CHARACTER RECOGNITION IN NATURAL IMAGES

Teófilo E. de Campos
Xerox Research Centre Europe, 6 chemin de Maupertuis, 38240 Meylan, France

Bodla Rakesh Babu
International Institute of Information Technology, Gachibowli, Hyderabad 500 032 India

Manik Varma
Microsoft Research India, "Scientia" 196/36 2nd Main, Sadashivnagar, Bangalore 560 080 India

Keywords: Object recognition, Camera-based character recognition, Latin characters, Digits, Kannada characters, Off-line handwritten character recognition.

Abstract: This paper tackles the problem of recognizing characters in images of natural scenes. In particular, we focus on recognizing characters in situations that would traditionally not be handled well by OCR techniques. We present an annotated database of images containing English and Kannada characters. The database comprises of images of street scenes taken in Bangalore, India using a standard camera. The problem is addressed in an object cateogorization framework based on a bag-of-visual-words representation. We assess the performance of various features based on nearest neighbour and SVM classification. It is demonstrated that the performance of the proposed method, using as few as 15 training images, can be far superior to that of commercial OCR systems. Furthermore, the method can benefit from synthetically generated training data obviating the need for expensive data collection and annotation.

1 INTRODUCTION

This paper presents work towards automatic reading of text in natural scenes. In particular, our focus is on the *recognition* of individual characters in such scenes. Figures 1, 2 and 3 highlight why this can be a hard task. Even if the problems of clutter and text segmentation were to be ignored for the moment, the following sources of variability still need to be accounted for: (a) font style and thickness; (b) background as well as foreground color and texture; (c) camera position which can introduce geometric distortions; (d) illumination and (e) image resolution. All these factors combine to give the problem a flavor of object recognition rather than optical character recognition or handwriting recognition. In fact, OCR techniques can not be applied out of the box precisely due to these factors. Furthermore, viable OCR systems have been developed for only a few languages and most Indic languages are still beyond the pale of current OCR techniques.

Many problems need to be solved in order to read text in natural images including text localization, character and word segmentation, recognition, integration of language models and context, *etc*. Our focus, in this paper, is on the basic character recognition aspect of the problem (see Figures 2, 3, 5 and 6). We introduce a database of images containing English and Kannada text[1]. In order to assess the feasibility of posing the problem as an object recognition task, we benchmark the performance of various features based on a bag-of-visual-words representation. The results indicate that even the isolated character recognition task is challenging. The number of classes can be moderate (62 for English) to large (657 for Kannada) with very little inter-class variation as highlighted by Figures 2 and 3. This problem is particularly acute for Kannada where two characters in the alphabet can differ just by the placement of a single dot like structure. Furthermore, while training data is readily available for some characters others might occur very in-

[1] Available at http://research.microsoft.com/˜manik/

E. de Campos T., Rakesh Babu B. and Varma M.
CHARACTER RECOGNITION IN NATURAL IMAGES.
DOI: 10.5220/0001770102730280
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (VISIGRAPP 2009)*, page
ISBN: 978-989-8111-69-2

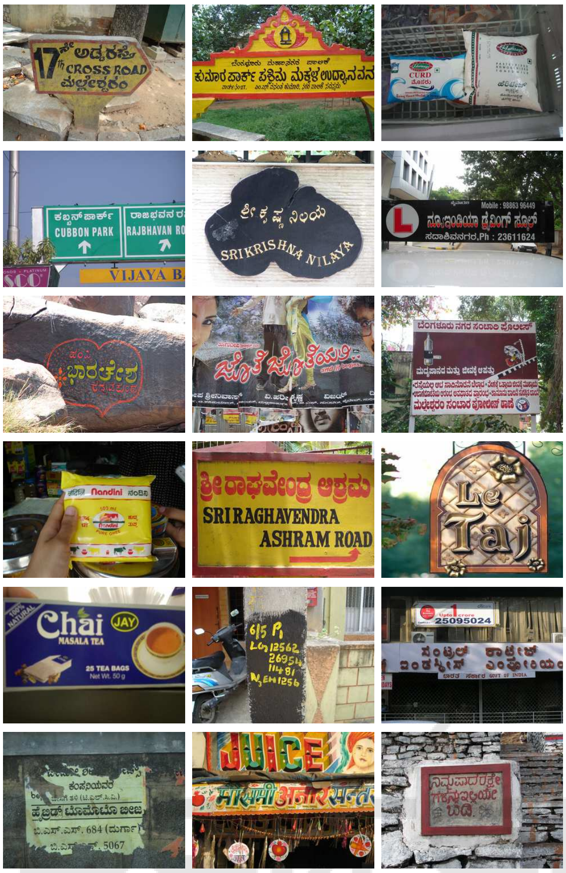

Figure 1: Sample source images in our data set.

frequently in natural scenes. We therefore investigate whether surrogate training data, either in the form of font generated characters or hand-printed characters, can be used to bolster recognition in such a scenario. We also present baseline recognition results on the font and hand-printed character databases to contrast the difference in performance when reading text in natural images.

2 RELATED WORK

The task of character recognition in natural scenes is related to problems considered in camera-based document analysis and recognition. Most of the work in this field is based on locating and rectifying the text areas (e.g. (Kumar et al., 2007), (Krempp et al., 2002), (Clark and Mirmehdi, 2002) and (Brown et al., 2007)), followed by the application of OCR techniques (Kise and Doermann, 2007). Such approaches are therefore limited to scenarios where OCR works well. Furthermore, even the rectification step is not directly applicable to our problem, as it is based on the detection of printed document edges or assumes that the image is dominated by text.

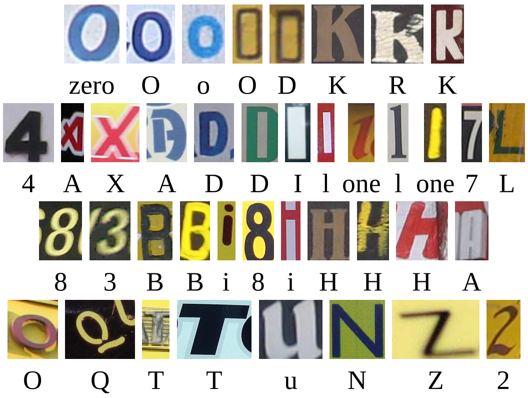

Figure 2: Examples of high visual similarity between samples of different classes caused mainly by the lack of visual context.

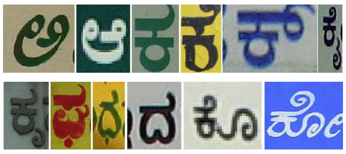

Figure 3: A small set of Kannada characters, all from different classes. Note that vowels often change a small portion of the characters, or add disconnected components to the character.

Methods for off-line recognition of hand-printed characters (Plamondon and Srihari, 2000), (Pal et al., 2007) have successfully tackled the problem of intra-class variation due to differing writing styles. However, such approaches typically consider only a limited number of appearance classes, not dealing with variations in foreground/background color and texture.

For natural scenes, some researchers have designed systems that integrate text detection, segmentation and recognition in a single framework to accommodate contextual relationships. For instance, (Tu et al., 2005) used insights from natural language processing and present a Markov chain framework for parsing images. (Jin and Geman, 2006) introduced composition machines for constructing probabilistic hierarchical image models which accommodate contextual relationships. This approach allows re-usability of parts among multiple entities and non-Markovian distributions. (Weinman and Learned Miller, 2006) proposed a method that fuses image features and language information (such as bi-grams and letter case) in a single model and integrates dissimilarity information between character images.

Simpler recognition pipelines based on classifying raw images have been widely explored for digits recognition (see (le Cun et al., 1998), (Zhang et al., 2006) and other works on the MNIST and USPS datasets). Another approach is based on modeling this as a shape matching problem (e.g. (Belongie et al., 2002)): several shape descriptors are detected and extracted and point-by-point matching is computed between pairs of images.

3 DATA SETS

Our focus is on recognizing characters in images of natural scenes. Towards this end, we compiled a database of English and Kannada characters taken from images of street scenes in Bangalore, India. However, gathering and annotating a large number of images for training can be expensive and time consuming. Therefore, in order to provide complementary training data, we also acquired a database of hand-printed characters and another of characters generated by computer fonts.

For English, we treat upper and lower case characters separately and include digits to get a total of 62 classes. Kannada does not differentiate between upper and lower case characters. It has 49 basic characters in its alpha-syllabary, but consonants and vowels can combine to give more than 600 visually distinct classes.

3.1 Natural Images Data Set

We photographed a set of 1922 images, mostly of sign boards, hoardings and advertisements but we also included a few images of products in supermarkets and shops. Some of these original images are shown in Figure 1.

Individual characters were manually segmented from these images. We experimented with two types of segmentations: rectangular bounding boxes and finer polygonal segments as shown in Figure 4. For the types of features investigated in this paper, it turned out that polygonal segmentation masks presented almost no advantage over bounding boxes. Therefore, all the results presented in Section 5 are using the bounding box segmentations.

Our English dataset has 12503 characters, of which 4798 were labeled as bad images due to excessive occlusion, low resolution or noise. For our experiments, we used the remaining 7705 character images. Similarly, for Kannada, a total of 4194 characters were extracted out of which only 3345 were used. Figures 5 and 6 show examples of the extracted characters. These datasets will be referred to as the *Img* datasets.

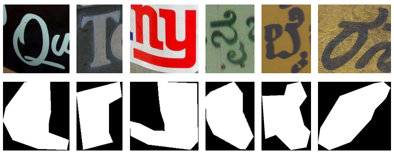

Figure 4: Sample characters and their segmentation masks.

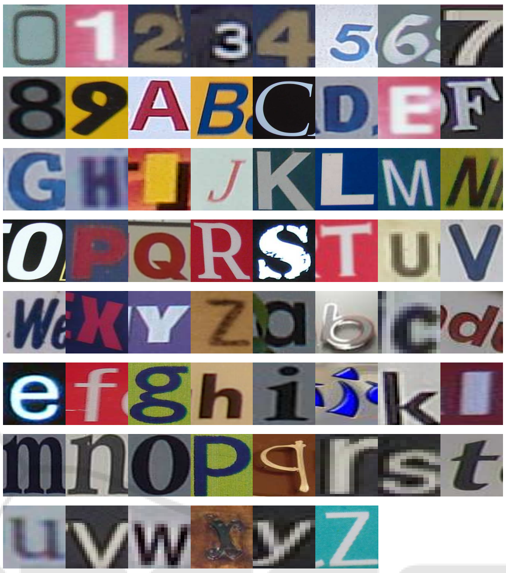

Figure 5: Sample characters of the English *Img* data set.

3.2 Font and Hand-printed Datasets

The hand-printed data set (*Hnd*) was captured using a tablet PC with the pen thickness set to match the average thickness found in hand painted information boards. For English, a total of 3410 characters were generated by 55 volunteers. For Kannada, a total of 16425 characters were generated by 25 volunteers. Some sample images are shown in Figure 7.

The font dataset was synthesized only for English characters. We tried 254 different fonts in 4 styles (normal, bold, italic and bold+italic) to generate a total of 62992 characters. This dataset will be referred to as the *Fnt* dataset.

4 FEATURE EXTRACTION AND REPRESENTATION

Bag-of-visual-words is a popular technique for representing image content for object category recognition. The idea is to represent objects as histograms of

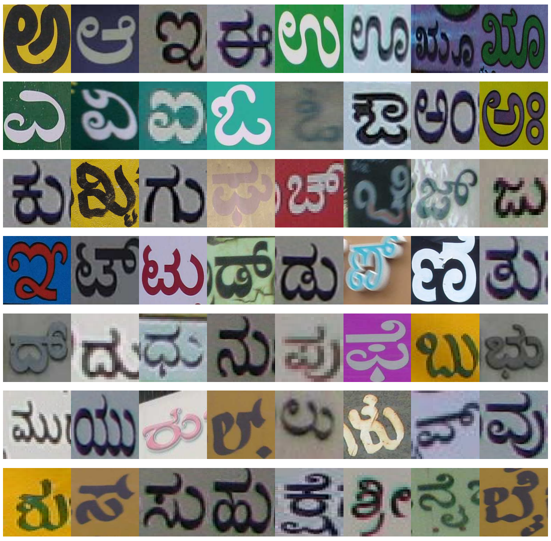

Figure 6: Sample characters of the Kannada *Img* data set.

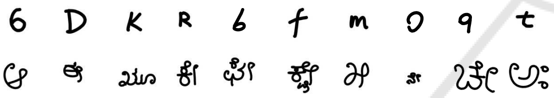

Figure 7: Sample hand-printed characters of the English and Kannada data sets.

feature counts. This representation quantizes the continuous high-dimensional space of image features to a manageable vocabulary of "visual words". This is achieved, for instance, by grouping the low-level features collected from an image corpus into a specified number of clusters using an unsupervised algorithm such as *K-Means* (for other methods of generating the vocabulary see (Jurie and Triggs, 2005)). One can then map each feature extracted from an image onto its closest visual word and represent the image by a histogram over the vocabulary of visual words.

We learn a set of visual words per class and aggregate them across classes to form the vocabulary. In our experiments, we learned 5 visual words per class for English leading to a vocabulary of size 310. For Kannada, we learn 3 words per class, resulting in a vocabulary of 1971 words.

4.1 Features

We evaluated six different types of local features. Not only did we try out shape and edge based features, such as Shape Context, Geometric Blur and SIFT, but also features used for representing texture, such as filter responses, patches and Spin Images, since these were found to work well in (Weinman and Learned Miller, 2006). We explored the most commonly used parameters and feature detection methods employed for each descriptor, with a little tuning, as described below.

Shape Contexts (SC) (Belongie et al., 2002) is a descriptor for point sets and binary images. We sample points using the Sobel edge detector. The descriptor is a log-polar histogram, which gives a $\theta \times n$ vector, where θ is the angular resolution and n is the radial resolution. We used $\theta = 15$ and $r = 4$.

Geometric Blur (GB) (Berg et al., 2005) is a feature extractor with a sampling method similar to that of SC, but instead of histogramming points, the region around an interest point is blurred according to the distance from this point. For each region, the edge orientations are counted with a different blur factor. This soothes the problem of hard quantization and allows its application to gray scale images.

Scale Invariant Feature Transform (SIFT) (Lowe, 1999) are extracted on points located by the Harris Hessian-Laplace detector, which gives affine transform parameters. The feature descriptor is computed as a set of orientation histograms on (4×4) pixel neighborhoods. The orientation histograms are relative to the key-point orientation. The histograms contain 8 bins each, and each descriptor contains a 4×4 array of 16 histograms around the key-point. This leads to feature vector with 128 elements.

Spin image (Lazebnik et al., 2005), (Johnson and Herbert, 1999) is a two-dimensional histogram encoding the distribution of image brightness values in the neighborhood of a particular reference point. The two dimensions of the histogram are d, distance from the center point, and i, the intensity value. We used 11 bins for distance and 5 for intensity value, resulting in 55-dimensional descriptors. The same interest point locations used for SIFT were used for spin images.

Maximum Response of filters (MR8) (Varma and Zisserman, 2002) is a texture descriptor based on a set of 38 filters but only 8 responses. This filter is extracted densely, giving a large set of 8D vectors.

Patch descriptor (PCH) (Varma and Zisserman, 2003) is the simplest dense feature extraction method. For each position, the raw $n \times n$ pixel values are vectorized, generating an n^2 descriptor. We used 5×5 patches.

5 EXPERIMENTS AND RESULTS

This section describes baseline experiments using three classification schemes: (a) nearest neighbor (NN) classification using χ^2 statistic as a similarity measure; (b) support vector machines (SVM); and (c) multiple kernel learning (MKL). Additionally, we show results obtained by the commercial OCR system

ABBYY FineReader 8.0[2]. For an additional benchmark, we provide results obtained with the dataset of the ICDAR Robust Reading competition 2003[3]. This set contains 11615 images of characters used in English. The images are more challenging than our English *Img* dataset and it has some limitations, such as the fact that only a few samples are available for some of the characters.

Most of our experiments were done on our *English Img* characters dataset. It is demonstrated that the performance of MKL using only 15 training images is nearly 25% better than that of ABBYY FineReader, a commercial OCR system. Also, when classifying the *Img* test set, if appropriate features such as Geometric Blur, are used, then a NN classifier trained on the synthetic *Fnt* dataset is as good as the NN classifier trained on an equal number of *Img* samples. Furthermore, since synthetic *Fnt* data is easy to generate, an NN classifier trained on a large *Fnt* training set can perform nearly as well as MKL trained on 15 *Img* samples per class. This opens up the possibility of improving classification accuracy without having to acquire expensive *Img* training data.

5.1 English Data Sets

5.1.1 Homogeneous Sets

This subsection shows results obtained by training and testing on samples from the same type – i.e. *Fnt*, *Hnd* and *Img* data. While our focus is on the *Img* dataset, training and testing on *Fnt* or *Hnd* provide useful baselines for comparison. For some classes, the number of available *Img* samples was just above 30, so we chose to keep the experiment sets balanced. The test set size was fixed to 15 samples per class for each of the three databases. The number of training samples per class was varied between 5 and 15 for the *Img* dataset. For *Fnt* and *Hnd*, we used 1001 and 40 samples per class respectively for training. Multiple training and testing splits were generated at random and the results averaged. Table 1 shows the results obtained with training sets of 15 samples per class. The performance of GB and SC is significantly better than all the other features. Also, there can be more than a 20% drop in performance when moving from training and testing on *Fnt* or *Hnd* to training and testing on *Img*. This indicates how much more difficult recognizing characters in natural images can be.

Table 1: Nearest neighbor classification results (%) obtained by different features on the English data sets. These were obtained with 15 training and 15 testing samples per class. For comparison, the results with the commercial software ABBYY are also shown. The bottom row indicates how many sets of training samples were taken per class to estimate mean and standard deviation of the classification results.

Feature	Fonts	Hand	Images
GB	69.71 ± 0.64	65.40 ± 0.58	47.09
SC	64.83 ± 0.60	67.57 ± 1.40	34.41
SIFT	46.94 ± 0.71	44.16 ± 0.79	20.75
Patches	44.93 ± 0.65	69.41 ± 0.72	21.40
SPIN	28.75 ± 0.76	26.32 ± 0.42	11.83
MR8	30.71 ± 0.67	25.33 ± 0.63	10.43
ABBYY	66.05 ± 0.00	–	30.77
# train splits	10	5	1

Table 2: Classification results (%) obtained with 1-vs-All SVM and with MKL (combining all the features) for the *Img* set with 15 training samples per class.

GB	52.58
SC	35.48
SIFT	21.40
Patches	21.29
SPIN	13.66
MR8	11.18
MKL	55.26

The features were also evaluated using SVMs with RBF kernels for the *Img* dataset, leading to the results shown in table 2. An additional experiment was performed with the multiple kernel learning (MKL) method of (Varma and Ray, 2007), which gave state-of-the-art results in the Caltech256 challenge. This resulted in an accuracy of 55.26% using 15 training samples per class.

As can be seen from these experiments, it is possible to surpass the performance of ABBYY, a state-of-the-art commercial OCR system, using 15 training images even on the synthetic *Fnt* dataset. For the more difficult *Img* dataset the difference in performance between MKL and ABBYY is nearly 25% indicating that OCR is not suitable for this task. Nevertheless, given that the performance using MKL is only 55%, there is still tremendous scope for improvement in the object recognition framework.

We also performed experiments with the ICDAR dataset, obtaining the results in Table 3. Due to the limitations of this dataset, we fixed the training set size to 5 samples per class and evaluated it in comparison to our dataset. As can be seen, the ICDAR results are worse than the *Img* results indicating that this might be an even tougher database. If we train on *Img* and test on ICDAR then the results can improve as more training data is added (see Table 4).

[2] http://www.abbyy.com

[3] http://algoval.essex.ac.uk/icdar

Table 3: Nearest neighbor results obtained with 5 training samples per class for some of the features. Here we compare our English *Img* dataset and with the ICDAR dataset.

Feature	*Img*	ICDAR
GB	36.9 ± 1.0	27.81
SC	26.1 ± 1.6	18.32
PCH	13.7 ± 1.4	9.67
MR8	6.9 ± 0.7	5.48

Table 4: Nearest neighbor results obtained by training with English *Img* and testing with the ICDAR dataset – using 15 training samples per class and using the whole *Img* set for training.

Tr. Spls.	15/class	all
GB	32.72	40.97
SC	27.90	34.51

5.1.2 Hybrid Sets

In this subsection we show experiments with hybrid sets, where we train on data from the *Fnt* and *Hnd* datasets and test on the same 15 images per class from the *Img* test set used in the previous experiments. The results are shown in Table 5 and indicate that for features such as Geometric Blur, training on easily available synthetic fonts is as good as training on original *Img* data. However, the performance obtained by training on *Hnd* is poor.

Table 5: Nearest neighbor results with mixed data: testing the recognition of natural images using training data from fonts and hand-printed sets, both with 15 training samples per class. These results should be compared with the *Img* column of Table 1.

Feature	Training on *Fnt*	Training on *Hnd*
GB	47.16 ± 0.82	22.95 ± 0.64
SC	32.39 ± 1.39	26.82 ± 1.67
SIFT	9.86 ± 0.91	4.02 ± 0.52
Patches	5.65 ± 0.69	1.83 ± 0.44
SPIN	2.88 ± 0.68	2.71 ± 0.33
MR8	1.87 ± 0.60	1.61 ± 0.11
# test splits	10	5

To aid visualization of the results, Figure 8 shows results of the experiments described above, separating panels for the top three methods: Geometric Blur, Shape Contexts and Patches. There is one curve for each type of experiment, where *FntImg* indicates training with *Fnt* and testing with *Img*, and *HndImg* indicates training with *Hnd* and testing with *Img*. The other curves show results by training and testing with the same kind of set (*Fnt*, *Hnd* and *Img*). Note that, for Geometric Blur, the NN performance when trained on *Fnt* and tested on *Img* is actually better than NN performance when trained and tested on *Img*.

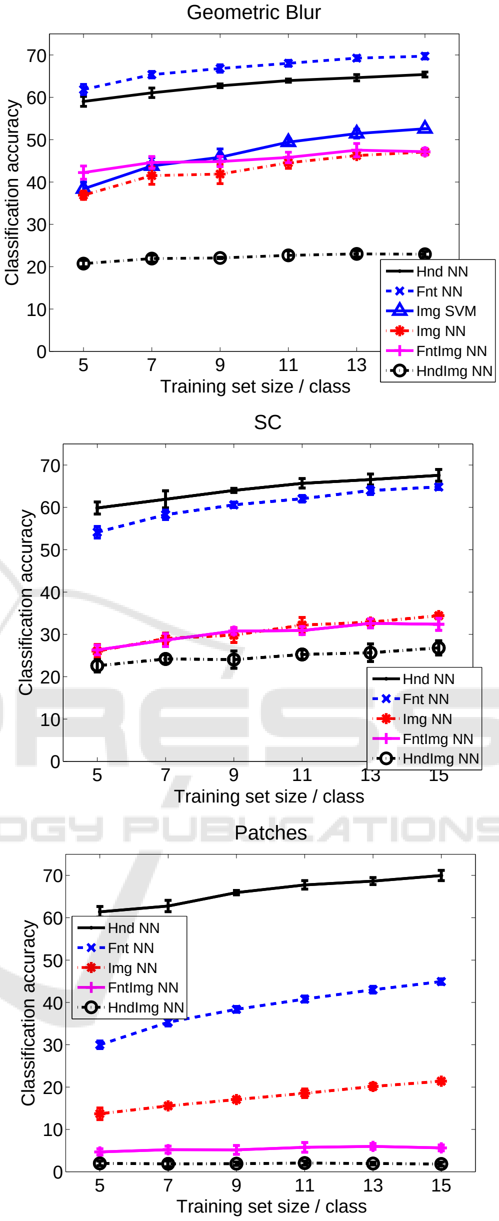

Figure 8: Classification results for the English datasets with GB, SC and Patch features.

In a practical situation, all the available fonts or hand-printed data could be used to classify images. Table 6 shows the results obtained by training with all available samples from *Fnt* and *Hnd* and testing with the same *Img* test sets of 15 samples per class described above. Note that for GB and SC, the NN classification results obtained by training with the entire *Fnt* training set were better than those obtained

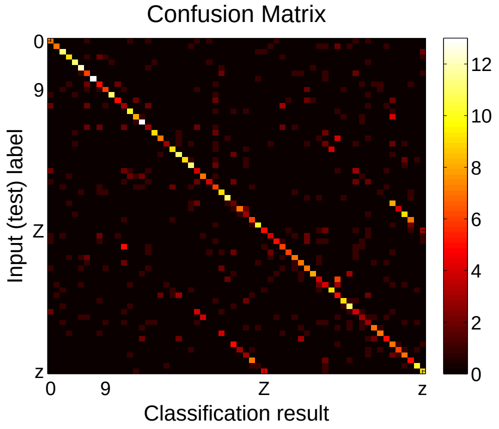

Figure 9: Confusion matrix of MKL for training and testing on *Img* with 15 training samples per class.

by training with 15 *Img* samples per class and, in fact, were nearly as good as MKL. This demonstrates the generalization power of these descriptors and validates the possibility of cheaply generating large sized synthetic sets and using them for training.

Table 6: Classification results (%) obtained with the same testing set as in table 5, but here the whole sets of synthetic fonts and hand-printed characters are used for training, i.e., 1016 and 55 samples per class, respectively.

Feature	Training on *Fnt*	Training on *Hnd*
GB	54.30	24.62
SC	44.84	31.08
SIFT	11.08	3.12
Patches	7.85	1.72
SPIN	3.44	2.47
MR8	1.94	1.51
Training set size	1016	55

Figure 6 shows the confusion matrix obtained for MKL when trained and tested on 15 *Img* samples per class. One can notice two patterns of high values in parallel to the diagonal line. These patterns show that, for many characters, there is a confusion between lower case and upper case. If we classify characters in a case insensitive manner then the accuracy goes up by nearly 10%.

5.2 Kannada Data Sets

The *Img* dataset of Kannada characters was annotated per symbol, which includes characters and syllables, resulting in a set of 990 classes. Since some of these classes occur rarely in our dataset, we did not perform experiments training and testing with *Img*. Instead, we only performed experiments on training with *Hnd* characters and testing with *Img*. We selected a subset of 657 classes which coincides with the classes acquired for the *Hnd* dataset.

Table 7: Nearest neighbor results (%) for the Kannada datasets: (i) training with 12 *Hnd* and testing with 13 *Hnd* samples, and (ii) training with all *Hnd* and testing with all *Img* samples.

Ftr.	Trn/tst on *Hnd*	Trn on *Hnd*, tst on *Img*
GB	17.74	2.77
SC	29.88	3.49
SIFT	7.63	0.30
Patches	22.98	0.12

Table 7 shows baseline results. For these experiments, random guess would have a 0.15% accuracy.

6 CONCLUSIONS

In this paper, we tackled the problem of recognizing characters in images of natural scenes. We introduced a database of images of street scenes taken in Bangalore, India and showed that even commercial OCR systems are not well suited for reading text in such images. Working in an object categorization framework, we were able to improve character recognition accuracy by 25% over an OCR based system. The best result on the *English Img* database was 55.26% and was obtained by the multiple kernel learning (MKL) method of (Varma and Ray, 2007) when trained using 15 *Img* samples per class. This could be improved further if we were not to be case sensitive. Nevertheless, significant improvements need to be made before an acceptable performance level can be reached.

Obtaining and annotating natural images for training purposes can be expensive and time consuming. We therefore explored the possibility of training on hand-printed and synthetically generated font data. The results obtained by training on hand-printed characters were not encouraging. This could be due to the limited variability amongst the writing styles that we were able to capture as well as the relatively small size of the training set. On the other hand, using synthetically generated fonts, the performance of nearest neighbor classification based on Geometric Blur features was extremely good. For equivalent size training sets, training on fonts using a NN classifier could actually be better than training on the natural images themselves. The performance obtained when training on all the font data was nearly as good as that obtained using MKL when trained on 15 natural image samples per class. This opens up the possibility of harvesting synthetically generated data and using it for training.

As regards features, the shape based features, Geometric Blur and Shape Context, consistently outperformed SIFT as well as the appearance based features. This is not surprising since the appearance of a character in natural images can vary a lot but the shape remains somewhat consistent.

We also presented preliminary results on recognizing Kannada characters but the problem appears to be extremely challenging and could perhaps benefit from a compositional or hierarchical approach given the large number of visually distinct classes.

ACKNOWLEDGEMENTS

We would like to acknowledge the help of several volunteers who annotated the datasets presented in this paper. In particular, we would like to thank Arun, Kavya, Ranjeetha, Riaz and Yuvraj. We would also like to thank Richa Singh and Gopal Srinivasa for developing tools for annotation.

REFERENCES

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Berg, A. C., Berg, T. L., and Malik, J. (2005). Shape matching and object recognition using low distortion correspondence. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, San Diego CA, June 20-25*.

Brown, M. S., Sun, M., Yang, R., yun, L., and Seales, W. B. (2007). Restoring 2d content from distorted documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Clark, P. and Mirmehdi, M. (2002). Recognising text in real scenes. *International Journal on Document Analysis and Recognition*, 4:243–257.

Jin, Y. and Geman, S. (2006). Context and hierarchy in a probabilistic image model. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, New York NY, June 17-22*.

Johnson, A. E. and Herbert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449.

Jurie, F. and Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*.

Kise, K. and Doermann, D. S., editors (2007). *Proceedings of the Second International Workshop on Camera-based Document Analysis and Recognition CBDAR*, Curitiba, Brazil. http://www.imlab.jp/cbdar2007/.

Krempp, A., Geman, D., and Amit, Y. (2002). Sequential learning of reusable parts for object detection. Technical report, Computer Science Department, Johns Hopkins University.

Kumar, S., Gupta, R., Khanna, N., Chaudhury, S., and Joshi, S. (2007). Text extraction and document image segmentation using matched wavelets and mrf model. *IEEE Transactions on Image Processing*, 16(8):2117–2128.

Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278.

le Cun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lowe, D. G. (1999). Object recognition from local scale-invariante features. In *Proc 7th Int Conf on Computer Vision, Corfu, Greece*.

Pal, U., Sharma, N., Wakabayashi, T., and Kimura, F. (2007). Off-line handwritten character recognition of devnagari script. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 496–500, Curitiba, PR, Brazil. IEEE.

Plamondon, R. and Srihari, S. N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84.

Tu, Z., Chen, X., Yuille, A. L., and Zhu, S. C. (2005). Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision, Marr Prize Issue*.

Varma, M. and Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil.

Varma, M. and Zisserman, A. (2002). Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 3, pages 255–271. Springer-Verlag.

Varma, M. and Zisserman, A. (2003). Texture classification: Are filter banks necessary? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Weinman, J. J. and Learned Miller, E. (2006). Improving recognition of novel input with similarity. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, New York NY, June 17-22*.

Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, New York NY, June 17-22*.