IMPROVEMENT OF ZONE CONTENT CLASSIFICATION BY USING BACKGROUND ANALYSIS

Yalin Wang[†], Robert Haralick[†], and Ihsin T. Phillips [‡]

[†] Department of Electrical Engineering
University of Washington Seattle, WA 98195 U.S.A.

[‡] Department of Computer Science/Software Engineering
Seattle University, Seattle, WA 98122 U.S.A.

{ylwang, haralick, yun@george.ee.washington.edu}

Abstract. This paper presents an improved zone content classification method. Motivated by our novel background-analysis-based table identification research, we added two new features to the feature vector from one previously published method [7]. The new features are the total area of large horizontal and large vertical blank blocks and the number of text glyphs in the zone. A binary decision tree is used to assign a zone class on the basis of its feature vector. The training and testing data sets for the algorithm include $1,600$ images drawn from the UWCDROM-III document image database. The classifier is able to classify each given scientific and technical document zone into one of the nine classes, 2 text classes (of font size $4-18$pt and font size $19-32$ pt), math, table, halftone, map/drawing, ruling, logo, and others. The improved zone classification method raised the accuracy rate to 97.53% from 96.67% and reduced the median false alarm rate to 0.12% from 0.27%.

1 Problem Statement

Let $\mathcal{A}$ be a set of zone entities. Let $\mathcal{L}$ be a set of content labels, such as text, table, math, etc. The function $f : \mathcal{A} \rightarrow \mathcal{L}$ associates each element of A with a label. The function $V : \mathcal{A} \rightarrow \Lambda$ specifies measurements made on each element of $\mathcal{A}$, where Λ is the measurement space.

The zone content classification problem can be formulated as follows: *Given a zone set $\mathcal{A}$ and a content label set $\mathcal{L}$, find a classification function $f : \mathcal{A} \rightarrow \mathcal{L}$, that has the maximum probability:*

$$P(f(\mathcal{A})|V(\mathcal{A})) \tag{1}$$

In our current approach, we assume conditional independence between the zone classifications, so the probability in Equation 1 may be decomposed as

$$P(f(\mathcal{A})|V(\mathcal{A})) = \prod_{\tau \in \mathcal{A}} P(f(\tau)|V(\tau)) \tag{2}$$

The problem can be solved by maximizing each individual probability $P(f(\tau)|V(\tau))$ in Equation 2, where $\tau \in \mathcal{A}$.

In our zone content classification experiment, the elements in set $\mathcal{A}$ are zone groundtruth entities from UWCDROM III document image database [6]. The elements of set $\mathcal{L}$ are text with font size $\leq$ 18pt, text with font size $\geq$ 19pt, math, table, halftone, map/drawing, ruling, logo, and others. $V(\tau)$ is a feature vector generated for τ, where $\tau \in \mathcal{A}$. We used a decision tree classifier to compute the probability in Equation 2 and make the assignment.

2 Related Work and Paper Organization

A complete document image understanding system can transform paper documents into a hierarchical representation of their structure and content. The transformed document representation enables document interchange, editing, browsing, indexing, filing and retrieval. The zone classification technique plays the key role in the success of such a document understanding system. Not only is it useful for successive applications such as OCR, table understanding, etc, but it can be used to assist and validate document segmentation.

In the literature, Sivaramakrishnan et. al [7] extracted features for each zone such as run length mean and variance, spatial mean and variance, fraction of the total number of black pixels in the zone, and the zone width ratio for each zone. They used the decision tree classifier to assign a zone class on the basis of its feature vector. They did their experiments on 979 document images from UWCDROM I image database. Liang et. al [5] developed a feature based zone classifier using only the knowledge of the widths and the heights of the connected components within a given zone. Le et. al [4] proposed an automated labeling of zones from scanned images with labels such as titles, authors, affiliations and abstracts. The labeling is based on features calculated from optical character recognition(OCR) output, neural network models, machine learning methods, and a set of rules that is derived from an analysis of the page layout for each journal and from generic typesetting knowledge for English text.

We developed a novel background-analysis-based table identification technique. We repeated Sivaramakrishnan et. al's work [7] on a larger database with a goal to improve its performance on table zone classification. Although some background analysis techniques can be found in the literature([1],[2]), none of them, to our knowledge, has been used in the table identification problem. We added two new features: the total area of large horizontal and vertical blank blocks and the number of text glyphs in the given zone, to the original feature vector. We improved the accuracy rate and reduced the false alarm rates for most of the nine classes.

The rest of this paper is divided into 5 sections. section 3 gives the definitions of large horizontal and large vertical blank blocks. The two new features are described in section 4. A brief introduction to the decision tree classifier is given in section 5. The experimental results are reported in section 6. Our conclusion and statement of future work are discussed in section 7.

3 Some Definitions

A *polygonal area* on a document page is given by a pair (θ, I), where $\theta \in \Theta$ specifies the label that designates the physical type of content, e.g. text block, text line, word, etc., and I is the area enclosed by boundary of the polygon. The boundary of a polygon is given as a sequence of line segments connecting the successive vertices of the polygon. The vertices are given as a clockwise ordered list of points.

We use white-pixels to represent background pixels and black-pixels for foreground pixels. In our background analysis, both foreground and background are represented by polygonal areas. Noting that, for the purpose of zone classification, the foreground polygonal areas are connected components and their bounding boxes and the background polygonal areas are blank blocks and their bounding boxes.

Definition 1: Let $\mathcal{Z}$ represent a *zone* with R rows and C columns. Let (x_1, y_1) be the coordinate of its lefttop vertex.

$$\mathcal{Z} = \{(r, c) \in Z \times Z | x_1 \leq r < x_1 + R, y_1 \leq c < y_1 + C\}$$

Definition 2: A *horizontal white run* p is a sequence of horizontally contiguous white-pixels whose starting and ending pixels must have a black-pixel or the zone border as its neighbor.

$$p = ((r_1, c_1), ..., (r_n, c_n))$$

where $(r_i, c_i) \in \mathcal{Z}$, $r_i = r_{i-1}$, $c_i = c_{i-1} + 1$, for $i = 2, ..., n$, and pixel (r_1, c_1), (r_n, c_n) must have a black pixel or the zone border on its left and right side, respectively. Similar to the run-length encoding, for each run, the location of the starting pixel of the run and its horizontal length must be recorded. We call them as Row(p), Column(p), and Length(p).

Definition 3: A *horizontal blank block* $\mathcal{HR}$ is a sequence of horizontal white runs on contiguous rows, whose starting point column coordinates and lengths are the same.

$$\mathcal{HR} = (p_1, ..., p_n)$$

where Row(p_i) = Row$(p_{i-1}) + 1$, Column(p_i) = Column(p_{i-1}), Length(p_i) = Length(p_{i-1}), for $i = 2, ..., n$.

Clearly, the same idea can be applied to define *vertical white run* and *vertical white blank block*. The difference will be that we use vertically contiguous pixels to define vertical white run and vertical runs on contiguous columns for vertical white blank block.

Definition 4: A horizontal blank block $\mathcal{HR} = b_r \times b_c$, with lefttop vertex coordinate (x_{b1}, y_{b1}), is a *large horizontal blank block* if and only if it satisfies the following conditions:

- its row numbers and column numbers are large enough compared with the current zone. Specifically, $\frac{b_c}{C} > \theta_1$, where θ_1 is 0.1. This number was statistically determined by our experiment on another table data set;

- It does not touch left or right side of the zone bounding box, i.e. $x_{b1} \neq x_1$ and $x_{b1} + b_c \neq x_1 + C$;

where C is the column number in the zone.

Definition 5: A vertical blank block $\mathcal{VR} = b_r \times b_c$, with lefttop vertex coordinate (x_{b1}, y_{b1}), is a *large vertical blank block* if and only if it satisfies the following conditions:

- Its row number and column number are large enough compared with the current zone. Specifically, $b_r \gg mh$ and $\frac{b_c}{mw} > \theta_2$, where mh and mw are the median height and median width of text glyphs in the zone. θ_2 is empirically determined as 1.4;
- It does not touch left or right side of the zone bounding box, i.e. $x_{b1} \neq x_1$ and $x_{b1} + b_c \neq x_1 + C$;

where C is the column number in the zone.

4 Two New Features for Zone Content Classifier

Our goal is to improve table zone classification using our background-analysis-based table identification technique. The most distinguished feature of a table zone is the large gaps between its columns. Even for the tables with rulings, we will have gaps if we remove the rulings. On the background analysis, we can expect to find both large horizontal and large vertical blank blocks inside the table zones.

On the other hand, we can also find many large horizontal and large vertical blank blocks in drawing/map zones. To discriminate table zones and drawing/map zones, we added one more feature, the number of text glyphs in a zone. The so-called text glyphs are not from any OCR output. They are outputs of a statistical glyph filter. The inputs of this filter are the glyphs after finding connected component operation. The filter classifies each glyph into one of two classes: text glyph and non-text glyph. The filter uses a statistical method to classify glyphs and was extensively trained on UWCDROM III document image database.

The newly added features are:

1. The total area of large horizontal and large vertical blank blocks, A.

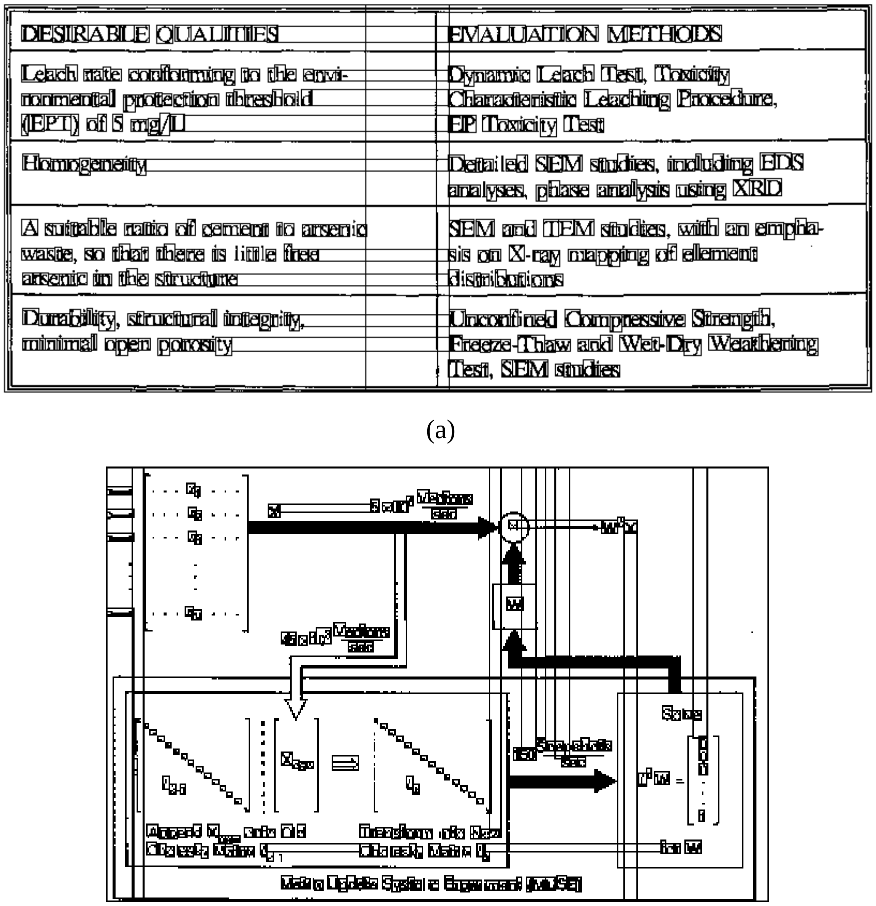
$$A = \sum_{R \in \mathcal{B}} Area(R),$$

   where $\mathcal{B} = \{R | R = \mathcal{HR}$ or $\mathcal{VR}, \mathcal{HR} \subset \mathcal{Z}$ and $\mathcal{VR} \subset \mathcal{Z}\}$.
2. The number of text glyphs in this zone, N, normalized by the zone area.

$$\frac{N}{Area(\mathcal{Z})}$$

Figure 1 illustrates two zone examples showing the overlayed bounding boxes of large horizontal blank blocks, large vertical blank blocks and text glyphs.

Together with the 67 features used in Sivaramakrishnan et. al's work [7], we have a total of 69 features. We feed the generated feature vector to a binary decision tree to get the zone classification.



DESIRABLE QUALITIES	EVALUATION METHODS
Leach rate conforming to the environmental protection threshold (EPT) of 5 mg/L	Dynamic Leach Test, Toxicity Characteristic Leaching Procedure, EP Toxicity Test
Homogeneity	Detailed SEM studies, including EDS analyses, phase analysis using XRD
A suitable ratio of cement to arsenic waste, so that there is little free arsenic in the structure	SEM and TEM studies, with an emphasis on X-ray mapping of element distributions
Durability, structural integrity, minimal open porosity	Unconfined Compressive Strength, Freeze-Thaw and Wet-Dry Weathering Test, SEM studies

(a)

(b)

Fig. 1. Illustrates the example bounding boxes of large horizontal blank blocks, large vertical blank blocks, and text glyphs. The so-called text glyphs are labeled by a statistical glyph filter. (a) a table zone example; (b) a map/drawing zone example.

5 Decision Tree Classifier

A decision tree classifier makes the assignment through a hierarchical decision procedure. The classification process can be described by means of a tree, in which at least one terminal node is associated with each class and nonterminal nodes represent various collections of mixed classes.

For the construction of a decision tree, we need a training set of feature vectors with true class labels. Let $U = \{u_k : k = 1, \cdots, N\}$ be a unit-training set to be used to design a binary tree classifier. Each unit u_k has an associated measurement X_k with known true class. At any non-terminal node, let Ω^n be the set of M^n classes still possible for a unit at node n. Let $U^n = \{u^n_k : k = 1, \cdots, N^n\}$ be the subset of N^n training units associated with node n. If the number of units for class c in node n is denoted by N^n_c, we must have $N^n = \sum_{c=1}^{M^n} N^n_c$.

Now we describe how the decision rule works at node n. Consider unit u^n_k which has measurement vector x^n_k. If the discriminant function $f(x^n_k)$ is less than or equal to a threshold, then u^n_k is assigned to class Ω^n_{left}, otherwise it is assigned to class Ω^n_{right}. An assignment to Ω^n_{left} means that a unit descends to the left child node and an assignment to Ω^n_{right} can be understood in a similar way. Given a discriminant function f, the units in U^n are sorted in such a way that $f(x^n_k) \leq f(x^n_{k+1})$ for $k = 1, \cdots, N^n - 1$. Let w^n_k be the true classes associated with the measurement vectors x^n_k. Then a set of candidate thresholds T^n for the decision rules is defined by

$$T^n = \left\{ \frac{f(x^n_{k+1}) - f(x^n_k)}{2} \mid w^n_{k+1} \neq w^n_k \right\}$$

For each threshold value, unit u^n_k is classified by using the decision rule specified above. We count the number of samples n^t_{Lc} assigned to Ω^n_{left} whose true class is c and we count the number of samples n^t_{Rc} assigned to Ω^n_{right} whose true class is c, that is,

$$\begin{aligned} n^t_{Lc} &= \#\,\{\, u^n_k \mid f(x^n_k) \leq t \ \text{ and } \ w^n_k = c \,\} \\ n^t_{Rc} &= \#\,\{\, u^n_k \mid f(x^n_k) > t \ \text{ and } \ w^n_k = c \,\} \end{aligned}$$

Let n^t_L be the total number of samples assigned to Ω^n_{left} and n^t_R be the total number of samples assigned to Ω^n_{right}, that is,

$$n^t_L = \sum_{c=1}^{M^n} n^t_{Lc} \qquad \text{and} \qquad n^t_R = \sum_{c=1}^{M^n} n^t_{Rc}$$

We define the impurity IP^t_n of the assignment made by node n to be

$$IP^t_n = \sum_{c=1}^{M^n} \left(-n^t_{Lc} \log \frac{n^t_{Lc}}{n^t_L} - n^t_{Rc} \log \frac{n^t_{Rc}}{n^t_R} \right)$$

The discriminant threshold t is chosen such that it minimizes the impurity value IP^t_n. The impurity is such that it gives a minimum value when the training samples are completely separable.

The learned discriminant function splits the training subset into two subsets and generates two child nodes. The process is repeated at each newly generated child node until a stopping condition is satisfied, and the node is declared as a terminal node based on a majority vote. The maximum impurity reduction, the maximum depth of the tree, and minimum number of samples are used as stopping conditions.

At the testing stage, a feature vector is the input to a decision tree, a decision is made at every non-terminal node as to what path the feature vector will take. This process is continued until the feature vector reaches a terminal node of the tree, where a class is assigned to it.

6 Experiments and Results

A hold-out method is used for the error estimation in our experiment. We divide the data set into N parts, train on the first $N-1$ parts and then test on the Nth part. Then train on the other $N-1$ parts, omitting the $(N-1)$th part. Continue the training and testing, each time omitting one part from the decision tree construction and then testing on the omitted part. Then the combined N part results are put together to estimate the total error rate [3].

The output of the decision tree is compared with the zone labels from the ground truth in order to evaluate the performance of the algorithm. A contingency table is computed to indicate the number of zones of a particular class label that are identified as members of one of nine classes. The rows of the contingency table represent the true classes and the columns represent the assigned classes. The cell at row *r* and column *c* is the number of zones whose true class is *r* while its assigned class is *c*. We also compute four rates here: *Correct Recognition Rate(CR)*, *Mis-recognition Rate(MR)*, *False Alarm Rate(FR)*, *Accuracy Rate(AR)*. Suppose we only have two classes: a and b. The possible true- and detected-state combination is shown in Table 1.

True Class	Assigned Class	
	a	b
a	P_{aa}	P_{ab}
b	P_{ba}	P_{bb}

Table 1. Possible true- and detected-state combination for two classes

We compute three rates for class a as follows:

$$CR = \frac{P_{aa}}{P_{aa} + P_{ab}} \qquad MR = \frac{P_{ab}}{P_{aa} + P_{ab}} \qquad FR = \frac{P_{ba}}{P_{ba} + P_{bb}}$$

And we compute the accurate rate as:

$$AR = \frac{P_{aa} + P_{bb}}{P_{aa} + P_{ab} + P_{bb} + P_{ba}}$$

We conducted a zone classification experiment on a significant sized data set. In our experiment, the training and testing data set was drawn from the scientific document pages in the University of Washington document image database III [6]. It has $1,600$

scientific and technical document pages with a total of $21,477$ zones. The class labels for each of the zones are obtained from the database. These zones were belonged to nine different classes: 2 text classes (of font size $4-18$pt and font size $19-32$pt), math, table, halftone, map/drawing, ruling, logo and others.

The two new features and 67 features from Sivaramakrishnan et. al.'s work [7] were computed for every zone in the document. Our hold-out method used N as 3. The set of feature vectors was divided into 2 parts, one part with two thirds of the vectors, used for creating the decision tree and the other part with the remaining one third of the vectors, used for testing the tree. The set of document images was randomly partitioned into three equal sized groups. The test was done three times each with a different third of the data set as test and two thirds of the data set for training.

Our experiment result is shown in Table 2. For a total of $24,177$ zones, the accuracy rate was 97.53% and the median false alarm rate was 0.12%. For the purpose of comparison, we also show Sivaramakrishnan's result in Table 3. They did their experiment on 979 page images in the University of Washington document image database I. For a total of $13,831$ zones, the accuracy rate was 96.67% and the median false alarm rate was 0.27%.

Besides the improvement in the accuracy rate in the larger data set, it is also interesting if we study the classifiers' performance on each class. We improved the correct recognition rates in most of classes except math, logo, and others classes and reduced the false alarm rates for most of classes except text with font size ≤ 18pt and others class. For the table class, the results show that the current classification rate remained about the same rate and the false alarm rate was reduced more than 50%. It proved the newly added two features can improve global and table zone classification results.

	T1	T2	M	T	H	M/D	R	L	O	CR	MR
T1	21416	18	55	7	3	6	1	0	3	99.57%	0.43%
T2	16	99	3	0	5	1	2	0	1	77.95%	22.05%
M	174	0	564	3	0	15	2	0	0	74.41%	25.59%
T	17	0	2	151	0	37	2	0	0	72.25%	27.75%
H	5	0	0	0	289	32	0	1	11	85.50%	14.50%
M/D	30	0	19	17	19	632	0	0	3	87.78%	12.22%
R	9	0	0	0	0	2	421	0	1	97.23%	2.77%
L	5	1	2	1	2	1	0	1	0	7.69%	92.31%
O	11	0	0	0	32	21	0	0	6	8.57%	91.43%
FR	10.01%	0.08%	0.35%	0.12%	0.26%	0.49%	0.03%	0.00%	0.08%		

Table 2. Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class in UWCDROM III. In the table, T_1, T_2, M, T, H, MD, R, L, O represent text with font size $\leq$ 18pt., text with font size $\geq$ 19pt., math, table, halftone, map/drawing zone, ruling, logo, others, respectively.

	T1	T2	M	T	H	M/D	R	L	O	CR	MR
T1	11974	29	54	14	7	16	11	4	1	98.88%	1.12%
T2	16	71	2	0	2	6	1	4	3	67.62%	32.38%
M	65	0	427	2	0	16	1	1	0	83.40%	16.60%
T	16	0	2	95	2	17	0	0	2	70.90%	29.10%
H	1	1	0	0	122	28	0	2	2	78.21%	21.79%
M/D	17	2	15	27	22	387	0	1	2	81.82%	18.18%
R	14	1	4	0	0	0	288	7	0	91.72%	8.28%
L	3	1	1	1	1	1	0	5	0	38.46%	61.54%
O	5	0	0	1	3	3	0	0	2	14.29%	85.71%
FR	7.96%	0.25%	0.59%	0.33%	0.27%	0.65%	0.10%	0.14%	0.07%		

Table 3. Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class reported in Sivaramakrishnan et. al [7]. In the table, T_1, T_2, M, T, H, MD, R, L, O represent text with font size ≤ 18pt., text with font size ≥ 19pt., math, table, halftone, map/drawing zone, ruling, logo, others, respectively

7 Conclusion and Future Work

Given the segmented document zones, correctly determining the zone content class is very important for the further processes. We improved one zone classification method by adding two new features and conducted the experiment on a larger data set. Our zone classification method can classify each given zone into one of the nine classes, 2 text classes (of font size $4 - 18$pt and font size $19 - 32$ pt), math, table, halftone, map/drawing, ruling, logo, and others. We raised the accurate rate to 97.53% from 96.67% and reduced the median false alarm rate to 0.12% from 0.27%.

Our future work will include improving math zone classification result, studying our zone classifier performance on a even larger image data set and developing an automatic zone segmentation method to produce the input to the classifier.

References

1. A. Antonacopoulos: Page segmentation using the description of the background. Computer Vision and Image Understanding, pages 350–369, June 1998.
2. H. S. Baird: Background structure in document images. Document Image Analysis, pages 17–34, 1994.
3. R. Haralick and L. Shapiro: Computer and Robot Vision. Addison Wesley, Vol 1, 1992.
4. D. X. Le, J. Kim, G. Pearson, and G. R. Thom: Automated labeling of zones from scanned documents. Proceedings SDIUT99, pages 219–226, 1999.
5. J. Liang, R. Haralick, and I. T. Phillips: Document zone classification using sizes of connected components. Document Recognition III, SPIE'96, pages 150–157, 1996.
6. I. Phillips: Users' reference manual. CD-ROM, UW-III Document Image Database-III, 1995.
7. R. Sivaramakrishnan, I. Phillips, J. Ha, S. Subramanium, and R. Haralick: Zone classification in a document using the method of feature vector generation. Proceedings of the 3rd ICDAR, pages 541–544, Aug. 1995.