Scanned Document Image Segmentation Using Back-Propagation Artificial Neural Network Based Technique

Nidhal Kamel Taha El-Omari, Ahmed H. Omari, Omran Fadeel Al-Badarneh, Hussein Abdel-jaber

***Abstract*—**Document images are composed of graphics, pictures, text, and background with varying number of colors. Based on the detected number of colors contained in a document image, a new approach for document image segmentation and classification using an Artificial Neural Network (ANN) technique is proposed. The ANN is designed to learn how to recognize some interesting color patterns from the labeled document images. Then, the unlabeled document images are classified according to these color patterns. This approach aims at segmenting the original image content into consistent and homogeneous four regions: picture, graphics, text, and background. In order to achieve better compression ratios, every component is compressed separately using the most appropriate compression technique.

***Keywords*—**Artificial Neural Network (ANN), Block-based encoding, Classification, Data Mining, Image Segmentation, Layered encoding.

I. Introduction

Scanned text documents are increasingly used in a wide range of applications, including but not limited to archiving and document management systems. Many of these documents, called compound or mixed documents, consist of a mixture of texts, pictures, graphics (drawing), and background. The storage requirements of uncompressed high quality color scanned documents are quite vast. And unfortunately, managing such uncompressed documents proves to be inefficient and creates the potential effect of substantially limiting their benefits and may perhaps never meet the ever-growing information demands of the users. Traditional compression mechanisms, which are generally developed with a particular image type and purpose, are facing many challenges with mixed documents [1,2]. As a standard A4 color page document, scanned with a resolution of 600 dpi, requires around 91 million bytes of storage space, assuming 24 bit-depth and a standard 8 X 12 inches sheet [1,2]. To reduce the needed storage or to speed up their transfer through the computers networks, the document images need to be compressed. Unfortunately, these documents do not compress well using classical image compression algorithms such as JPEG-2000. This is due to the presence of sharp edges on top of the smooth surfaces of the text and graphics, typically found in natural images. Moreover, compression algorithms for text facsimiles, such as JBIG2, are not suited for color or gray level images [2,3,4,5].

Segmentation is used to break down the scanned image into smaller segments, called regions, then saving each segment using the appropriate compression for that segment. It has better compression and quality than just one standard compressed file. However, selecting the best segmentation for a given image can be seen as a classification problem, analogous to the compression problem of selecting the best encoding algorithm for that segment. The simplicity and modularity is at the cost of some encoding overhead. [2,3]

ANN is a data-modeling tool that has the capability to capture and model complex input/output relationships and make knowledge generalization in a form of patterns [2,7,8]. These patterns may be quantitative or structural description of the components that compromise an image [2,7,8]. Each pattern is formed by one or more descriptors which are the features that characterized each block of the image [2,3]. It can be used to remap the original mixed (compound) digital documents into four different types of components: picture, graphics, text, and background image. In order to achieve better compression ratios, every component is compressed separately using the most off-the-shelf appropriate compression technique.

This work is a continuation of the previous works in the area of document image segmentation based ANN where the average of accuracy is raised from 87% to 89% [2,3]. The rest of the paper is organized as follows. Section 2 looks at the related work and background of the algorithms used. In Section 3, the developed approach and the algorithm of this research is presented. Section 4 shows the training results and describes the analysis of the results. Finally, Sections 5 and 6 provide conclusions and offers avenues for future work, respectively.

II. Background

Segmentation of a mixed document aims to separate background, text, pictures, and graphical components of a document image [1,2,3]. There are different techniques

Nidhal Kamel Taha El-Omari, The World Islamic Sciences & Education University (WISE)), Amman, Jordan, omari@wise.edu.jo
Ahmed H. Omari, Applied Science University (ASU), Amman, Jordan, a.omari@asu.edu.jo
Omran Fadeel Al-Badarneh, Arab Academy for Banking and Financial Sciences (AABFS), Amman, Jordan, omran99@hotmail.com
Hussein Abdel-jaber, The World Islamic Sciences & Education University (WISE)), Amman, Jordan, habdelja@wise.edu.jo

proposed in the literature to solve the problem of segmenting and compressing compound documents. These techniques can be classified into three different categories that Fig. 1 illustrates.

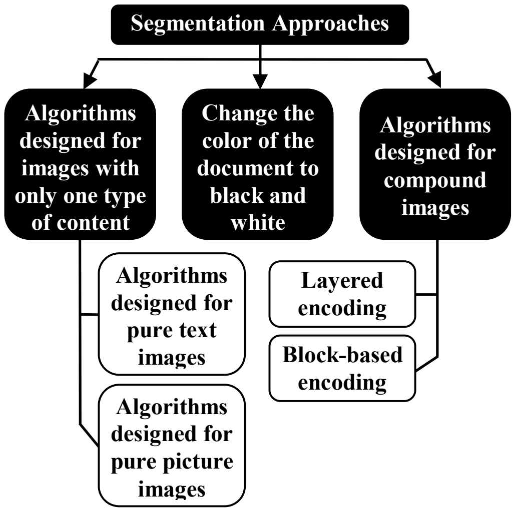

Fig. 1: Three Approaches for Compressing Compound Documents

The first category of algorithms scans and transforms the document into a black and white image. Then decode it using lossless decoders, such as “Fax Group 3” and “Fax Group 4”. Although these algorithms achieve high compression rate and preserve text legibility, they lead to the losing of contrast and color information. They may suit some business and technical documents, but not other document types such as magazines or historical documents. [9,11,14,5]

The other category uses algorithms that are only designed for one type of content. Some of them are designed to compress pure text images which contain only text on pure color background of the whole image. These algorithms show bad performance on pure picture images. An example of them is Lempel-Ziv algorithm. Other algorithms are designed for pure picture images which do not have any text; they show bad performance on pure text images. An example of them is JPEG. [4,5,15,16]

Since no single algorithm gives good results across all image types or applications, a third category of algorithms is needed to compress compound images with different content types: picture, graphics, and text. Although these algorithms are proposed to solve the drawbacks of the previous two categories, they do not reach the ideal situation. The algorithms of this category are categorized into two groups: the Layered encoding and the Bock-based encoding. [2,1]

The Layered encoding methods separate the images into different layers where each layer is encoded independently from the other layers. Most Layered encoding algorithms use the standard three layers Mixed Raster Content (MRC). As illustrated in Fig. 2, the three layers are: an image BackGround layer (BG), an image ForeGround layer (FG), and a binary mask layer. The mask; classified the image components as either foreground or background components. While the foreground components are coded by the foreground coder, the background components are coded by the background coder. Examples of this group of methods are LuraDocument and DjVu techniques. [9,18,11,13,14]

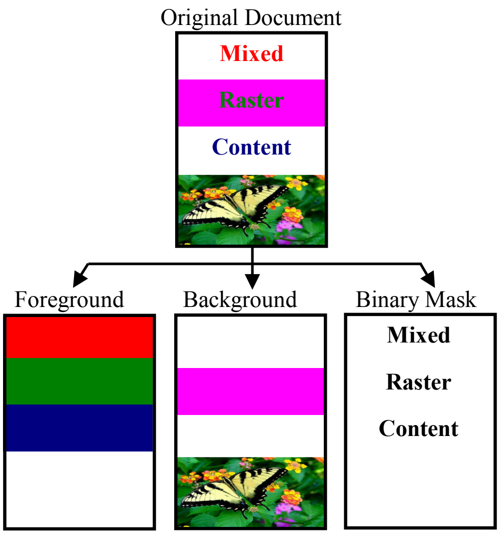

Fig. 2: Example on MRC

The Layered encoding methods still have some drawbacks; the complexity of layer generation is high, that makes it unsuitable for many embedded and real time applications [14,16,15]. Binary representation of the mask layer, that encodes the text and the graphics contents, tends to distort some fine document details, such as text edges and thin lines [13,11,9]. Although foreground and background layers may not be used, they should also be coded; this adds some inefficiency [18,11]. Unfortunately, some foreground components may be classified as belonging to the background layer [9,13,18,11]. By contrast, some background components may be classified as belonging to the foreground layer [9,13,18,11].

Moreover, layer based approaches work well on simple compound images. But when the content is very complex, they show poor performance. For example, it is difficult to separate text from backgrounds when the text overlaps with background or the text has surrounding shadow. [15,16, 11,13,9,18]

The Block-based encoding methods classify the compound image into blocks of different types. Then each type is compressed individually with the most appropriate encoder. Although these methods give better results than the previous group, there are still some drawbacks. In case of strong edges in the textual area, they lead to hybrid blocks. These hybrid blocks contain mixed text and pictures and cannot be handled effectively. Even if the block contains boundary between two regions, all its pixels are classified in the same manner and given the same label. Although the complexity is lower than Layered encoding techniques, both the classification and compression algorithms of Block-based encoding still have high calculation complexity, which makes them not suitable for real time applications [16,15,17].

Accordingly, there is still much room for improving existing algorithms or coming up with new effective algorithms and techniques which is described in this research paper. However, there is a need for an effective way to classify image components and to compress its content.

III. THE NEW TECHNIQUE DESCRIPTION

The idea behind the proposed segmentation technique is to analyze the image in order to exploit the inconsistency and hidden patterns present in the data that are relevant to the different components of the image. Therefore, the network is designed to learn how to recognize some interesting patterns from the labeled data, and then the unlabeled data is classified according to these patterns.

3.1. Supervised Learning

The purpose of this trained ANN is to generalize some knowledge about color histogram of a block; the network matches an input color histogram against other patterns that have been learned before. An output label is produced determining the type of the block: picture, graphics, text, or background. After sufficient training, the network is able to classify a color histogram for the new block based on the degree of matching with learned histograms.

Each input of this technique is interlinked with a target entity for which the class label is known and previously defined to be one of the four mentioned labels. This target works as a teacher or supervisor who classifies the training examples into classes. Since the training examples are previously classified into classes and the ANN is ending up with a label that point to one of these predefined classes, the learning process is called “Supervised Learning”. [12,20,3]

The difference between the real outputs and the desired outputs, called targets, is used by the ANN algorithm to adapt the interconnection weights of the network. By this adjustment between elements of the connections, the ANN can be trained to perform a particular function. These weights are adjusted automatically by training the network according to a specified learning rule until the desired result is achieved. Eventually, the neural network will be learned to associate certain inputs with certain outputs. [12,20,3]

Therefore, the network is continuously adjusted, based on a comparison of the actual output and the desired output, until the output matches the target. Typically, many such large sets of sequence input/target data pairs are used in training the neural networks for this computation. Each run through the examples is called a cycle, epoch, or a pass [2]. The number of training cycles should be sufficient to determine the overall performance of neural network. Each cycle traverse through all of the specified training input and target vectors, it carries out a loop of calculations.

Fig. 3 illustrates the steps in the ANN training process. The examples of the training data, however, are divided up into three independent disjointed sets: [20,2,3]

1. **Training set:** A set of examples used to adjust the weights. The ANN has to be trained on a representative set of input/target pairs and get good results without training it on all possible input/output pairs. Many training examples are required until the network is capable of classifying a new block. Unless these examples are selected carefully, the network will not operate perfectly.
2. **Validation set:** After training, the network is simulated to see whether it has learned to respond correctly or not. This set of examples used to tune up the architecture (not the weights); it is any data set that is used to choose the best of two or more ANN architectures. For example, to choose the number of hidden units.
3. **Test set:** Well trained networks tend to give reasonable answers when presented with new inputs that have never seen before. The testing data is usually a new data set which reflects variety of documents. It is a set of examples used only to assess the performance on practice and to generate generalization in deriving knowledge. It is never used to choose among two or more ANNs.

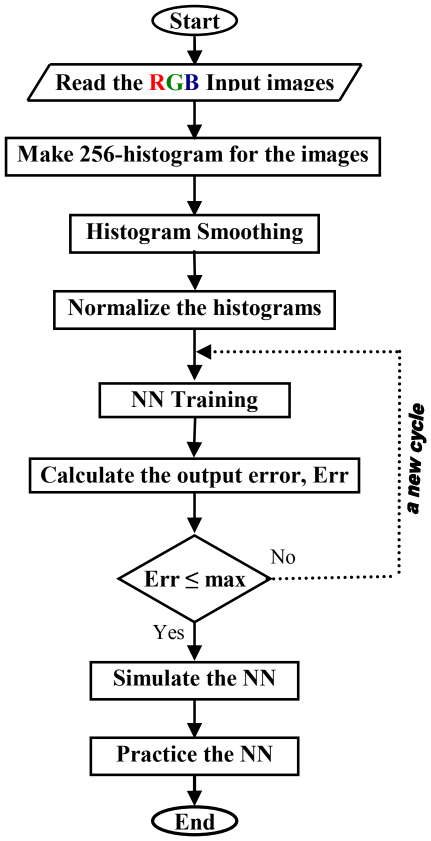

Fig. 3: ANN **Training Flowchart**

3.2. The Algorithm

The ANN learning process intends to map the input block into one of the “K” classes of the classification space. As Fig. 4 illustrates, this mapping includes three main stages [20,12]:

1. **Pre-processing stage**: As the real world, data may be ambiguous and error prone. Thus, this stage is the most important step before feeding to the network. It determines the success of any neural network model. The system is

driven by a set of training samples which are used for learning and building the relationship between scanned images. However, a sufficient amount of training samples is available for every images category. In order to build the "learning approach", they are previously classified into four categories: picture, graphics, text, and background. This stage includes data collection and partitioning, pre-processing, post-processing (i.e. Denormalization), and all the primary operations that are used for reducing noise or variations inside the scanned image.

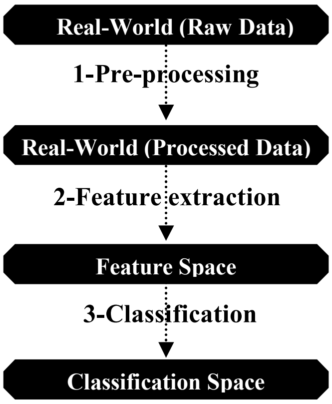

Fig. 4: ANN Phases

2 . **Feature extraction stage**: Feature extraction is essential for data representation and extracting meaningful features for later processing. This stage is the key property of mapping from the real world to the feature space.

Although, each image in the training set is different from the others, all images of the same category type may share certain common characteristics. The system captures these characteristics and uses them in order to classify new images. The training system has the capability to generate information from this set of images and store this information in a training set.

3 . **Classification stage:** The next stage is the classifying of objects according to their features. It brings related things together by assigning each block to one of the predefined classes. It is the process of mapping the feature space to the classification space. However, classified blocks point out to the class of each block.

Four supervised ANNs are built in this proposed technique, one network for each type:

1 . **Picture ANN**: This network decides whether the block is assigned to the picture segment or not.
2 . **Graphics ANN**: This network decides whether the block is assigned to the graphics segment or not.
3 . **Text ANN**: This network decides whether the block is assigned to the text segment or not.
4 . **Background ANN**: This network decides whether the block is assigned to the background segment or not.

Each ANN has a single element output vector (1 x 1). It gives a continuous output activation value, "v", between zero and one that corresponds to the degree that belongs to. This value is used by the ANN to classify a given input block. For the continuous output between "v=0" and "v=1", the value "v=0.5" is used as the threshold to make the decision [20,2]. To be more adequate, the threshold value, "v=0.4", is used instead of "v=0.5". However, this value is determined here empirically.

The reason to use four different ANNs, one for each single element output vector (1 x 1), instead of using one ANN of 4-element output vector (4 x 1) is that: If only one ANN with 4 outputs is used, then any output element of the 4-element output may be affected and biased by the decision of the other elements. This requires making the decision by comparing the four results and taking the maximum value which may not coincide with the wanted priority.

Whereas, using four separated ANNs provides an unbiased estimate decision toward the other decisions. It gives the decision according to the priority which is: picture, graphics, text, background; once the desired threshold value, "**v**", is met, regardless the other results.

For simplicity, picture, graphics, text, and background ANN are abbreviated here as **PNN**, **GNN**, **TNN**, and **BNN** respectively. Their corresponding ANN outputs are abbreviated as **PNNO**, **GNNO**, **TNNO**, and **BNNO**, respectively.

In order to decide the type of each block, the following procedure is applied:

If **PNNO** is greater than "v" **Then**
"*Label the block as Picture*"
Else If **GNNO** is greater than "v" **Then**
"*Label the block as Graphics*"
Else If **TNNO** is greater than "v" **Then**
"*Label the block as Text*"
Else If **BNNO** is greater than "v" **Then**
"*Label the block as Background*"
Else
"Label the block as Picture"
End If

However, this procedure means that any block of type "picture" are determined only by the **PNN** without bypassing the block to the other ANNs. Similarly, any block of type "graphics" are determined from the previous two ANNs **(PNN** and **GNNO**) without bypassing the block to the other two ANNs. Moreover, while any block of type "text" are determined from the previous three ANNs **(PNN** and **GNNO** and **TNNO**) without bypassing the block to the last ANN, any block of type "background" are determined by using the four ANNs. Finally, if all the ANNs could not determine the type of the block, the block is labeled as "picture". Table 1 summarizes the above procedure.

Table 1: The Four Used ANNs

PNNO	GNNO	TNNO	BNNO	Block Type
>v	not used			Picture
≤ v	>v	not used		Graphics
≤ v	≤ v	>v	not used	Text
≤ v	≤ v	≤ v	>v	Background
≤ v	≤ v	≤ v	≤ v	Picture

Consequently, the input image is segmented into four segments according to the labels presented in the previous procedure. However, the proposed technique is demonstrated through the flowchart shown in Fig. 5.

Fig. 5: ANN Proposed Technique Flowchart

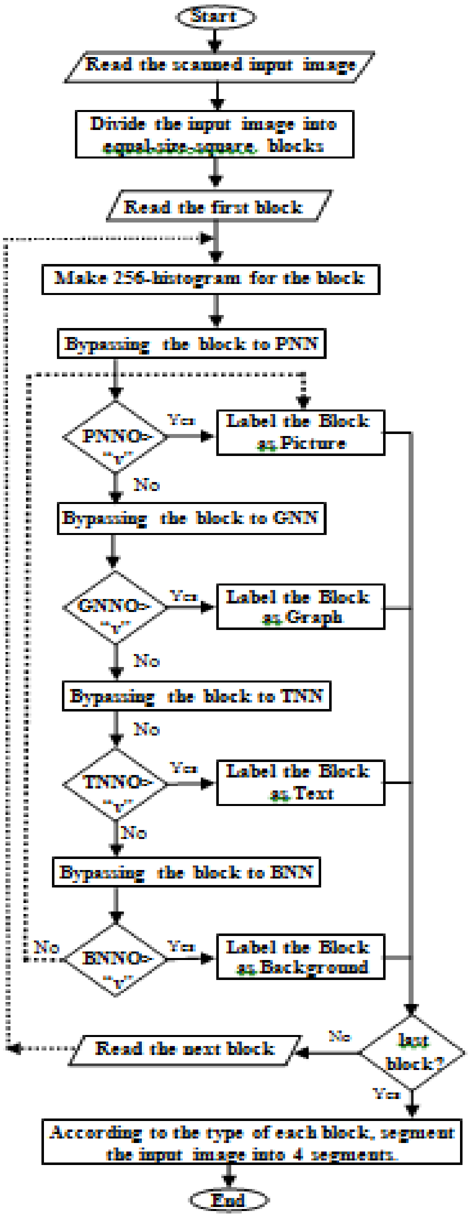

Fig. 6 shows an example using this technique. The original image of this example contains the four types. It is segmented into four segments by the proposed technique. The four segments: picture, graphics, text, and background are determined by the four ANNs: **PNNO**, **GNNO**, **TNNO**, and **BNNO**, respectively.

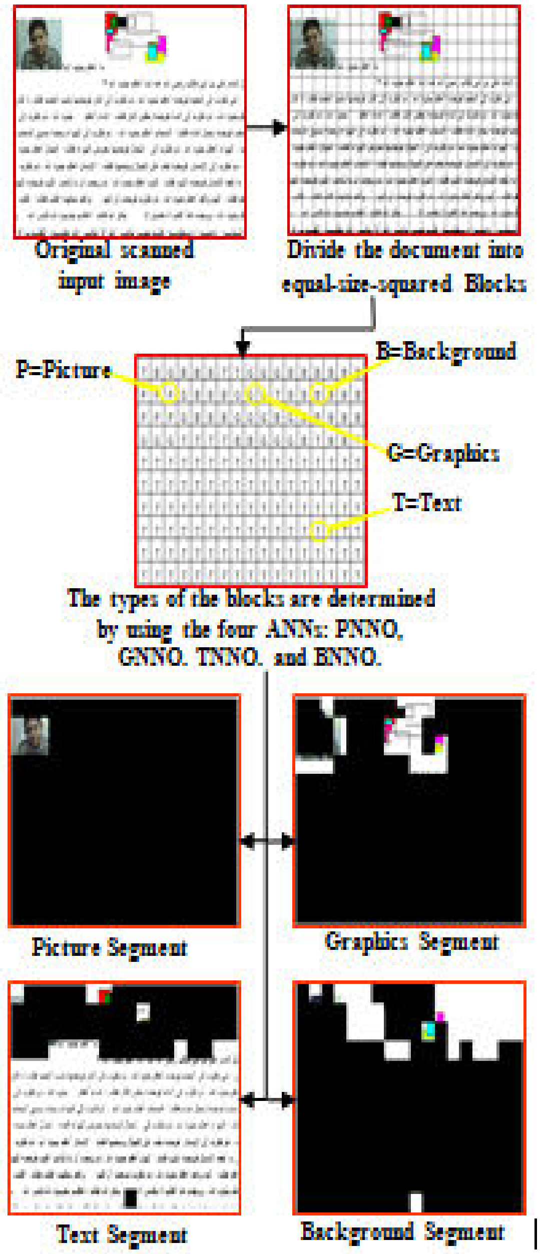

Fig. 6: An example on this proposed technique

Since ANN is sensitive to the number of hidden layers and the number of neurons per layer, the main concern is to determine these values. On the way to detect the best suitable number of hidden layers and units in each layer, many training experiments are conducted until general lack of improvement (i.e. no benefit can be derived from performing more attempts). Table 2 gives a brief description of the architecture of this proposed technique and its four networks. It is noted from the Table 2, that the **GNN** and **BNN** reach their goals with more accuracy (performance goal) than the other **PNN** and **TNN**. Moreover, the number of their needed epochs is also lower than the other two ANNs.

Table 2: ANN Parameters.

Parameter	Artificial Neural Network			
	BNN	GNN	TNN	PNN
Number of hidden layers	2	3	2	2
Size of the i^{th} layer	{16,12,1}	{3,6,9,1}	{12,16,1}	{6,10,1}
Transfer function of the i^{th} layer	**A nonlinear logarithmic-sigmoid, "logsig"**			
Training function	Gradient descent with adaptive learning rate back-propagation "trai gda"		Levenberg Marquardt with adaptive learning rate back-propagation "trainlm"	
Performance function	**Mean Square Error "MSE"**			

IV. EXPERIMENTAL RESULTS & EVALUATION

Since ANNs learn by deriving knowledge from examples and a reliable system should be experimented on a large number of samples, a special database that contains different images was created as illustrated in Table 3. This database contains 2987 24-bit-RGB-bitmap images of different resolutions distributed among five classes. Except the last class that is only used for testing, the other four classes are used in training, validation, and testing the ANN proposed technique. While one half of the stored images are used for training, the other half are used for validation, and testing .

Table 3: Classes of the Special Database.

Image Class	**Training**	**Validation**	**Testing**	**Total No. of images**
Pure Background	170	80	79	**329**
Pure Text	350	175	173	**698**
Pure graphics	290	145	139	**574**
Pure Picture	390	190	187	**767**
Mixed Image	×	×	619	**619**
Total	**1200**	**590**	**1197**	**2987**

These different images came from three main sources:

1. **Internet resources**: Those are available on image processing at literature review of the third chapter.
2. **Scanned images**: Images those are created by different scanners. The scanner should have the ability to digitize a mixed document and convert the image into a digital image that can be used as an input for one of the proposed techniques. The quality of the primary scanning should be as high as possible.
3. **Computer-generated images**: Images that are obtained by simulation using one or more of the traditional packages like "Microsoft Office Word" or "Microsoft Paint".

The proposed technique has been implemented using MATLAB® version 7.0 release 14 environments. The selected images of this database are used to test the performance of the proposed algorithm. Since it is able to distinguish between the four types of blocks, it is ready for working on and can be used as a practical block/image type detector. It successfully produced the proper segmentation with an average accuracy of 89%. The best results are for the cases where the documents consist mainly of texts and graphics images with a success rate of about 95%. The worst was found for the mixed images with rate of about 83%. However, the performance of this proposed algorithm, as most other algorithms, depends on the statistics of the file to be compressed.

Finally, this technique has the capability to learn from examples and to generalize knowledge that is capable to classify new unclassified block. This knowledge is simply the patterns that are inferred from the information.

V. CONCLUSION

ANN of this proposed technique, specifically back-propagation learning algorithm, is used successfully in developing a model for segmenting a compound scanned image into coherent regions and classifies them as background, text, graphics or pictures. Although ANNs normally take more time to get training (not testing or simulation) results, this proposed technique shows a clear advantages over the others. One better thing is that the training process can be done off-line.

The proposed approach combines different techniques in order to achieve better compression of the scanned documents. It is a Block-based approach where the input image is divided into equal-size-square blocks. It is a region-based approach where the input image is divided into homogenous regions based on the number of colors. Furthermore, it also uses different compression concepts to improve the compression ratio.

In order to generalize, the ANN should be able to generate the correct output data on testing samples that is not included in the learning set. Accordingly, this technique is tested on a set of 2987 documents of different resolutions, 619 of them are mixed documents. The experiments show that results are excellent and the average accuracy is around 89%.

Based on the result of this technique, it can be concluded that the ANN can acquire knowledge through learning and capable of exhibiting some degree of intelligence in determining the type of a new block using a historical data that belongs to pre-classified images.

VI. FUTURE WORK

On this proposed technique, searching for more input parameters for the ANN proposed model may contribute a lot in the speed and accuracy of segmentation. This may result in increasing the number of input variables. Due to that, the number of pixels in a digital image and its color gamut are the main distinguishing features of a digital image. Followings some recommended new parameters:

- Number of colors per image: Table 4 illustrates the effect of the number of colors on **PNN** performance.
- The mean and median of pixels.
- The variance or the standard deviation of pixels.
- The minimum and maximum values of pixels.

However, variance of the color histogram was also considered but this turned out not to be a good measure [2].

Table 4: The effect of the Number of Colors on the **PNN** Performance.

Epoch Number	MSE
000	2.61092 e-001
050	5.88220 e-004
100	7.61908 e-024
150	1.18528 e-024
200	6.49016 e-025
250	4.47023 e-025
300	1.320994e-025

As the speed is an important factor, this technique may work by conjoining some predefined “rule-based” approaches [20,2]. Table 5 defines some obvious rules that may contribute a lot in the speed and accuracy of the segmentation. These rules are calculated for each image using the number of color frequencies that are greater than zeroes. Furthermore, to determine whether the ANN has acquired knowledge through learning or not Table 6 defines some other rules that may be used after the ANN is used.,

Table 5: ANN Predefined Conjoining Rules.

Rule	The block is Identified as:
(Number of colors =1)	Background
(Number of colors =2) .and. (the distance between these two colors < 10)	Background
(Number of colors =2) .and. (the distance between these two colors > 150)	Text
0.067 of the number of pixels are of distinct colors (0.067 is determined empirically).	Picture

Table 6: ANN Predefined Testing Rules.

Rule	The block is Identified as:
(Number of colors =2)	Not a Picture type and not a graphics type
(Number of colors > 2) .and. (the difference of intensity between the first two highest frequency colors > 100)	Not a Background type

What's more, after the separation of text and none text (background, graphics, and pictures) from mixed content is done, the addition of text layout analysis and optical character recognition (OCR) will make it possible to index and edit text extracted from the encoded documents.

Additionally the analysis of scanned documents is important in the construction of digital paperless offices, digital libraries or other digital versions of originally printed documents. It can be used in search engines; it gives the capability to search engines by means of images, that have text and none text, and not only with keywords. The picture recognition of isolating pictures from the image is used in the facility of “image-archive”.

Finally, the proposed techniques can be used as a pre-processing step for automatic indexing and retrieval of relevant documents from large collections of document images. Its successes depend firstly on the availability of an efficient OCR system and then on the indexing program that stores the keys automatically alternative to the manual data entry process [21,2].

VII. ACKNOWLEDGEMENT

This work is supported by WISE University, www.wise.edu.jo

REFERENCES

[1] Acharyya, M. and Kundu, M.K., “*Document Image Segmentation Using Wavelet Scale-Space Features*”, IEEE Transactions Circuits Syst. Video Technol., Volume 12, Issue 12, pp. 1117–1127, December 2002.

[2] Nidhal Kamel Taha El Omari, “*A Hybrid Approach for Segmentation and Compression of Compound Images*”, PhD Dissertation, The Arab Academy for Banking and Financial Sciences, 2008.

[3] Nidhal Kamel Taha El-Omari and Arafat A. Awajan, “*Document Image Segmentation and Compression Using Artificial Neural Network Based Technique*”, International Conference on Information and Communication Systems (ICICS09), pp. 320-324, Amman, Jordan, December 20-22, 2009.

[4] Kai Uwe Barthel, Simon McPartlin, and Michael Thierschmann, “*New Technology for Raster Document Image Compression*”, SPIE The International Society for Optical Engineering, Volume 3967, pp. 286-290, San Jose, CA, January 2000.

[5] Patrice Y. Simard, Henrique S. Malvar, James Rinker, and Erin Renshaw, “*A Foreground/Background Separation Algorithm for Image Compression*”, IEEE Data Compression Conference (DCC), pp. 498–507, Snowbird, UT, USA, March 23-25, 2004.

[6] Alexey Minin, “*The Neural-Network Analysis & its Applications Data Filters*”, Saint-Petersburg State University, Joint Advanced Student School (JASS), St. Petersburg, Russia, 2006.

[7] Robyn Ball and Philippe Tissot, “*Demonstration of Artificial Neural Network in Matlab*”, Division of Nearhsore Research, Texas A&M University, 2006.

[8] Julien M.J. Racca and Robert Racca, “*PaleoNet: Software for Building Neural Network-based Paleoecological Transfer Functions*”, User guide Version 1.01, http://www.cen.ulaval.ca/paleo/PaleoNet/PaleoNet_manual.pdf, September 2005.

[9] Ricardo L. de Queiroz, Robert Buckley, and Ming Xu, “*Mixed Raster Content (MRC) Model for Compound Image Compression*”, SPIE The International Society for Optical Engineering, Volume 3653, pp. 1106-1117, February 1999.

[10] Kai Uwe Barthel, Simon McPartlin, and Michael Thierschmann, “*New Technology for Raster Document Image Compression*”, SPIE The International Society for Optical Engineering, Volume 3967, pp. 286-290, January 2000.

[11] Ricardo L. de Queiroz, “*Pre-Processing for MRC Layers of Scanned Images*”, Proceedings of the International Conference on Image

Processing (ICIP), Atlanta, Georgia, USA, pp. 3093–3096, October 8-11, 2006.
[12] Lihong Zheng and Xiangjian He, "*Edge Detection Based on Modified BP Algorithm of ANN*", Conferences in Research and Practice in Information Technology (RPIT), Volume 36, pp. 119–122, 2004.
[13] Guotong Feng and Charles A. Bouman, "*High Quality MRC Document Coding*", IEEE Transactions Image Processing, Volume 15, Issue 10, pp. 3152-3169, October 2006.
[14] Leon Bottou, Patrick Haffner, Paul G. Howard, Patrice Y. Simard, Yoshua Bengio, and Yann Lecun, "*High Quality Document Image Compression with DjVu*", Journal of Electronic Imaging, Volume 07, Issue 3, pp. 410-425, July 1998.
[15] Wenpeng Ding, Yan Lu, Feng Wu, and Shipeng Li, "*Rate-Distortion Optimized Color Quantization for Compound Image Compression*", Visual Communications and Image Processing Conference, SPIE Proceedings, Volume 6508, pp. 65082Q1-65082Q9, San Jose, CA, USA, January 30, 2007.
[16] Tony Lin and Pengwei Hao, "*Compound Image Compression for Real Time Computer Screen Image Transmission*", IEEE Transactions on Image Processing, Volume 14, Issue 8, pp. 993-1005, August 2005.
[17] Wenpeng Ding, Dong Liu, Yuwen He, and Feng Wu, "*Block-based Fast Compression for Compound Images*", ICME, paper ID 1722, pp. 809–812, 2006.
[18] Debargha Mukherjee, Christos Chrysafis, and Amir Said, "*JPEG2000-Matched MRC Compression of Compound Documents*", IEEE International Conference on Image Processing (ICIP), Volume 3, pp. 225-228, June 2002.
[19] Cheng H. and Bouman C. A., "*Document Compression Using Rate-Distortion Optimized Segmentation*", Journal of Electronic Imaging, Volume 10, Issue 2, pp. 460–474, April 2001.
[20] Fu Limin, "*Neural Network in Computer Intelligence*", Tata McGraw Hill, New York, USA, 2003.
[21] Sachin Rawat, K.S. Sesh Kumar, Million Meshesha, Indraneel Deb Sikdar, A. Balasubramanian, and C.V. Jawahar, "*A Semi-automatic Adaptive OCR for Digital Libraries*", Proceedings of Seventh IAPR Workshop on Document Analysis Systems (LNCS 3872), pp. 13–24, January 13-15, 2006.