On Multifont Character Classification in Telugu

Venkat Rasagna, K. J. Jinesh, and C. V. Jawahar

International Institute of Information Technology,
Hyderabad 500032, INDIA.

Abstract. A major requirement in the design of robust OCRs is the invariance of feature extraction scheme with the popular fonts used in the print. Many statistical and structural features have been tried for character classification in the past. In this paper, we get motivated by the recent successes in object category recognition literature and use a spatial extension of the histogram of oriented gradients (HOG) for character classification. Our experiments are conducted on 1453950 Telugu character samples in 359 classes and 15 fonts. On this data set, we obtain an accuracy of 96-98% with an SVM classifier.

1 Character Classification

Large repositories of digitized books and manuscripts are emerging worldwide [1]. Providing content-level access to these collections require the conversion of these images to textual form with the help of Optical Character Recognizers (OCRs). Design of robust OCRs is still a challenging task for Indian scripts. The central module of an OCR is a recognizer which can generate a class label for an image component. Classification of isolated characters and thereby recognizing a complete document is still the fundamental problem in most of the Indian languages. The problem becomes further challenging in presence of diversity in input data (for example, variations in appearance with fonts and styles.)

Characters are first segmented out from page or word images. A set of appropriate features are then extracted for representing the character image. Features could be structural or statistical. Structural features are often considered to be sensitive to degradations in the print. A feature-vector representation of the image is then classified with the help of a classifier. Multilayer neural network, K nearest neighbour, support vector machines (SVM) etc. are popular for this classification task. Classification of Indian scripts is challenging due to (i) large number of classes (compared to Latin scripts) (ii) many pairs of very similar characters. (See Figure 1.)

In a recent work, Neeba and Jawahar [2] had looked into the success rates of character classification problem in an Indian context. Though their results are primarily on Malayalam, they are directly extendible to other scripts. They successfully solved the character classification problem (even in the presence of large number of classes) for limited number of fonts popularly seen in print. They had

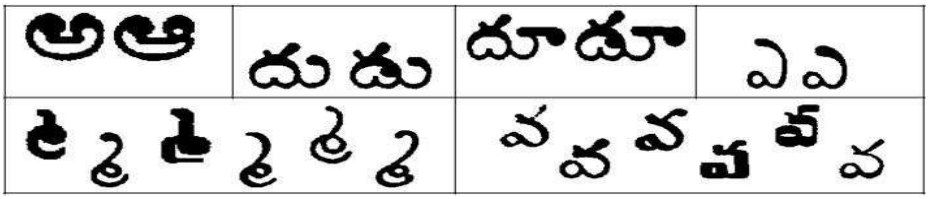

Fig. 1 Challenges in character classification of Telugu. First row shows similar character pairs from Telugu. Second row shows how the same character gets rendered in different fonts

argued that (i) multiclass classification solution can be made scalable by designing many pair-wise classifiers. (ii) use of large number of features (of the order of few hundred) makes the problems better separable and solvable with simple classifiers. (iii) when the dimensionality of the feature is made reasonably high, even the simple features like raw-pixels or PCA-projections provide satisfactory results.

A strong requirement of any robust character recognition system is the high classification accuracy, in the presence of multiple and diverse font sets. In this paper, we explore the problem of character classification in a multifont setting. Though our studies are for Telugu script, we believe that these results are also extendible to other languages. Our objective is to demonstrate the utility of the histogram of gradients (HoG) [3] sort of features for character classification. We also show that the linear SVM with DDAG sort of classifier fusion strategy provides equivalent results to an Intersection kernel SVMs. We validate our experimental results on 1453950 Telugu character samples in 359 classes and 15 fonts.

Telugu Script: Telugu is a south Indian language with its own script. Like most other Indian scripts, there are consonants, vowels and vowel-modifiers. In addition, there are also half consonants which get used in consonant clusters. Though the script is getting written from left to right in a sequential manner, many of these modifiers often gets distributed in a 1.5D (not purely left to right; they are also written top-to-bottom at places) manner. Compared to most other Indic scripts, Telugu has large number of basic characters/symbols. Many of them are also similar in appearance. This makes the character classification problem in Telugu very challenging. (See Figure1)

Telugu character recognition has been attempted in the past with various features. Negi et al. [4] used fringe maps as the feature. The method was tested on 2524 characters. Jawahar et al. [5] did a more thorough testing (of close to one Million samples) of the character classifiers but with limited font variations as well as degradations. They used PCA, LDA etc. as the possible feature extraction scheme for Telugu character classification.

2 Features and Classifiers

Recent years have witnessed significant attention in development of category level object recognition schemes with many interesting features. Histogram of oriented gradients (HOG), which was successfully used for detecting pedestrians [3], is one

of the prominent and popular features for capturing the visual data, when there are strong edges. Naive histogram representation looses the spatial information in the image. To address this, spatial pyramid matching was proposed [6]. Similar to [7], we also employ a feature vector which captures spatial information and histograms of oriented gradients.

We are motivated by the recent classification experiments in multifont data sets [8] and handwritten MNIST and USPS digit data sets [7]. Many of these studies are limited to handwritten digits. There have been many studies in this area (i) focusing on generalization of classification results to unknown fonts, and thereby solving the character 'category' recognition problem [8]. (ii) accurately solving the handwritten digit recognition with many machine learning concepts [7]. (iii) development of recognition algorithms with fewer training data or lesser resource usage.

Character/Symbol images are first normalized to a fixed size of 28 × 28 and histograms are constructed by aggregating the pixel responses within the cells of various sizes. Our cell sizes include 14 × 14, 7 × 7 and 4 × 4, with overlap of half the cell size. The histograms at different levels are multiplied by weights 1, 2 and 4. The entire sets of histograms are finally concatenated to form a single histogram. We refer this feature as SPHOG in the paper.

Based on the conclusions obtained in our earlier work on character classification [2], we use SVM classifiers. SVM classifiers are the state of the art in machine learning, to produce highly accurate and generalizable classifier. The classification rule for a sample x is

$$sign(\sum_{i=1}^{nSV} \alpha_i \kappa(x, s_i) + b)$$

where s_i s are the support vectors and $\kappa()$ is the kernel used for the classification. The Lagrangians α are used to weigh the individual kernel evaluations. The complexity of classification linearly increases with the number of support vectors. To make the classification fast, we can do the following [9]: (i) Use linear kernels instead of nonlinear ones. (ii) Store the weight vector instead of the support vectors (iii) Use binary representation as well as appropriate efficient data structures and (iv) Simplification of repeating support vectors in a decision making path consisting of multiple pair wise classifiers.

It was shown that the intersection kernel can be evaluated fast in many practical situations [10]. However, the comparisons are with that of complex kernels like RBF Kernel. Such classifiers are appropriate when the classes are not well separable. In the case of large class character recognition data set, most of the pair wise classifiers could be linearly separable. The overall classification accuracy reduces due to (i) cascading effects in the multiple classifier systems (ii) some of the pairs are difficult to separate with simple features. In this work, we compare the IKSVM with linear SVM and prefer to go for linear SVMs due to the computational and storage advantages of the linear SVM over IKSVM.

Based on the experimental results presented in the next section, we argue that (i) object category recognition features are useful for the character recognition especially in presence of multiple fonts. (ii) linear SVMs perform very similar to

IKSVM for most of the character classification tasks. (iii) Use of SPHOG sort of features can successfully solve the multifont character classification problem in Indic scripts.

3 Results and Discussions

We start by investigating the deterioration of performance with the number of fonts. For this purpose, we collected a character level groundtruthed Telugu data set in fifteen fonts. Number of classes which is common to all these fonts is 359. We first investigate the utility of raw pixels as a feature with a linear SVM classifier. For this experiment, we consider only the first 100 classes.

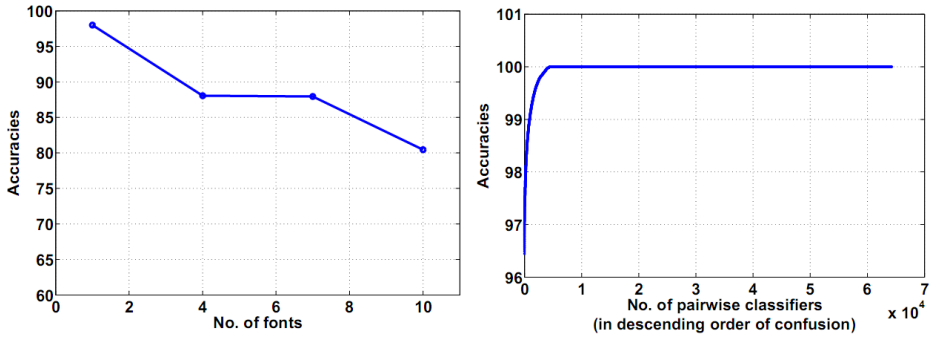

Fig. 2 (a) Variations of accuracies with increase in fonts (b) Accuracy and confused pair wise classifiers

Results of the variation of accuracy are plotted in Figure 2(a). It may be seen that with only one or limited fonts, the accuracies are acceptable, however, with the number of fonts increasing, the accuracy comes down significantly.

We now quantitatively show the results on a 100 class subset of the Telugu characters in 15 different and popular fonts. We show that the naive features, like raw pixels or PCA, are unable to address the significant font variation present in the dataset.

Table 1. Comparative results on a smaller set of Telugu Multifont Data Set

Classifier	RawPixels	PCA	SPHOG	PCA-SPHOG
	d=784	d=500	d=2172	d=500
LSVM(OneVs All)	91.91	90.19	98.10	96.70
LSVM(DDAG)	94.19	93.84	97.25	97.51
IKSVM(OneVs All)	92.26	96.369	98.71	98.39

Table 1 compares the performance of the four features in presence of two different SVM classifiers – Linear SVM(LSVM) and Intersection Kernel SVM (IKSVM). Linear SVMs are also implemented as One Vs All as well as DDAG [2]. It may be noted that the raw image features are not able to perform well when the

number of fonts increases. This is expected because of the variation in the styles and shapes of the associated glyphs. It is surprising that the PCA, which was performing reasonably well for limited number of fonts [2] is also not able to scale well for the multifont situation. A graph which shows the variation of the number of eigen vectors (principal components) selected Vs the accuracy obtained is shown in Figure 3 (a). The plot of magnitudes of eigen values of the covariance matrix (used in PCA) is shown in Figure 3(b).

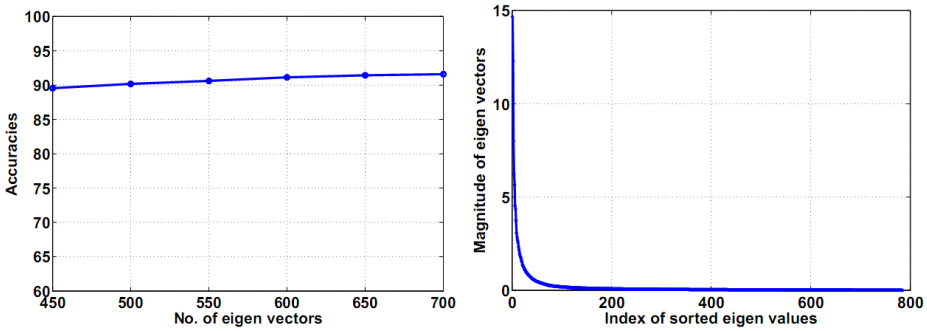

Fig 3. (a) Accuracy and number of eigen vectors (b) Eigen vectors and their magnitude

These graphs explain that with an increase in the number of PCs the accuracy monotonically improves. However, the accuracy saturates at a level 91%, which is not an acceptable level of accuracy, we are looking for an OCR. On the contrary, the SPHOG features are performing consistently well for the large font data set, as can be seen in Table 1. PCA has been applied on the SPHOG feature as the dimensionality of the feature is large. Even with 23% of the SPHOG feature vector, accuracy close to the SPHOG result has been obtained.

In short, it is clear from the experiments conducted on a 100 class data set, that SVM classifier with SPHOG and PCA-SPHOG features provide the most accurate classifiers. We have extended the results obtained for a full Telugu character set consisting of 359 classes. They summarize as follows:

Obtaining an accuracy of 96.4 on a truly challenging multifont data is significant. However, we would like to see the possibility of enhancing the accuracy further. For this, we analyze the confusions associated with all the pair wise classifications. As

Table 2. Classification accuracy: No of classes = 359, No of samples = 1453950

Raw pixels with Linear SVM classifier results in an accuracy	81.05
SPHOG with Linear SVM classifier results in an accuracy	96.41
PCA-SPHOG with Linear SVM classifier results in an accuracy	92.95

can be seen from Figure 2(b), the errors are associated with only certain pairs. In Figure 2(b), we plot the cumulative accuracy over all pair wise confusions. If we can address the errors in these pairs with the help of an additional classifier (we call them as post-processing classifier), we can enhance the accuracy. We propose to use an

RBF based SVM classifier for this purpose. The detailed design and analysis of the post-processing classifier are beyond the scope of this paper. It is observed that with the help of a few robust post-processing classifiers, one can enhance the accuracy to 98%.

4 Conclusions

We show that high classification accuracies can be obtained for character classification problem with the help of SPHOG-SVM combination. Left out confusions is associated only to a small percentage of the classifier and a post-processing classifier with an uncorrelated feature set can successfully boost the overall classification performance.

References

1. Pramod Sankar K, Ambati V, Pratha L and Jawahar C.V: Digitizing a Million Books: Challenges for Document Analysis. In: Proceedings of Document Analysis Systems VII (DAS), Springer-Verlag New York Inc (2006) 425–436
2. Neeba N.V and Jawahar C.V: Empirical evaluation of character classification schemes. In: Seventh International Conference on Advances in Pattern Recognition (ICAPR), IEEE (2009) 310–313
3. Dalal N and Triggs B: Histograms of oriented gradients for human detection. In: Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. IEEE (2005) 886–893
4. Negi A, Bhagvati C and Krishna B: An OCR system for telugu. In: Proceedings of Sixth International Conference on Document Analysis and Recognition (ICDAR), 2001, IEEE (2002) 1110–1114
5. Jawahar C.V, Kumar P, Kiran R and others: A bilingual OCR for hindi-telugu documents and its applications. In: Proceedings of Seventh International Conference on Document Analysis and Recognition (ICDAR), IEEE (2003) 408–412
6. Lazebnik S, Schmid C, and Ponce J: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2, IEEE (2006) 2169–2178
7. Maji S and Malik J: Fast and accurate digit classification. Technical Report UCB/EECS-2009-159, EECS Department, University of California, Berkeley (2009)
8. De Campos T.E, Babu B.R and Varma M: Character recognition in natural images. In: Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP), INSTICC (2009) 273–280
9. Ilayaraja P, Neeba N.V and Jawahar C.V: Efficient implementation of SVM for large class problems. In: 19th International Conference on Pattern Recognition (ICPR), IEEE (2009) 1–4
10. Maji S, Berg A.C and Malik J: Classification using intersection kernel support vector machines is efficient. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2008) 1–8