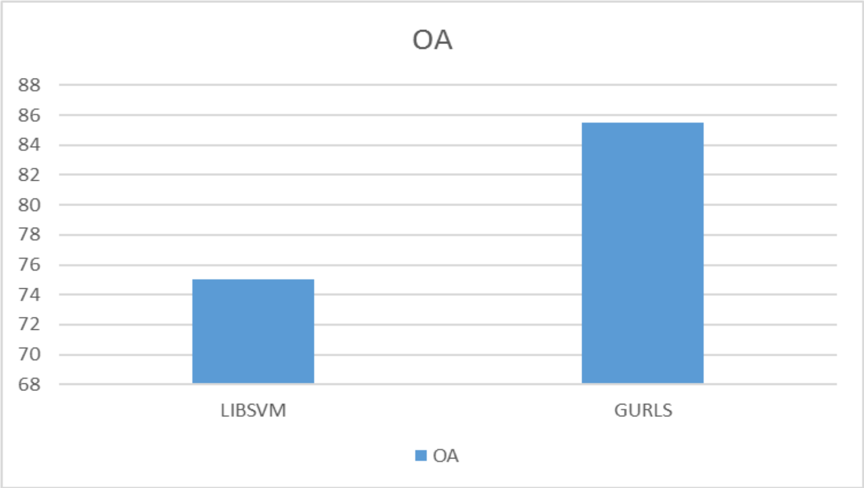

Fig: 4. Overall accuracy of LIBSVM and GURLS

THE IIOAB JOURNAL
ISSN: 0976-3104

CONCLUSION AND FUTURE WORK

We observed that for digit classification, using MNIST dataset, LIBSVM with polynomial classification function gave good classification with overall accuracy of 75%. But, a way better classification is provided by GURLS when used Gaussian kernel, which gave an accuracy of 85.5%. From this, we can infer that for digit classification using SVM, Gaussian kernel is ideal. Also, if handwritten data has minimum rotation based error then the accuracy of the proposed algorithm will be far better. Decay factor would vary according to different datasets. Tuning to a good decay factor enhances computational speed. Error rate can be further reduced by selectively increasing the sampling of structurally similar prototypes such as digit 1 and digit 9. Supervised learning for handwritten digit is good in computational perspective. Without any iteration the accuracy obtained is 85.5%. Including more scaled and rotated prototypes in the training set could improve the accuracy. Ideally if all such scaled and rotated prototypes are included, we should be able to obtain highly accurate classification for train any test data prototypes.

CONFLICT OF INTEREST

Authors declare no conflict of interest.

ACKNOWLEDGEMENT

None.

FINANCIAL DISCLOSURE

No financial support was received to carry out this project.

REFERENCES

[1] Y LeCun, L Jackel, L Bottou, et al. [1995] Learning algorithms for classification: a comparison on handwritten digit recognition, Neural Network: *The Statistical Mechanics Perspective* 261–276.

[2] KP Soman, R Loganathan, and V Ajay. [2009] Machine Learning with SVM and other Kernel methods. PHI Learning Pvt. Ltd.

[3] Rafael M O Cruz, George DC Cavalcanti ,Tsang Ing Ren. [2010] Handwritten Digit Recognition Using Multiple Feature Extraction Techniques and Classifier Ensemble. 17th International Conference on Systems, Signals and Image Processing.

[4] Berkant Sawas, Lars Elden.[2007] Handwritten digit classification using higher order singular value decomposition. *Pattern recognition* 40: 993–1003.

[5] Nikhila Haridas, V Sowmya, K P Soman.[2015] GURLS vs LIBSVM: Performance Comparison of Kernel Methods for Hyperspectral Image Classification. *Indian Journal of Science and Technology*, 8(24).

[6] C.-C. Chang, C.-J. Lin. [2011]Libsvm: A library for support vector machines. ACM *Transactions on Intelligent Systems and Technology (TIST)*, 2(3): 27.

[7] Patrice Y Simard, Yann A Le Cun, John S Denker, Bernard Victorri.[2000]Transformation Invariance in Pattern Recognition-Tangent Distance and Tangent Propagation. *Internat J Imag Systems Technol 11 (3)*: 181–197.

[8] A Tacchetti, PK Mallapragada, M Santoro, L Rosasco. [2013]Gurls: a least squares library for supervised learning. *The Journal of Machine Learning Research*, 14(1): 3201–320.

ABOUT AUTHORS

Deepthi Praveenlal Kuttichira *received her B.Tech degree in Computer Science in 2015. Currently she is pursuing her M.Tech degree in Computational Engineering and Networking (CEN) from Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidaypeetham, Amrita University, India. Her research interests include Machine Learning, Pattern Recognition and Big Data analysis.*

Sowmya V *currently serves as Assistant Professor at Amrita Centre for Computational Engineering and Networking (CEN), Coimbatore campus. Her research area include Image processing, Hyperspectral Image Classification, Pattern Recognition and Machine Learning.*

Dr. K P Soman *currently serves as Head and Professor at Amrita Centre for Computational Engineering and Networking (CEN), Coimbatore campus. His research interest include Software Defined Radio, Wireless Sensor Networks (WSN), High Performance Computing, Statistical Digital Signal Processing (DSP) on Field Programmable Gate Array (FPGA), Machine learning Support Vector Machines, Signal Processing and Wavelet & Fractals.*