A Texture-based Method for Document Segmentation and Classification

Ming-Wei Lin, Jules-Raymond Tapamo, Baird Ndovie

A Texture-based Method for Document Segmentation and Classification

M-W Lin, J-R Tapamo, B Ndovie

School of Computer Science, University of KwaZulu-Natal, Durban 4041, South Africa

ABSTRACT

In this paper we present a hybrid approach to segment and classify contents of document images. A Document Image is segmented into three types of regions: Graphics, Text and Space. The image of a document is subdivided into blocks and for each block five GLCM (Grey Level Co-occurrence Matrix) features are extracted. Based on these features, blocks are then clustered into three groups using K-Means algorithm; connected blocks that belong to the same group are merged. The classification of groups is done using pre-learned heuristic rules. Experiments were conducted on scanned newspapers and images from MediaTeam Document Database.

KEYWORDS: Information Retrieval, Document Image Analysis, Texture segmentation, Grey Level Co-occurrence Matrix (GLCM), K-Means Clustering, Feature extraction

1 INTRODUCTION

Paper has been and still is one of the major medias used by people to share knowledge around the world. Although the large amount of documents are nowadays created using computer, there are still situations (structuring of archives documents, processing of scanned documents) where there is a need to process document images in order to use them in a more adequate way. Storing document into an appropriate digital format makes it easier to retrieve and analyze the content.

Document segmentation is a critical step in document processing, which is done according to some criterion of homogeneity; the criterion of homogeneity used, in the context of document analysis, is the connected zones of text, graphics and space. This process often acts as a prelude to other document processes, amongst which we have OCR (Optical Character Recognition) of text regions, graphics extraction from newspaper [1], text strings extraction from engineering drawings [2, 3] and pornography graphics analysis [4]. To get a good performance from these processes, document segmentation is one of the important factors, hence robust and efficient techniques are required for this process.

Several approaches for text/graphics separation in a document have been proposed and examined [2, 5, 6, 7, 8, 9]. Texture analysis is one of the possible solutions to segment different contents of the document; it has been used in many applications to detect, segment and classify images based on local spatial variations of intensity and color [10, 11, 12]. Texture is believed to be an important feature to discriminate the contents of zones. Different local textures in an image can describe different physical characteristics corresponding to different parts of a surface.

Human vision can distinguish the zones of text and graphics far away before clear graphics and text letters can be read, and this shows that contents of the document can be recognized by using texture features.

Many systems based on these approaches have some limitations. Most of them only perform classification into text and non-text zones. In some cases the processing time is relatively high or there are lots of assumptions. In many applications it is important to be able to extract text, graphics and space zones. For instance it could be interesting to find a correlation between the text and the graphics that are on the same document, or to classify a document based on graphics of documents.

In this paper we present a new approach, based on GLCM texture characterization, K-means clustering and adaptive heuristic rules, that segments and classifies zones of documents into one of the 3 categories: text, graphics, space. The rest of the paper is organized as follows: related previous works for document segmentation and zone classification are presented in section two, section three describes our approach, experimental results are presented in section four, and future works and conclusion are provided in the last section.

2 RELATED WORKS

There are three main approaches to document segmentation: bottom-up [6, 13], top-down [14] and hybrid [5, 7, 8, 9, 15, 16].

Email: M-W Lin linm@ukzn.ac.za, J-R Tapamo tapamoj@ukzn.ac.za, B Ndovie ndovieb@ukzn.ac.za

The process using bottom-up techniques starts from pixel level, pixels are then merged into larger components such as homogenous square blocks; connected blocks that have the same characteristics are then merged to form homogeneous regions.

When compared to top-down techniques, bottom-up techniques are more efficient when it comes to handling complex layout documents, but have the disadvantage of having a high processing time.

Top-down techniques, like X-Y cuts [14], proceed by starting from the whole image and split it recursively into different zones until regions in the zone share the same features. Top-down techniques are efficient for good layout structured documents but often fail in complex layout. There are also hybrid techniques that mix the two previously mentioned techniques. Segmentation using texture analysis falls under the latter category.

Zone classification of a document aims to discriminate the content of the zones into one of the predefined categories, such as graphics, text, space, noise and tables. The value for each feature is extracted from samples and trained using different Artificial Intelligence techniques. The knowledge extracted is then used to classify zones of the document. Duong et al. [16] simply use some criteria on entropy to classify zones into text or non text zones. Problems for text region extraction have been well studied. Other than traditional bottom-up and top-down approaches, various ideas have been proposed to extract text regions from different document categories. Zheng et al. [7] segment and classify black-white noisy document into three different categories: machine printed text, handwriting text and noise; connected components are extracted based on geometric proximity and size. Components are then classified into one of the three categories using 31 features selected from 140 features and trained by Fisher classifier. Features like region size, stroke orientation, stroke length, run-length histogram and texture of the component, etc. are widely examined. Duong et al. [16] extract text regions from printed document images; a binary image is built using features of cumulative horizontal gradient, bounding boxes of the components in the binary image are then extracted as potential text regions, potential regions are classified as text or non text regions using geometric and entropy texture features. Yuan and Tan [6] extract text regions from a gray scale document image using edge information by performing Canny edge detection on the image; and edges that have similar orientations are then merged into larger edge by performing horizontal smoothing. Straight lines are fitted into these edges to construct a bounding box by using some defined heuristics rules to find the best matching pair of the straight lines. Such approaches show efficiency and fast processing speed, but it is assumed that all texts are oriented in the same direction in the image. In the next section we present our method.

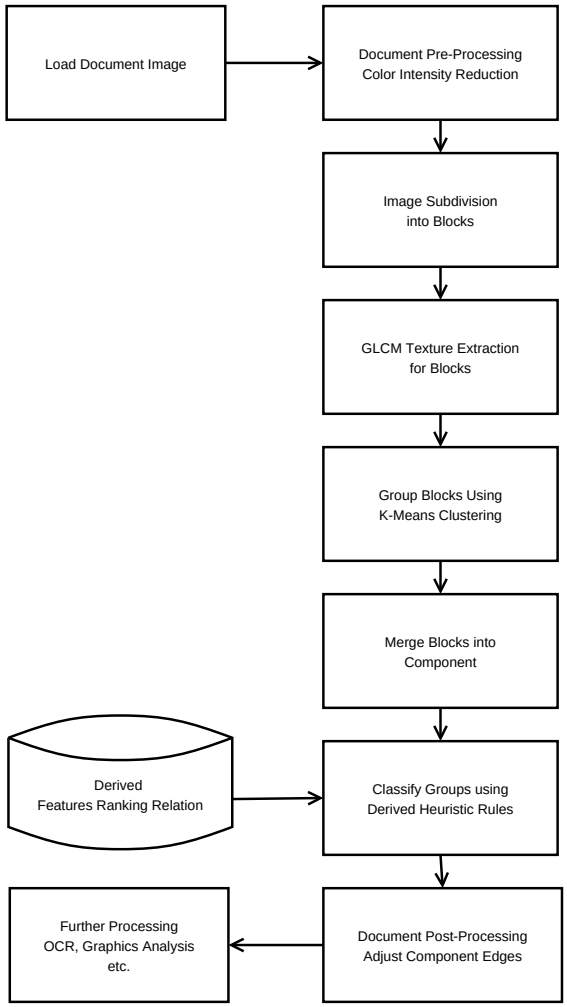

Figure 1: Flowchart of the system

3 METHOD DESCRIPTION

The algorithm segments and classifies regions of the document into three categories (text, graphics, space) using GLCM texture features. Document segmentation is done by performing texture analysis and zone classification is done by using heuristic rules learned from sample images.

For segmentation, an image is subdivided into small blocks, 5 GLCM texture features for each block are then extracted to form a feature vector of 5 components for each block. Standardization and normalization are then performed on feature vectors. The next step consists of using K-Means clustering to classify these blocks into 3 different classes. Connected components operation is then performed to merge connected blocks into a single region.

For zone classification, K-Means clustering classifies blocks into one of the 3 groups, each group is further classified into one of the predefined categories by examining centroid values using heuristic rules, a centroid value is obtained when performing K-Means clustering. The flowchart of the system is shown in Figure 1.

Based on result of the zone classification, bounding boxes are then put around the components of the text and graphics regions and labeled accordingly.

3.1 Image subdivision

The document is subdivided into blocks; block size is predefined and changes correspond to the size of the image document. Each block becomes the smallest unit for further processing. For an image Img is defined as

$$Img = \{p_{ij},\ 0 \leq i < H, 0 \leq j < W\} \tag{1}$$

where p_{ij} is a pixel at location i, j; H and W are respectively the height and the width of the image. The subdivision of Img into blocks can be expressed as

$$Img = \{b_{IJ},\ 0 \leq I < \frac{H}{h}, 0 \leq J < \frac{W}{w}\} \tag{2}$$

where b_{IJ} is a block at I_{th} row and J_{th} column; h and w are respectively the height and the width of the blocks. A block b_{IJ} is defined as

$$b_{IJ} = \{p_{ij},\ h{\times}I \leq i < h{\times}(I{+}1),\ w{\times}J \leq j < w{\times}(J{+}1)\} \tag{3}$$

3.2 Feature extraction

Co-occurrence features [17, 18, 19] are a popular and effective texture descriptor using statistical approach. Given an image of n gray levels, characteristics of images are estimated from the second-order statistical features by considering the spatial relationship of pixels in the image. A GLCM element $P_{\theta,d}(i,j)$ is the joint probability of the gray level pairs i and j in a given direction θ separated by distance of d units. For each region of interest (ROI) in this work, five features are determined for texture discrimination: *Energy(ENR), Entropy(ENT), Sum Entropy(SEN), Difference Entropy(DEN), and Standard Deviation(STD)*, their definitions are given by the equations(4-8). Each subdivided block is an independent ROI. Multi-distance and multi-direction can be used to extract a large number of features. In our method we extract GLCM features using one distance $d = \{1\}$, and four direction $\theta = \{0^o, 90^o, 180^o, 270^o\}$, which result in 20 i.e ($1 \times 4 \times 5$) features extracted for each block.

$$ENR = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} P_d^2(i,j) \tag{4}$$

$$ENT = -\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} P_d(i,j) log_2 P_d(i,j) \tag{5}$$

$$SEN = -\sum_{k=0}^{2n-2} P_{x+y}(k) log_2 P_{x+y}(k) \tag{6}$$

$$DEN = -\sum_{k=0}^{n-1} P_{x-y}(k) log_2 P_{x-y}(k) \tag{7}$$

$$STD = \sqrt{\frac{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(P_d(i,j)-\mu)^2}{n \times n}} \tag{8}$$

where

$$\mu = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} P_d(i,j)}{n \times n} \tag{9}$$

$$P_{x+y}(k) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} P_d(i,j) \tag{10}$$

$$for \quad i+j=k, \quad k=0,1\ldots 2n-2$$

$$P_{x-y}(k) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} P_d(i,j) \tag{11}$$

$$for\ |i-j|=k, k=0,1\ldots n-1$$

GLCM features extraction *(GFE)* can be expressed as

$$GFE : Img \rightarrow Img \times \mathbb{R}^p \tag{12}$$

where

$$GFE(b_{IJ}) = F(GLCM(b_{IJ})) = (b_{IJ}, f_p) \tag{13}$$

where f_p is the feature vector and p is the number components (features extracted from b_{IJ}).

The $GLCM$ function introduced in (13) can be defined as

$$GLCM : Img \rightarrow \mathcal{M}_{n\times n}(\mathbb{N}) \tag{14}$$

where $M_{n\times n}(\mathbb{N})$ is the set of square matrices with the dimension $n \times n$. The function in (14) takes a block b_{IJ} of the image Img and returns its $GLCM$, given a direction and a distance.

The F function introduced in equation (13) can be defined as

$$F : \mathcal{M}_{n\times n}(\mathbb{R}) \rightarrow Img \times \mathbb{R}^p \tag{15}$$

where the function of equation (15) takes a GLCM of a block b_{IJ} and returns (b_{IJ}, f_p).

3.3 Feature Standardization and Normalization

The scales of individual features can differ drastically. This disparity can be due to the fact that each feature is computed using a formula that can produce various range of values. Another problem is that, features may have the same approximate scale, but the distribution of their values has different means and standard deviation. In this work we use statistical normalization(standardization), that independently transforms each feature in such a way that each transformed feature distribution has means equal to 0 and variance equal to 1. A further normalization is performed to enable all the features to have the same range of values that will result in an equal contribution of weight for the similar measure when classifying blocks [20, 21].

Let p be the number of features and m the size of the distribution, features matrix X is defined as .

$$X = \begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mp} \end{bmatrix} \quad (16)$$

where X_{ij} is the j^{th} feature of the i^{th} candidate for $i = 1, 2, \dots m$ and $j = 1, 2, \dots p$)
the corresponding standardized value is $newX_{ij}$ and is defined as (17)

$$newX_{ij} = \frac{(X_{ij} - \overline{X_j})}{\sigma_j} \quad (17)$$

where $\overline{X_j}$ is the mean defined in equation (18) and σ_j the standard deviation defined in equation (19)

$$\overline{X_j} = \frac{1}{m}\sum_{i=1}^{m} X_{ij} \quad (18)$$

$$\sigma_j = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(X_{ij} - \overline{X_j})^2} \quad (19)$$

3.4 Grouping Same Content Blocks Using K-Means clustering

Document segmentation is to segment the document into different contents. In this system, segmentation is done by the observation that same content share similar texture information. Texture of the areas of same content in the document are assumed to be similar, e.g. texture for text regions should be similar to each other, texture of the areas of space should be similar and texture of the graphics areas should be close as well. From this observation, the result, after clustering blocks into three groups using K-Means with the GLCM texture information as the clustering criteria, each group should be one of these identities: Graphics, Text or Space.

K-means clustering [22, 23, 24] is a basic and effective algorithm. It splits a set of objects into a selected number of groups; every object is compared to the centroid value of each group and assigned to the group that has most similar properties with the object. The centroid value is updated by taking the average of all the objects that belong to the group. This process is repeated until there is no change from the previous round compared to the current one. For K-Means clustering, K is specified by the user; in our case, K is 3, representing the three major different contents inside document images: text, graphics and space.

The clustering (CM_k) of blocks of an image Img defined at equation (2) using K-Means after GLCM features have been extracted from blocks can be expressed in equation (20):

$$CM_k : Img \times \mathbb{R}^p \rightarrow Img \times \mathbb{R}^p \times \{1, 2, \dots, k\} \quad (20)$$

The K-Means clustering function CM_k takes blocks (b_{IJ}, f_p) and returns it with a label (l) that specifies the number of the cluster to which it belongs. In other words $CM_k(b_{IJ}, f_p) = (b_{IJ}, f_p, l)$ means that the block b_{IJ} belongs to the cluster l. The K-means clustering is then a process of partitioning Img into g_l, where g_l is expressed as

$$g_l = \{b_{IJ} \in Img | d_l(b_{IJ}) < d_m(b_{IJ}) \; for \; l \neq m, 1 \leq m \leq k\} \quad (21)$$

where $d_l(b_{IJ})$ is a distance between block the b_{IJ} and the cluster l
We then have $Img = \cup_{i=1}^{k} g_i$ and $g_i \cap g_j = \emptyset$ for all i and j elements of $\{1, 2, \dots, k\}$ such that $i \neq j$. Given the parameter k (number of clusters) the K-Means algorithm can be performed as in the following steps:

1. Initial k prototypes $\{u_1^o, u_1^o, ...u_k^o\}$ by randomly selecting from feature vectors.
2. Assign each feature vector to its nearest prototype.
3. Update each prototypes as an average of feature vectors that belong to the cluster.
4. Repeat step 2. and 3. until there are no changes or maximum iteration reached.

3.5 Homogeneous Region Detection

Three contents of interest may contain component regions separated spatially in the document; for example, a document may contain two or more separated graphics or text paragraphs. The process of the document segmentation is then done by merging all the adjacent blocks into components. Location, width, height and class of each merged component is then extracted for later processing.

In our context, two blocks b_i and b_j are said to be adjacent if they both belong to the same cluster and there exist pixels $x_i \in b_i$ and $x_j \in b_j$ such that x_i and x_j are 8-neighbor[25]. The connectivity is further considered according to this adjacency.

Homogeneous region detection is the process that merges all the 8-connected blocks into a single connected component[25, 26]. One of the general algorithms used is to scan through the document image from top-left to bottom-right, when the algorithm encounters an unlabeled block, a new label C_i is then assigned to the block and recursively to all the adjacent blocks that belong to the same cluster. After one pass of scan, segmentation of image Img is obtained by having the connected components $C_i (i = 1, 2, \dots, ncc)$ where ncc is the number of components), and each component belongs to a specific cluster group l where $l = 1, 2 \; or \; 3$

3.6 Zone Classification

After the process of the clustering and connected components operation, the content of the clustered groups are classified into text, graphics and space zone by

using the rank of texture features between different contents. Instead of training a classifier to obtain a threshold to classify each content of the 3 clustered groups in high dimensional spaces; rank relationship of the texture features between three different contents is studied and used for classification. The visual criteria are usually similar for documents from the same category, i.e. similar text font, paper material and graphics resolutions; hence texture information from different contents may each converge to a different value. The rank of the contents of the converged values may also tend to be stable for the documents within the same category. From this intuition, a statistical approach is used to study the rank relation of the texture features between different contents.

Twenty sample image documents are selected, each document is subdivided into blocks and classified into three groups. We studied the centroid values produced by K-Means clustering; the results showed that the rank order of 5 selected GLCM features are highly related to the contents of the groups. By studying the 3 centroid values produced from sample images, GLCM texture features for the three different categories are well relate to each other, heuristic rules are defined and shown in Table 1. By applying these heuristic rules to the sample documents, error rates are 5% for Standard Deviation, 10% for Difference Entropy and 0% for other features. Zone classification is then performed using a scoring function based on the relations in table 1 as explained in the following paragraph.

After K-means clustering, three classes are formed and represented by the outputs f_{ij}, where $i = 1, 2, 3$, $j = ENR, ENT, SEN, DEN, STD$, and f_{i*} represents the components of the centroid of the class C_i, and each centroid has five components(Energy, Entropy, Sum Entropy, Difference Entropy, Standard Deviation).

Table 1 represents the most probable ranking of the feature values for the three different categories(text, space, graphics). For simplicity in the SCORING function below, Energy, Entropy, Sum Entropy, Difference Entropy, and Standard Deviation will be represented by 1, 2 , 3, 4 and 5 respectively. The expression $Scr(C, Cl)$ represents the score of the class Cl for the category C.

The scoring function is defined as follows

SCORING(C_i, $i = 1, 2, 3$)

for $c \in$ { Text,Graphics, Space} **do**
  for j=1 **to** 3 **do**
    $Scr(c, j) = 0$
  end for
end for
for i = 1 **to** 5 **do**
  $S = \{C_1, C_2, C_3\}$
  j = 1
  while $|S| > 0$ **do**
    $C_k = Argmin_{C_i}(f_i(S))$
    $S = S - \{C_k\}$
    $Scr(h(i, j), C_k) = Scr(h(i, j), C_k) + 1$
    j = j+1
  end while
end for

- $f_i(S)$ is the function that returns the set of the i^{th} features of the elements of the set S.

- $Argmin(f_i(S))$ is the function that gives the class C_i minimizing the function $f_i(S)$.

- $h(i, j)$ gives the category for the rank j of the i^{th} feature.

After construction of $Scr(C, Cl)$ a label: text, graphics or space is assigned to each class by choosing the category that achieved the highest score. For example a document with the centroids as shown in table 2 will produce the rankings as shown in table 3 and scores as shown in table 4, which results in classes: Classes 1, 2 and 3 being labeled text, space and graphics respectively.

Table 1: Features Relation

Features	Category Relation
Energy	Image < Text < Space
Entropy	Space < Text < Image
Sum Entropy	Space < Image < Text
Difference Entropy	Space < Image < Text
Std Deviation	Image < Text < Space

Table 2: A sample of centroid of classes

GLCM Feature	Class 1	Class 2	Class 3
Energy	1	7	6
Entropy	8	9	5
Sum Entropy	1	8	2
Difference Entropy	7	1	6
Std Deviation	6	8	4

Table 3: Classes ranking

GLCM Feature	Class 1	Class 2	Class 3
Energy	Image	Space	Text
Entropy	Text	Image	Space
Sum Entropy	Space	Text	Image
Difference Entropy	Text	Space	Image
Std Deviation	Text	Space	Image

Table 4: Table of scores

	Class 1	Class 2	Class 3
Text	3	1	1
Graphics	1	1	3
Space	1	3	1

4 EXPERIMENTAL RESULTS

Our method is implemented in C++ Builder 5.5. Preprocessing and post-processing are used to reduce run-

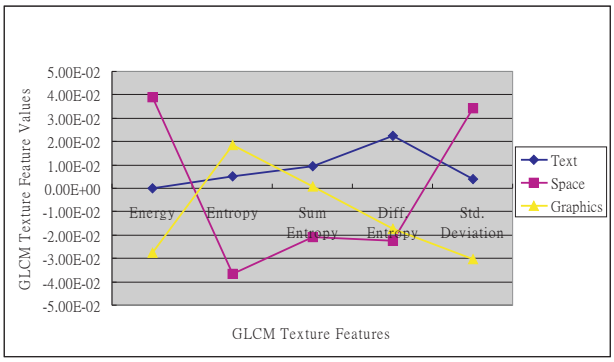

Figure 2: Three centroid GLCM texture values of the result obtained from figure 3

ning time and to obtain better edges of the components from blocks. Generally, segmented documents using the texture approach have the disadvantage of high computation time. Texture processing is very demanding computationally. To alleviate the high computation costs, the bunching technique is used to reduce the range of the color intensity in the image. Tests have shown the conversion of 24-bit color images into 64 gray level images does not affect performance, but at the same time the running time is reduced from minutes to seconds. Edge adjustment is used to tune the edges for each component to remove noise introduced by the subdivision of the image into blocks; small isolated regions that are identified as space or text components within graphics regions are set as graphics. Fifty document images are selected from scanned newspapers, documents and samples from the *articles* section of *MediaTeam Document Database*. Results show that our method can correctly extract and classify text and graphics regions from image documents. Test samples contain 158 separated graphics and 209 text paragraphs; within the 50 images, most of graphics regions are correctly segmented and classified as *graphics zone*, and most text portions are segmented and classified as *text zone*. The results also show that scratch graphics can easily be misclassified as part of the *text zone* as the texture of the scratch graphics is closer to the text: an example is shown in Figure 6. Some small isolated texts like page number may be misclassified due to post-processing that changes the isolated graphics and text blocks into space blocks. Two performance measurements are performed for both graphics and text zones: Extraction Rate(ER) , and Misclassification Rate(MR) and are defined in equations 22 and 23.

$$ER = \frac{Number\ of\ blocks\ correctly\ extracted}{Number\ of\ Expected\ Correct\ Blocks} \quad (22)$$

$$MR = \frac{Number\ of\ misclassified\ blocks}{Number\ of\ Expected\ Correct\ Blocks} \quad (23)$$

For graphics zones, the *Number of blocks correctly extracted* is the number of graphics blocks that have been correctly classified and the *Number of misclassified blocks* is the number of graphics blocks that have

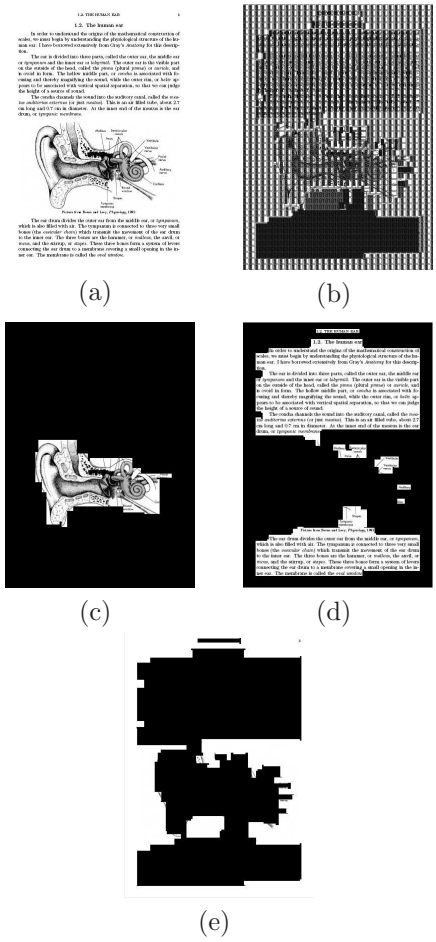

Figure 3: A scientific document (a)Original document (b) Image subdivided in blocks (c)Graphics Zone, (d)Text Zone, (e)Space Zone

been classified as *text* or *space* zones as well as the text and space blocks that have been classified as image zone. Table 5 shows the result of the performance. The extraction rate of the graphics zone is lower than text zone and the misclassification rate of the graphics zone is higher than the text zone because some portions of the graphics that contain text or a large portion of monochromatic color are easily misclassified as text and space, as shown in Figure 6.

Size of the test images vary from 380 × 528 to 2100 × 3200, running time for 1449 × 2021 image is about 3 seconds on a Pentium 4 computer. Some experimental results for separation and classification can be seen from Figure 3 to Figure 6.

Table 5: Experimental Results, NEC = Number of expected correct of blocks, NCE= Number of blocks correctly extracted, NMB = Number of misclassified blocks.

	NEC	NCE	NMB	ER	MR
Graphics	28737	26009	3603	90.51%	12.54%
Text	42059	40557	3752	96.43%	8.92%
Average	70796	66566	7355	94.03%	10.39%

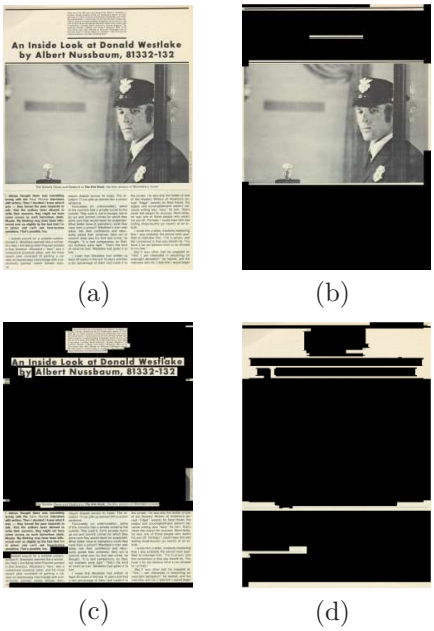

Figure 4: Experiment result for P00545 from Article Section, MediaTeam Document Database, resized to 734 × 1024(a)Original document (b) Graphics Zones (C) Text Zones (d) Space Zones

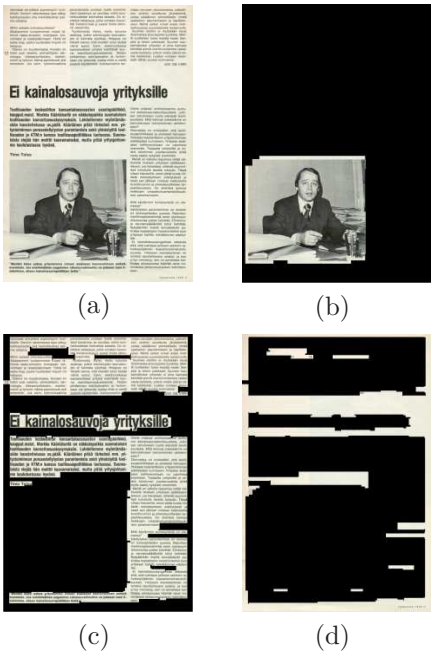

Figure 5: Experiment result for P00836 from Article Section, MediaTeam Document Database, resized to 702 × 1024 (a)Original document (b) Graphics Zones (C) Text Zones (d) Space Zones

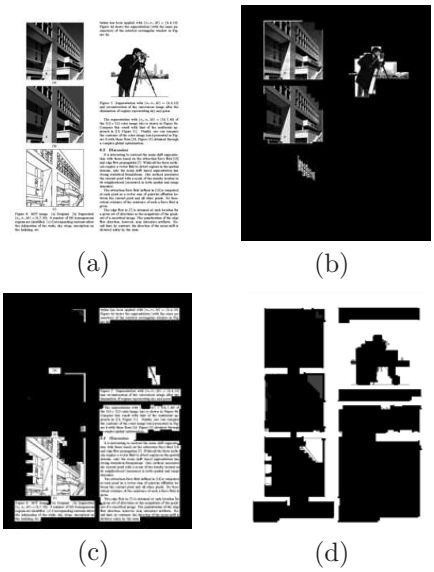

Figure 6: One of the experiment results, scratch graphics is misclassified into Text Zones and portions of monochromatic color graphics are misclassified as space. (a)Original document (b) Graphics Zones (C) Text Zones (d) Space Zones

5 FUTURE WORKS AND CONCLUSION

We have presented a method that starts of by subdividing a document image into blocks and then clusters them in order to separate and classify the contents of this document. To perform this task two parameters, the block width and the block height are needed. If block size is too small, it may not have sufficient information for classification, and if block size is too large, different types of components may be mixed within the same block. After some experiments we have found it suitable to set 3 different default block sizes, 8×8, 16×32 and 40×60 for small, medium and large documents respectively. Knowledge on the size of the contents and the appropriate block size is one of the important factors for successful separation. Future work will involve finding an adaptive method based on the context to set appropriate block sizes to increase the accuracy of the results. One of the remaining problems is the frontier of each component, since each block has been treated as a unit which may result in a block containing two or more different contents and this usually occurs around the frontier of the components. Post-processing for finding better frontiers is needed for the increase in accuracy of the segmentation.

The approach used to classify the content type of the clustered groups is by using the rank of the texture features between different contents. It will be interesting to investigate if the similar heuristic rules can be extracted from the different categories of the documents.

We have presented a hybrid method using texture approaches to extract text, graphics and space zones from a gray scale document image. Tests result have shown that tasks are carried out in a reasonable time and fairly accurate for separation and classification in

document images.

6 ACKNOWLEDGMENTS

The authors would like to thank National Research Foundation of South Africa for their financial support for the project and the anonymous reviewers for their constructive comments.

REFERENCES

[1] D. Wang and S. N. Srihari. "Classification of newspaper image blocks using texture analysis". *Computer Vision, Graphics, and Image Processing*, vol. 47, pp. 327–352, January 1989.

[2] Z. Lu. "Detection of Text Regions From Digital Engineering Drawings". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 431–439, 1998.

[3] C. Tsai and Y. L. Chi. "An Extractor for Understanding Text Strings from Digital Engineering Drawings". In *Proceedings of the SCI 2001/ISAS 2001, World Multi-Conference on Systemics, Cybernetics and Informatics*, vol. XIV. Orlando, Florida, 2001.

[4] M. M. Fleck, D. Forsyth and C. Bregler. "Finding Naked People". In *ECCV (2)*, pp. 593–602. 1996.

[5] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy and P. Dosch. "Text/Graphics Separation Revisited". In *DAS '02: Proceedings of the 5th International Workshop on Document Analysis Systems V*, pp. 200–211. Springer-Verlag, London, UK, 2002.

[6] Q. Yuan and C. Tan. "Text extraction from gray scale document images using edge information". In *Proceedings of the ICDAR 2001*, pp. 302–306. 2001.

[7] Y. Zheng, H. Li and D. Doermann. "Machine printed text and handwriting identification in noisy document images". Tech. rep., LAMP Lab, University of Maryland, College Park, 2002.

[8] A. Jain and S. Bhattacharjee. "Text segmentation using Gabor filters for automatic document processing". *Machine Vision and Applications*, vol. 5, no. 3, pp. 169–184, 1992.

[9] T. Randen and J. Husoy. "Segmentation of text/image documents using texture approaches". In *Proceedings of the NOBIM-konferansen-94*, pp. 60–67. Asker, Norway, June 1994.

[10] Q. Zhang, P. G. J. Wang and P. Shi. "Study of urban spatial patterns from SPOT panchromatic imagery using textural analysis". *International Journal of Remote Sensing*, vol. 24, no. 21, pp. 4137–4160, November 2003.

[11] S. Grigorescu, N. Petkov and P. Kruizinga. "Comparison of texture features based on gabor filters". *IEEE Transactions on Image Processing*, vol. 11, no. 10, October 2002.

[12] C. T. Wu, Y. C. Chen and K. S. Hsieh. "Texture features for Classification of Ultrasonic Liver Images". *IEEE Transactions on Medical Imaging*, pp. 141–152, April 1992.

[13] A. Jain and B. Yu. "Document Representation and Its Application to Page Decomposition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 294–308, March 1998.

[14] G. Nagy, S. Seth and S. Stoddard. "Document Analysis with an Expert System". In *Pattern Recognition in Practice II*, pp. 149–155. Elsevier Science, 1984.

[15] T. Pavlidis and J. Zhou. "Page segmentation and classification". *Graphical Models and Image Processing*, vol. 54, pp. 484–496, 1992.

[16] J. Duong, M. Ct, H. Emptoz and C. Suen. "Extraction of Text Areas in Printed Document Images". In *ACM Symposium on Document Engineering: DocEng'01, November 9-10*, pp. 157–165. Atlanta, USA, November 2001.

[17] R. M. Haralick, K. Shanmugam and I. Dinstein. "Textural features for image classification". *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, November 1973.

[18] R. M. Haralick. "Statistical and structural approaches to texture". *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[19] M. Tuceryan and A. Jain. "Texture Analysis". In L. P. C.H. Chen and P. Wang (editors), *The Handbook of Pattern Recognition and Computer Vision(2nd Edition)*, pp. 207–248. World Scientific Publishing, 1998.

[20] S. Teeuwsen. "Feature selection for small-signal stability assessment". In *Proceedings of the Dresdner Kreis 2002*. Werningerode, Germany, March 2002.

[21] D. Frandkin and I. Muchnik. "A Study of K-means Clustering for Improving Classification Accuracy of Multi-Class SVM". Tech. Rep. TR: 2004-02, DIMACS, April 2004.

[22] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. 1967.

[23] A. Belaïd and Y. Belaïd. *Pattern Recogition : Methods and Applications*. Dunod Informatique, 1992.

[24] A. K. Jain, M. N. Murty and P. J. Flynn. "Data clustering: a review". *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999.

[25] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[26] M. Sonka, V. Hlavac and R. Doyle. *Image Processing, Analysis and Machine Vision*. PWS, September 1998.