Choosing Shape Features by means of Genetic Algorithms for Glyph-Clustering of Historical Documents

Jan-Hendrik Worch
Centre for Computing Technologies (TZI)
University of Bremen
Am Fallturm 1
28359 Bremen
Germany

Björn Gottfried
Centre for Computing Technologies (TZI)
University of Bremen
Am Fallturm 1
28359 Bremen
Germany

ABSTRACT

The solution for a feature selection problem is presented in the field of document image processing. The choice of shape features for describing glyphs of historical documents is a non-trivial task since the variations of glyphs in different documents is innumerable. Hence, the manual selection of shape features would be a cumbersome task. To select a subset of features from a given set a genetic algorithm is used which optimises the result of a clustering process by x-means. The result of x-means is evaluated by using different quality measures. The optimisation methodology is illustrated within a case study, in which the selection of an appropriate set of features is a crucial part of the system. The intended application supports a user who is transcribing historical documents by showing him similar occurrences of a given glyph.

Keywords:

Document Image Processing, Genetic Algorithms, Feature Selection, Shape Descriptions, Glyph Clustering, X-Means

1. INTRODUCTION

In the last two decades application systems based on computer vision became more and more specialised. At the same time, a great variety of different features have been introduced in order to describe the image contents. The selection of features to be employed usually depends on the specific application and context. As a consequence, each software system to be developed faces the problem of how to choose an appropriate set of features in order to solve a given problem.

In this paper it is investigated how genetic algorithms can help in feature selection in such a specialised application context, and more specifically in the analysis of historical document images. More precisely, an appropriate set of shape features is to be found which describes glyphs in printed and even in handwritten document images. Glyphs describe the visual appearance of characters. The visual description of characters can in particular be applied to the problem of glyph spotting, which is an important step in the analysis of document images where established techniques of optical character recognition (OCR) fail. Instead, visually similar glyphs are clustered into equivalence classes without referring to any background knowledge.

In the following, the clustering method which has been used is explained, followed by the description of a genetic algorithm in order to find a valuable set of features. A case study is presented together with an evaluation resulting in a few sets of feature configurations for glyph clustering. In the last two sections the results are discussed and a summary completes the paper.

2. METHOD

2.1 Clustering using X-Means

Typically OCR processes employ classification algorithms. Within the context of analysing historical documents, however, there is frequently a lack of a priori knowledge, so that classification algorithms fail. Then, the solution consists in clustering visually similar glyphs. In the present case, a clustering algorithm is used which is able to calculate the ideal number of clusters. This number is determined by the different character classes of the document at hand.

A clustering algorithm which is able to calculate the number of necessary clusters is the x-means algorithm by Pelleg and Moore [17]. It is an extension of the well-known k-means algorithm by MacQueen [14]. Whereas k-means is bounded to k clusters, x-means calculates the best-scoring model within an upper and a lower bound for k.

The rough process of x-means is given by Algorithm 1. The call of `Improve-Params` performs k-means until convergence, whereas `Improve-Structure` determines whether and where to split the existing clusters. Thus, for each cluster a local k-means with $k = 2$ is performed. If the score of the result is better than before, the result is used for further processing, otherwise the former partitioning is restored. The score is calculated based on the Bayesian Information Criterion [17].

Algorithm 1 X-Means [17]

```
Require: lower bound $k_{min}$, upper bound $k_{max}$
  $k = k_{min}$
  while $k \leq k_{max}$ do
    Improve-Params
    Improve-Structure
  end while
  return best-scoring model
```

Later on, a method is required in order to compare how good different clustering results are. For this purpose, a widely used measure is applied. namely the accuracy α which is defined as

$$\alpha(x) = \frac{n-e}{n} \tag{1}$$

with x being a clustering result, n the number of glyphs to be clustered, and e the number of errors in the result. Using just the accuracy, the result of a clustering process would lead to a high rating only for clusters of size one. Hence, there is a need for a measure, which takes the compression of the result into account. The compression rate γ is defined as

$$\gamma(x) = 1 - \frac{c}{n} \tag{2}$$

where c is the number of clusters in the result and n is the number of glyphs.

The quality of the clustering can in particular be calculated by looking at the number of errors made in a sequence of 100 words (ε). This can be done by using

$$\varepsilon(x) = \frac{e \cdot 100}{\frac{n}{a}} \tag{3}$$

where e is the number of errors, n is the number of glyphs, and a the average number of glyphs per word. In the present case it holds that $a = 5.9$.

2.2 Feature Selection

In principle, it is imaginable to use a set of features of arbitrary size to cluster objects. However, the more features deployed, the more time is necessary for the clustering process. Another problem which might arise is that a large number of features could have a generalising effect, i. e. the features would not be discriminatory any longer. To avoid those problems, a genetic algorithm is employed to select a subset of features from a given set.

Genetic algorithms were first introduced by Holland [12], who tried to analyse natural selection and who wanted to integrate those mechanisms into computer systems. The development of genetic algorithms is inspired by Darwin's evolution theory, i. e. the underlying mechanism include *selection*, *recombination* and *mutation* [15]. Additionally, genetic algorithms consist of populations and individuals. Each individual is part of a population and represents one possible solution for a given problem.

In contrast to many other feature selection algorithms, theoretically genetic algorithms can find the optimal solution for a problem. In fact, this is the reason why it has been decided to use such an algorithm to select a subset of features.

2.2.1 The Genetic Algorithm. The implementation of the genetic algorithm is based on the genuine algorithm shown in Algorithm 2.

Algorithm 2 Genetic Algorithm

```
Require: SizeOfPopulation $p$, NumberOfChilds $c$: $c$ mod 2 = 0,
  probabilityOfRecombination $p_r$
  $t \leftarrow 0$
  $P(t) \leftarrow$ initialisePopulation($p$)
  evaluate($P(t)$)
  while $t < stop$ do
    $P' \leftarrow$ selectForVariation($P(t)$)
    $P'' \leftarrow \emptyset$
    for $i = 1 \rightarrow k/2$ do
      $r \leftarrow random([0,1])$
      if $r < p_r$ then
        $(B,C) \leftarrow$ Recombination($A^{2i-1}$,$A^{2i}$)
      else
        $(B,C) \leftarrow (A^{2i-1},A^{2i})$
      end if
      $B \leftarrow$ Mutation($B$)
      $C \leftarrow$ Mutation($C$)
      $P'' \leftarrow P'' + B, C$
    end for
    evaluate($P''$)
    $t \leftarrow t+1$
    $P(t) \leftarrow$ selectForSurvival($P(t-1), P''$)
  end while
  return best individual of $P(t)$
```

2.2.1.1 Encoding. An individual consists of several *genes*. Those genes represent the genetic code of an individual. Here, the encoding of an individual is a string of '0's and '1's which means that a feature is used for a solution when the corresponding value is set to '1'.

2.2.1.2 Fitness. The fitness function is a crucial part of each genetic algorithm. On the basis of the fitness function one determines how good (*fit*) an individual is. The fitness function depends on the specific problem. In the present case, the result of the clustering is used to determine the fitness of an individual i, which leads to the following fitness function

$$f(i) = \alpha + \gamma - \frac{1}{590}\varepsilon \tag{4}$$

where α is the accuracy of the clustering result, γ is the compression rate and $\frac{1}{590}\varepsilon$ is the error ratio. ε is normalised with $\frac{1}{590}$ because 590 is the maximum number of errors per 100 words (see section 2.1).

2.2.1.3 Selection. For the presented approach it has been decided to deploy the so-called *Tournament-Selection* [4]. Using this selection one randomly choose $q > 1$ individuals whose fitness will be compared. Only the strongest individual will survive, but any loosing individuals can still be selected for later comparisons.

The problem of this selection method is that one might loose the best individuals. This is why it has been decided to combine this selection strategy with *Elitist Selection*. Using the elitist selection method one assures that the best individual(s) will survive.

2.2.1.4 Recombination. The main operator of any genetic algorithm is the recombination. It is responsible for the creation of the child population. A process which is often used to generate children is the *One-Point-Crossover* [15] approach. This approach splits each individual at a specific chromosome and two children are generated by recombining the divided parts of two individuals. The following gives an example:

Parents	Crossover	Children
$A = \mathbf{0100}\|\mathit{1101}$	$\rightarrow$	$A' = \mathbf{0100}\|\mathbf{0001}$
$B = \mathit{1011}\|\mathbf{0001}$		$B' = \mathit{1011}\|\mathit{1101}$

2.2.1.5 Mutation. The mutation randomly modifies the chosen genes. From the possible methods the so-called *Bit-Flip-Mutation* is applied. For this purpose, a random value for each gene is generated. If the value is below a certain threshold, the gene is flipped, i. e. '0' gets '1' and vice versa.

Without the mutation the genetic algorithm will presumably get stuck in a local optimum.

3. EVALUATION

3.1 Case Study

The offline analysis of mediaeval handwritings is a challenging task due to numerous kinds of degradations found in old documents. Hence, the standard pipeline in document image processing can hardly be realised. Instead, an assistance system is under development which supports a user in transcribing a given handwriting.

When trying to automatically process mediaeval documents one is confronted with the problem of separating handwritten glyphs. In [22] a first approach is presented on how to spot single glyphs in medieval handwritings. Spotting for similar glyphs is useful when transcribing a glyph whose meaning is not obvious within the given context. So, similar occurrences might be taken into account to transcribe the glyph at hand. That is, those occurrences help the user by determining the meaning of the according glyph.

So far the present system allows a user to extract a single glyph manually and it lets him search for similar occurrences. For this purpose the correlation coefficient is applied with postprocessing filters to sort out false positives. The mentioned processing is based on the comparison of different shape descriptions and a filtering by size. In contrast to that, this paper deals with several experiments based on the aforementioned feature based clustering approach combined with the genetic algorithm. However, since the correlation coefficient combined with the filtering by size showed to be a useful base for further processing, this approach is going to be used as a first step within the whole methodology.

3.2 Application

For the experiments a set of 56 features has been implemented, out of which the genetic algorithm has to select successful configurations. The set of features consists of 29 features which are extracted from the original sized but greyscale glyph image and 27 features which are calculated for glyph images which are normalised in size to 32 × 32 pixel by preserving the image aspect ratio. Those features were extracted for two different document images: a handwritten page of the 11th century [2] and a printed one from 1869 [1]. Figures 1 and 2 depict examples for both document images. From the former image 1523 glyph images have been extracted and 2233 glyph images from the latter (cf. Figure 2).

The genetic algorithm was used to optimise the result of the feature based clustering. Table 5 in the appendix lists all features including references and the five configurations which led to the best results. As mentioned above, there are altogether 56 different features for the genetic algorithm. This leads to a number of

$$2^{56} = 72.057.594.037.927.936 \tag{5}$$

possible combinations of features. For the experiments 100 generations with 20 individuals per population are computed. In total the best results of the genetic algorithm use 21 out of the 56 features (distinguishing between original image size features and size normalised image features). With the exception of one configuration, the configurations are a combination of region based and polygon based features. In fact, this appears obvious since features of the insides of objects and their contours describe fundamentally different aspects which complement each other.

As mentioned in section 3.1, in [22] an approach is presented to spot similar glyphs based on the comparison of different shape features. The results of that approach can be seen in Tables 1 and 3 whereas the results of the new approach are shown in Tables 2 and 4. The tables show the precision and recall values of the different approaches. In order to improve the possibility of comparing the results, the $f_{0.5}$-measure has been calculated for each approach. Since the focus of the application is on precision, it gets double-weighted. A high recall is not as necessary as a high precision because when searching for similar glyphs, even just a few similar occurrences might be helpful in order to get familiar with a selected glyph.

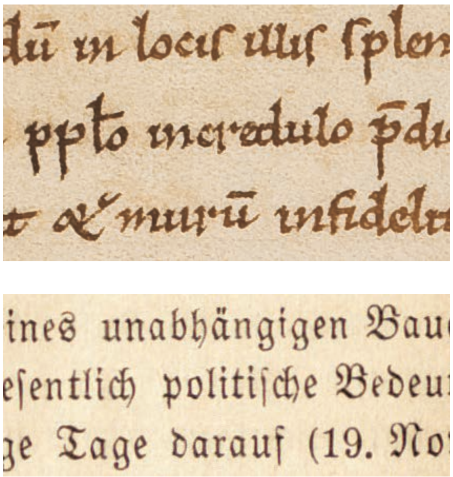

Fig. 1: Samples from [2] (top) and [1] (bottom)

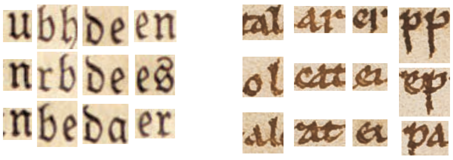

Fig. 2: Glyphs from [1] (left) and [2] (right)

One might notice the different recall values, even though this evaluation is concerned with clustering. After clustering only the cluster in which the template is located is evaluated, since the clustering operates as some kind of filtering process. This means that the cluster in which the template is located is going to be displayed.

4. RESULTS AND DISCUSSION

In [22] different thresholds for the correlation coefficient have been investigated. It shows that low thresholds result in too many possible matches where most of the matches were false positives. By contrast, high thresholds were too restrictive and the recall fell down too steep.

As tables 1 and 3 show, the results in [22] suffer from a low recall. Nevertheless the results are good enough to handle the problem of spotting similar glyphs to support a user in transcribing single glyphs, since already a few similar occurrences can help a user to recognise the glyph.

Table 2 shows the results of the different configurations achieved by the genetic algorithm for [2]. As the $f_{0.5}$-measure values show, the best result for clustering after applying the correlation coefficient with a threshold of 175 outperforms the approach presented in [22]. By contrast, using a threshold of 200 for template matching,

Table 1. : Results when applying the approach presented in [22] to [2].

	Threshold 175			Threshold 200		
	Precision	Recall	$F_{0.5}$-measure	Precision	Recall	$F_{0.5}$-measure
TM	0.12	0.78	0.14	0.56	0.53	0.55
TM+Size+Skeleton	0.89	0.12	0.39	0.90	0.09	0.32
TM+Size+Moments	**0.43**	**0.42**	**0.43**	**0.80**	**0.31**	**0.61**
TM+Size+Polyline	0.70	0.14	0.39	0.88	0.13	0.41

Table 2. : Results when applying the optimised feature based clustering approach to [2].

	Threshold 175			Threshold 200		
	Precision	Recall	$F_{0.5}$-measure	Precision	Recall	$F_{0.5}$-measure
TM+Size	0.39	0.53	0.41	**0.76**	**0.37**	**0.63**
Config 1	0.71	0.36	0.59	0.93	0.20	0.54
Config 2	0.67	0.29	0.53	0.85	0.20	0.52
Config 3	0.63	0.32	0.53	0.87	0.19	0.51
Config 4	**0.75**	**0.36**	**0.62**	0.87	0.20	0.52
Config 5	0.70	0.34	0.58	0.88	0.19	0.51

the following clustering result becomes poorer than when taking template matching and size filtering alone. The low improvement rate can be explained by the high intraclass variance in mediaeval handwritings (as shown in Figure 2).

Table 4 shows the results of the different configurations achieved by the genetic algorithm for [1]. In that case all of the different clustering results outperform the approach presented in [1]. For example the best result (Configuration 1) improves the result of [22] by 48%.

Figure 3 shows the five configurations sorted from the left to the right by the number of features involved. The numbers represent the shape features according to Table 5. For configurations 2 and 5 the white slices predominate and for configurations 1 and 4 the black slices predominate. This shows that neither the original nor the size normalised images are more relevant for the best five configurations found. Moreover, the number of features contained in the configurations varies from 2 to 14. Compared to the 56 possibilities the best configurations are confined to only a few of the possible features. Some shape features, such as the betweenness feature of the largest polygon [7] and the projections along the half of the images [10] only occur in a single configuration, while the other features which are present in the best configurations occur more often in different configurations. The most frequent feature is the vector of Hu-moments [13] that occurs in four configurations, however altogether seven times because in three configurations it occurs two times, applied to the original image and also applied to the size normalised image. Such feature doublets also occur for the enclosedness feature [21] as well as for the stroke direction distribution [16].

5. SUMMARY

In conclusion, it has been shown that a genetic algorithm for feature selection is a successful approach in the domain of clustering glyphs of historical documents. The genetic algorithm employed selects a subset from a set of 56 implemented features. The algorithm led to useful results and reduced the number of necessary features to create a meaningful shape description.

Compared to the best results in [22] the precision is improved up to 43%, the recall up to 63%, and in total, according to f-measure, an improvement of up to 48% is achieved.

Acknowledgements

This work has been funded by the German Research Foundation, DFG, under Grant number GO 2023/3-2.

6. REFERENCES

[1] Die Grenzboten, 28. Jahrgang, 2. Semester 1. Band, 1869. Scan 27 von der Staats- und Universitätsbibliothek Bremen.

[2] SBPK Berlin, Philllipps 1870, fol. 11v, 1870.

[3] W. Burger and M. J. Burge. *Principles of digital image processing: Core algorithms*. Springer, London, 2009.

[4] D. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan-Kaufmann, 1991.

[5] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Addison-Wesley, Reading, Mass., [3. ed.] reprint. with corr. edition, 1992.

[6] B. Gottfried. Qualitative similarity measures - the case of two-dimensional outlines. *Computer Vision and Image Understanding*, 110(1):117–133, 2008.

[7] B. Gottfried. *Representing Material Objects by Qualitative Spatial Representations*. Universität Bremen, 2008. Unpublished Habilitation.

[8] B. Gottfried, A. Schuldt, and O. Herzog. Extent, extremum, and curvature: Qualitative numeric features for efficient shape retrieval. In Joachim Hertzberg, Michael Beetz, and Roman Englert, editors, *KI 2007: Advances in Artificial Intelligence*,

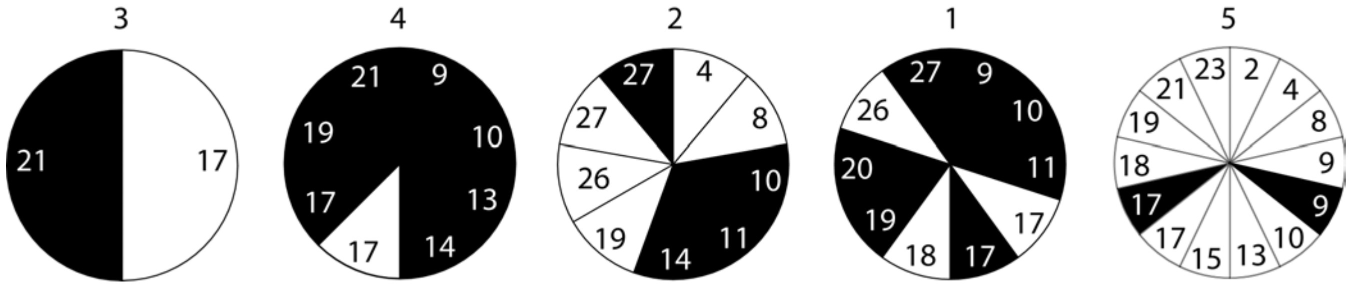

Fig. 3: The five best configurations. The number of pie slices indicates the number of shape features of the according configuration. The numbers within each slice index the features according to Table 5. A white slice stands for a feature applied to the original image, while a black slice shows that the according feature has been applied to a size normalised image.

volume 4667 of *Lecture Notes in Computer Science*, pages 308–322. Springer Berlin / Heidelberg, 2007.

[9] T. K. Ho. Random decision forests. In *Proceedings of the second International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

[10] T. K. Ho and H. S. Baird. Perfect metrics. In *Proceedings of the second International Conference on Document Analysis and Recognition*, pages 593–597, 1993.

[11] T. K. Ho and H. S. Baird. Large-scale simulation studies in image pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1067–1079, 1997.

[12] J. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.

[13] M.-K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[14] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp.*, volume 1, pages 281–297, 1967.

[15] P. Merz. *Memetic Algorithms for Combinatorial Optimization Problems*. Dissertation, Universität-Gesamthochschule Siegen, 2000.

[16] S. Mori, C. Y. Suen, and K. Yamamoto. Historical review of ocr research and development. In *Proceedings of the IEEE*, volume 80, pages 1029–1058, July 1992.

[17] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proc. 17th Int. Conf. Machine Learning*, pages 727–734, 2000.

[18] T. H. Reiss. The revised fundamental theorem of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):830–834, August 1991.

[19] A. Schuldt, B. Gottfried, and O. Herzog. Towards the visualisation of shape features the scope histogram. In C. Freksa, M. Kohlhase, and K. Schill, editors, *KI 2006: Advances in Artificial Intelligence*, volume 4314 of *Lecture Notes in Computer Science*, pages 289–301. Springer Berlin / Heidelberg, 2007.

[20] G. Vamvakas, B. Gatos, and S. J. Perantonis. A novel feature extraction and classification methodology for the recognition of historical documents. In *10th International Conference on Document Analysis and Recognition*, pages 491–495, 2009.

[21] J.-H. Worch. VaBene – Validierung eines Benchmarks zur Evaluation von Formmerkmalen für Glyphen. Diploma thesis, Universität Bremen, September 2011.

[22] J.-H. Worch, M. Lawo, and B. Gottfried. Glyph spotting for mediaeval handwritings by template matching. In *Proceedings of the 12th ACM symposium on Document engineering*, DocEng '12, New York, NY, USA, 2012. ACM.

[23] R. Xu and O. A. Di Guida. Comparison of sizing small particles using different technologies. *Powder Technology*, 132(2-3):145 – 153, 2003.

Table 3. : Results when applying the approach presented in [22] to [1].

	Threshold 175			Threshold 200		
	Precision	Recall	$F_{0.5}$-measure	Precision	Recall	$F_{0.5}$-measure
TM	0.05	1.00	0.06	0.15	1.00	0.18
TM+Size+Skeleton	0.55	0.26	0.45	0.62	0.26	0.49
TM+Size+Moments	0.28	0.66	0.32	0.36	0.63	0.39
TM+Size+Polyline	**0.65**	**0.36**	**0.56**	**0.74**	**0.36**	**0.61**

Table 4. : Results when applying the optimised feature based clustering approach to [1].

	Threshold 175			Threshold 200		
	Precision	Recall	$F_{0.5}$-measure	Precision	Recall	$F_{0.5}$-measure
TM+Size	0.28	0.80	0.32	0.45	0.79	0.49
Config 1	**0.93**	**0.59**	**0.83**	**0.92**	**0.57**	**0.82**
Config 2	0.84	0.65	0.79	0.88	0.52	0.77
Config 3	0.80	0.56	0.74	0.88	0.56	0.79
Config 4	0.86	0.57	0.78	0.90	0.55	0.80
Config 5	0.87	0.58	0.79	**0.91**	**0.58**	**0.82**

Table 5. : The 5 best configurations found by the genetic algorithm. OI means that a feature is extracted for the original sized image, whereas SN means that a feature is extracted for a size normalised image. For polygon based features, (a) means the feature is extracted for the largest polygon (e. g. the main body of the letter 'i' instead of its dot at the top) and (b) means the average of all features from all polygons is taken (in most cases there is only one polygon).

	Config 1		Config 2		Config 3		Config 4		Config 5		
Feature	OI	SN	OI	SN	OI	SN	OI	SN	OI	SN	Reference
01. Aspect Ratio	–	–	–	–	–	–	–	–	–	–	[11]
02. Betweenness (a)	–	–	–	–	–	–	–	–	–	✓	[7]
03. Betweenness (b)	–	–	–	–	–	–	–	–	–	–	[21]
04. Black Pixel Density	–	–	✓	–	–	–	–	–	✓	–	[11]
05. Contour-Boundingbox Distance	–	–	–	–	–	–	–	–	–	–	[10]
06. Curvature (a)	–	–	–	–	–	–	–	–	–	–	[6, 8]
07. Curvature (b)	–	–	–	–	–	–	–	–	–	–	[21]
08. Eccentricity	–	–	✓	–	–	–	–	–	✓	–	[5]
09. Enclosedness (a)	–	✓	–	–	–	–	–	✓	✓	✓	[21]
10. Enclosedness (b)	–	✓	–	✓	–	–	–	✓	–	✓	[21]
11. Euler Number	–	✓	–	✓	–	–	–	–	–	–	[5]
12. Extent (a)	–	–	–	–	–	–	–	–	–	–	[6, 8]
13. Extent (b)	–	–	–	–	–	–	–	✓	–	✓	[21]
14. Extremum (a)	–	–	–	✓	–	–	–	✓	–	–	[6, 8]
15. Extremum (b)	–	–	–	–	–	–	–	–	–	✓	[21]
16. Heywood Diameter	–	–	–	–	–	–	–	–	–	–	[23]
17. Hu-Moments	✓	✓	–	–	✓	–	✓	✓	✓	✓	[13]
18. Image Width, Height	✓	–	–	–	–	–	–	–	✓	–	[11]
19. Orientation	–	✓	✓	–	–	–	–	✓	–	✓	[3]
20. Pixel Correlation	–	✓	–	–	–	–	–	✓	–	–	[9]
21. Pixel Values	–	–	–	–	–	✓	–	–	–	✓	[9]
22. Projection Full	–	–	–	–	–	–	–	–	–	–	[3]
23. Projection Half	–	–	–	–	–	–	–	–	✓	–	[10]
24. Projection Half Overlapping	–	–	–	–	–	–	–	–	–	–	[21]
25. Reiss Moments	–	–	–	–	–	–	–	–	–	–	[18]
26. Scope Histogram	✓	–	✓	–	–	–	–	–	–	–	[19]
27. Stroke Direction Distribution	–	✓	✓	✓	–	–	–	–	–	–	[16]
28. Subsample	–	–	–	–	–	–	–	–	–	–	[11]
29. Vamvakas	–	–	–	–	–	–	–	–	–	–	[20]