Recognizing Handwritten Characters with Local Descriptors and Bags of Visual Words

Olarik Surinta[1(✉)], Mahir F. Karaaba[1], Tusar K. Mishra[2], Lambert R.B. Schomaker[1], and Marco A. Wiering[1]

[1] Institute of Artificial Intelligence and Cognitive Engineering (ALICE), University of Groningen, Nijenborgh 9, Groningen, The Netherlands
{o.surinta,m.f.karaaba,l.r.b.schomaker,m.a.wiering}@rug.nl
[2] Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Odisha, India
tusar.k.mishra@gmail.com

Abstract. In this paper we propose the use of several feature extraction methods, which have been shown before to perform well for object recognition, for recognizing handwritten characters. These methods are the histogram of oriented gradients (HOG), a bag of visual words using pixel intensity information (BOW), and a bag of visual words using extracted HOG features (HOG-BOW). These feature extraction algorithms are compared to other well-known techniques: principal component analysis, the discrete cosine transform, and the direct use of pixel intensities. The extracted features are given to three different types of support vector machines for classification, namely a linear SVM, an SVM with the RBF kernel, and a linear SVM using L2-regularization. We have evaluated the six different feature descriptors and three SVM classifiers on three different handwritten character datasets: Bangla, Odia and MNIST. The results show that the HOG-BOW, BOW and HOG method significantly outperform the other methods. The HOG-BOW method performs best with the L2-regularized SVM and obtains very high recognition accuracies on all three datasets.

Keywords: Handwritten character recognition · Feature extraction · Bag of visual words · Histogram of oriented gradients · Support vector machines

1 Introduction

In this paper we propose the use of several feature descriptors for handwritten character recognition. Obtaining high accuracies on handwritten character datasets can be difficult due to several factors such as background noise, many different types of handwriting, and an insufficient amount of training examples. Our motivation for this study is to obtain high recognition accuracies for different datasets even when there are not many examples in these datasets. There are currently many character recognition systems which have been tested on the

© Springer International Publishing Switzerland 2015
L. Iliadis and C. Jayne (Eds.): EANN 2015, CCIS 517, pp. 255–264, 2015.
DOI: 10.1007/978-3-319-23983-5_24

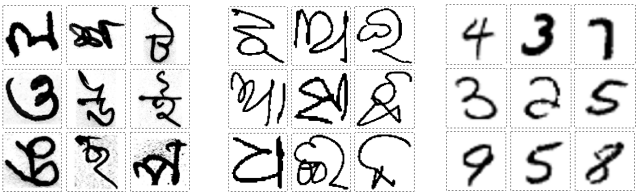
standard benchmark MNIST dataset [16]. MNIST consists of isolated handwritten digits with a size of 28×28 pixels and contains 60,000 training images and 10,000 test images. Compared to other handwritten datasets such as the Bangla and Odia character datasets, MNIST is simpler as it contains much more training examples, the diversity of handwritten digits is smaller in MNIST, and the number of digits is much smaller than the number of characters in the Odia and Bangla datasets. Therefore it is not surprising that since the construction of the MNIST dataset a lot of progress on the best test accuracy has been made.

Currently the best approaches for MNIST make use of deep neural network architectures [20]. The deep belief network (DBN) [11] has been investigated for MNIST in [11], where different restricted Boltzmann machines (RBMs) are stacked on top of each other to construct a DBN architecture. Three hidden layers are used where the sizes of each layer are 500, 500 and 2000 hidden units, respectively. The recognition performance with this method is 98.65% on the MNIST dataset.

In [17], a committee of simple neural networks is proposed for the MNIST dataset, where three different committee types comprising majority, average and median committees are combined. Furthermore, deslanted training images are created by using principal component analysis (PCA) and the elastic deformations are used to create even more training examples. The trained 9-net committees obtained 99.61% accuracy on MNIST. This work has been extended in [4] where 35 convolutional neural networks are trained and combined using a committee. This approach has obtained an accuracy of on average 99.77%, which is the best performance on MNIST so far. This technique, however, requires a lot of training data and also takes a huge amount of time for training for which the use of GPUs is mandatory.

Although currently many deep learning architectures are used in the computer vision and machine learning community, in [6] an older method from computer vision, namely the bag of visual words approach [7] was used on the CIFAR-10 dataset and obtained a high recognition accuracy of 79.6%. This simpler method requires much less parameter tuning and much less computational time for training the models compared to deep learning architectures. Also many other feature extraction techniques have been used for different image recognition problems, such as principal component analysis (PCA) [9], restricted Boltzmann machines [14], and autoencoders [12].

The cleanliness of MNIST and lack of variation may make MNIST a bad reference for selecting feature extractor techniques that are suitable for Asian scripts. Different feature extraction methods have been used for the Bangla and Odia handwritten character datasets. In [13], the celled projection method is proposed. The recognition performance obtained on the Bangla digit dataset was 94.12%. In [21], the image pixel-based method (IMG) which uses directly the intensities of pixels of the ink trace is used. The IMG is shown to be a quite powerful method [21] when the training set size is increased and obtained a recognition accuracy of 96.4% on the Bangla digit dataset.



(a) (b) (c)

Fig. 1. Illustration of the handwritten character datasets. (a) Some examples of Bangla and (b) Odia handwritten characters and (c) MNIST handwritten digits.

Contributions: In this paper, we propose the use of histograms of oriented gradients (HOG), bags of visual words using pixel intensities (BOW), and bags of visual words using HOG (HOG-BOW) for recognizing handwritten characters. These methods are compared to the direct use of pixel intensities, the discrete cosine transform (DCT), and PCA on three datasets, namely Bangla, Odia and MNIST, shown in Fig. 1. There are some challenges in the Bangla and Odia character datasets, such as the writing styles (e.g., heavy cursivity and arbitrary tail strokes), as shown in Fig. 2, similar structures of different characters, background noise, and a lack of a large amount of handwritten character samples. We have evaluated the six feature extraction techniques with three types of support vector machines [23] as a classifier, namely a linear SVM, an SVM with a radial basis function (RBF) kernel, and a linear SVM with L2-norm regularization (L2-SVM).

The results show that the HOG-BOW method obtains the highest accuracies on the three handwritten datasets. Also the HOG and BOW feature descriptors work much better than the more traditional techniques such as PCA, DCT and the direct use of pixel intensities. The results also show a very high performance of HOG-BOW with the L2-SVM on the MNIST dataset. Its recognition performance is 99.43% without the use of elastic distortions to increase the dataset, without the use of ensemble learning techniques, and without the need for a large amount of training time.

Paper outline: This paper is organized in the following way. Section 2 describes the feature extraction techniques. Section 3 describes the handwritten character datasets which are used in the experiments. The experimental results of the feature extraction techniques and the classifiers are presented in Section 4. The conclusion is given in the last section.

2 Feature Extraction Methods

We study different kinds of feature extraction techniques to deal with the handwritten character datasets as described below.

(a) (b)

Fig. 2. Illustration of the Bangla and Odia handwritten characters, in the first and second row, respectively. Some examples of (a) heavy cursive and (b) arbitrary tail strokes writing styles.

2.1 Image Pixel-Based Method (IMG)

The IMG method directly uses the pixel intensities of the handwritten character image. It is a simple method to construct a feature vector. In this method, the character image is resized to a fixed size in pixels [21] and this resulting image is treated as a feature vector. In this study, a 36×36 pixel resolution is selected, so that the feature-vector size becomes 1,296 dimensions.

2.2 Principal Component Analysis (PCA)

PCA is a well-known dimensionality reduction method, which extracts an effective number of uncorrelated variables (called 'principal components') from high-dimensional correlated variables (input data). In fact, a small number of principal components is sufficient to represent the actual data. Here, eigenvectors are computed from the training data which are used as a model which is applied on an image to compute the feature vectors. After conducting preliminary experiments, we have selected 80 eigenvectors for this approach.

2.3 Discrete Cosine Transform (DCT)

The DCT technique transforms the data from a spatial domain (image) into frequency components using only a cosine function. We use 2D-DCT in our experiments since 2D-DCT is more suitable for 2D image data. Here, the highest coefficient values are stored in the upper left and the lowest valued coefficients are stored in the bottom right of the output array [15]. The highest coefficient values are extracted in a zigzag form [18] and then represented as feature vectors. In the experiment, 60 coefficient values were selected in the feature vectors.

2.4 Histograms of Oriented Gradients (HOG)

The HOG descriptor was proposed in [8] for the purpose of human detection from images. To compute the HOG descriptor, the handwritten character image is divided into small regions [22], called 'blocks', $\eta \times \eta$. To compute the horizontal G_x and vertical G_y gradient components at every coordinate x, y of the

handwritten character image, we use a simple kernel $[-1, 0, +1]$ as the gradient detector (*i.e.* Sobel or Prewitt operators) [2].

The gradient detector can be calculated as $G_x = f(x+1, y) - f(x-1, y)$ and $G_y = f(x, y+1) - f(x, y-1)$, where $f(x, y)$ is the intensity value at coordinate x, y. The gradient magnitude M and the gradient orientation θ are calculated as:

$$\begin{aligned} M(x, y) &= \sqrt{G_x^2 + G_y^2} \\ \theta(x, y) &= \tan^{-1} \frac{G_y}{G_x} \end{aligned} \tag{1}$$

Furthermore, the image gradient orientations within each block are weighted into a specific orientation bin β of the histogram. Then, the HOG descriptors from all blocks are combined and normalized by applying the L2-norm [8]. In this experiment, we employed rectangular HOG (R-HOG) with non-overlapping blocks. The best η and β parameters for recognizing our handwritten character datasets use 36 rectangular blocks ($\eta = 6$) and 9 orientation bins, yielding a 324-dimensional feature vector.

2.5 Bag of Visual Words with Pixel Intensities (BOW)

The bag of visual words [7] has been widely used in computer vision research. In this approach, local patches that contain local information of the image are extracted and used as a feature vector. Then, a codebook is constructed by using an unsupervised clustering algorithm. In [1], some novel visual keyword descriptors for image categorization are proposed and it was shown that the soft assignment schemes outperform the more traditional hard assignment methods. In [6], it was shown that the BOW method outperformed other feature learning methods such as RBMs and autoencoders. In [5], the method was applied to text detection and character recognition in scene images. We will now explain the BOW method consisting of patch extraction, codebook computation, and feature extraction.

Extracting Patches from the Training Data. The sub-image patches X are extracted randomly from the unlabeled training images, $X = \{x_1, x_2, ..., x_N\}$ where $x_k \in \mathbf{R}^p$ and N is the number of random patches. The size of each patch is defined as a square with $(p = w \times w)$ pixels. In our experiments we used $w = 15$, meaning 15×15 pixel windows are used.

Construction of the Codebook. The codebook is constructed by clustering the vectors obtained by randomly selecting patches. Here, the codebook C is computed by using the K-means clustering method on pixel intensity information contained in each patch. Let $C = \{c_1, c_2, ..., c_K\}, c \in \mathbf{R}^p$ represent the codebook [24], where K is the number of centroids. In our experiments we used 400,000 randomly selected patches to compute the codebooks.

Feature Extraction. To create the feature vectors for training and testing images, the soft-assignment coding scheme from [6] is used. This approach uses the following equation to compute the activity of each cluster given a feature vector x from a patch:

$$i_k(x) = max\left\{0, \mu(s) - s_k\right\} \tag{2}$$

where $s_k = \|x - c_k\|_2$ and $\mu(s)$ is the mean of the elements of s [6].

The image is split into four quadrants and the activities of each cluster for each patch in a quadrant are summed up. We use a sliding window on the train and test images to extract the patches. Because the stride is 1 pixel and the window size is 15×15 pixels, the method extracts 484 patches from each image to compute the cluster activations. The feature vector size is $K \times 4$ and because we use $K = 600$ clusters, the feature vectors for the BOW method have 2,400 dimensions.

2.6 Bag of Visual Words with HOG Features (HOG-BOW)

In the previous BOW method, the intensity values in each patch are extracted and used for clustering. With HOG-BOW, however, feature vectors from patches are computed by using the state-of-the-art HOG descriptor [8], and then these feature vectors are used to compute the codebook and the cluster activities. The HOG descriptor captures the gradient structure of the local shape and may provide more robust features. In this experiment, the best HOG parameters used 36 rectangular blocks and 9 orientation bins to compute feature vectors from each patch. As in BOW, HOG-BOW uses 4 quadrants and 600 centroids, yielding a 2,400 dimensional feature vector.

3 Handwritten Character Datasets and Pre-processing

The handwritten character images in the Bangla and Odia datasets were scanned into digital images at different pixel resolutions. The details of the handwritten character datasets and pre-processing steps will now be described.

3.1 Bangla Character Dataset

The Bangla basic character consists of 11 vowels and 39 consonants [3]. In the experiment, the dataset includes 45 classes and contains 5,527 character images. The dataset is divided into training and test sets, containing 4,627 and 900 samples, respectively. Samples of the Bangla characters are shown in Fig 1(a).

3.2 Odia Character Dataset

The Odia handwritten dataset was collected from 50 writers using a take-note device. This dataset consists of 47 classes, 4,042 training and 987 test samples. Some examples of the Odia characters are shown in Fig. 1(b).

3.3 MNIST Dataset

The standard MNIST dataset [16] is a subset of the NIST dataset. The handwritten images were normalized to fit into 28×28 pixels. The anti-aliasing technique is used while normalizing the image. The handwritten images of the MNIST dataset contain gray levels. The dataset contains 60,000 handwritten training images and 10,000 handwritten test images, see Fig. 1(c) for some examples.

An overview of the handwritten datasets is given in Table 1.

Table 1. Overview of the handwritten character datasets

Dataset	Color Format	No. of Writers	No. of Classes	Train	Test
Bangla character	Grayscale	Multi	45	4,627	900
Odia character	Binary	50	47	4,042	987
MNIST	Grayscale	250	10	60,000	10,000

3.4 Dataset Pre-processing

In order to prepare the handwritten character images from the Bangla and Odia datasets, a few pre-processing steps which include background removal, basic image morphological operations and image normalization are employed. First, the Bangla handwritten dataset contains different kinds of backgrounds and is stored in gray-scale images. On the other hand, the Odia handwritten dataset is stored in binary image format as shown in Fig 1(b). Hence, the background removal is applied only to the Bangla handwritten dataset. In this study, due to its simplicity and yet robustness feature, we selected Otsu's algorithm [19] for removing background noise and making a binary image.

Next, a basic morphological dilation operation is applied to the binary handwritten images from the previous step. Finally, many researchers investigated the effect of scale differences for handwritten character recognition [17]. In this study, we normalize the handwritten image into 36×36 pixels with the aspect ratio preserved.

4 Experimental Results

We evaluated the feature extraction techniques on the Bangla, Odia and MNIST datasets by using three different SVM algorithms [23]. We used an SVM with a linear kernel, an SVM with an RBF kernel and a linear L2-regularized SVM (L2-SVM) [10]. The results are shown in Table 2, Table 3 and Table 4. In each table we show the results of the feature extraction techniques with a different SVM. In all the tables, the results show that the HOG-BOW, the HOG and the BOW method significantly outperform the other methods. Furthermore, on all 9 experiments the HOG-BOW method performs best (highly significant differences according to a student's t-test are indicated in boldface).

Table 2. Results of training (10-fold cross validation with the standard deviation) and testing recognition performances (%) of the feature descriptors when combined with the linear SVM.

Algorithms	Bangla dataset		Odia dataset		MNIST dataset	
	10-cv	Test	10-cv	Test	10-cv	Test
PCA	54.87 ± 0.20	53.67	56.57 ± 0.32	53.60	93.29 ± 0.02	92.69
DCT	59.33 ± 0.32	52.33	60.77 ± 0.40	54.81	92.51 ± 0.06	91.32
IMG	56.25 ± 0.22	54.33	56.12 ± 0.57	56.23	94.13 ± 0.05	94.58
BOW	77.96 ± 0.21	77.17	79.30 ± 0.34	78.01	98.71 ± 0.02	98.47
HOG	81.17 ± 0.30	80.11	79.86 ± 0.20	80.45	98.62 ± 0.01	99.11
HOG-BOW	**82.07 ± 0.24**	82.44	**81.74 ± 0.49**	82.43	**99.09 ± 0.03**	99.16

Table 3. Results of training (10-fold cross validation with the standard deviation) and testing recognition performances (%) of the feature descriptors when combined with the SVM with the RBF kernel.

Algorithms	Bangla dataset		Odia dataset		MNIST dataset	
	10-cv	Test	10-cv	Test	10-cv	Test
IMG	63.25 ± 0.28	60.00	57.95 ± 0.42	60.28	96.95 ± 0.02	97.27
PCA	64.08 ± 0.30	61.11	60.57 ± 0.57	59.87	96.86 ± 0.02	96.64
DCT	70.18 ± 0.27	61.33	69.91 ± 0.34	63.63	98.18 ± 0.09	97.51
BOW	78.76 ± 0.38	77.17	81.29 ± 0.42	80.65	98.98 ± 0.01	98.97
HOG	83.11 ± 0.25	83.00	82.16 ± 0.27	83.38	99.13 ± 0.01	99.12
HOG-BOW	83.14 ± 0.18	83.33	**83.62 ± 0.17**	83.56	**99.30 ± 0.02**	99.35

Table 4. Results of recognition performances (%) of the methods when used with the L2-SVM.

Algorithms	Feature dimensionality	Handwritten character dataset		
		Test Bangla	Test Odia	Test MNIST
DCT	60	51.67	56.94	90.84
PCA	80	50.33	53.90	91.02
IMG	1,296	31.33	42.65	91.53
HOG	324	74.89	74.27	98.53
BOW	2,400	86.56	84.60	99.10
HOG-BOW	2,400	87.22	85.61	99.43

In terms of SVM algorithms, we can see different results. Here, the linear SVM obtains a worse performance compared to the SVM with the RBF kernel. The linear SVM seems, however, to better handle low dimensional input vectors, compared to the L2-SVM. The L2-SVM yields significantly better results if high dimensional feature vectors [10] are used such as with the HOG-BOW and the BOW method. In fact, the best results have been achieved with the L2-SVM

with the HOG-BOW method. It is followed by the BOW and the HOG method, respectively. The HOG method outperforms the BOW method when using the SVM with an RBF kernel (see Table 3). The feature vector size of each feature extraction technique is shown in Table 4.

5 Conclusion

In this paper, we have demonstrated the effectiveness of different feature extraction techniques from computer vision for handwritten character recognition. We have shown that the HOG-BOW method combined with an L2-regularized SVM outperforms all other methods. The obtained accuracies with this method can be considered very high. On the MNIST dataset for example, HOG-BOW combined with the L2-regularized SVM obtains a recognition accuracy on the test set of 99.43% which is a state-of-the-art performance. The best method for MNIST [4] uses an ensemble of 35 convolutional neural networks and elastic deformations to increase the dataset and obtains around 99.77% accuracy. The proposed HOG-BOW method, however, is much faster, needs less training data and we have not yet evaluated its performance in an ensemble of different classifiers.

In future work we want to research different ways to improve the HOG-BOW even more. We are interested in examining other soft assignment coding schemes to compute cluster activities and we also want to construct an ensemble method to obtain even higher accuracies.

References

1. Abdullah, A., Veltkamp, R., Wiering, M.: Ensembles of novel visual keywords descriptors for image categorization. In: 2010 11th International Conference on Control Automation Robotics Vision (ICARCV), pp. 1206–1211, December 2010
2. Arróspide, J., Salgado, L., Camplani, M.: Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. Visual Communication and Image Representation **24**(7), 1182–1190 (2013)
3. Bhowmik, T.K., Ghanty, P., Roy, A., Parui, S.: SVM-based hierarchical architectures for handwritten Bangla character recognition. Document Analysis and Recognition (IJDAR) **12**(2), 97–108 (2009)
4. Cireşan, D.C., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649, June 2012
5. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D., Ng, A.: Text detection and character recognition in scene images with unsupervised feature learning. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 440–445, September 2011
6. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: 2011 International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 215–223, April 2011
7. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: 2004 8th European Conference on Computer Vision (ECCV), pp. 1–22 (2004)

8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 The IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886–893, vol. 1, June 2005
9. Deepu, V., Madhvanath, S., Ramakrishnan, A.: Principal component analysis for online handwritten character recognition. In: 2004 The 17th International Conference on Pattern Recognition (ICPR), vol. 2, pp. 327–330, August 2004
10. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A library for large linear classification. Machine Learning Research **9**, 1871–1874 (2008)
11. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (2006)
12. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length and Helmholtz free energy. In: Cowan, J., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems, vol. 6, pp. 3–10. Morgan-Kaufmann (1994)
13. Hossain, M., Amin, M., Yan, H.: Rapid feature extraction for bangla handwritten digit recognition. In: 2011 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 4, pp. 1832–1837, July 2011
14. Karaaba, M.F., Schomaker, L., Wiering, M.: Machine learning for multi-view eye-pair detection. Engineering Applications of Artificial Intelligence **33**, 69–79 (2014)
15. Lawgali, A., Bouridane, A., Angelova, M., Ghassemlooy, Z.: Handwritten Arabic character recognition: Which feature extraction method? Advanced Science and Technology **34**, 1–8 (2011)
16. LeCun, Y., Cortes, C.: The MNIST database of handwritten digits (1998)
17. Meier, U., Ciresan, D., Gambardella, L., Schmidhuber, J.: Better digit recognition with a committee of simple neural nets. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 1250–1254, September 2011
18. Mishra, T., Majhi, B., Panda, S.: A comparative analysis of image transformations for handwritten odia numeral recognition. In: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 790–793, August 2013
19. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics **9**(1), 62–66 (1979)
20. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61**, 85–117 (2015)
21. Surinta, O., Schomaker, L., Wiering, M.: A comparison of feature and pixel-based methods for recognizing handwritten bangla digits. In: 2013 International Conference on Document Analysis and Recognition (ICDAR), pp. 165–169, August 2013
22. Takahashi, K., Takahashi, S., Cui, Y., Hashimoto, M.: Remarks on computational facial expression recognition from HOG features using quaternion multi-layer neural network. In: Mladenov, V., Jayne, C., Iliadis, L. (eds.) EANN 2014. CCIS, vol. 459, pp. 15–24. Springer, Heidelberg (2014)
23. Vapnik, V.N.: Statistical Learning Theory. Wiley, September 1998
24. Ye, P., Kumar, J., Kang, L., Doermann, D.: Unsupervised feature learning framework for no-reference image quality assessment. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1098–1105, June 2012