Context-Aware Handwritten and Optical Character Recognition Using a Combination of Wavelet Transform, PCA and Neural Networks

Ngoc Hoang Phan(✉) and Thi Thu Trang Bui

Faculty of Information Technology, Ba Ria-Vung Tau University,
Truong van Bang Street 01, Vung Tau city, Ba Ria-Vung Tau province, Vietnam
{hoangpn285, trangbt.084}@gmail.com

Abstract. This paper proposes a novel context-aware handwritten and optical character recognition algorithm using a combination of wavelet transform, PCA and neural networks. At first, the features of character are extracted using combination of wavelet transform and PCA. Then multi-layer feed-forward neural networks will be used to classify these extracted features. In this algorithm, we use one neural network for each training character. This neural network is used to determine whether an input character is training character or not. The paper experimental results show that the proposed algorithm gives an effective performance of character recognition on noisy images and competes with state-of-the-art algorithms.

Keywords: Character recognition · Wavelet transform · PCA · Neural network · Image processing

1 Introduction

In recent years, pattern recognition problem is one of the most widely studied tasks in the field of image processing. The solution of pattern recognition is demanded in various areas of modern society. In addition, character recognition is one of the urgent pattern recognition tasks. The solution of this task can be used to solve other tasks, such as license plate recognition, text recognition and so on.

Wavelet transform is an effective method used to extract image features. By using wavelet transform, we will obtain the necessary information about the image. In addition, the wavelet transform is also quickly enough to be calculated. In the algorithms [1–6], wavelet transform is used to solve problem of image classification. The experimental results of these algorithms show that image features extracted by using wavelet transform give 76–99.7 % accuracy rate of image classification.

In addition, the experimental results of the algorithms [7–12] show that wavelet transform is effectively used to solve the pattern recognition tasks, in particular face recognition task on noisy images. Accuracy rate of face recognition in this case is 90–98.5 %.

P.C. Vinh and V. Alagar (Eds.): ICCASA 2015, LNICST 165, pp. 254–263, 2016.
DOI: 10.1007/978-3-319-29236-6_25

Thus, using wavelet transform is perspective way for development of novel context-aware character recognition algorithm. In this paper we propose a novel context-aware algorithm for character recognition based on combination of wavelet transform, PCA and neural networks. In case of image processing, context is any information about an image such as: image pixel, contour, noise and so on.

2 Proposed Algorithm

The proposed algorithm for character recognition consists of two main steps. The first step is training neural networks. In the second one character is recognized by using trained neural networks. The proposed character recognition algorithm works as follows.

Step 1: Training neural networks

1.1. Using wavelet transform to extract features of characters of training set.
1.2. Using PCA to reduce dimension of vectors of extracted feature.
1.3. Using obtained feature vectors to train neural networks.

Step 2: Recognizing character

2.1. Using wavelet transform to extract features of testing character.
2.2. Using PCA to reduce dimension of vector of extracted feature.
2.3. Using obtained vector and trained neural networks to recognize character.

2.1 Character Feature Extraction

Extracting feature of character using wavelet transform works as follows. Firstly, the image of character is resized to 64 × 64 pixels. Then wavelet transform is applied to obtained image and the low-frequency wavelet coefficients are extracted. In the result, we have matrix that consists of 32 × 32 low-frequency wavelet coefficients.

In order to extract local features of character, the image of character is divided to 12 parts with the same size 32 × 32 pixels (Fig. 1). Then wavelet transform is applied to each part and the low-frequency wavelet coefficients are extracted. In the result we have 12 matrixes, each of which consists of 16 × 16 low-frequency wavelet coefficients.

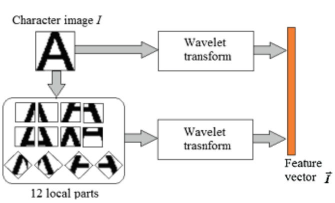

Fig. 1. Example of feature extraction of character “A”.

Finally, character feature vector is formed using low-frequency wavelet coefficients obtained in the previous steps. In the result we have character feature vector that consists of 32 × 32 + 12 × 32 × 32 = 4096 elements (Fig. 1).

2.2 Dimension Reduction

Before submission to the inputs of neural networks, dimension of feature vector is reduced. In order to solve this problem we use PCA. At first, we create eigenspace for characters (eigencharacter) using M images of characters. Creation of character eigenspace is carried out as follows.

At first, extraction feature process is applied to each of M images. In the result we have a set of $\vec{I}_1, \ldots, \vec{I}_M$. feature vectors. Then we form the mean vector, the value of each element of which is calculated by the formula (1):

$$\vec{I}_{avg} = \frac{1}{M}\sum_{n=1}^{M}\vec{I}_n. \tag{1}$$

Then each of the M feature vectors is subtracted mean vector by formula (2):

$$\vec{\Phi}_n = \vec{I}_n - \vec{I}_{\mathrm{cp}},\ n = 1, \ldots, M. \tag{2}$$

After that we create the eigenspace consisting of K eigenvectors of the covariance matrix C (3), that in the best way describe the distribution of M feature vectors ($K < M$).

$$C = \frac{1}{M}\sum_{n=1}^{M}\vec{\Phi}_n\vec{\Phi}_n^T = AA^T, \quad A = \left\{\vec{\Phi}_1, \ldots, \vec{\Phi}_M\right\}. \tag{3}$$

In this case k-th vector $\vec{u}_k$ satisfies maximization of the following (4):

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}\left(\vec{u}_k^T\vec{\Phi}_n\right)^2. \tag{4}$$

and an orthogonality condition (5):

$$\vec{u}_l^T\vec{u}_k = \begin{cases} 1, & l = k \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

Vectors $\vec{u}_k$ and values λ_k are eigenvectors and eigenvalues of covariance matrix C.

In order to create this eigenspace, at first, we calculate M eigenvectors $\vec{u}_l$ of covariance matrix C by using eigenvectors of other matrix $L = A^TA$. Each vector $\vec{u}_l$ is calculated by the following formula (6):

$$\vec{u}_l = \frac{1}{M}\sum_{k=1}^{M} v_{lk}\Phi_k, l = 1, \ldots, M. \tag{6}$$

Then we select K eigenvectors with the largest eigenvalues from M obtained eigenvectors. The eigenspace is the set of K selected eigenvectors (Fig. 2).

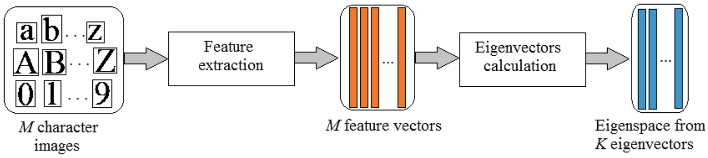

Fig. 2. Creation of character eigenspace.

After the character eigenspace is created, reducing of dimension of character feature vector $\vec{I}_{\text{in}}$ is carried out as follows.

At first, the character feature vector is decomposed on K eigenvectors $\vec{u}_i$ and corresponding decomposition coefficients are calculated by the following formula (7):

$$w_i = \vec{u}_i^T(\vec{I}_{\text{in}} - \vec{I}_{\text{avg}}),\ i = 1, \ldots, K. \tag{7}$$

Then we obtain a vector (8):

$$\vec{\Omega}^T = \{w_1, \ldots, w_K\}. \tag{8}$$

This vector describes the distribution of each eigenvectors in presentation of character feature vector. In the result of dimension reduction we have a new character feature vector $\vec{\Omega}$ consisting of K elements. In this case $K < <4096$ (Fig. 3).

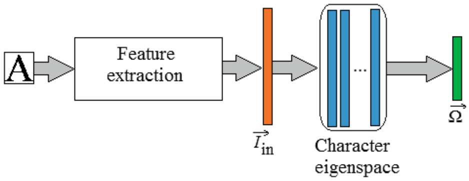

Fig. 3. Reducing of dimension of character feature vector.

2.3 Character Recognition

Back-propagation feed-forward neural networks are used for classifying character based on obtained feature vectors. In this proposed algorithm we create one multi-layered feed-forward neural network for each character of training set. These neural networks are trained by back propagation method. The input of these neural networks is the character feature vector $\vec{\Omega}$ (8), that consists of K elements. The output layer of these neural networks has one neuron, which returns a value from 0 to 1. Using one neural network for each character of training set can speed up the process of neural network training.

Each neural network determines the similarity between input character and the only one character of training set. The input character is recognized by neural networks as follows. At first, we extract feature vector of the input character and reduce its dimension. Then obtained feature vector is used as the input of all trained neural networks. Input character is recognized as a character of training set, neural network of which returns the largest value (Fig. 4.).

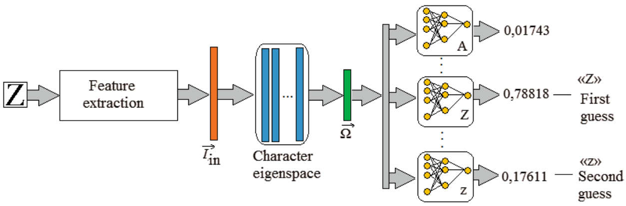

Fig. 4. Recognizing character by neural networks.

Besides, using one neural network for each character of training set allows us to include the second guess in recognition result. The second guess is a character of training set, neural network of which returns the second largest value. Using the second guess has an advantage when recognizing characters, that similarly written, such as {c, C}, {o, O}, {p, P}, {s, S}, {u, U}, {v, V}, {w, W}, {x, X} и {z, Z} (Fig. 4).

3 Experimetal Results

The proposed algorithm was tested using handwritten character and printed character images. All experiments were performed on a laptop with the processor Intel Core Duo P7350 2.0 GHz and 2.0 GB of RAM.

3.1 Handwritten Character Recognition

In the first experiment the proposed algorithm was tested on handwritten character recognition task. In order to carry out this experiment we use the known data set of handwritten digits MNIST [13, 14]. This data set consists of 60000 images for training and 10000 images for testing. All images have the same size 28 × 28 pixels and all digits are centered within the images.

In this experiment we created additional test data set by adding to images of original test set MNIST "salt and pepper" noise with probation 5, 10, 15, 20, 25, 30 %.

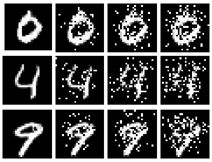

Fig. 5. Examples of test images of handwritten digits.

Examples of using images are shown in Fig. 5 (from left to right: images of handwritten digit with 10, 20 and 30 % noise).

The recognition results of handwritten digits MNIST depending on dimension of feature vector are shown in Fig. 6. Handwritten digits recognition accuracy (δ, %) is presented on vertical axis, and number of features (K) – on horizontal axis. It is shown that the recognition accuracy depends on dimension of character feature vector. When using more number of features, the recognition accuracy is higher. The recognition accuracy becomes stable and ranges 97–97.5 % (*1st guess*) and 98.8–99 % (*2nd guess*) when number of features is more than 37. The highest recognition accuracy 97.5 % (*1st guess*) and 99 % (*2nd guess*) is obtained when using the vector of 49 features. So the vector of 49 features will be used for testing with noisy data set.

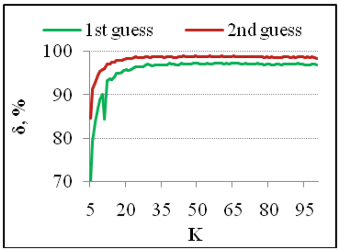

Fig. 6. Results of recognition of handwritten digits MNIST.

Table 1. Comparison of proposed algorithm and HTM network.

Processor	Algorithm	δ, %	Trainning time	Testing time
Intel coreTM 3.47 GHz	HTM network (Greedy)	97.3	05:34:12	01:38:43
	HTM network (AHC)	97.6	05:15:17	01:30:56
	HTM network (MTC)	98.5	05:21:47	01:32:35
Intel core Duo P3750 2.0 GHz	Proposed algorithm (1st guess)	97.5	00:24:36	00:06:08
	Proposed algorithm (2nd guess)	99.0	00:24:36	00:06:08

Recognition results of handwritten digits MNIST by proposed algorithm and *Hierarchical Temporary Memory* (HTM) network are shown in Table 1. In this case HTM network was trained by various algorithms, such as *Greedy*, *Aglomerative Hierarchical Clustering* (AHC) и *Maximum Temporal Connection* (MTC).

It is shown that proposed algorithm is 13 times trained and 15 times performs more rapidly, than HTM network. The highest recognition accuracy 99 % obtained by using proposed algorithm (*2nd guess*).

The obtained results were also compared with the results of other algorithms tested on the handwritten digits data set MNIST [13, 14]. The comparison results of different algorithms are shown in Table 2. It is shown that the recognition accuracy of proposed algorithm is comparable with other recognition algorithms.

Table 2. Recognition results of different algorithms tested on MNIST data set.

Algorithm	Minimum error, %	Maximum error, %
Linear clssifiers	7.6	12
Non-linear classifier	3.3	3.6
HTM network	1.5	1.5
Boosted stumps	0.87	7.7
K-nearest neighbors	0.63	5
SVMs	0.56	1.4
Neural nets	0.35	4.7
Convolutional nets	0.23	1.7
Proposed algorithm (1st guess)	2.5	3
Proposed algorithm (2nd guess)	1	1.2

In this experiment the proposed algorithm was also tested with noisy images of handwritten digits using vector of 49 features. The testing results are shown in Fig. 7. It is shown that the proposed algorithm allows recognizing handwritten digits on noisy images. The recognition accuracy considerably decreases when the noise level exceeds 20 %.

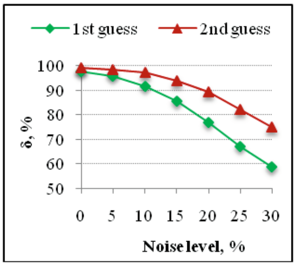

Fig. 7. Results of recognition of handwritten digits MNIST on noisy images.

3.2 Optical Printed Character Recognition

In this experiment the proposed algorithm was tested with printed characters. In order to train the proposed algorithm we created the training set consisting of 1488 images of 10 digits (0–9) and 52 characters (a–z, A–Z). Each character was presented by two fonts Times New Roman and Arial with bold and normal styles, and 16, 18, 20, 22, 24 and 26 sizes. In this case each character of training set has 24 images.

In order to test the proposed algorithm we used images of printed characters of 8 popular fonts: 4 serif fonts – Times New Roman, Garamond, Courier New and Bookman Old Style, 4 sans-serif fonts – Arial, Lucida Sans, Tahoma and Verdana. For each font we created one test set consisting 2480 images of 10 digits (0–9) and 52 characters (a–z, A–Z). Each character was presented with bold and normal styles, and 12, 14, 16, 18, 20, 22, 24, 26, 28 and 36 sizes.

Figure 8 shows the recognition results of printed characters of different fonts depending on the dimensionality of feature vector. The vertical axis is recognition accuracy (δ, %), and the horizontal axis is the number of features (K).

The experiment results show that the propose algorithm, that trained by only characters of two fonts, may recognize characters of other fonts. It is shown that the recognition accuracy of proposed algorithm for all fonts is acceptable when the number of features is between 20 and 60. The recognition accuracy of sans-serif fonts is better and more stable than the recognition results of serif fonts. The best recognition result for most fonts obtained using vector of 27 features. So this feature vector is being used to test proposed algorithm with printed characters on noisy images.

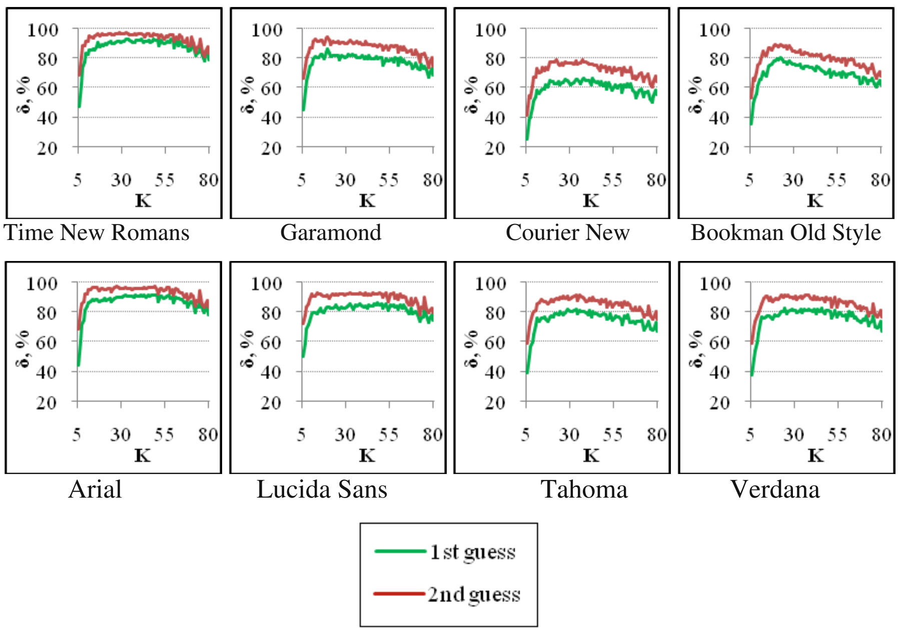

Fig. 8. Results of recognition of printed characters.

Fig. 9. Examples of optical printed character images.

For each font we created additional test set by adding 5, 10, 15, 20, 25 and 30 % of salt and pepper noise to images of existing test sample. Examples of printed character image and his noisy images with level 10, 20, and 30 % are shown in Fig. 9 from left to right.

The results of recognition printed character on noisy images are presented in Fig. 10. It is shown that the proposed algorithm is able to effectively recognize printed character of different fonts on noisy images. Recognition accuracy depends on the noise level.

The comparison results of proposed algorithm and systems ABBYY FineReader 11 and Tesseract OCR on recognition of printed character with two fonts Times New Roman and Arial on noisy images are shown in Fig. 11.

It is shown that when noise level is increased the recognition accuracy of systems FineReader 11 and Tesseract OCR significantly decreased, but the recognition accuracy

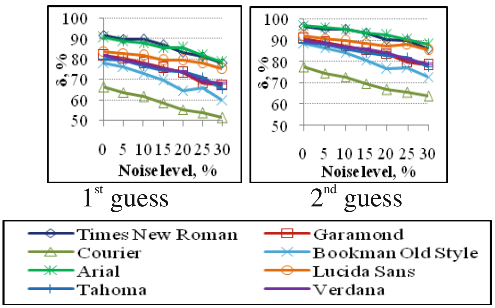

Fig. 10. Results of optical printed character recognition on noisy images.

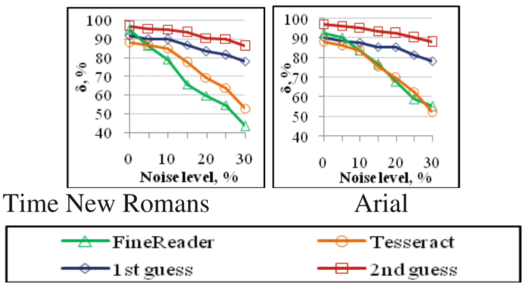

Fig. 11. Comparision results of printed character recognition on noisy images.

of proposed algorithm is more slowly decreased. The proposed algorithm more effectively recognize printed character on noisy images than systems FineReader 11 and Tesseract OCR. When the noise level is more than 15 %, the difference between their recognition results becomes more noticeable.

4 Conclusion

In this paper we developed a novel algorithm for handwritten and printed character recognition based on wavelet transform, principal component analysis and neural networks. Developed algorithm allows effectively recognizing handwritten and printed on noisy images.

References

1. Mehdi, L., Solimani, A., Dargazany, A.: Combining wavelet transforms and neural networks for image classification. In: 41st Southeasten Symposium on System Theory, Tullahoma, TN, USA, pp. 44–48 (2009)
2. Weibao, Z., Li, Y.: Image classification using wavelet coefficients in low-pass bands. In: Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, pp. 114–118 (2007)
3. Chang, T., Jay, K.: Texture analysis and classification with tree-structured wavelet transform. IEEE Trans. Image Process. **2**(4), 429–440 (1993)
4. Daniel, M.R.S., Shanmugam, A.: ANN and SVM based war scence classification using wavelet features: a comparative study. J. Comput. Inf. Syst. **7**(5), 1402–1411 (2011)
5. Park, S.B., Lee, J.W., Kim, S.K.: Content-based image classification using a neural network. Pattern Recogn. Lett. **25**(3), 287–300 (2004)
6. Riveron, E.M.F., Sossa, J.H., Gonzalez, A.C., Pogrebnyak, O.: Histograms, wavelets and neural networks applied to image retrieval. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 820–827. Springer, Heidelberg (2006)
7. Lai, J.H., Yuen, P.C., Feng, G.C.: Face recognition using holistic Fourier invariant features. Pattern Recogn. **34**, 95–109 (2001)
8. Kakarwal, S., Dsehmuhk, R.: Wavelet transform based feature extraction for face recognition. Informatica **15**(2), 243–250 (2004)
9. Zhang, B.-L., Zhang, H.: Face recognition by applying wavelet subband representation and kernel assosiative memory. IEEE Trans. Image Process. **4**(11), 1549–1560 (1995)
10. Gumus, E., Kilic, N., Sertbas, A., Ucan, O.N.: Evulation of face recognition techniques using PCA, wavelets and SVM. Expert Syst. Appl. **37**, 6404–6408 (2010)
11. Wadkar, P.D., Wankhade, M.: Face recognition using discrete wavelet transform. Int. J. Adv. Eng. Technol. **III**(I), 239–242 (2012)
12. Mazloom, M., Kasaei, K.: Face recognition using PCA, wavelets and neural networks. In: Proceeding of the First International Conference on Modeling, Simulation and Applied Optimization, Sharjah, UAE, pp. 1–6, 1-3 February (2005)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: International Symposium on Circuits and Systems, Paris, pp. 253-256 (2010)