Optical english font recognition in document images using eigenfaces

Reconocimiento óptico de fuentes en inglés en documentos de imágenes utilizando eigenfaces

Author:

Hasan S. M. Al-Khaffaf*[1]
Nadia A. Musa[2]

SCIENTIFIC RESEARCH

How to cite this paper:
Al-Khaffaf H, Musa N, Optical English Font Recognition in Document Images Using Eigenfaces, Duhok, Kurdistan Region of Iraq. Innovaciencia. 2018; 6(1): 1-11.
http://dx.doi.org/10.15649/2346075X.466

Reception date:
Received: 21 September 2018
Accepted: 15 November 2018
Published: 28 December 2018.

Keywords:
Font Recognition; EigenFaces; EigenFonts; PCA.

ABSTRACT

Introduction: In this paper, a system for recognizing fonts has been designed and implemented. The system is based on the Eigenfaces method. Because font recognition works in conjunction with other methods like Optical Character Recognition (OCR), we used Decapod and OCRopus software as a framework to present the method. **Materials and Methods**: In our experiments, text typeset with three English fonts (Comic Sans MS, DejaVu Sans Condensed, Times New Roman) have been used. **Results and Discussion**: The system is tested thoroughly using synthetic and degraded data. The experimental results show that Eigenfaces algorithm is very good at recognizing fonts of synthetic clean data as well as degraded data. The correct recognition rate for synthetic data for Eigenfaces is 99% based on Euclidean Distance. The overall accuracy of Eigenfaces is 97% based on 6144 degraded samples and considering Euclidean Distance performance criterion. **Conclusions**: It is concluded from the experimental results that the Eigenfaces method is suitable for font recognition of degraded documents. The three percentage incorrect classification can be mediated by relying on intra-word font information.

*[1] Software Engineering and Embedded Systems (SEES) Research Group, Department of Computer Science, University of Duhok, hasan.salim@uod.ac.
[2] Department of Physics, University of Duhok, nadia.musa@uod.ac

1. INTRODUCCTION

Optical Font Recognition (OFR) is concerned with the recognition of fonts for a text image. It can be used as a post processing step after OCR in order to faithfully recreate the document with original look. It can also be used prior to OCR process to improve the OCR quality[1,2]. Like any pattern recognition problem, font recognition is based on the extraction of a group of features from document images. Typographical features in font recognition are found in two types: global features and local features[2]. The former type of features can be extracted from large text entities like words, lines, or paragraphs. Examples of these features are height of line, word orientation, word space, word height, word width, regardless of their content. The change in font can be easily detected even by non-expert in typography. The global features depend on the length of the text (number of characters) more than the characters content and can be extracted from binary images scanned at lower solution[3]. In the rest of this section, some papers related to the *field* are reviewed. Global typographical features and Gabor filter methods use block of text without prior knowledge of the content making them useful prior to OCR process. One disadvantage of these methods is their inaccuracy because text block will not give us enough information about the font of individual character and block text must belong to the same font type. Zramdini and Ingold[4] presented a method that is based on identifying the global typographical features of the text such as type face, size, slope and weight of the text from an image block without knowing the contents of the text. They used a multi variate Bayesian classifier for the recognition purpose. Another method based on analyzing global texture of the document images is presented by Zhu *et al.*[5]. While Emptoz[6] presented a method to analyze the font at texture level instead of pixel level by finding the frequency and orientation of the texture.

On the other hand, local features can be extracted from individual characters, for example Serif shape, slope (roman versus italic) width (normal versus expanded) or size of vertical lines. In order to retrieve this type of feature from the real document, a careful processing is needed because it focuses on small and special character parts (like Serifs) and it is affected by some factors like noise, skew, low resolution, and binarization thresholds. Using local features also need the prior knowledge of the character class[3]. Cutter *et al.*[7] presented a work that is based on clustering similar tokens into clusters of candidate fonts. They used unsupervised method based on token occurrence. Their algorithm however, is suitable for reconstructing representative fonts rather than recognizing used fonts. The search engine of Solli and Lenz[2] is able to recognize fonts in very large data base of fonts. The recognition system is based on Eigenimage and use data base of 2763 different fonts of English alphabet. The evaluation shows the correct font name is one of the best *five* matches.
Recently, Sevik *et al.* used deep learning to recognize Turkish fonts[8]. Bharath *et al.* used Support Vector Machines (SVM) to recognize fonts of Roman text[9]. Jaiem *et al.* used steerable pyramid method to recognize fonts of Arabic documents[10]. Tao *et al.* used Principal Component Analysis (PCA) to recognize fonts of single Chinese characters[11].
In this paper Eigenfaces is also used for the purpose of font recognition. However, unlike Solli and Lenz[2], it is used to find the font name of each character glyph of the scanned document in order to reconstruct a PDF with the same font type used previously during the time of creating the original document. Breuel *et al.* developed OCRopus and Decapod, an open source OCR system and a low cost book digitization project, respectively. The Decapod project provides a framework for researchers to experiment and develop OCR and font recognition methods. Figure 1 shows the block diagram of Decapod framework steps. The proposed recognition font system (grey background rectangle) is shown as Font Recognition module *fitted* within Decapod pipeline.

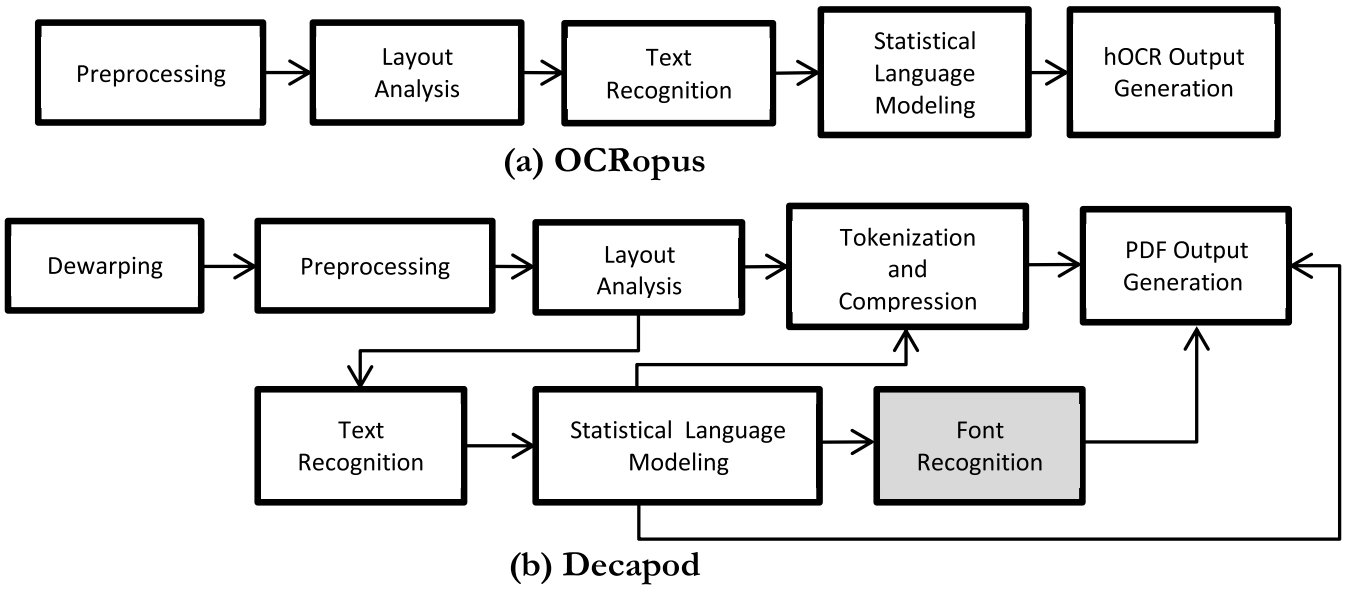


Figure 1. The relationship between the Decapod project and OCRopus. (a) The standard OCRopus processing pipeline. (b) The Decapod book scanning pipeline with OCRopus steps blended-in.

Source: Taken and adapted from Shafait *et al.*(12).

2. MATERIALS AND METHOD

2.1 EIGENFACES: BACKGROUND AND DEFINITIONS

Eigenfaces is an approach that converts face images into small group of characteristic feature images. Eigenvector can be shown as a sort of spectral face which we call Eigenfaces[13,14] or as a group of Eigenvectors used in computer vision problem of human face recognition[13,15]. The approach of using Eigenfaces for recognition was developed by Sirovich and Kirby[16] and used by Turk and Pentland in face classification and recognition[14]. Eigenfaces is a Principle Component Analysis (PCA) based on face recognition method[13,15,16]. The number of potential Eigenfaces is equal to the number of face image in the training set. The main reason for using fewer Eigenfaces is the computational efficiency[13,14,15].

In mathematics, Eigenvector (x) of a linear transformation is a non-zero vector. When that transformation is applied to that vector, it may change the magnitude but not the direction as shown in Fig. 2. For each eigenvector of a linear transformation there is a corresponding scalar value called an eigenvalue (λ) of that vector. The eigenvector is scaled under the linear transformation.

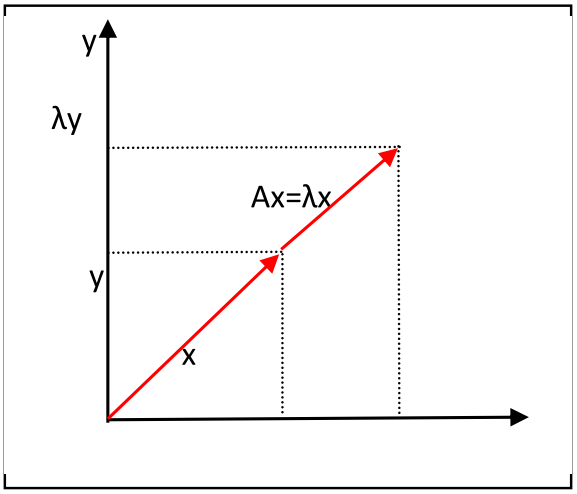


Figure 2. Eigenvector (x) and Eigenvalue (λ).

$$A\,x \quad \lambda x \tag{1}$$

$$(A - \lambda I)x = 0 \tag{2}$$

where A is a vector function, and is the I identity matrix.

This is a homogeneous system of equations and a non-trivial solution exists if-and-only-if $Det(A-\lambda I)=0$ where *Det* is determinant.

Then it is called characteristic polynomial of A. For a matrix $N\,X\,N$ thre are N eigenvectors. It can be thought that the eigenvector is a set of features that together characterize the variation between corresponding images[13].

2.2 PRINCIPLE COMPONENT ANALYSIS (PCA)

PCA is a statistical method used to reduce the dimensionality. It transforms the number of possible correlated variables into small number of uncorrelated variables called (Principle Component). The PCA is useful when you want to reduce the number of variables and is being used to explore, sort and group data. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate. PCA was presented by Karl Pearson according to Jolliffe[17]. The PCA is unsupervised feature reduction algorithm[18]. PCA has not been widely used before developing the computer because it is not easy to do PCA by hand when the number of variables is larger than four, but it is the best choice when the number of variables is large. PCA was used in many fields like (image processing, machine learning, signal processing, communication, etc.).
The goal of PCA is the analysis of data to identify patterns and finding patterns to reduce the dimensions of the dataset with minimal loss of information. The advantage of the PCA when used in the Eigenfaces is to reduce the size of the database for the recognition of a new image[17,18,19,20,21].

2.3 FEATURE REDUCTION

Feature reduction refers to the mapping of the original high dimensional data into a lower-dimensional space. The goal of feature reduction is to obtain a compact, exact representation of the data by removing statistically redundant components. The reasons of using the feature reduction is visualization (projecting of high-dimensional data into 2D or 3D), data compression (efficient storage and retrieval), and noise removal (positive effect on query accuracy). Feature reduction is used in many applications such as face recognition, image retrieval, handwriting

digit recognition, etc.[17, 18, 22].

2.4 THE EIGENFACES ALGORITHM STEPS

1. Preparing the training set. The training set Γ should be prepared for processing. All images should have the same resolution and dimensions (same size). Each image is converted to a vector simply by concatenating the rows of pixels in the original image (size of each vector is N^2 X 1. All images of the training set are stored in a single matrix (the size of matrix is N^2 X M) where M is the number of images in the training set (Database).

2. Normalizing the face vector to leave only the unique features of each face image by (i) Calculating the average (Mean) face vector:

$$\Psi = \frac{1}{M}\Sigma_{n=1}^{M}\Gamma_n \tag{3}$$

(ii) Subtracting the average (Mean) face vector from each face vector:

$$\Psi = \frac{1}{M}\Sigma_{n=1}^{M}\Gamma_n \tag{4}$$

where Φ is the normalized (Mean Centered) face image.

3. Calculating Eigenvectors and Eigenvalues. In this step the Eigenvalues and Eigenvectors (Eigenfaces) should be calculated.

$$\lambda_k = \frac{1}{M}\,(u_k^T\Phi_n)^2 \tag{5}$$

where u_k and scalars λ_k are Eigenvectors and Eigenvalues, respectively, of the covariance matrix C

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n \cdot \Phi_n^T \tag{6}$$

$$C = A \cdot A^T \tag{7}$$

where the matrix $A = (\Phi_1, \Phi_2, \Phi_3 \ldots, \Phi_M)$ and the covariance matrix is $N^2\ X\ N^2$.

The size of covariance matrix is very large $N^2\ X\ N^2$ which is a severe problem because it will occupy a lot of memory space and it is very time consuming for calculations, so it produces N^2 Eigenvectors and Eigenvalues. For example: to find Eigenvectors and Eigenvalues for images of size 102 X 102 , a covariance of size 10404 X 10404 has to be computed and 10404 Eigenvectors and Eigenvalues should be calculated. Normally, this is not trivial and also not efficient especially when we know that most of those Eigenfaces are not useful for our task (because it may include noise). Hence, the dimensions of the covariance and Eigenfaces are reduced[13].

First Eigenvectors of $M\ X\ M$ matrix is solved then taking appropriate linear combination of face images Φ_i . Consider the Eigenvectors v_i of $A^T A$ such that

$$A^T A v_i = u_i v_i \tag{8}$$

Multiply both sides by , we have

$$AA^T Av_i = u_i Av_i \tag{9}$$

From which we see that are the Eignevectors of $C = A - A^T$
We construct an $M X M$ matrix:

$$L = A^T . A \tag{10}$$

where $L_{mn} = \Phi_m^T \Phi_n$ and find M Eigenvectors vl of L. These vectors determine the linear combinations of the M training set face images to form the Eignefaces ul

$$u_{l=} \sum_{k=1}^{M} v_{lk} \Phi_k \tag{11}$$

where $l = 1, 2, 3,..., M$ where corresponds to all Eigenvectors of L and u represents Eigenvectors of C (Eigenfaces). The advantage of this method is that one has to evaluate only numbers instead of N^2. Usually $M < N^2$ and only M a few principle components (Eigenfaces) will be relevant.

4. Ideally not all eigenvectors are needed from the M eigenvectors (Eigenfaces) u_i . Only M' should be chosen which have the highest Eigenvalue (The M Eigenvectors are sorted in descending order Eigenvalue and chosen to represent Eigenspace). The higher eigenvalue has more characteristic features of the face image Eigenfaces with low eigenvalues will be removed.

5. Calculating Omega (Ω) by projecting each of the train images into Eigenspace:

$$\omega_k = u_k^T (\Gamma_k - \Psi) \tag{12}$$

where $k = 1, 2, 3, \ldots, M'$ and $\Gamma_k - \Psi$ is the mean centered image (normalized image). Note that each projection of image can be obtained as ω_1 of image$_1$ and ω_2 for projection of image$_2$ and so on, where ω_k consists of M values and $\Omega^T = [\omega_1, \omega_2 \ldots \omega_{M'}]$. This step is the last step in the training phase.

6. Test a new image by first normalizing test image Γ_{test}

$$\Phi = \Gamma_{test} - \Psi \tag{13}$$

second by projecting the new image into the Eigenface space to obtain a vector that contains weights as

$$\omega_{test} = u_{test}^T (\Gamma_{test} - \Psi) \tag{14}$$

where is a vector that contains values as weights of the test image.

7. Calculating the Euclidean Distance (ED)

$$\varepsilon_k^2 = ||\omega_{test} - \Omega_k||^2 \tag{15}$$

Then the image with minimum Euclidean distance is taken as the identified image.

3. EXPERIMENTAL SETUP

In this section the proposed font recognition system is being evaluated. A thorough experiment has been executed to test the accuracy of the system to recognize fonts in synthetic clean images and degraded images. The system is implemented using Visual Basic .Net programing language on a PC running Windows 8.1. The PC is equipped with Core i7 4770HQ processor, 2.20 GHz, 4 CPUs (8 Threads) and 16GB RAM. Training images were extracted directly from font files. The test images are taken from electronic book typeset with three different fonts: Comic Sans MS (Comic), DejaVu Sans Condensed (DejaVu), Times New Roman (Times). The dataset is divided into two parts clean-synthetic and degraded images[23]. Images were degraded using Kanungo *et al.* method[24] by means of Qgar library (http://www.qgar.org). We used OCRopus and Decapod software to segment pages into lines and lines into separate character images. The ground truth data are generated by extracting glyphs bitmaps using FontForge library (http://fontforge.org). The largest possible bitmap-glyph width and height that do not require scaling-down was selected as the preferred size (102*102). Hence, all TI glyphs are scaled up to 102*102 image size to avoid degradation shape quality. The authors of the dataset also used PNG file format since it can be processed by Decapod and OCRopus software. The original dataset is a book of around 60 pages of text and available in synthetic and degraded formats. In our experiment we have used 6 pages only. The first three pages of the book were used as test images (TI) and the last three pages were used as training images. The ground truth data (GT) are extracted from the original True Type Font (TTF) files. Figure 3 shows a synthetic and degraded sample of the three digital fonts.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Figure 4 shows the Euclidean distance comparison between (TIC and GTC, GTD, GTT) for the small and capital letters. Figure 5 shows the Euclidean distance between (TID and GTC, GTD, GTT) for the small and capital letters. Figure 6 shows the Euclidean distance comparison between (TIT and GTC, GTD, GTT) for the small and capital letters for the synthetic image. Figure 7 shows the percentage of correct recognition rate for the font type (Comic, DejaVu, Times) using Euclidean distance for degraded images.

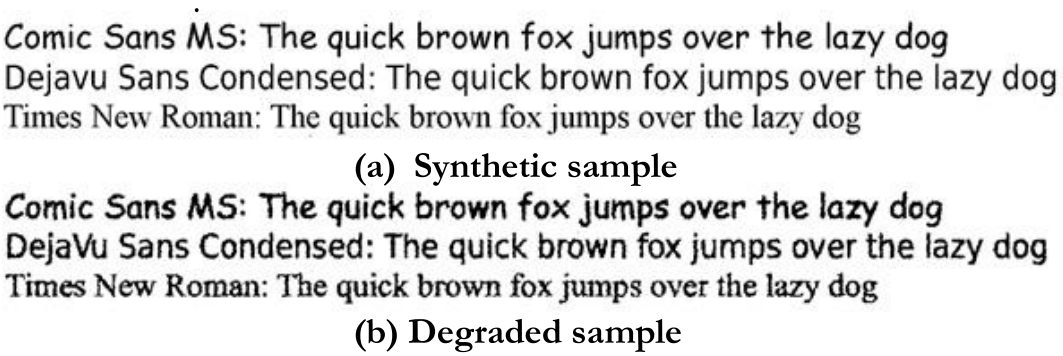
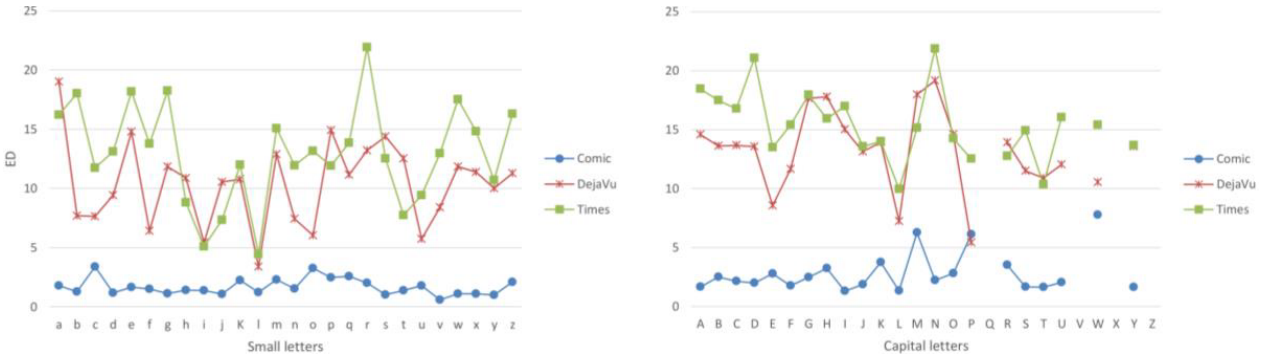
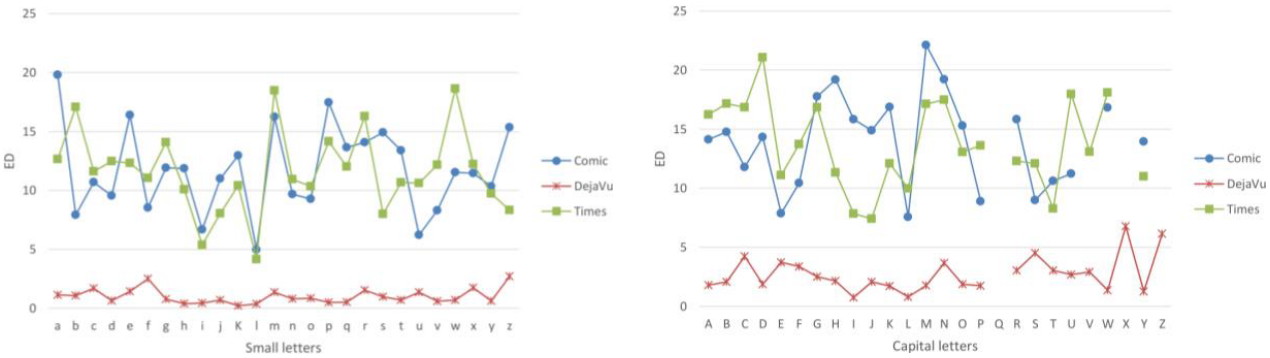


Figure 3. Samples of synthetic and degrade



(a) Small (b) Capital

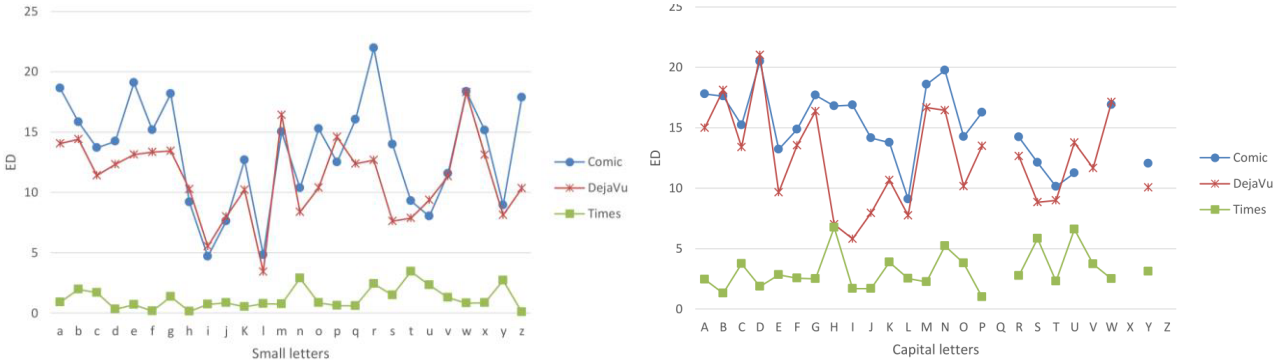
Figure 4. Result of EigenFaces on Comic synthetic test data (lower is better). (a) Lower case letters. (b) Upper case letters.



(a) Small (b) Capital

Figure 5. Result of EigenFaces on DejaVu synthetic test data (lower is better). (a) Lower case letters. (b) Upper case letters.

The performance of our developed EigenFaces sub-module is explained next. The low ED values showed in Figs. 4-6 mean that the EigenFaces method is able to detect the font of TI images for all characters of the alphabet (capital and small letters) with nearly 100% accuracy. This is also consistent for the three tested fonts (Comic, DejaVu, and Times). This accuracy stems from the fact that synthetic TI images are very close in shape to their GT counterparts.



(a) Small (b) Capital

Figure 6. Result of EigenFaces on Times synthetic test data (lower is better). (a) Lower case letters. (b) Upper case letters.

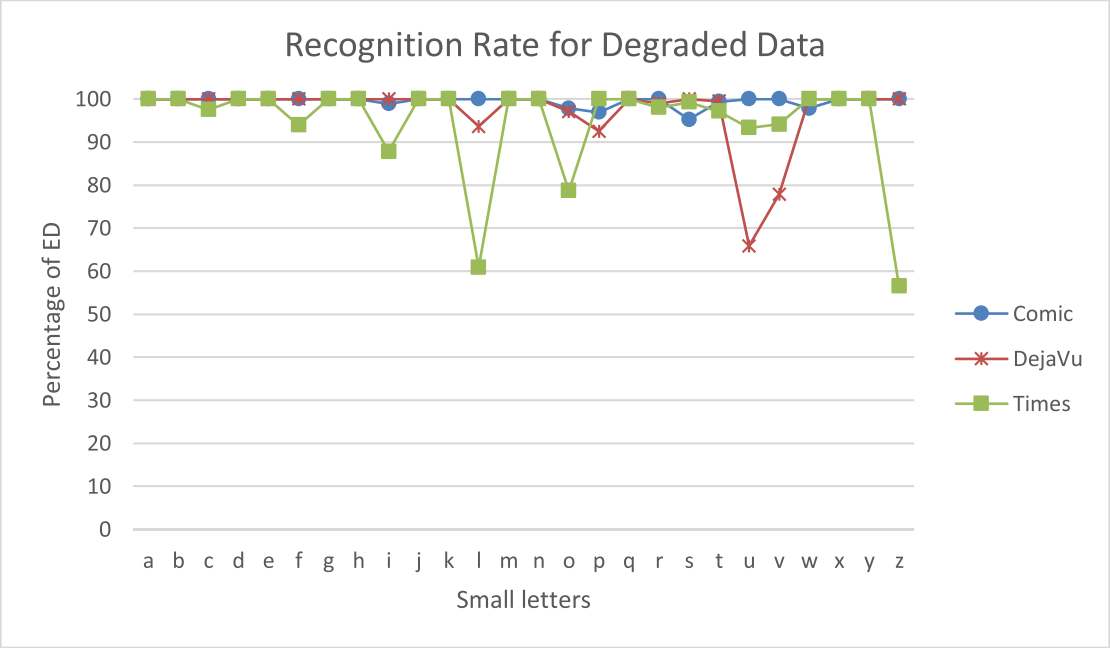


Figure 7. Percentage of the correct classification of EigenFaces on degraded test data (higher is better) for small letters.

For degraded data, the values in Fig. 7 represent the percentage of correct classification of the TI compared with the ground truth (GTC, GTD, GTT) with the number of occurrences. For example, considering Comic font, it is shown that the percentage of correct classification is between (95.24% - 100%) relying on ED for the number of sample images between (14 - 234). One reason for the high recognition rate for the Comic font is that the shape of glyphs is quite distinguishable and different than that of DejaVu and Times which differentiate the glyphs of this font against other fonts.

For DejaVu, the recognition percentage is between (65.85% - 100%) based on ED for the number of samples between (13 - 293). While most letters have high recognition rate, only one letter has low recognition rate: u (65.85%). The reason for this is due to the deformation of the character after degradation as shown in Fig. 8 where degraded letter u belong to DejaVu font looks like letter u of

Figure 8. Superimposition u glyph. The green color is for TI, red color is for GT, and olive-green color for the intersection area between TI and GT. (a) superimposition of u-DejavuGT and u-DejaVuTI degraded (b) superimposition of u-ComicGT and u-DejaVuTI degraded.

comic font due to degradation. For Times font, the percentage of correct classification is between (60.87%-100%) for the number of sample images between (13 - 303). Almost all letters got high recog-

nition rate, except the letter l which due to deformation of the character after degradation the classification is dropped to only (60.87%). To have an insight on the quality of our results we can refer to the work of Bharath[9] where he got 80% average accuracy with English fonts using Support Vector Machines. The difficulty in recognition of some letters can be compensated by utilizing an intuitive intra-word typesetting fact that is: characters of a word are usually typeset with same font type. From the results on three fronts, we conclude that recognition rate for Comic font is better than the other fonts due to its glyph shapes which are much different than DejaVu and Times.

5. CONCLUSION

In this paper we presented a Eigenfaces-based system for English font recognition. The system is tested and evaluated with synthetic and degraded data. Experimental results showed that the recognition of synthetic data is correct for all samples and the percentage of correct classification of degraded data is 97% (overall accuracy based on 6144 samples). It is concluded from the experimental results that the Eigenfaces method is suitable for font recognition of degraded documents. The three-percentage incorrect classification can be corrected by utilizing noise removal algorithms[25,26,27] and/or relying on intra-word font information. In addition, the good overall accuracy of the Eigenfaces module suggests that adding/porting it to Decapod system is feasible and will enable the creation of type 5 PDF files, i.e. files with original TTF fonts, hence it will lead to pleasant viewing experience.

REFERENCES

1. Nagy G. Twenty years of document image analysis in PAMI. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2000; 22(1):38-62. https://doi.org/10.1109/34.824820
2. Solli M, Lenz R. FyFont: find-your-font in large font databases. In: Ersbøll B.K., Pedersen K.S. (eds) Image Analysis. SCIA 2007. Lecture Notes in Computer Science, Vol 4522. Springer, Berlin, Heidelberg; 2007, pp 432-441. https://doi.org/10.1007/978-3-540-73040-8_44
3. Ramakrishnan AG, Urala KB. Global and local features for recognition of online handwritten numerals and Tamil characters. In: Proceedings of the 4th International Workshop on Multilingual OCR; 2013 Aug 24; Washington, D.C., USA. ACM; 2012. p. 16. Available from: ACM Digital Library. https://doi.org/10.1145/2505377.2505391
4. Zramdini A, Ingold R. Optical font recognition using typographical features. IEEE Transactions on pattern analysis and machine intelligence. 1998; 20(8): 877-82. https://doi.org/10.1109/34.709616
5. Zhu Y, Tan T, Wang Y. Font recognition based on global texture analysis. IEEE Transactions on pattern analysis and machine intelligence. 2001;23(10):1192-200. https://doi.org/10.1109/34.954608
6. Allier B, Emptoz H. Font Type Extraction and Character Prototyping Using Gabor Filters. In International Conference on Document Analysis and Recognition; 2003 August 3-6; Washington, DC, USA. IEEE; 2002. p. 799-803.
7. Cutter MP, Beusekom JV, Shafait F, Breuel TM. Unsupervised font reconstruction based on token co-occurrence. In: Proceedings of the 10th ACM symposium on Document engineering; 2010 Sep 21-24; Manchester, United Kingdom. ACM; 2010. p. 143-150. Available from: ACM. https://doi.org/10.1145/1860559.1860589
8. Sevik A, Erdogmus P, Yalein E. Font and Turkish Letter Recognition in Images with Deep Learning. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT); 2018 Dec 3-4; Ankara,Turkey. IEEE;2018. p. 61-64. Available from: IEEE Xplore. https://doi.org/10.1109/IBIGDELFT.2018.8625333
9. Bharath V, Rani NS. A font style classification system for English OCR. In: 2017 International Conference on Intelligent Computing and Control (I2C2); 2017 Jun 23-24; Coimbatore, India. IEEE; 2017, p. 1-5. Available from: IEEE Xplore. https://doi.org/10.1109/I2C2.2017.8321962
10. Jaiem FK, Slimane F, Kherallah M. Arabic font recognition system applied to different text entity level analysis. In: 2017 International Confer-

ence on Smart, Monitored and Controlled Cities (SM2C); 2017 Feb 17; Sfax, Tunisia. IEEE; 2017, p. 36-40. Available from: IEEE Xplore. https://doi.org/10.1109/SM2C.2017.8071847
11. Tao D, Lin X, Jin L, Li X. Principal component 2-D long short-term memory for font recognition on single Chinese characters. IEEE transactions on cybernetics. 2016;46(3):756-65. https://doi.org/10.1109/TCYB.2015.2414920
12. Shafait F, Cutter MP, Van Beusekom J, Bukhari SS, Breuel TM. Decapod: A flexible, low cost digitization solution for small and medium archives. In: International Workshop on Camera-Based Document Analysis and Recognition; 2011 Sep 22; Beijing, China: Springer, Berlin, Heidelberg; 2011. p. 101-111. https://doi.org/10.1007/978-3-642-29364-1_8
13. Turk M, Pentland A. Eigenfaces for recognition. Journal of cognitive neuroscience. 1991;3(1):71-86. https://doi.org/10.1162/jocn.1991.3.1.71
14. Turk MA, Pentland AP. Face recognition using eigenfaces. In Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 1991 Jun 3-6; Maui, HI, USA. IEEE; 1991. p.586-591. Available from: IEEE Xplore.
15. Lata YV, Tungathurthi CK, Rao HR, Govardhan A, Reddy LP. Facial recognition using eigenfaces by PCA. International Journal of Recent Trends in Engineering. 2009;1(1):587.
16. Sirovich L, Kirby M. Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A. 1987;4(3):519-24. https://doi.org/10.1364/JOSAA.4.000519
17. Jolliffe I. Principal component analysis, Wiley Online Library, 2002.
18. Kambhatla N, Leen TK. Dimension reduction by local principal component analysis. Neural computation. 1997;9(7):1493-516. https://doi.org/10.1162/neco.1997.9.7.1493
19. Yang J, Zhang DD, Frangi AF, Yang JY. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE transactions on pattern analysis and machine intelligence. 2004;26(1):131-137. https://doi.org/10.1109/TPAMI.2004.1261097
20. Saabni R, El-Sana J, Efficient Generation of Comprehensive Database for Online Arabic Script Recognition, In 10th International Conference on Document Analysis and Recognition; 2009 Jul 26-29; Barcelona, Spain. IEEE; 2009. Available from: IEEE Xplore. https://doi.org/10.1109/ICDAR.2009.258
21. Pearson K. "Liii. On lines and planes of closest fit to systems of points in space." The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901;2(11):559-72. https://doi.org/10.1080/14786440109462720
22. Burges CJ, Geometric methods for feature extraction and dimensional reduction. In: Maimon O, Rokach L, editors. Data mining and knowledge discovery handbook. Boston, MA, USA. IEEE: Springer; 2005. p. 59–91. https://doi.org/10.1007/0-387-25465-X_4
23. Al-Khaffaf HSM, Shafait F, Cutter MP, Breuel T. M. On the performance of Decapod's digital font reconstruction. 21st International Conference on Pattern Recognition (ICPR); 2012 Nov 11-15;Tsukuba, Japan. IEEE; 2012, p. 649-652, Available from: IEEE Xplore.
24. Kanungo T, Haralick RM, Baird HS, Stuezle W, Madigan D. A statistical, nonparametric methodology for document degradation model validation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000;22(11):1209-23. https://doi.org/10.1109/34.888707
25. Ko SJ, Lee YH. Center weighted median filters and their applications to image enhancement. IEEE Transactions on Circuits and Systems. 1991;38(9):984–993. https://doi.org/10.1109/31.83870
26. Al-Khaffaf HSM, Talib AZ, Salam RA. Removing Salt-and-Pepper Noise from Binary Images of Engineering Drawings. In: 19th International Conference on Pattern Recognition; 2008 Dec 08-11; Tampa, Florida, USA. IEEE;2008. p. 1271–1274. Available from: IEEE Xplore. https://doi.org/10.1109/ICPR.2008.4761425
27. Al-Khaffaf HSM, Talib AZ, Salam RA. Enhancing Salt-and-Pepper Noise Removal in Binary Images of Engineering Drawing. IEICE Transactions on Information and Systems. 2009;E92-D(4):689–704. https://doi.org/10.1587/transinf.E92.D.689