ISSN: 0976-9102(ONLINE)
DOI: 10.21917/ijivp.2015.0162

APPLYING PRINCIPAL COMPONENT ANALYSIS, MULTILAYER PERCEPTRON AND SELF-ORGANIZING MAPS FOR OPTICAL CHARACTER RECOGNITION

Khuat Thanh Tung[1] and Le Thi My Hanh[2]
DATIC Laboratory, The University of Danang, University of Science and Technology, Vietnam
E-mail: [1]thanhtung09t2@gmail.com, [2]ltmhanh@dut.udn.vn

Abstract
Optical Character Recognition plays an important role in data storage and data mining when the number of documents stored as images is increasing. It is expected to find the ways to convert images of typewritten or printed text into machine-encoded text effectively in order to support for the process of information handling effectively. In this paper, therefore, the techniques which are being used to convert image into editable text in the computer such as principal component analysis, multilayer perceptron network, self-organizing maps, and improved multilayer neural network using principal component analysis are experimented. The obtained results indicated the effectiveness and feasibility of the proposed methods.

Keywords:
Optical Character Recognition, Principal Component Analysis, Multilayer Perceptron, Self-Organizing Maps

1. INTRODUCTION

The computer has become the first choice in terms of data storage because of its enormous utilities such as easing to manage, search, edit, and store. Nevertheless, papers are still the irreplaceable materials used to storage the documents due to its necessity in life such as books, newspapers, magazines, book notes, contracts, etc. As a result, people want to put the documents stored on paper in particular and other materials in general to the computer by typing the content for computer directly in order to exploit them effectively. For the small number of documents, this work can be achieved. For the huge volume of documents, however, this is a difficult problem because it will take a lot of time.

Optical Character Recognition (OCR) is the process of extracting characters from the image to create the editable and searchable documents. OCR is a branch of image processing. Though this is still new field compared to many other areas of science, it quickly achieved significant progress. Based on the actual demand is to put the documents which are stored on paper into the computer without typing, many OCR programs were released and widely applied in the fields related to recognition. With the development of recent studies, the accuracy rate of recognition result can reach to around 99% [1] for image clarity and regular typeface. Nonetheless, for poor quality images, special typeface, handwriting, or Vietnamese documents, the exact percentage is not high. Therefore, in addition to OCR there are some attractive research fields such as Intelligent Character Recognition [2], Optical Mark Recognition [3], Magnetic Ink Character Recognition [4], and Barcode Recognition [5].

In this paper, we will study the ways to detect text in a photo or scanned image. Several machine learning algorithms such as principal component analysis (PCA), Multi-layer Perceptron (MLP), Kohonen's Self-organizing Feature Maps, and the combination of MLP with PCA are employed to recognize the optical character recognition. The focus of this paper is to study the process of detect text in the image, preprocessing and verify the effectiveness of four proposed approaches for optical character recognition with respect to the various examined cases.

Some Open source or commercially available engines such as FreeOCR tool can only detect text in an image containing only paper and text. They cannot recognize text in the image with the complex background including table surfaces, floors, *etc*. With regard to images that have various tilts, text is also not detected and recognized. In our work, we will enhance some preprocessing techniques to handle these drawbacks and text in images with complicated background or having tilts can be recognized successfully.

The rest of paper is further organized as follows: Section 2 introduces preprocessing and the methods to detect text in the image. The way to build the training set is represented in section 3. Proposed approaches for optical character recognition are shown in section 4. Section 5 is experiments and obtained results. Conclusion and future work will be discussed in the section 6.

2. PREPROCESSING AND DETECTING TEXT IN THE IMAGE

The process of recognition of paper frame in the image is a large problem. Therefore, in this work, we only take into account image with paper frame tilting on a dark background (partial or full text).

The process of detecting text and preprocessing for image is illustrated in Fig.1 including the following steps:

a) Input image.

b) Dynamic threshold binarization for image: iteration means defines the threshold of a pixel with the grey level values of its own and neighbor's pixels and the coordinate of the pixel [6]. Grey image is split into the small parts; each part is applied different binary thresholds. Therefore, the difference of grey level between the paper frame and background can be recognized.

c) Find border surrounding paper frame.

d) Employing mask image to separate the paper frame from the background.

e) Background neutralization for the image by using morphological methods such as erosion and dilation.

f) Dividing image (d) by image (e) to remove noise and find floating points on the background color. These are characters.

g) Image binarization and finding the frame around the text in the paper frame to reduce time processing and data for steps later.

h) The frame surrounding text allows us knowing the tilt and center of document on the image. Then, the image will be rotated with obtained tilt. It can be used Bicubic interpolation method to rotate image.

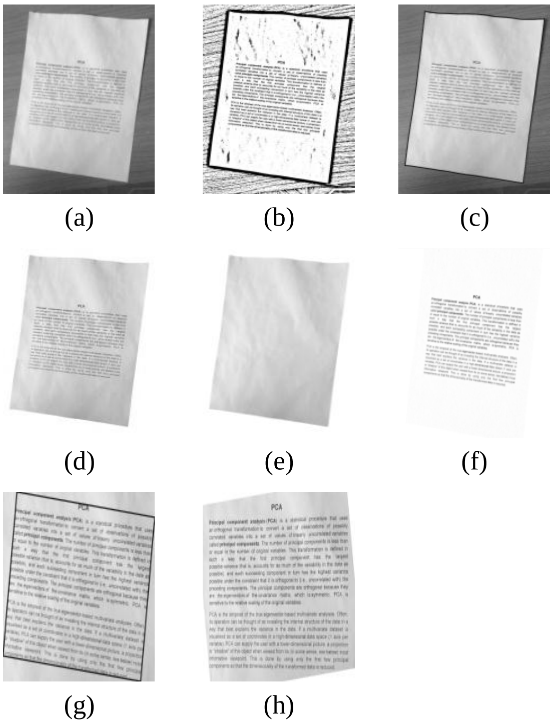


Fig.1. The process of detecting text and preprocessing for the image

However, before carrying out the optical character recognition step, image Fig.1(h) must be separated into individual lines and characters in each line. Herein, we use the approach of traveling through pixels, so there are some limitations in the type of characters which cohere vertically or points underneath the current line coincide with the points of the next line.

It is difficult to determine the format of original document. In this work, the space between words is recognized using erosion method for each line of text with the size computed by the following formula (this formula is created by *trial and error*):

$$Size = \frac{2*H}{7} \tag{1}$$

where, H is the height of a line of text.

3. CONSTRUCTING A TRAINING SET OF OPTICAL CHARACTERS

In this work, we create the list of training set automatically in order to draw the image of character based on training typeface in option. This work can be done by using Graphics class in Java. The training database used in this work is 64 Latin characters with Arial typeface. The details of training set are represented as follows:

- Lowercase letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z.
- Uppercase letters: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- A number of special characters:), (, #, %,@.

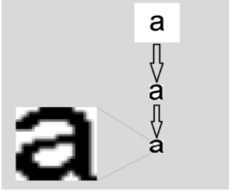


Fig.2. Constructing the training character

However, the image drawn by Graphics class has the large white background. In order to train and recognize, training images need to be narrowed background to barely contain training character and same size for all. An example of the training set construction is shown in Fig.2.

4. THE PROPOSED APPROACHES FOR OPTICAL CHARACTER RECOGNITION

4.1 MULTILAYER PERCEPTRON

Artificial neural networks [7] are a family of statistical learning models inspired by biological neural networks and are employed in many fields such as medicine, economics, recognition systems, etc. The calculation of the output value is the sum of products of inputs and weights and then using this value for activation function. The activation function consists of some features such as lower and upper bounded, being monotone, continuity, and smoothness. In this paper, activation function is the sigmoid function given by,

$$\sigma(\gamma) = \frac{1}{1+e^{-\gamma}} \tag{2}$$

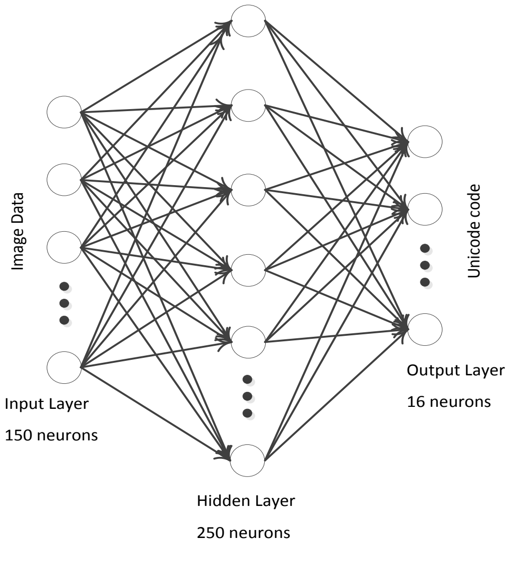


Fig.3. Multilayer perceptron diagram

Consider a single layer of N_0 artificial neuron units, with each unit fully connected to each of the N_i common inputs (plus the bias input unit). This network provides mappings of the input patterns to multi-dimensional output vectors $y = (y_1, \ldots, y_M)$ and is mentioned as a single-layer perceptron network. The Multi-Layer Perceptron network allows the additional extension to multiple layers of units, with the outputs of the first layer of units becoming the inputs of the next, and so on. Whatever the structure of the network, information flows in one direction, from the network inputs, forward towards the output layer of units, and produces the network output associated with that input.

For optical character recognition problem, a 3-layer neural network is designed: 150 neurons of input layer corresponding to the size of binary image 10 × 15, hidden layer with 250 neurons which is determined by trial and error method, 16 neurons of output layer corresponding to 16-bits Unicode of training character. The Fig.3 illustrates the multilayer neural network designed for the problem in this work.

In the process of optical character recognition through image, because input binary image will be skewed in pixels when comparison of training image as well as the limitation of training set, so the output of network is 16 bits Unicode unexpected. This leads to the recognition error and confusion among characters. In order to overcome this problem, instead of using a set of linking weights between layers, we will store each training image with specific weights and employ error coefficient as a threshold to analyze the features of the image.

The neural network training process is to find the set of weight values that will cause the output from the neural network to match the actual target values as closely as possible. A learning rule is applied in order to improve the value of the MLP weights over a training set according to a specific algorithm. The algorithm which is used to tune weights in this paper is Backpropagation.

The Backpropagation learning algorithm can be divided into two phases: propagation and weight update.

- **Phase 1:** Forward propagation of a training pattern's input $X_s = \{x_1, x_2, \ldots, x_n\}$, $s = \overline{1, N}$ (N is the number of training patterns) through the neural network in order to generate the propagation's output activations.
  - Output of j^{th} neuron in hidden layer

$$y_j = \sigma\left(\sum_{i=1}^{n} w_{ij} x_i\right) \quad (3)$$

  - Output of k^{th} neuron in output layer

$$y_k = \sigma\left(\sum_{j=1}^{m} w_{jk} y_j\right) \quad (4)$$

- **Phase 2:** Calculate Mean Squared Error for training pattern s:

$$E_s = \frac{1}{p}\sum_{k=1}^{p}(y_k - t_k)^2 \quad (5)$$

In which t_k is expected value of k^{th} neuron at the output layer. backward propagation of the error through the neural network to adjust weights at l^{th} iteration.

- Deviation of linking weights between hidden layer and output layer:

$$\Delta w_{jk} = \eta \delta_k(l) y_j(l) \quad (6)$$

where,

$$\delta_k(l) = (t_k - y_k)(1 - y_k)y_k \quad (7)$$

- Deviation of linking weights between input layer and hidden layer:

$$\Delta w_{ij} = \eta \delta_j(l) x_i \quad (8)$$

where,

$$\delta_j(l) = y_j(1 - y_j)\sum_{k=1}^{p} \delta_k w_{jk} \quad (9)$$

After updating weights, pattern X_s continues to be input of network at $(l + 1)^{\text{th}}$ iteration and weights are updated until $E_s < \varepsilon$ or the number of predefined iterations has been reached.

The above process is conducted until the network learns by heart training set. After that, network configuration is stored to use later.

4.2 SELF-ORGANIZING MAPS

Kohonen neural network [8] is a typical network in self-organizing map model (SOM) based on competitive unsupervised learning. This one is the modeling of the behavior of the human brain. The different kinds of neural networks are usually only concerned with the value and input of the network but not exploited the structural relationship in the neighborhood of the sample data or entire sample space. However, Kohonen network takes into account to these factors.

The main idea of this algorithm is to map topological features of self-organizing to preserve the orderly arrangement of the sample in multidimensional space into new space with dimensionality less, usually two dimensions. This is a non-linear projection providing a "mapping feature" two-way, which is applied to detect and analyze the characteristics of the input space.

The output from the Kohonen neural network does not consist of the output of several neurons. When a pattern is represented to a Kohonen network, one of the output neurons is chosen as a "winner". This "winning" neuron is the output from the Kohonen network. Often these "winning" neurons show groups in the data that is presented to the Kohonen network.

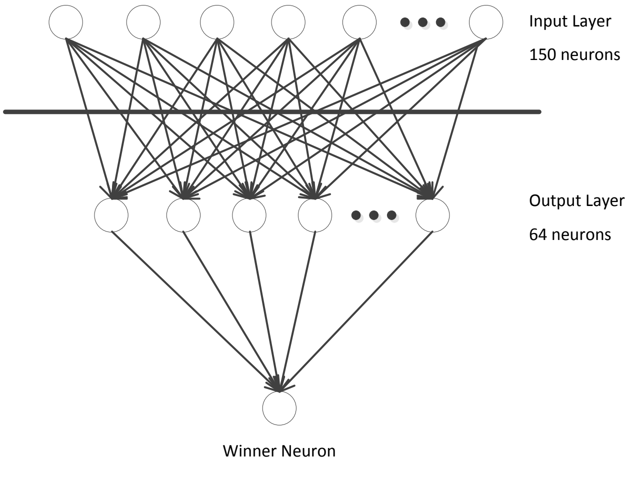


Fig.4. Self-organizing neural network model

For optical character recognition problem, we construct the network model with 150 neurons for input layer corresponding to the size of binary image 10 × 15. The training process will classify the input data set randomly, and each character in training set corresponds to each of the specific output layer neurons. Training set for each character is only one image, so the number of output neurons is equal to the size of training set (64 neurons). The Fig.4 is a model of the Kohonen neural network employed in this work.

Kohonen Learning Algorithm [8] is presented as follows:

Step 1: *Initialization*. Choose the different random values for initial weight vectors $w_j = [w_{j1}, w_{j2},\ldots, w_{jM}]$, $j = 1, 2, ..., L$, where L is the total number of neurons in the lattice, M is the dimension of the input space.

Step 2: *Sampling*. Draw a sample x from input space with a certain probability; the vector x presents the activation pattern that is applied to the lattice. The dimension of x is equal to M.

Step 3: *Similarity Matching*. Find the winning neuron at a time step n by using minimum distance Euclidian criterion:

$$i(x) = \arg\min_j \|x - w_m\| \quad j = \overline{1, L} \tag{10}$$

Step 4: *Updating*. Tuning the synaptic weight vectors of all neurons by employing the update formula:

$$w_j(n + 1) = w_j(n) + \eta(n)h_{j,i(x)}(n)[x(n) - w_j(n)] \tag{11}$$

where, $\eta(n)$ is the learning rate parameter, and $h_{j,i(x)}(n)$ is the neighborhood function centered around the winning neuron $i(x)$; both $\eta(n)$ and $h_{j,i(x)}(n)$ fluctuated dynamically during learning for best results.

Step 5: Continue with step 2 until no noticeable changes in the feature map are observed.

4.3 PRINCIPAL COMPONENT ANALYSIS

PCA is a mathematical procedure that employs transformation to convert a set of observations of possibly correlated features into a set of values of uncorrelated features called principal components [9]. PCA is a well-known technique for extracting representative features for character recognition and is used to reduce the extent of the data with minimal loss of information. In this algorithm, the entire dataset (d dimensions) is projected onto a new subspace (k dimensions where $k < d$). This method of projection is useful in order to reduce the computational costs and the error of parameter estimation. The new subspace is composed of k unit vector with dimension N. Each this vector is called as an eigenvector.

Let M is the number of input images, each image is converted to vector of dimension N (the image area), and we obtain the set of inputs $X = \{x_1, x_2, \ldots, x_M\}\{x_i \in R^N\}$. Then, input vectors are translated to the center of training set $A = [\Phi_1, \Phi_2, \ldots, \Phi_M]$. The formula determining this center is represented as follows:

$$x_{tb} = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{12}$$

$$\Phi_i = x_i - x_{tb} \tag{13}$$

The covariance matrix of A is $C = A.A^T$ has eigenvalues sorted by descending $\lambda_1 > \lambda_2 > \ldots \lambda_m$ and corresponding eigenvectors $u_1, u_2, \ldots, u_m$. Each u_i is an eigenvector. A set of original vectors is presented in the space created by n eigenvectors as follows:

$$x - x_{tb} = w_1u_1 + w_2u_2 + \cdots + w_Nu_N = \sum_{i=1}^{N} w_iu_i \tag{14}$$

Selecting K eigenvectors u_i and corresponding K maximal eigenvalues λ_i, we have:

$$x - x_{tb} = w_1u_1 + w_2u_2 + \cdots + w_Ku_K = \sum_{i=1}^{K} w_iu_i \tag{15}$$

where, $K << N$

The vector of coefficients $[w_1, w_2, \ldots, w_k]$ is the new representation of image created in the space PCA. It can be seen that the number of dimensions of input data is reduced, but most of the basic features are retained based on eigenvectors corresponding to large eigenvalues.

K eigenvalues in M values are chosen based on the following formula [10]:

$$\frac{\sum_{i=1}^{K}\lambda_i}{\sum_{i=1}^{N}\lambda_i} \geq \theta \quad (\text{e.g. } \theta \text{ is } 0.9 \text{ or } 0.95) \tag{16}$$

As a result, each image Φ_i in the training set can be presented as a linear combination of K maximal eigenvectors $w_i = u_j^T\Phi_i,\ j = 1,2\ldots K.$

4.4 USING PRINCIPAL COMPONENT ANALYSIS FOR MULTI-LAYER PERCEPTRON

In the process of principal component analysis as mentioned above, let Ω_i is the i^{th} training image after translating to the center of training set Φ_i. This image is represented in new space as follows:

$$\Omega_i = \begin{bmatrix} u_1^T & \Phi_i \\ u_2^T & \Phi_i \\ u_3^T & \Phi_i \\ \vdots & \vdots \\ u_K^T & \Phi_i \end{bmatrix}$$

We apply PCA to extract the features of the image and create the vectors of the training image, and then use this vector of features for the input layer of MLP network. PCA is employed to extract the features from image and then these features are vectors of input layer to enhance the quality of MLP. The vectors of training images presented in PCA's space in the dimensionality less than original ones must be partitioned on determining domain [-1, 1] to significantly reduce the coefficient of error in the training process and increase the effectiveness of Backpropagation algorithm. Only the direction of these vectors is considered and magnitude is ignored (magnitude of the vector is 1, or called the unit vector).

The transformation of a vector to unit vector is also known as vector normalization and using the formula as follows:

$$\hat{\Omega}_i = \frac{\Omega_i}{\|\Omega_i\|} \tag{17}$$

Vectors in training set normalized have become the input vectors for MLP network. The dimensionality of the training set which is reduced by using PCA results in the decrease of the number of input neurons K << 150. The identification number of neurons in the hidden layer is very important because the number of neurons of input layer is not fixed. Determining the number of hidden layers without affecting to training time and ensuring the sufficiency of information of training set are carried out by the following formula [11]:

$$N_H = \frac{2(N_O + N_I)}{3} \qquad (18)$$

where,

N_O is the number of neurons of output layer

N_I is the number of neurons of input layer

N_H is the number of neurons of hidden layer

5. EXPERIMENTATION

5.1 THE OBJECTIVE OF THE EXPERIMENTS

Using image acquisition devices to scan, photograph the document and put the images into the computer to handle cannot avoid the noise or tilt of paper due to Impact from outside. Therefore, in this experiment, the effectiveness of 4 proposed approaches which are PCA, MLP, SOM, and MLP using PCA is evaluated on standard input image and image with different tilting.

5.2 EXPERIMENT FOR THE CASE OF STANDARD INPUT

In this experiment, standard input image with no tilt and no noise is employed to recognize the optical characters. Fig.5 is an example for a standard input image and Fig.6 is the obtained results of experiments.

Some inputs:

- The number of characters in the empirical image is 1253 including lowercase letters, bold letters, and special characters.
- Tilting ~ 0 degrees.
- The average height of each line is 45 px.
- The training set contains 64 Latin characters with Arial typeface.

Fig.5. Standard input image

The Table.1 shows the obtained results of 4 approaches for the standard input image. The above results have not yet been performed post-processing step to repair the errors such as confused between characters which have similar shapes or between lowercase and uppercase letters.

Table.1. The obtained result for recognition in the case of standard input

Sl. No.	Approach	No. Wrong Characters	Error Rate (%)
1	SOM	61	4.9
2	MLP	46	3.7
3	PCA	1	0.08
4	PCA-MLP	9	0.7

- **Results for SOM:**	- **Results for MLP:**
pca principal component analysis (pca) is a statistic6l procedure that uses an o1hogonal transformation to conv6r a s6t of obserations of possibly corr6lated variablos into a sot of values of linearly uncorrelated variables called principal components, thc number of principal components fs foss than or equat to tho number of orfgfnal variables, this transformation fs d6fin6d in such a way that the first principal compon6nt has the lauest possible varianco (that is, accounts for as much of the variabitity fn th6 data as possible), and each succeeding componont in turn has the highest varfance possible undor the oonstraint that it is orhogonaf to (i,e,, uncorrefated with) the preceding components. the princfpal components are ohhogonaf because they are the eigenvectors of the covariance matrix, which is symmetric, pca fs sensitive to the relative scaling ofthe original variables. pca is tho simpfost of the true eigenvectofbased multivariato analyses. cxcn, jts operatfon can be thought of as revealing the fnternal structure of the data jn a way that best explains the variance in the data, lf a mujtivariate datasot is vfsualised as a set of coordinates in a high.dimensional data space (1 axis pcr variable), pca can supply th6 user with a lower.dimensional picture, a pr(6ction or ..shadow.. of this ohect when vfewed from its (in some sense, soo below) most inhrmativo viowpoint, this is done by using only the first fow principalcompononts so that the dimonsionality of tho transhfmod data js foduced.	lcl lrllclpll colpolelt lllllll (lcl) is a statisticll procedure that uses an o1hogonal transformation to conver a set of obserations of possibly correlated variables into a set of values of linearly uncorrelated variables called lrllcllll collolelt l, the number of principal components is less than or egual to the number of original variables, this transformltion is defined in such a way that the first principal component hle the lalest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn hls the highest variance possible under the constraint that it is olhogonal to (i,e,, uncorrelated with) the precedlng components. the principal components lre ohhogonll because they are the eigenvectors of the covariance matrix, which is symmetric, pca is sensitive to the relative sclling ofthe original varilblee. pca is the simplest of the true eigenvectofbased multivarilte analvses. cxen, its operation can be thought of as revealing the internal structure of the data in a wav that best explains the variance in the data, lf a multivariate dataset is visualised as a set of cocrdinates in a high.dimensional data space (1 axis per variable), pca can supply the user with a lower.dimensional picture, a prhection or ..shadow.. of this ohect when viewed from its (in some sense, see below) most inhrmative viewpoint, this is done bv using onlv the first few principal components so that the dimensionality of the transhrmed data is reduced.

- **Results for PCA:**	- **Results for PCA-MLP:**
pca principal component analysis (pca) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components, the number of principal components is less than or equal to the number of original variables, this transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i,e,, uncorrelated with) the preceding components. the principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric, pca is sensitive to the relative scaling ofthe original variables. pca is the simplest of the true eigenvector-based multivariate analyses. often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data, lf a multivariate dataset is visualised as a set of coordinates in a high.dimensional data space (1 axis per variable), pca can supply the user with a lower.dimensional picture, a projection or ..shadow.. of this object when viewed from its (in some sense, see below) most informative viewpoint, this is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.	pca principal component analysis (pca) is a statistical procedure that uses an o4hogonal transformation to conve4 a set of obsewations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components, the number of principal components is less than or equal to the number of original variables, this transformation is defined in such a way that the first principal component has the lamest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is okhogonal to (i,e,, uncorrelated with) the preceding components. the principal components are o4hogonal because they are the eigenvectors of the covariance matrix, which is symmetric, pca is sensitive to the relative scaling of the original variables. pca is the simplest of the true eigenvectofbased multivariate analyses. o#en, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data, lf a multivariate dataset is visualised as a set of coordinates in a high.dimensional data space (1 axis per variable), pca can supply the user with a lower.dimensional picture, a pr(ection or ..shadow.. of this onect when viewed from its (in some sense, see below) most inhrmative viewpoint, this is done by using only the first few principal components so that the dimensionality of the transhrmed data is reduced.

Fig.6. The recognition results of the algorithms

5.3 OTHER TEST CASE

In order to achieve more objective assessment, this section will examine the results of four proposed methods with respect to input images that have various tilts and the partial image of page. The empirical results are evaluated by observing and being statistic for the error rate. The Fig.7 is the cases of an input image and the empirical results are presented in Table.2.

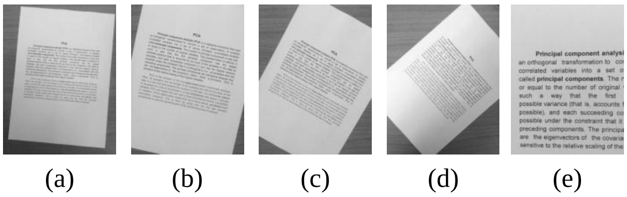


Fig.7. The input image for examined cases

Table.2. The error rate of the examined cases

Sample Method	Tilted image 5º (a)	Tilted image 15º (b)	Tilted image 30º (c)	Tilted image 50º (d)	Partial image (e)
SOM	6.38%	6.78%	10.38%	100%	0.53%
MLP	3.19%	3.03%	5.1%	100%	6.97%
PCA	0.08%	0.16%	0.48%	100%	0%
PCA-MLP	1.52%	0.72%	0.96%	100%	1.88%
FreeOCR	N/A	N/A	N/A	N/A	0.1%

Because test cases in Fig.7(a), 7(b), 7(c), 7(d) include the floor background and have the tilts, FreeOCR, which is one of the famous OCR tools, cannot detect and recognize characters in the document. In contrast, by using the preprocessing techniques to remove the background and handle the tilt of the image, our work can detect characters with the low error rate. This indicates that our methods improved the accuracy of optical character recognition compared with some open source products in the same field.

The Fig.8 is the case of an image with a luminous paper frame. In this case, luminous areas will create connected regions to other external details of the paper frame when employing dynamic threshold binarization. This results in noise and affects to the process of preprocessing for the image. Therefore, we cannot use the proposed approach to detect text in the image.

Fig.8. The fault image

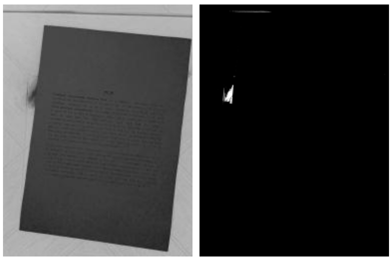
In order to eliminate the gloss of the paper frame, we can convert original image from RGB to HSV color space. HSV color space gives us to know the information: the hue, saturation, and lightness of a specific range of colors. The range of glare in the image can be recognized based on lightness value received. The Fig.9 illustrates the way to eliminate the luminance noise in the image.

5.4 EVALUATION THE OBTAINED RESULTS

We can see that PCA recognizes the bold characters effectively. This approach is based on the specific features of character and not depends on the background color of page.

In general, multilayer neural network recognizes characters exactly; however, this method depends on results from the image binarization and background color of paper. For MLP network, the result of the recognition of bold characters without being training is not good, but with the use of SOM model this is resolved despite untrained bold characters.

The volatility of traveling through pixels employed in this paper results in the case of some characters cohered is misidentified. Nonetheless, the use of PCA can overcome this disadvantage by training with images which have two cohering characters. The output layer of MLP model is fixed 16-bit Unicode of each of individual characters, so cannot train two ligatures simultaneously. This is the reason why the result of MLP using PCA is not as high as that of PCA.



(a) (b)

Fig.9. Eliminating luminance noise in a paper frame: (a) HSV image, (b) Noise areas is detected

Applying PCA for multi-layer neural network enables the process of training network faster, speeds up for Backpropagation algorithm, reduces the number of employed neurons significantly, and increases computing speed but still provides positive results.

The problem of how to rotate text skewed has not been fully resolved yet. With the large tilt of document, the recognition results are not correct. In addition, the results depend on the quality of image. From there, it can be seen the important role of preprocessing to narrow the diversity in the image.

6. CONCLUSION

Based on the obtained results of four proposed approaches, it can be seen that the accuracy of optical character recognition in the image is fairly high. Principal component analysis not only give the best result in recognizing text from the image but it also significantly enhance the propriety of multi-layer neural network. However, the proposed means have not been good yet when applying them for accented letters such as Vietnamese characters.

From the empirical results, the problem of optical character recognition needs the post-processing for the document as well as the effective approach for preprocessing before recognition to improve the efficiency. The proposed methods in this paper can be applied for other recognition problems such as faces, license plates, and traffic signs.

REFERENCES

[1] L.R. Blando, J. Kanai and T.A. Nartker, “Prediction of OCR Accuracy Using Simple Image Features”, *Proceedings of the Third International Conference on Document Analysis and Recognition*, Vol. 1, pp. 319-322, 1995.

[2] D. Deodhare, N.N.R.R. Suri and R. Amit, “Preprocessing and Image Enhancement Algorithms for a Form-based Intelligent Character Recognition System”, *International Journal of Computer Science and Applications*, Vol. 2, No. 2, pp. 131-144, 2005.

[3] Sabyasachi Das, “Optical Mark Recognition Technology for Rural Health Data Collection”, Technical Report, IKP Centre for Technologies in Public Health, 2010.

[4] S. Souza, J.M. Abe and K. Nakamatsu, “MICR Automated Recognition based on Paraconsistent Artificial Neural Networks”, *Procedia Computer Science*, Vol. 22, pp. 1083–1091, 2013.

[5] R. Adelmann, M. Langheinrich and C. Floerkemeier, “A Toolkit for Bar Code Recognition and Resolving on Camera Phones – Jump Starting the Internet of Things”, *Proceedings of the Workshop on Mobile and Embedded Interactive Systems at Informatik 2006*, pp. 366-373, 2006.

[6] Jagroop Kaur and Rajiv Mahajan, “A Review of Degraded Document Image Binarization Techniques”, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 3, No. 5, pp. 6581-6586, 2014.

[7] S. Haykin, “*Neural Networks. A Comprehensive Foundation*”, 2nd Edition, Prentice-Hall, 1999.

[8] T. Kohonen, “Self-Organized Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, Vol. 43, No. 1, pp. 59-69, 1982.

[9] Munish Kumar, M. K. Jindal and R.K. Sharma, “PCA-based Offline Handwritten Character Recognition System”, *Smart Computing Review*, Vol. 3, No. 5, pp. 346-357, 2013.

[10] D.H Jeong, C. Ziemkiewicz, W. Ribarsky and R. Chang, “Understanding Principal Component Analysis Using a Visual Analytics Tool”, Technical Report, UNC Charlotte, 2009.

[11] Jeff Heaton, “*Introduction to Neural Networks for Java*”, 2nd Edition, Heaton Research Inc., 2008.