# An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition

G.Vamvakas, B.Gatos, S. Petridis and N.Stamatopoulos

*Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,*
*National Center for Scientific Research "Demokritos", GR-153 10 Agia Paraskevi, Athens, Greece*
*{gbam, bgat, petridis, nstam} @iit.demokritos.gr*

## Abstract

*In this paper, we present an off-line methodology for isolated Greek handwritten character recognition based on efficient feature extraction followed by a suitable feature vector dimensionality reduction scheme. Extracted features are based on (i) horizontal and vertical zones, (ii) the projections of the character profiles, (iii) distances from the character boundaries and (iv) profiles from the character edges. The combination of these types of features leads to a 325-dimensional feature vector. At a next step, a dimensionality reduction technique is applied, according to which the dimension of the feature space is lowered down to comprise only the features pertinent to the discrimination of characters into the given set of letters. In this paper, we also present a new Greek handwritten database of 36,960 characters that we created in order to measure the performance of the proposed methodology.*

## 1. Introduction

Optical Character Recognition (OCR) systems aim at transforming large amount of documents, either printed or handwritten, into electronic form for further transformation. Nowadays, although recognition of printed isolated characters is performed with high accuracy, recognition of handwritten characters still remains an open problem in the research arena. A widely used approach in isolated character recognition is to follow a two step schema: a) represent the character as a vector of features and b) classify the feature vector into classes [1].

Selection of a feature extraction method is most important in achieving high recognition performance [2]. A feature extraction algorithm must be robust enough such that for a variety of instances of the same symbol, similar feature sets are generated, thereby making the subsequent classification task less difficult [3]. In the literature, feature extraction methods have been based on three types of features: a) statistical, b) structural and c) global transformations [4].

The most common statistical features used for character representation are: a) zoning, where the character is divided into several zones and features are extracted from the densities in each zone [5] or from measuring the direction of the contour of the character by computing histograms of chain codes in each zone [6], b) projections [7] and c) crossings and distances [8].

Structural features are based on topological and geometrical properties of the character, such as maxima and minima, reference lines, ascenders, descenders, cusps above and below a threshold, cross points, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom, etc . In [9], a structural approach for recognizing Greek handwritten characters is introduced. A 280-dimensional vector is extracted consisting of histograms and profiles. The horizontal and vertical histograms are used in combination with the radial histogram, out-in radial and in-out radial profiles.

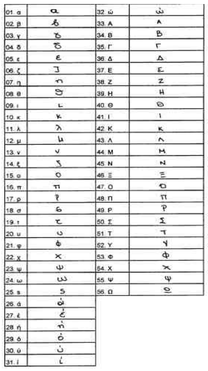
Due to the fact that a continuous signal contains more information than needed representing this signal by a linear combination of a series of simpler well-defined functions is easier. The coefficients of the linear combination provide a compact encoding known as transformation. Fourier Transforms (FT) is most popular among global transformation. Since the first *n* coefficients of the FT can be used in order to reconstruct the image [10], then these n coefficients are considered to be a *n*-dimesional feature vector that represents the character. Other transformations used are the Discrete Cosine Transform (DCT) [11], moments [12] etc.

The existence of standard databases for the comparison of the results is essential for the evaluation of off-line recognition techniques. There are widely used databases for handwriting recognition, such as NIST [13], CEDAR [14], CENPARMI [15] and UNIPEN [16]. As far as Greek characters are concerned the only database one can find in the literature is the GRUHD database [17], which consists of Greek characters, text, digits and symbols in unconstrained handwriting mode.

In this paper we present a database consisting of 36,960 isolated Greek Handwritten Characters. Furthermore, an off-line OCR methodology for these letters is proposed based on a feature extraction scheme followed by a suitable dimensionality reduction step. The remaining of the paper is organized as follows. In Sections 2 and 3 the data acquisition procedure and the proposed methodology are presented respectively. Experimental results are discussed in Section 4 and finally conclusions are drawn in Section 5.

## 2. Greek Handwritten Character Database

The database comprises samples of 56 Greek handwritten characters (24 uppercase, 24 lowercase, the final “ς” and the accented vowels “ά”, “έ”, “ή”, “ ί”, “ύ”, “ό”, “ώ ”) written by 132 Greek writers. Every writer contributed 5 samples of each letter, thus resulting in a database of 660 variations of each letter and an overall of 36,960 isolated labelled character samples.



| | | | |
|---|---|---|---|
| 01. α | α | 32. ώ | ώ |
| 02. β | β | 33. Α | Α |
| 03. γ | γ | 34. Β | Β |
| 04. δ | δ | 35. Γ | Γ |
| 05. ε | ε | 36. Δ | Δ |
| 06. ζ | ζ | 37. Ε | Ε |
| 07. η | η | 38. Ζ | Ζ |
| 08. θ | θ | 39. Η | Η |
| 09. ι | ι | 40. Θ | Θ |
| 10. κ | κ | 41. Ι | Ι |
| 11. λ | λ | 42. Κ | Κ |
| 12. μ | μ | 43. Λ | Λ |
| 13. ν | ν | 44. Μ | Μ |
| 14. ξ | ξ | 45. Ν | Ν |
| 15. ο | ο | 46. Ξ | Ξ |
| 16. π | π | 47. Ο | Ο |
| 17. ρ | ρ | 48. Π | Π |
| 18. σ | σ | 49. Ρ | Ρ |
| 19. τ | τ | 50. Σ | Σ |
| 20. υ | υ | 51. Τ | Τ |
| 21. φ | φ | 52. Υ | Υ |
| 22. χ | χ | 53. Φ | Φ |
| 23. ψ | ψ | 54. Χ | Χ |
| 24. ω | ω | 55. Ψ | Ψ |
| 25. ς | ς | 56. Ω | Ω |
| 26. ά | ά | | |
| 27. έ | έ | | |
| 28. ή | ή | | |
| 29. ό | ό | | |
| 30. ύ | ύ | | |
| 31. ί | ί | | |

**Figure 1.** Sample of the forms used

Collection of characters has been facilitated by requesting writers to fill specially crafted forms, as shown in Fig.1, afterward converted to binary images with the use of a scanner. In order to segment the form into cells containing the characters, detection of horizontal and vertical lines present in the forms was necessary. The form is scanned horizontally and when the first black pixel is detected we calculate the number consecutive black pixels. If the number exceeds a predefined number $D_{min}$ then it is presumed that the start ($Y_s$) of a line has been detected. The end ($Y_e$) of the line is calculated as $Y_s + d$ where $d$ is the expected width of the line. A likewise procedure is followed for detecting the vertical lines.

Notice that, due to the fact that the form may be slightly skewed, a problem that can occur is that a line may not be detected, as shown in Fig.2.

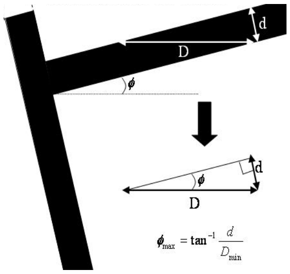


**Figure 2.** Error in line detection

Specifically, if $D_{min}$ is the minimum number of consecutive black pixels and $d$ is the expected width of the line, then, the maximum angle of skew, for which the line can be detected, is: $\varphi_{max} = \tan^{-1}(d/D_{min})$.

Once lines detection has been completed, we proceed into determining the cells, in which characters are written. A simple method would be to define the co-ordinates of the cells as lines and columns intersections. However, if the lines are slightly skewed, then the co-ordinates of the intersections are incorrectly calculated, which fact may have severe effects for the extraction of the character from the cell (i.e. part of the character, which will be located at the boundaries of the defined “cell”, can be fragmented). To avoid this problem, we redefine the co-ordinates for every cell separately.

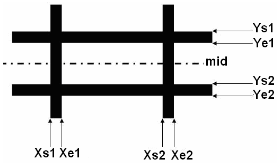


**Figure 3.** Redefinition of the co-ordinates for cell extraction

Consider two consecutive horizontal lines, $H_1(Y_{s1}, Y_{e1})$, $H_2(Y_{s2}, Y_{e2})$, and two consecutive vertical lines, $V_1(X_{s1}, X_{e1})$ and $V_2(X_{s2}, X_{e2})$ ( Fig. 3). To define the cell formed by these lines we start from the centre of the initial cell with vertical offset *mid*:

$$mid = Y_{e1} + \frac{Y_{s2} - Y_{e1}}{2} \qquad (1)$$

and scanning upwards until a line $Y_{c1}$ is detected as follows:

$$f(x, Y_{c1}) = 1 \text{ where } x_1 \leq x \leq x_2,\ x_2 \text{ - } x_1 \geq L_{\min} \qquad (2)$$
$$\text{and } mid \geq Y_{c1} \geq Y_{e1}$$

Similarly downwards we detect a line $Y_{c2}$ as follows:

$$f(x, Y_{c2}) = 1 \text{ where } x_1 \le x \le x_2,\, x_2 - x_1 \ge L_{\min} \quad (3)$$
$$\text{and } mid \le Y_{c2} \le Y_{s2}$$

where $L_{min}$ is a predefined number of consecutive black pixels used to find the lines of a cell. So the co-ordinates of the cell are $Y_{c1}, Y_{c2}, X_{e1}, X_{s2}$.

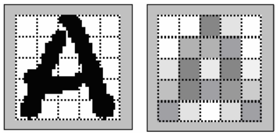
The bounding box of the character inside the cell defines the character image that will be stored in the database.

## 3. Feature Extraction Method

Before the feature extraction algorithm takes place, we first normalize all binary character images to a *N*x*N* matrix with respecting the original aspect ratio.

### 3.1. Feature Extraction

In our approach, we employ four types of features. The first set of features is based on zones. The image is divided into horizontal and vertical zones, and for each zone we calculate the density of the character pixels (Fig. 4).

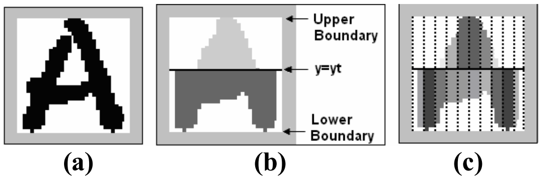


**(a)** **(b)**

**Figure 4.** Feature extraction of a character image based on zones. (a) The normalized character image. (b) Features based on zones. Darker squares indicate higher density of character pixels.

In the second type of features, the area that is formed from the projections of the upper and lower as well as of the left and right character profiles is calculated. Firstly, the center mass $(x_t, y_t)$ of the character image is found.

Upper/lower profiles are computed by considering, for each image column, the distance between the horizontal line $y=y_t$ and the closest pixel to the upper/lower boundary of the character image (Fig. 5b).

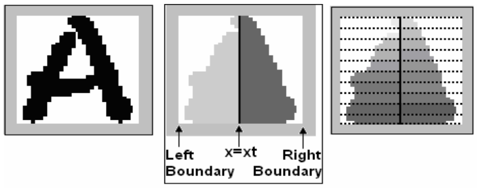


**(a)** **(b)** **(c)**

**Figure 5.** Feature extraction of a character image based on upper and lower character profile projections. (a) The normalized character image. (b) Upper and lower character profiles. (c) The extracted features. Darker squares indicate higher density of zone pixels.

This ends up in two zones (upper, lower) depending on $y_t$. Then both zones are divided into vertical blocks. For all blocks formed we calculate the area of the upper/lower character profiles. Fig. 5c illustrates the features extracted from a character image using upper/lower character profiles.

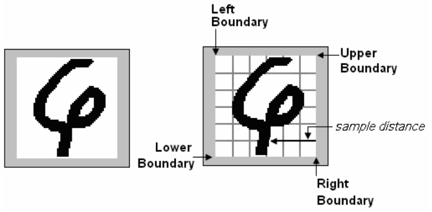
Similarly, we extract the features based on left/right character profiles (Fig. 6).



**(a)** **(b)** **(c)**

**Figure 6.** Feature extraction of a character image based on left and right character profile projections. (a) The normalized character image. (b) Left and right character profiles. (c) The extracted features. Darker squares indicate higher density of zone pixels.

The third feature set is based on the distances [8] of the first image pixel detected from the upper and lower boundaries of the image, scanning along equally spaced vertical lines as well as from the left and right boundaries scanning along equally spaced horizontal lines (Fig .7)



**(a)** **(b)**

**Figure 7.** Feature extraction of a character image based on distances. (a) The normalized character image. (b) A sample distance from the right boundary.

The forth set, calculates the profiles of the character from the upper, lower, left and right boundaries of the image [7], as shown in Fig. 8. The profile counts the number of pixels between the edges of the image and the contour of the character. These features are used because they describe well the external shape of the characters.

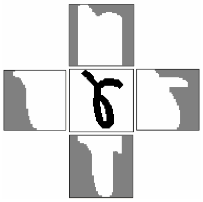


**Figure 8.** Features extraction of the character image based on profiles.

### 3.2. Dimensionality reduction

Once the character has been represented as a feature vector, a classification rule has to be defined in order to classify it into one among the letters. Even though machine learning literature provides us a wide choice of learning algorithms to construct classifiers based on training data, these have limited capabilities to construct the optimal one, depending on particularities of the distributions of the classes in the feature space. An important such limitation, evident to algorithms giving each feature direction equal importance, concerns the sensitivity to the number of features comprising the feature vector. These classifiers, such as the K Nearest Neighbors (K-NN) [18] and the Support Vector Machine (SVM) using the Radial Basis Function (RBF) kernel [19], tend to be disturbed by noisy features, i.e. features that contain no useful information concerning the separability of classes.

To that end, our methodology for character recognition also considered a dimensionality reduction step, according to which the dimension of the feature space, engendered by the features extracted using the methodology described in section 3.1, is lowered down to comprise only the features pertinent to the discrimination of characters into the given set of letters. In particular, we employed the Linear Discriminant Analysis (LDA) method [19], according to which the most significant linear features are those where the samples distribution has important overall variance while the samples per class distributions have small variance. Formally, this criterion is represented as

$$\mathrm{LDA}(w) = \frac{w^T \,\mathrm{Cov}(X) w}{w^T \mathrm{E}_c\left[\mathrm{Cov}(X \mid c)\right] w} \tag{4}$$

where $w$ represents a linear combination of the original features, $X$ the original feature vector, $c$ the class, $Cov$ is a the covariance matrix that has to be estimated from the samples and $E_c$ is the expectation in respect to the classes. It turns out that finding the linear features that maximize the LDA criterion comes down to solving a generalized eigenvalue/eigenvector problem and keeping the eigenvectors that have greater eigenvalues. Moreover, the ratio of the sum of the eigenvalues kept to the overall eigenvalues sum provides as an index of quality of the feature subspace kept.

## 4. Experimental Results

For our experiments the Greek handwritten character database was used. After size normalization some characters such as the uppercase "O" and the lowercase "o" are considered to be the same. So for having meaningful results we merged these two classes into one, by randomly selecting 660 characters form both classes. This was done to a total of 10 pair of classes, as shown in Table 1. This concluded in having 46 classes each one containing 660 patterns and the database now has 46x660=30,360 characters. Moreover, 1/5 of each class was used for testing and the 4/5 for training.

**Table 1.** Merged Classes

| | Uppercase | Lowercase |
|---|---|---|
| **1** | **Ε** | **ε** |
| **2** | **Θ** | **θ** |
| **3** | **Κ** | **κ** |
| **4** | **Ο** | **ο** |
| **5** | **Π** | **π** |
| **6** | **Ρ** | **ρ** |
| **7** | **Τ** | **τ** |
| **8** | **Φ** | **φ** |
| **9** | **Χ** | **χ** |
| **10** | **Ψ** | **ψ** |

As it has already been described in section 3 we have used a size normalization step before feature extraction. During this step, the size of the normalized character images used is $x_{max}$=60 and $y_{max}$=60. In the case of features based on zones, the character image is divided into 5 horizontal and 5 vertical zones forming a total of 25 blocks with size 12x12 (see Fig. 4). Therefore, the number of features based on zones is 25. In the case of features based on character (upper/lower) profile projections the image is divided into 10 vertical zones (see Fig. 5). Similarly, the normalized image is divided into 10 horizontal zones (see Fig. 6). Therefore, the total number of features based on profile projections is 40. For the third type of features the distances from upper and lower boundaries and from left and right boundaries are calculated along 5 vertical and along 5 horizontal lines respectively (see Fig.7). So, the number of corresponding features is 20. Finally, since the character image is normalized to a 60x60 matrix, the total number of features extracted from the profiles is 4x60=240 (see Fig. 8). Combining all the above feature extraction schemes led to a 325-dimensional feature vector.

For the classification step three well known classifiers were used; the Naïve Bayes [20], the k-NN and the SVM. As shown in Fig. 9 their performance depends on the number of features used. Moreover, all three classifiers perform better when they use approximately 50 out of 325 features. Finally, the SVM seems to achieve considerable higher levels of recognition accuracy than the other two classifiers. The best performance (92,05 %) is attained with the SVM using 40 features.

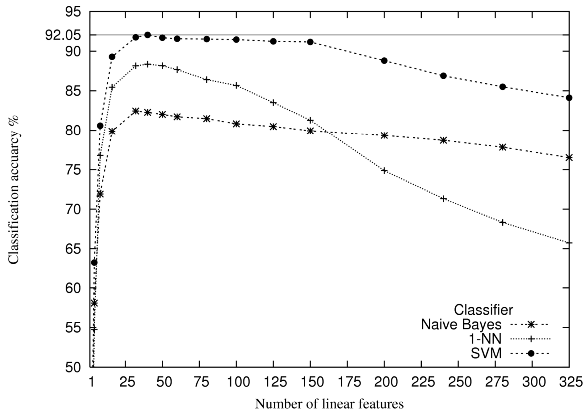


**Figure 9.** Classification accuracy for all three classifiers depending on the number of features used.

## 5. Conclusions

In this paper, we propose an off-line OCR methodology for isolated Greek handwritten characters based on feature extraction scheme followed by a dimensionality reduction step, which seems to improve the recognition accuracy.

Our future research will focus on exploiting new features as well as fusion methods to further improve as well as on the creation of new hierarchical classification schemes based on rules extracted after examining the corresponding confusion matrix.

## Acknowledgements

This research is carried out within the framework of the Greek Ministry of Research funded R&D project POLYTIMO [21] which aims to process and provide access to the content of valuable historical books and handwritten manuscripts.

## References

[1] A. S. Britto, R. Sabourin, F. Bortolozzi, C. Y. Suen, “Foreground and Background Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings”, *9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9*), 2004, pp. 371-376.

[2]O. D. Trier, A. K. Jain, T.Taxt , “Features Extraction Methods for Character Recognition – A Survey”, *Pattern Recognition*, 1996, Vol.29, No.4, pp. 641-662.

[3] J. A. Fitzgerald, F. Geiselbrechtinger, and T. Kechadi, “Application of Fuzzy Logic to Online Recognition of Handwritten Symbols”, *9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 9)*, 2004, pp. 395- 400.

[4] N. Arica and F. Yarman-Vural, “An Overview of Character Recognition Focused on Off-line Handwriting”, *IEEE Transactions on Systems, Man, and Cybernetics*, Part C: Applications and Reviews, 2001, 31(2), pp. 216 - 233.

[5] Luiz S. Oliveira, F. Bortolozzi, C.Y.Suen, “Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy”, *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2001, Vol. 24, No. 11, pp. 1448-1456.

[6] K. M. Mohiuddin and J. Mao, “A Comprehensive Study of Different Classifiers for Handprinted Character Recognition.” *Pattern Recognition*, Practice IV, 1994, pp. 437- 448.

[7]A. L. Koerich, “Unconstrained Handwritten Character Recognition Using Different Classification Strategies.” *International Workshop on Artificial Neural Networks in Pattern Recognition* (ANNPR), 2003.

[8] J. H. Kim, K. K. Kim, C. Y. Suen, “Hybrid Schemes Of Homogeneous and Heterogeneous Classifiers for Cursive Word Recognition”, *7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, 2000, pp 433 - 442.

[9] E. Kavallieratou, N. Fotakis, G Kokkinakis, “Handwritten Character Recognition Based on Structural Characteristics”, *ICPR, 16th International Conference on Pattern Recognition (ICPR'02),* 2002, pp. 139-142.

[10] Kuhl, F.P., Giardina, “Elliptic Fourier features of a closed contour”, *Comput. Vis. Graphics Image Process.* 18, 1982, pp 236-258.

[11] R. C. Gonzalez, R. E. Woods, “Digital Image Processing, Second Edition”, *Prentice Hall*, 2002.

[12] Hu, “Visual pattern recognition by moment invariants”. *In IRE Trans. Inf. Theory* 8, 1962, pp. 179-187.

[13] Wilkinson, J. Geist, S. Janet, P. Grother, C. Burges, R. Creecy, B. Hammond, J.Hull, N. Larsen, T. Vogl, and C. Wilson. The first census optical character recognition systems conf. #NISTIR 4912, The U.S. Bureau of Census and the National Institute of Standards and Technology, Gaithersburg, MD, 1992.

[14]Hull, “A database for handwritten text recognition research”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1994, Volume 16, Issue 5, pp. 550 – 554.

[15] C. Y. Suen, C. Nadal, R. Legault, T. Mai, and L. Lam , “Computer recognition of unconstrained handwritten numerals”, *Proc. of the IEEE*, 1992, Volume 7, Issue 80,pp. 1162-1180.

[16] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, “Unipen project of on-line data exchange and benchmarks”, *Proc. of the 12th IAPR Int. Conf on Pattern Recognition*, Jerusalem, Israel, Oct. 1994, pp. 29-33.

[17]E.Kavallieratou, N.Liolios, E.Koutsogeorgos, N.Fakotakis, G.Kokkinakis, “The GRUHD database of Modern Greek Unconstrained Handwriting”, *In Proc. ICDAR*, 2001.

[18] Theodoridis, S., and Koutroumbas, K., *Pattern Recognition*, Academic Press, 1997.

[19] Cortes C., and Vapnik, V., “Support-vector network”, *Machine Learning*, vol. 20, pp. 273-297, 1997.

[20] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification, Second Edition*, Wiley, 2000.

[21]POLYTIMO project,
http://iit.demokritos.gr/cil/Polytimo, 2007.