International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 04, April 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

RECOGNITION OF HANDWRITTEN CHARACTERS USING SVM

Naveenkumar Yerra[1], Radha Varanasi[2], Hima Adapaka[3], Harikrishna Surumalla[4], Janakiram Dantha[5]

[1] Assistant Professor, M.Tech, [2,3,4,5]Student, B.Tech, Electronics and Communication, SSCE, Srikakulam, Andhra Pradesh, (India)

ABSTRACT

Recognition of characters, Digits and other symbols are become an enthusiastic and challenge research problem in the field of pattern recognition. The style and shape of the digits written by the same person may vary and entirely different for different persons. Exact digit recognition of different persons highly complex and challenging tasks. In this paper proposes a method for recognizing English digits and Numerals are formulated as pattern classification algorithm namely Support Vector Machine (SVM) has been employed to construct the model, which is implemented in Matlab environment using machine learning algorithm. It does not require training samples. The present algorithm is tested on a large data base of hand written and printed digits and its performance and accuracy are also determined.

Keywords: Handwritten and Printed digits database, Machine learning algorithm, Pattern recognition, Support Vector Machine.

I. INTRODUCTION

Nowadays many different applications need some Handwritten Characters recognition and because of that it represents an active research field. Character recognition is a part of computer vision, which refers to the problem of recognition of specific character in digital image or digital video. Digit recognition is used in post offices for sorting the mail, in banks for reading checks, for license plate recognition, street number recognition, etc.

Task of digit recognition can be divided into two groups, printed digit recognition and handwritten digit recognition. Recognition of printed digits is easier compared to the handwritten digit recognition because printed digits have regular shape and difference between images of the same number are just in the angle of view, size, color, etc. On the other hand, there are numerous handwriting styles which mean that the same digit can be written in many different ways, hence more effort is required to find similarity between instances of the same digit.

In general, nowadays digit recognition contains three parts, preprocessing, feature extraction and classification. Preprocessing prepares image for feature extraction. Some of the common preprocessing steps are binarization, centering, morphological operations and more. Feature extraction is very important step and success of the classification strongly depends on it. Many different features were proposed in literature. In [JSDK13]

horizontal and vertical projection with dynamic thresholding was proposed. Projection histograms are usually used for printed digit recognition and combined with other feature sets.

One of the most important parts of object recognition algorithms and handwritten digit recognition algorithms is classification. Classification in computer science represents prediction of class or label for an object based on its similarity with previous objects. In machine learning, each object or instance is represented with same set of features. Based on the learning algorithm, classifiers can be divided to unsupervised and supervised classifiers. Supervised learning uses knowledge of labels for instances used for building the model while instances for unsupervised learning are unlabeled. Today, many techniques for building a classification model are used. One of the simplest machine learning algorithms is SVM.

In this paper we propose using SVM for handwritten digits recognition. Support vector machine has a few parameters that should be adjusted. First parameter is parameter of soft margin C that allows outliers to be misclassified. In real life data, outliers are common and also data usually are not linearly separable. In that case some kernel functions need to be used and parameter of this functions also need to be tuned. One of the common kernel functions is Gaussian radial basis function with parameter g. Tuning parameters of SVM is a hard optimization problem.

II. SUPPORT VECTOR MACHINE

Support vector machine was proposed by Vapnik as binary classifier [CV95]. It represents one of the latest supervised learning classifiers and it was used in numerous applications. SVM discovers a hyperplane that separates data from different classes. Each instance is labeled with one of existing classes and they are represented as points in space. SVM builds a model based on instances from training set and further classification of unknown instances is done by that model.

Hyperplane that separates labeled instances from the training set is defined by the next equation:

$$yi(w.\ xi+b) \geq 1 \quad \text{for } 1 \leq i \leq n \qquad (1)$$

where xi ∈ Rd are instances represented as vectors in d-dimensional space, n is the number of instances, yi ∈ (-1;1)g are classes of corresponding instances and w and b are parameters of the hyperplane. This hyperplane is determined by the nearest instances that are called support vectors. Hyperplane should be as far as possible from instances of both classes. The distance that should be maximized is $\frac{2}{||w||}$.

The described model has a problem to classify real life data, because all instances must be on the correct side of the hyperplane. Real world data contains some noise and usually a few outliers. The previous model is not able to separate such data. As a solution for this problem, using of soft margin was proposed. Soft margin is used instead of Eq. (1). The idea is to introduce a slack variable e that allows some instances to be misclassified i.e. to be on the wrong side of the hyperplane. This is defined by the following expression:

$$yi(w.\ xi+b) \geq 1 - \in i, \quad \in\ i \geq 0,\ 1 \leq i \leq n \qquad (2)$$

Finding this hyperplane is done by solving the following quadric programming problem:

$$\text{Min } \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \in i \qquad (3)$$

where C is the soft margin parameter. Increasing the value of parameter C asymptotically leads to the model with hard margin. Selecting appropriate value for this parameter has major influence on classification accuracy. Another problem with this model, when it comes to real world data, is the assumption that instances are linearly separable. In order to make SVM suitable for nonlinearly separable data kernel function is used instead of dot product. Theoretically, any function that satisfies Mercer's condition can be used as kernel function. In practice, usually Gaussian radial basis function (RBF), polynomial function and sigmoid function are used. Kernel function projects data into higher dimensional space in order to make it linearly separable. In this paper we used RBF as kernel function. RBF is defined by the next equation:

$$K(xi;xj) = \exp(-\gamma||xi - xj||^2) \quad (4)$$

where γ is the parameter of kernel function. This parameter has influence on the quality of classifier, so tuning the value of it is an important task. Too large value of γ will reduce benefits gained by introducing the kernel function and too small value will make decision boundary sensitive to the noise in training data.

Selecting optimal values for SVM's parameters is very important task. In most common technique for parameters tuning, grid search with cross validation, was described. Grid search builds models for different values of parameters and checks the accuracy of these models. Cross validation is used for determination of the model's accuracy. Training set is divided into v distinct subsets. For training v-1 subsets were used and the accuracy was checked on the remaining subset.

All subsets are used as test set once and the accuracy is the average value of v obtained accuracies. This method requires huge computational time and the search for optimal pair of values for (C; γ) is limited to predefined set of values.

2.1 EXPERIMENTS SETTING

In order to evaluate the efficiency of the suggested system based on SVM classifier, we investigated its performance for training and recognizing characters of database. To effectively train the model on more data so as to perfectly handle the variability of handwriting, the size of the training set is extended ten times by the elastic deformation technique suggested by Simard et al. Technical execution specifics of the of the selected system are given in the next subdivision.

- **Pre-processing:** The pre-processing phase where the database utilized in this experimental study does not require to be normalized. It is noticeable that a few fundamental pre-processing activities are vitally to be conducted throughout the database development. selected system are given in the next subdivision.
- **Parameters setting**: For the setting architecture, we must determine the optimal kernel parameter and penalty parameter of SVM. The value of the tradeoff parameter C and σ parameter in SVM are chosen empirically.
- **Feature extraction**: Features are the information extracted from the image of a word or character, and they are used to build classifiers for classification. The challenge is to determine which features are more suitable for classification.

2.2 EXPERIMENTS USING SVM MODEL

In our experiments, we investigated the performance of the SVM model for training and recognizing characters. For the setting architecture, we need to determine about SVM classifier essentially two parameters of the RBF kernel; Gamma (y) and C. We selected the pertinent parameters for SVM model on the basis of empirical tests.

We inaugurated an experimental study so that we can assess the suggested model. Our selection of parameters is on the basis of the criterion of the error classification rate on the train dataset. We also used the one-versus-all method with 66-way for the multi-class RBF kernel SVM as it provides a more valuable discrimination than the linear kernel. Yet, a less parameter than the polynomial kernel was utilized.

III. RESULTS AND DISCUSSION

3.1 Statistics and Machine Learning Toolbox

This example shows how to classify digits using HOG features and a multiclass SVM classifier. Object classification is an important task in many computer vision applications, including surveillance, automotive safety, and image retrieval. For example, in an automotive safety application, you may need to classify nearby objects as pedestrians or vehicles. Regardless of the type of object being classified, the basic procedure for creating an object classifier is:

- Acquire a labeled data set with images of the desired object.
- Partition the data set into a training set and a test set.
- Train the classifier using features extracted from the training set.
- Test the classifier using features extracted from the test set.

To illustrate, this example shows how to classify numerical digits using HOG (Histogram of Oriented Gradient) features [1] and a multiclass SVM (Support Vector Machine) classifier. This type of classification is often used in many Optical Character Recognition (OCR) applications

The example uses the fitcecoc function from the Statistics and Machine Learning Toolbox™ and the extract HOG Features function from the Computer Vision System Toolbox.

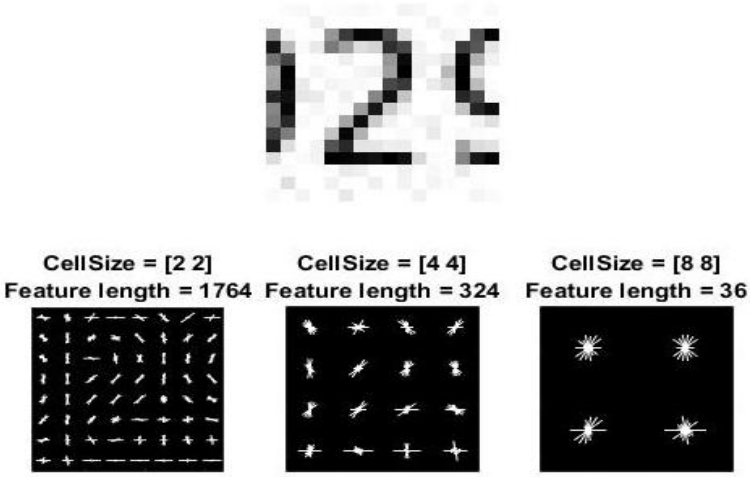

Fig 1: Extracting the hog features of digit to from an array for different block lengths

Fig 2: identifying the query digit after feature extraction using classifier

ISSN (O) 2454-9665
ISSN (P) 2454-0663

Fig 3: identifying the query digit in an array

3.2 Confused Matrix

digit	0	1	2	3	4	5	6	7	8	9
0	0.25	0.00	0.08	0.00	0.00	0.00	0.58	0.00	0.08	0.00
1	0.00	0.75	0.00	0.00	0.08	0.00	0.00	0.08	0.08	0.00
2	0.00	0.00	0.67	0.17	0.00	0.00	0.08	0.00	0.00	0.08
3	0.00	0.00	0.00	0.58	0.00	0.00	0.33	0.00	0.00	0.08
4	0.00	0.08	0.00	0.17	0.75	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.33	0.58	0.00	0.08	0.00
6	0.00	0.00	0.00	0.00	0.25	0.00	0.67	0.00	0.08	0.00
7	0.00	0.08	0.08	0.33	0.00	0.00	0.17	0.25	0.00	0.08
8	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.08	0.67	0.17
9	0.00	0.08	0.00	0.25	0.17	0.00	0.08	0.00	0.00	0.42

IV. CONCLUSION

In this paper we proposed a novel algorithm for handwritten digit recognition. The goal was to use simple feature set as input for support vector machine that was used for classification. This can be extending to Alphabets, engineering and scientific, mathematical symbols, currency symbols. The research studies presented in this paper reflect various aspects and models of handwritten character recognition systems. Different functions and working procedures have been used for recognizing a character in developing different models. Every model is thus based upon a unique technique of feature extraction. We compared our method with other methods proposed in literature and our proposed method obtained better accuracy with rather simple feature set. This establishes this approach as very robust and by using more complex features the results could be further improved.

REFERENCES

[1.] Network, Pp. 1-7, 2010. [2] Johann-adolf Muller, Frank Lemke, “Self Organizing Data Mining-an Intelligent Approach To Extract Knowledge From Data”, (Book) Pub. Libri, Hamburg, 2000.

[2.] Nunes, Alceu de S. Britto Jr, Celso A. A. Kaestner and Robert Sabourin, “An Optimized Hill Climbing Algorithm for Feature Subset Selection: Evaluation on Handwritten Character Recognition”, In Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, pp. 365-370, 2004.

ISSN (O) 2454-9665
ISSN (P) 2454-0663

[3.] Travieso, Jesus B. Alonso and Miguel A. Ferrer, “Combining different off-line handwritten character recognizers”, 15th International Conference on Intelligent Engineering Systems, pp. 315-318, June 2325, 2011.

[4.] Alihodzic, A. and Tuba, M. Improved hybridized bat algorithm for global numerical optimization. The 16th IEEE International Conference on Computer Modelling and Simulation, UKSim-AMSS 2014, pp. 57–62, 2014.

[5.] S.A Zeem, H. Ahmed, "Effective technique for the recognition of oftline Arabic handwritten words using hidden Markov models," TJDAR, vol. 16(4), pp. 399-4 12,20 13.

[6.] A Lawgali, M. Angelova, A Bouridane, "HACDB: Handwritten Arabic characters database for automatic character recognition," EUropean Workshop on Visual Information Processing (EUVIP), pp.255-259, 2013.

[7.] M. Elleuch, N. Tagougui, and M. Kherallah, "Towards Unsupervised Learning for Arabic Handwritten Recognition Using Deep Architectures," Neural Information Processing - 22nd International Conference, ICONIP 20 15, Istanbul, Turkey, part. (I), pp. 363-372, 20 15.