Zone Classification in a Document using the Method of Feature Vector Generation

Ramaswamy Sivaramakrishnan, Ihsin T. Phillips, Jaekyu Ha
Suresh Subramanium, and Robert M. Haralick

Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195, U.S.A.
E-mail : ramas@brownie.cs.unlv.edu
suresh@shasta.ee.washington.edu,
{yun, ha, haralick}@george.ee.washington.edu

Abstract

A document can be divided into zones on the basis of its content. For example, a zone can be either text or non-text. This paper describes an algorithm to classify each given document zone into one of nine different classes. 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 are extracted. Run length related features are computed along four different canonical directions. A decision tree classifier is used to assign a zone class on the basis of its feature vector. The performance on an independent test set was 97 %.

1 Introduction

A document is varied in content. It can contain text, math, figure zones, etc. Each of these zones has its own characteristic features. For example, a math zone may contain symbols like $=$, $+$, $\sum$, $\int$, $\cdots$, which a text zone may not contain. On the other hand, figure zones may not contain any symbols or text. Captions and pure text vary in font size and shape.

Every page contains numerous zones. Each zone is specified by a unique zone identification number, a rectangular box which encloses the zone and is represented by the coordinates of the leftmost-top and rightmost-bottom points, and the zone type. This paper describes an algorithm for determination of the zone type given the coordinates of the leftmost-top and rightmost-bottom points, and the document image.

In the design of a zone classifier, a set of measurements are first done and properties of the zone are calculated along the rows, columns, right diagonals and left diagonals of the zone. The method used to obtain these properties of the zone is described in Section 3. A feature vector that consists of all these properties as fields is formed for each zone. Statistical pattern recognition is used to classify each zone on the basis of its feature vector[1][2][3]. The performance of the classifier is calculated from a contingency table indicating the number of zones in the document image data base that are identified in the ground truth as class i and assigned by the decision rule to a different class j.

2 Decision Tree

A decision tree classifier makes the assignment through a hierarchial deçision 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 the 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 $U^n = \{u_k^n : k = 1, \cdots, N^n\}$ 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_c^n, we must have $N^n = \sum_{c=1}^{M^n} N_c^n$.

0-8186-7128-9/95 $4.00 © 1995 IEEE

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

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

$$T^n = \left\{ \frac{f(x_{k+1}^n) - f(x_k^n)}{2} \mid w_{k+1}^n \neq w_k^n \right\} \qquad (1)$$

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

$$\begin{aligned} n_{Lc}^t &= \#\{u_k^n \mid f(x_k^n) \leq t \text{ and } w_k^n = c\} \\ n_{Rc}^t &= \#\{u_k^n \mid f(x_k^n) > t \text{ and } w_k^n = c\} \end{aligned}$$

Let n_L^t be the total number of samples assigned to Ω_{left}^n and n_R^t be the total number of samples assigned to Ω_{right}^n, that is,

$$n_L^t = \sum_{c=1}^{M^n} n_{Lc}^t \qquad \text{and} \qquad n_R^t = \sum_{c=1}^{M^n} n_{Rc}^t$$

We define the purity PR_n^t of the assignment made by node n to be

$$PR_n^t = \sum_{c=1}^{M^n} \left(n_{Lc}^t \log \frac{n_{Lc}^t}{n_L^t} + n_{Rc}^t \log \frac{n_{Rc}^t}{n_R^t} \right) \qquad (2)$$

The discriminant threshold t is chosen such that it maximizes the purity value PR_n^t. The purity is such that it gives a maximum value when the training samples are completely separable. The expansion of the tree is stopped if the decision rules used at the nodes cannot be applied to smaller training sets or if the number of feature vectors goes smaller than a predetermined value.

In the simplest form of a linear decision rule (f is linear), one of the components of the measurement vector is taken and a set of candidate thresholds, T, are calculated for that component. The one that gives the maximum purity is chosen. This process is repeated for all the components in the measurement vector. Out of the thresholds computed for all the components in the measurement vector, the one that yields maximum purity is chosen.

When a feature vector is 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.

3 Zone Classification

Every zone in the document is considered to be rectangular. Properties of each zone are used by the classifier for the process of classification. For a zone, features are computed for each line along four different canonical directions, horizontal, vertical, left-diagonal, and right-diagonal.

3.1 Line Features

In each zone the black pixels are assumed to be foreground pixels and the white pixels are assumed to be background pixels. For any line belonging to a zone the line projection along a particular direction (row, column or diagonal) is the number of foreground pixels belonging to that line.

For every line in a particular direction (row, column or diagonal), the sum of run lengths in the foreground gives the line projection. The starting position of the current run is stored, and the number of runs on the foreground along the line is calculated. From these quantities, the run length mean, the run length variance, the spatial mean and the spatial variance can be calculated. The line features calculated for each line are stored in a separate data structure. This process is repeated along all the remaining directions. Section 3.2 gives a step by step procedure that was followed to extract the zonal features.

3.2 Zone Features

For a zone, the following features are evaluated along each of the directions with the help of the line features computed earlier.

1) The number of runs on the foreround and background are calculated by adding up the number of runs along each line in the zone. When calculated along all four directions, these add up to 8 features in the feature vector of the zone.

2) The total run length for all the runs along the background is calculated and is divided by the total number of background runs to give the run length mean

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

of the background. This can be written as

$$rlmean_0 = \frac{rl_0}{totalruns_0} \tag{3}$$

where $rlmean_0$ is the run length mean of the background, $totalruns_0$ is the total number of background runs in the zone and rl_0 is the total run length of the backround in the zone. The run length mean for the foreground can be calculated similarly. This process is carried out along all four directions. The run length means represent a total of 8 fields in the feature vector of the zone.

3) Run length variance along the background can be obtained by calculatingthe mean of the squares of all the run lengths in the zone along the background and subtracting it by the square of the run length mean. This can be expressed as

$$rlvar_0 = \frac{m_0^2}{totalrun_0} - (rlmean_0)^2 \tag{4}$$

where $rlvar_0$ is the variance of the background and m_0^2 is obtained by taking the sum of the squares of the lengths of each background run in the zone. The run length variance for the foreground can be calculated similarly. This process is carried out along all four directions. We get a total of 8 features in the feature vector of the zone due to run length variances.

4) Spatial mean is the mean line of a zone along any direction. For example, along the rows it can be written as

$$r' = \frac{1}{A} \sum_{(r,c)\in R} r \tag{5}$$

where (r, c) is the coordinates of the point belonging to the zone represented by R, and A is the area of the zone. A similar notation can be used to determine the spatial mean along any other direction. In the algorithm, the product of the projections of every line with its corresponding line number is calculated along a particular direction. The sum of all these products, divided by the area of the zone gives the spatial mean for the zone. This can be given as

$$spmean = rP/A \tag{6}$$

where $spmean$ is the spatial mean of the zone and rP is the product of the line projections and line numbers along a particular direction. A is the total number of foreground pixels in the zone. Spatial mean along the other directions is calculated similarly to give a total of 4 features for the zone.

5) Spatial variance μ_{rr} along the direction of the rows can be given by

$$\mu_{rr} = \frac{1}{A} \sum_{(r,c)\in R} (r - r')^2 \tag{7}$$

The spatial variance along the other directions, can be described by a similar notation. In the algorithm, spatial variance is calculated by using the following equation,

$$spvar = \sum_{y_1 \in R} \left[\frac{(y_1 - spmean)^2}{lineproj} \right] \tag{8}$$

where $spvar$ is the spatial variance of the zone, $lineproj$ is the line projection of the line whose line number is given by y_1 and R is the zone. Spatial variance along other directions is calculated similarly to give a total of 4 features for each zone.

6) Line projection, number of foreground runs, spatial mean and run length mean are represented as functions of the line number. Autocorrelations of these functions are calculated using the formula

$$R_{xx}(\tau) = \int f(x) f(x + \tau) dx \tag{9}$$

where $R_{xx}(\tau)$ is the autocorrelation of the function f, and τ is the index of the autocorrelation. The index τ for which the autocorrelation function goes to 10 % of its maximum value is calculated. Another feature of interest is the slope of the tangent to the autocorrelation function when $\tau \rightarrow 0$. When these two features are calculated for all the four functions mentioned and along all the directions, we get a total of 32 features.

7) A fraction of the black pixels to the total number of pixels in the zone is another feature that reveals information about the class of a zone. This feature contributes 1 field to the feature vector of the zone.

8) The area covered by the zone is calculated by multiplying the length of the zone with its width. The length and width of a zone can be obtained from the coordinates of the zone. This features contributes 1 field to the feature vector.

9) It is a common observation that math zones and figure zones have a smaller width compared to text zones. For every zone, the quotient of the the zone width and the width of its column is calculated. This feature contributes 1 feature to the feature vector of the zone.

When all these features are computed the feature vector has a total of 67 fields.

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

3.3 Feature Vector Generation and Decision Rule

Section 3.2 describes how a 67 dimensional feature vector is formed. The class labels for each of the zones are obtained from the database. After forming the labeled feature vector for every zone, the following procedure is followed for the process of classification.

1) The zones in the entire document are divided into two distinct parts: a training part consisting of two thirds of the zones in the document image database, and a testing part consisting of the remaining zones in the document.

2) A decision tree is created using the labelled feature vectors for the zones belonging to the training set. The decision tree uses the thresholding decision rule and the minimum entropy purity condition.

3) The unlabeled feature vectors in the testing set are used to test the decision tree.

4) The process is repeated by changing the training set and the testing set until the algorithm is tested on the entire database.

5) 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.

6) A contingency table is computed to indicate the number of zones of a particular class label that are identified as members of another class.

Example

The University of Washington document image data set[4][5] has 979 distinct images with a total of 13831 zones. These zones belonged to nine different classes: 2 text classes (of font size 4-18pt and font size 19-32 pt), math, table, halftone, map/drawing, ruling, logo and other. These classes were assigned labels $1-9$. The features mentioned in Section 3.3 were calculated for every zone in the document and the resulting feature vector had 67 features. The total number of terminating nodes in the decision tree was chosen to be 25. For the purposes of creating a decision tree, 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 testing is done three times each with a different third of the tree data set as test and two thirds of the data set for training. A contingency table for the result of the classification algorithm is shown in Table 1. This table shows the number of zones of a particular class that are identified as members of a different class. The performance of this algorithm is 97 %.

	T1	T2	M	T	H	MD	R	L	O
T1	11974	29	54	14	7	16	11	4	1
T2	16	71	2	0	2	6	1	4	3
M	65	0	427	2	0	16	1	1	0
T	16	0	2	95	2	17	0	0	2
H	1	1	0	0	122	28	0	2	2
MD	17	2	15	27	22	387	0	1	2
R	14	1	4	0	0	0	288	7	0
L	3	1	1	1	1	1	0	5	0
O	5	0	0	1	3	3	0	0	2

Table 1: Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class: 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.

4 Conclusion

If a document is divided into zones, the identification of the class of a zone can be done by calculating the region properties of each zone and using statitical pattern recognition. The performance of this algorithm is high. The largest classification error occurs in confusing math zones and text zones. The performance of the algorithm can be improved by adding features that will help identify the math zones from text zones. We are now exploring the utility of the left offset of a zone for this purpose.

References

[1] R.M. Haralick and L.G. Shapiro, "Computer and Robot Vision," Addison Wesley, Vol 1 1992

[2] Abend, K. and T.J. Harley, "Comments on the Mean Accuracy of Statistical Pattern Recognizers," IEEE Transactions on Information Theory, Vol. IT-14, 1968, pp. 420-421

[3] Chen, C. H., " Statistical Pattern Recognition," Spartan Books, Rochelle Park, NJ, 1973

[4] I.T. Phillips, S. Chen, J. Ha and R.M. Haralick, "English Document Database Design and Implementation Methodology," *Proc. of the second Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, April 26-28, 1993, pp. 65-104.

[5] I.T. Phillips, S. Chen and R.M. Haralick, "English Document Database Standard", *Proc. of the Second ICDAR* , Japan, October 20-22, 1993, pp. 478-483.

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE