LAMP-*** MDA****
CAR-TR-*** May 2007
CS-TR-***
UMIACS-2007-***

Multi-Class Document Layout Classification using Random Chopping

Mei Huang

Laboratory for Language and Media Processing (LAMP)
University of Maryland, College Park, MD 20742-3275

Abstract

This paper proposes a multi-class document layout classification/recognition system using a method called random chopping. A scanned document image undergoes text line extraction and is represented as a set of quadrilaterals for every pair of text lines. For compact representation, a dictionary of quadrilateral clusters is built beforehand, and a document image is then represented as a word occurrence histogram by looking up its quadrilaterals in the dictionary. The training process iteratively chops all training classes into two partitions and trains a linear classifier for this split. A binary coordinate space is built from all chops, and every document's histogram descriptor is then projected to this space to form a binary signature. Layout similarity is reduced to distance computation between two signatures. Our experiments demonstrate that this multi-class classification system achieves very good performance not only on trained classes but also on instances from layout classes never seen in the training.

1 Introduction

Document classification and retrieval based on layout similarity is gaining interests to some researchers with the increasing need both in government and companies to searching through very large collections

of scanned materials. For example, in the discovery phase of litigation cases, companies are often asked to provide all the documents they produced in decades, resulting huge databases. Content search based on keywords input by hand is formidable, while OCR can provide some help, it falls short of those mixed with handwritings and/or low quality photocopies. Treating scanned documents as ordinary images and applying content based image retrieval techniques for searching special local or global patterns have been regarded as a promising way, at least for helping decreasing the searching range for regular keywords search. For example, there are situations for search based on document layout, because similar layouts are often indicative of a particular source-deposit forms for a given bank, letterhead from a given company, etc. While using other search capabilities, or in the process of examining the collection, a user may find a document (form, invoice, letter, etc) with a layout of particular interest. The user would show the system one or more examples of documents with the layout of interest, and from these examples the system scores the documents and returns a list of document images with the most relevant on top, as would an internet search engine. A system based on layout is robust even when OCR fails to decode the content, as is the case with handwritten documents or old photocopies.

We design a system for scanned textual document classification based on layout similarity. A scanned document image undergoes text line extraction and is represented as a set of quadrilaterals formed by every pair of text lines. A dictionary of quadrilateral clusters is built beforehand, and a dictionary word occurrence histogram is computed by looking up all quadrilaterals of a document in the dictionary. We then train the system through iteratively chopping all training classes into two partitions and training a linear classifier for this chop. A binary coordinate space is build when training is done, and every document is then projected to this space. Layout similarity is reduced to distance computation between two points in this space.

The following sections are organized as follows: Section 2 gives a brief introduction on our previous document ranking system according to layout similarity. Section 3 explains the theoretic foundation of the methods, the random chopping method for pattern recognition, used in our multi-class classification system. Section 4 describes in detail how we represent a document, how we build the new coordinate system from training, and how is retrieval carried out. Section 5 shows our experimental results and

Section 6 is a short conclusion.

2 Previous Work

A document ranking system using layout has been developed in LAMP lab in 2006 [1]. The system takes in some samples of the target layout class as positive training samples and also samples from a large range of layout classes different from the target class as negative training samples. The training process clusters the document constitutional elements' descriptors from positive samples and from negative samples separately, and then find for each positive cluster the neighbor negative clusters with preset neighborhood range radius. Each positive cluster is weighted using the ratio of its member size relative to the total members in the neighborhood. Given a query, its constitutional elements are distributed to their respective nearest positive clusters within a given range and the weights of the clusters are added up and normalized to use as the query's similarity score to the target layout class. As for the constitutional element, four kinds of elements have been proposed and tested, which are all based on text line extraction on a binarized document image. Two of them use quadrilaterals formed by text line pairs as the constitutional elements, one describing the quadrilateral using turning functions and the other using 5D shape vectors. Another two use text lines as constitutional elements. Experiments showed that the representation using text line pair as constitutional elements and 5D shape vectors achieves the best performance when taking both the precision and the time cost into account, so we also adopt this representation scheme in our new system (See Section 4.1 for detail).

The disadvantage of the document ranking system is obvious: each time the target layout is changed, training sample need to change accordingly and training needs to restart. Since training is the most time consuming part compared to retrieval, this system is not favorable in situations where too many target layout classes need to deal with. Also, as the weight of a positive cluster is a kind of relative occurrence ratio against negative clusters, it sometimes becomes a burden to decide the appropriate numbers of positive and negative training samples. Too many negative samples may shadow some distinct positive features, while too few negative samples may exaggerate the importance of indistinct

features. And since this ”too many”/”too few” cannot be determined beforehand at the input time, and also difficult to determine even after training without applying visualization methods on the training results, the detection of inappropriate input and the correction of this problem cannot be automated in a single system.

Our new system is aimed at the above mentioned disadvantages of the previous system. Through a fast learning on more than one target class at one time, the new system does not need to restart training for each single target class, instead, it can classify multiple classes at once. It even can rank documents by similarity to an unseen layout to some degree, due to the generalizability of the combination of classifiers. As the classifiers do not take as input the relative feature occurrence ratio from each target class samples, they do not suffer from the second disadvantage mentioned above.

3 Pattern Recognition by Chopping

Generally, there are two methods for pattern recognition. One is based on a priori knowledge and the other on statistical information extracted from some training data set. Extensive research has been carried out on the latter, which is the central topic of machine learning. Patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. A complete pattern recognition system consists of a feature extraction mechanism that computes numeric or symbolic information from the observations; and a classifier or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

In [3], Fleuret et. al. proposed a generic method, called chopping, for one-example-learning based pattern recognition. Their research was to investigate the learning of the appearance of an object from a single image, i.e., they want to decide if two images are picturing the same object and the first image is seen as a single training sample and the second one a test. For this purpose, they propose to build a large number of binary splits of the object space, which form a binary coordinate system with each coordinate being the output of a binary classifier. Ideally all images of any given object would be assigned the same binary label, which we call a signature. In essence, the method is a kind of classifier combination.

The general framework is as follows:

1. Randomly assigning label 1 to N_1 objects appearing in the training set and 0 to the others. Note the assignment is in the object space, not the image space, i.e., if an object is labeled 1, all images of this object are assigned 1 in this iteration.

2. Treat all these objects as 2 categories (one is class 1, one is class 0) and train a binary classifier.

3. Repeat the above two steps M times.

We use this framework in our system, and choose to use logistic regression linear classifier for each split. While Fleuret only accept balanced split, i.e., the number of objects with label 1 is always equal to that with label 0, we accept quasi-balanced chops (See Section 4.2.4).

4 Multi-Class Document Layout Classification

4.1 Document Representation

A scanned document image is originally represented as a 2D pixel matrix of typical size 1700*1500. Low level features resulted from operations at pixel level, such as pixel difference at each pixel site, cannot provide enough information for structural characterization, and thus not a good representation for the task of efficient layout comparison. So we choose to use statistical measurements and represent an image as a histogram with each entry recording the occurrence of a predefined set of layout related features. A histogram vector is later called a document descriptor. To prepare the predefined set of layout related features, a dictionary of features has been created, and after feature extraction on a given document, each feature is consulted against the dictionary and the corresponding histogram bin is increased by one.

For layout recognition purpose, a feature is defined as the shape of the quadrilateral constituted of a pair of text lines, and a document is thus characterized by a set of quadrilaterals. As all pairs of text lines are considered, the layout of a textual document is uniquely defined by such a set.

4.1.1 Text line Extraction

We used the module *getLines* in *DocLib* [4] for text line extraction, which is based on connected components grouping and text de-skewing on a document image. The output of this module includes the 2D coordinates of the two end points of a text line, the font height and the rotation angle relative to the horizontal line.

Figure 1: *Top*: Text lines detected in tax form by grouping connected components of black pixels. *Bottom*: Set of quadrilaterals formed by considering all pairs of text lines.

4.1.2 Quadrilateral Shape Descriptor

Two text lines, the line connecting the two left end points and the line connecting the two right end points form a quadrilateral. To compare the similarity of two quadrilaterals, we define a shape descriptor for a quadrilateral as a 5D vector, two of them being the length of the two text lines, two of them the diagonal lengths and the last the length of the line connecting the midpoints of the two text lines, see Fig. 2. Every quadrilateral has a unique 5D shape descriptor and vice versa.

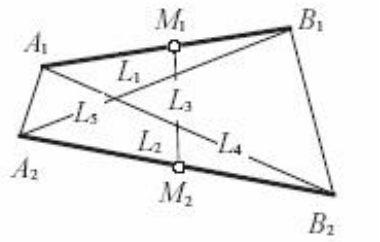

Figure 2: 5D Shape Descriptor for a quadrilateral, the five dimensions are: the lengths of the two text lines, the two diagonals, and the connection between the two text line midpoints.

4.1.3 Dictionary Generation and Document Histogram Descriptor

As mentioned above, we need first to build a dictionary of quadrilateral shapes (words) so that we can represent a document as a word occurrence histogram. To do so, we gather a random set of document images and extract quadrilateral shape vectors from them. We then cluster these vectors by fixing the radius of a cluster. Specifically, each time we distribute a new vector to its nearest cluster by examining the distance between the new point and the center of a cluster. If the distance is no larger than the prefixed radius and is the minimum, we attribute the point to the cluster and update the cluster center to be the mean of the cluster members. If no cluster is found, we create a new cluster whose only member is the quadrilateral under question.

The fixed radius is chosen empirically, and the clustering process consists of several iterations. We also tried k-means clustering with the number of clusters specified close to the number of clusters we got in the above method. The resultant performance does not change too much, but the time cost for k-means clustering with the same number of iterations is much higher than the above clustering method. From our experimental data, k-means spends nearly 4 hours to build a 730-word dictionary, and 5.5 hours for a 1000-word one; while the above clustering method only uses several minutes to get the job done.

After clustering, only the cluster centers are recorded and become the words of the dictionary. To represent a document image, we consult the dictionary and find for each quadrilateral in the document the index of word/cluster that the quadrilateral belongs to, and increment the corresponding histogram bin. There are cases that some quadrilateral may not find its corresponding word in the dictionary, and we just neglect those quadrilaterals as we can believe the dictionary has gathered the majority of

the important features and thus those not appearing in the dictionary are of minor effect in the layout comparison at large.

4.2 Training

As mentioned in Section 2, we train a bunch of layout classes at one time. Suppose there are m training classes, each having m_i training samples, then the classes are randomly divided into two sides and a feature selection procedure using Conditional Mutual Information Minimization criteria (CMIM) [2] is conducted to deduct the dimension into manageable range. A binary classifier is trained on the selected feature dimensions using Logistic Regression (LR). The chopping process is repeated several times. When all chops are done, every single chop is evaluated through a validating process, i.e., every trained LR classifier is applied to a validating data set, and the classification precision is recorded and used as the weight of this chop in retrieving phase.

4.2.1 CMIM Feature Selection

Feature selection is not only for dimension reduction to gain computation efficiency, but also for reducing overfitting of learning methods. In our document layout classification, every document is a vector of nearly a thousand dimensions. Such high dimension is easy to cause the classifiers to suffer from the curse of dimensionality in machine learning process, especially for learning from few data samples. Dimensionality deduction methods that are based on eigen decomposition to transform the data set to a new coordinate system, usually orthogonal, such as PCA, is very demanding in computation when the original dimension is high, and thus not a good choice for real time applications. We turn to feature selection methods that can bypass coordinate system transforming while achieving correlation deduction between selected features.

Fleuret proposed in [2] a fast feature selection criterion called CMIM (Conditional Mutual Information Maximization) to ensure a good tradeoff between individual discrimination and weak in-between dependency. The way to achieving this is to iteratively pick features which maximize their predictive power with the class and minimize the conditional mutual information with the class given the features

already picked. Their experiments demonstrate that CMIM outperforms other feature selection methods that based on mutual information, and that it is robust when challenged by noisy training sets. We thus choose to use this method for feature selection.

Conditional mutual information is defined as the difference of conditional entropy $H(C|v_1)$, which quantifies the uncertainty of variable *C* (in our case, it is the class label to predict) when feature v_1, a dimension of the 976 dimension histogram vector, is known, and conditional entropy $H(C|v_1, v_2)$, which quantifies the uncertainty of itC given both v_1 and v_2.

$$I(C; v_2|v_1) = H(C|v_1) \tag{1}$$

It can be seen that if v_2 is closely correlated to v_1, and thus share a large part of information about *C*, the uncertainty of the two terms are not very different, and the resultant conditional mutual information is small, even if v_1 and v_2 are both individually informative, i.e., $H(C|v_1)$ and $H(C|v_2)$ are both high/low. Whereas, if v_2 bears information about *C* which is not already contained in v_1, the mutual information is high and thus v_2 is a good choice for selection. This is the theoretic foundation of the feature selection method.

Generally speaking , the goal of feature selection is to select a small subset of features $v_i, 1 \leq i \leq K$ to minimize $H(C|X_{v_i})$, X_{v_i} being the value of v_i. This formulation requires $n - choose - k$ evaluations of conditional entropy of $k + 1$ variables, which is not practical and will make the minimization of $H(C|X_{v_i})$ computationally intractable. Fleuret's solution is to compare each new feature with each of those already picked. A feature v is a good choice if and only if $I(C; v|w)$ is large for every w already picked. The iterative scheme is formalized as follows:

$$v_1 = argmax_n I(C; X_n) \tag{2}$$

for any k, $1 \leq k < K$,

$$v_{k+1} = argmax_n min_{l<=k} I(C; X_n|X_{n_l}) \tag{3}$$

So, v_1 is the feature that brings most information about class label variable *C*, and the following features v_k are selected one by one that the minimal conditional mutual information between *C* and v_k is maximized. $I(C; X_n|X_{n_l})$ is small if v_n itself is non informative about *C* or if the information carried in v_n already borne by some already picked feature.

Now the evaluation of $I(C; X_n|X_{n_l})$ only requires estimation of the entropy of triples. And a fast implementation is given in [2], which is based on the observation that $I(C; X_n|X_{n_l})$ is decreasing with the iterative process and thus a partial computation is applied.

In our application, each feature is a dimension in the histogram vector representation of a document, so X_i takes integer values in $[0, MAXQUAD]$, where ***MAXQUAD*** is the preset upper limit of the number of quadrilaterals in a document image. The typical value of this upper limit is about 10000 for a 100-text line document, which cause the computation of the entropies time consuming as we need to count the number of (x_i, c_i) for every possible combination of (x_i, c_i) across all training samples and in the case of triples we need to consider (x_i, x_j, c) which leads to 10000^2 combinations. To accelerate, we binarized each feature by choosing a division point adaptively. Specifically, we find the optimal division point in $[0, MAXQUAD]$ in the sense that this division leads to the best classification performance of this single feature for current chop. After determination of the division point, the values become zero if they are below the division point; and 1 otherwise. Note the division point is different from feature to feature and from chop to chop.

Two criteria are used to stop the selection process. One is that the preset maximum number of chops being reached. The other is when the information gain from adding an additional feature is below a preset threshold. The two are combined using *Or*.

4.2.2 Logistic Regression

Regression is a process of learning a continuous target variable. Most regression models can be turned into classifiers using logistic trick of logistic regression. Logistic regression (LR) is a discriminative classifier as it learns $P(Y|X)$ directly from training data by assuming a sigmoid functional form for $P(Y|X)$, where Y is the binary class label variable and X is the feature variable:

$$P(Y = 1|X = x) = 1/(1 + e^{-w*x}) \tag{4}$$

$$P(Y = 0|X = x) = e^{-w*x}/(1 + e^{-w*x}) \tag{5}$$

where w is the weight vector to be learned. Let $p_1(x; w) = P(Y = 1|X = x)$ and $p_0(x; w) = 1 - p_1(x; w)$, the log odds of class label 1 is $\log p_1(x; w)/p_0(x; w) = w * x$, which is a linear function of x, that is why LR belongs to linear classifier category.

Let S be the training set. The goal of learning is to find hypothesis h that maximizes $P(h|S)$, i.e., to fit a conditional probability distribution given the training data. For parameterized representation of hypothesis, we use h_θ. Assuming uniform h_θ, we have

$$\begin{aligned} argmax\,_{h_\theta} P(h_\theta|S) &= argmax\,_{h_\theta} P(S|h_\theta)P(h_\theta)/P(S) \\ &= argmax\,_{h_\theta} \log P(S|h_\theta) \end{aligned}$$

In the case of logistic regression, the parameters are the weights w. $L(S; w) = P(S|h_\theta)$, and thus the w that maximizes the likelihood of the training data is the target.

4.2.3 Computing the Likelihood

Assuming each training sample (x_i, y_i) is drawn independently from the same underlying (though unknown) probability distribution $P(x, y)$. This means that the log likelihood of S is the sum of the log likelihoods of the individual training examples:

$$\begin{aligned} \log P(S|h) &= \log \Pi_i P((x_i, y_i)|h) \\ &= \textstyle\sum_i \log P((x_i, y_i)|h) \end{aligned}$$

$$
\begin{aligned}
argmax_{\ h} \log P(S|h) \qquad &= argmax_{\ h} \textstyle\sum_i \log P((x_i, y_i)|h) \\
&= argmax_{\ h} \textstyle\sum_i \log P(y_i|x_i, h)P(x_i|h) \\
&= argmax_{\ h} \textstyle\sum_i \log P(y_i|x_i, h)P(x_i) \\
&= argmax_{\ h} \textstyle\sum_i \log P(y_i|x_i, h)
\end{aligned}
$$

The optimal w^* can be found using standard optimization algorithm.

4.2.4 Random Chopping

Random chopping is implemented by randomly assigning zeros and ones to the training layout class label array. Those with the same value (zero or one) are treated as one class for a chop. We enforce a quasi-balanced chopping by discarding those partitions that the difference between the number of zeros and the number of ones is larger than 4, i.e., we only accept a chop candidate that belongs to one of the $5-5$, $4-6$, $3-7$ partitions when the total number of training classes is 10. The consideration behind this is to deduct the learning bias caused by the unbalance sample size of each side.

4.2.5 Validating Each Single Classifier

After training, each LR classifier is evaluated on a validating set different from training set. This is to assign a weight to each chop for the combination of all chops during retrieving time. See the next section for details.

4.3 Retrieval

When training is done, all chops form a binary coordinate system. Every training sample has a coordinate, which is called signature later, when projected to the chopping space. For each training layout class, the mean coordinate of training samples forms the reference of this class and is used in computing distance from a given query. For a given query, the class label is determined through finding out the

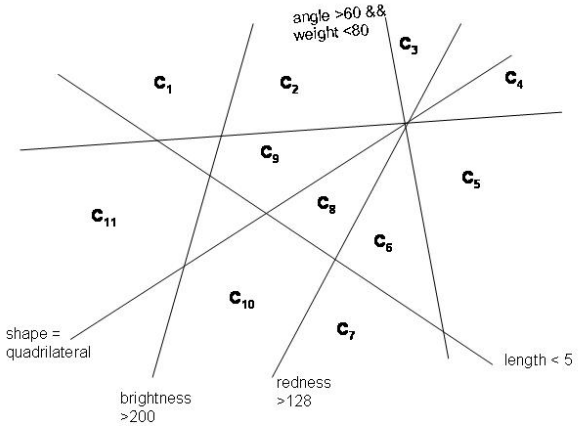

Figure 3: A chopping space example. There are 11 object categories and they are divided by 6 chops. The criterion set used is angle, weight, redness, length, brightness, shape, height, number of legs.

training class mean point that is closest to the query in the binary chopping space. Closeness is quantified using a distance function, which is designed to take into consideration both the classifiers?consistency on training samples and their precision on validating data set.

4.3.1 Signature Computation

After training, the features selected for each chop and the LR classifier parameters are stored. Any document image, represented as a m-D histogram vector is then projected to the chopping space through the following steps: for each chop the corresponding selected features (among the m dimensions) are input to the classifier, the output 0 or 1 is the coordinate of current chop. After projection, each document is now represented as a binary vector with dimension equal to the number of chops. Figure 4 shows a simple 3D chopping coordinate system; the coordinates/siguature of any document can only be at the corners of the cube, but the averaged signatures can be distributed inside the cube.

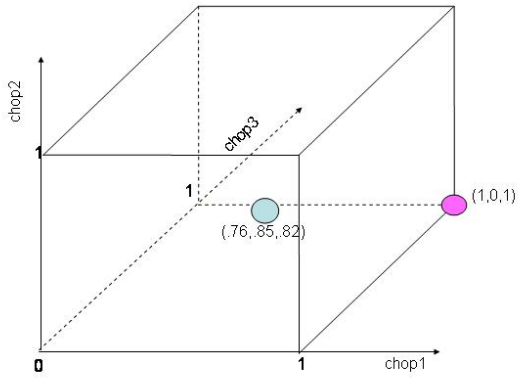

Figure 4: Projecting a document shape descriptor into a 3D chopping space. The pink dot at (1,0,1) is the coordinate or signature of a document after projection. The blue dot inside the cube is the averaged signature of a layout class.

4.3.2 Distance function

The mean signature of training samples in each class is a numeric vector with each dimension in $[0, 1]$. It is easy to see that the more one dimension is close to one or zero, the more consistent the classifier is on the training sample, on the contrary, if one dimension is close to 0.5, the classifier is very inconsistent and should not be trusted too much in the retrieval phase. Similarly, the precision of each classifier on the validating data reflects their discriminative power and is used as the weights of the classifier. Combining these two considerations, we design the distance function as follows:

$$D(Q, C_i) = \sum_k F(Q_k, C_{i,k}) * P_k \tag{6}$$

where Q is the binary signature of a query document, Q_k is the k-th dimension of the signature, C_i is the mean signature of the i-th training class, $C_{i,k}$ the k-th dimension of it, and P_k is the precision of the k-th classifier on the validating set. $F(Q_k, C_{i,k})$ is the function that computes the difference of the output of a single classifier between the query and the i-th training class, and is defined as follows:

$$F(Q_k, C_{i,k}) = (1 - Q_k)(1 - C_{i,k}) + Q_k C_{i,k} \tag{7}$$

Since Q_k can only be one or zero, only one term in the right hand side will be large than zero. When the absolute difference between Q_k and $C_{i,k}$ is less than 0.5, the maximum of $1 - C_{i,k}$ and $C_{i,k}$ will be assigned to the function, and participates the computation of the score of this single bit k. It can be seen that the distance between Q and C_i is the weighted sum of difference of every signature dimension.

The class label of the given query is determined as the training class which has the minimum distance to the query and also the distance is above a preset threshold. If the minimum distance is above the threshold, we label the query as from unseen layout class.

5 Experiments

We carried out four experiments. The first tests the multi-class classification performance of the system on natural data set and on simulated data set. The second examines the optimal number of chops. The third compares the random chopping method with the biclassification for multiple class classification method. At last, we test the generality of the system on the task of ranking documents with untrained layouts.

5.1 Multi-Class Classification

We train the system using 100 training instances from 10 layout classes, each classes having 10 instances. A total of 1304 documents from these 10 classes were selected by hand and used as test set. Sample documents for each of these classes are shown in Figure 5, and a description of these classes is given in the caption. Among the ten classes considered in turn, the first six classes group papers printed in English, where the differentiating visual features are the numbers of text columns, the lack of symmetry, the presence of a single or double-column abstract in a double-column document, the presence of a title, etc. The last four classes contain handwritten Arabic text, and respectively contain (1) forms, (2) notes, (3) résumés, and (4) formal letters with headers. The confusion matrix of the ten classes is shown below,

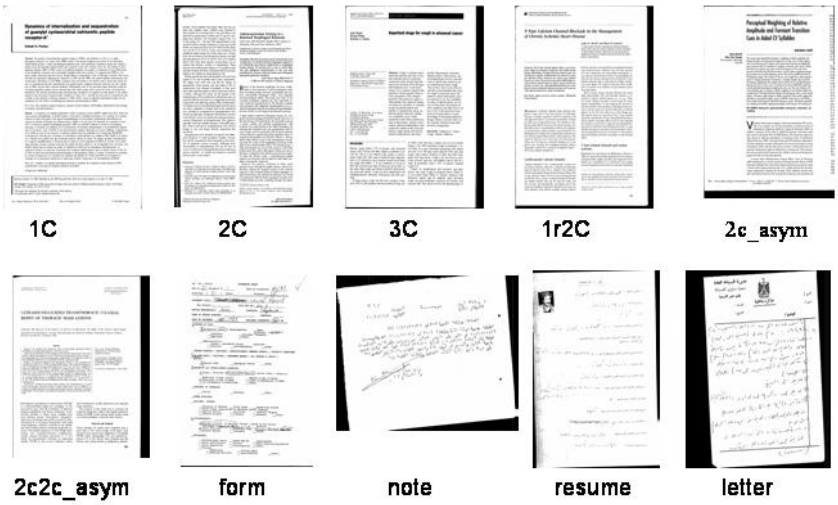

Figure 5: Document samples from the ten classes used in our experiments. The first six classes group papers printed in English, where the differentiating visual features are the numbers of text columns, their symmetry, the presence of a single or double-column abstract in a double-column document, the presence of a title, etc. The last four classes contain handwritten Arabic text, and respectively contain (1) forms, (2) notes, (3) résumés, and (4) formal letters with headers.

in which the numbers after the class name in the first column are the number of test documents of each class. Empty entries in the matrix are zeros. The result is derived from 32 random chops.

Table 1: 10-class Classification Confusion Matrix

	1c	**2c**	**1r2c**	**3c**	**2c-asym**	**2c2c-asym**	**form**	**note**	**résumé**	**letter**
1c(113)	87	8	16		2					
2c(144)		133	4	1		5	1			
1r2c(431)	9	168	246			8				
3c(23)				23						
2c-asym(6)					3	3				
2c2c-asym(45)		1				44				
form(62)							62			
note(264)	3					2	3	230	2	24
resume(121)	1			1			13	2	101	3
letter(95)				1		1	17	27	7	52

The diagonal dominance demonstrates that most instances are correctly classified. But there are cases with low precision. E.g., more than $1/3$ test instances from Class *1r2c* (two column printout with title zone) are misclassified as Class *2c* (two column printout), as some *1r2c* samples' title zone is not that dominant and easily dominated by their two-column feature. For similar reason, *Class2* and *Class4* have an evident misclassification to each other.

We did a further test on closeness of trained binary coordinate system. Several simulated document layout classes were made by combining halves of two original classes. Fig. 6 shows some examples. The layout distances from the synthesized document to the training classes show without exception that the two nearest training classes for each synthesized instance are the two source classes from which the synthesized one is made. This phenomenon demonstrates that the coordinate system built from the chops is reasonable under the layout similarity measure define in equation 6.

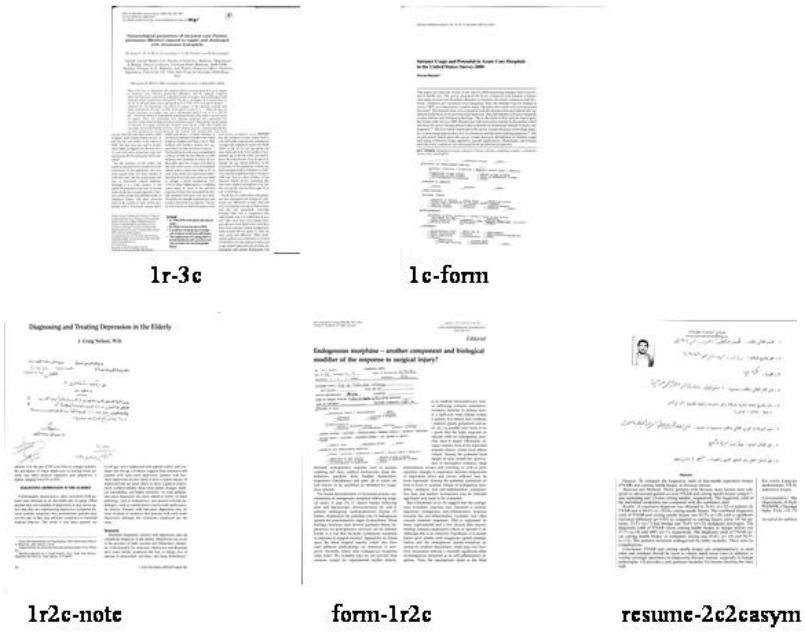

Figure 6: Document samples from the four synthesized classes used in our experiments. The name of each synthesized class indicates how the synthesization is carried out. E.g., class *1r-3c* is made of original class *1r* in the upper part and original class *3c* in the bottom part.

5.2 the Optimal Number of Chops

When the number of training classes is n, the maximum number of chops can be $\sum_{i=1}^{n/2} C_n^i$, which is quite large even when only those quasi-balanced chops are selected. Naturally, one can assume there exists an optimal number of chops, as intuitively when the chops are too few, it is equally that the judges are too few, and the discriminative power will not be strong, whereas when there are too many chops, the consistency of the judges suffers. We did experiment to test several number of chops, and find the performance follows a Gaussian-like curve. See Fig. 7 below.

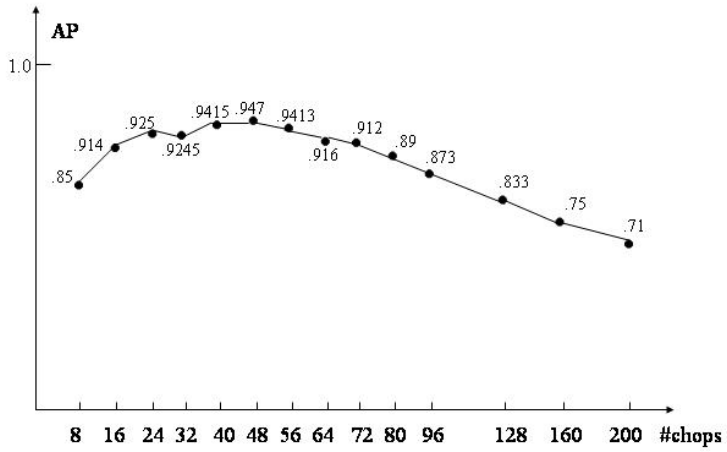

Figure 7: Classification performances under different number of chops. The performance measure is the averaged classification precision.

Although it is impractical to test every single number of chops in real application to select the optimal number and it is possible that the optimal number is not unique, we can sample some number and test their performance on a small test set to determine a potentially good number.

5.3 BiClassification vs. Random Chopping

The essence of this random chopping multiple class classification is a combination scheme on a set of binary classifiers. A popular multi-class classification method is to iteratively treat one class as one side and all others as the other side and train a binary classifier, and samely, these binary classifiers are combined for determine the class label of a given query. Thus the essential difference between this method and the random chopping based method is on the keyword ”random”. We thus carried experiments to compare the performance of the deterministic way and the random way of chopping. The results demonstrate that ”random” chopping achieves better performance as (1)it is quasi-balanced chopping, whereas the deterministic one-against-others scheme suffers from structural unbalance since the number of training samples from each layout class are similar; (2)the maximum number of random chops has a large value(see the above sub-section), and for quasi-balanced chopping we choose a number around 3 times the training classes, while for the deterministic chops, the total number is limited to the number of training classes. The reason(2) is more important as when we decrease the number of random chops to the number of training classes, the performance predominance disappears.

5.4 Ranking Documents with Untrained Layouts

This experiment is to test the generalizability of the trained binary coordinate system. Taking a document whose layout is unseen before as the target, we rank the test documents according to their similarity to this target. In this experiment, seven classes from the previous training set are kept to train the binary system, while the other three were used as test target in turn. The two performance measures, Mean Average Precision (MAP) and Mean Average Normalized Rank(MANR) **??**, show that the system is capable of recognize new layouts (See the table below).

6 Conclusion

We propose in this technical report a method for multiple class textual document layout classification system based on random chopping. As multiple layout classes are trained at one time, we overcome

Table 2: Document Ranking results

	AP	**NAR**
1r1r2c(83)	0.998	0.001
form(52)	0.966	0.008
letter(46)	0.651	0.112

the disadvantage of the previous system which requires restart training from scratch each time a new target layout class is introduced. Besides, this system gains generalizability through combination of a set of generalizable classifiers. It can tell whether a given set of instances of unseen layouts are from same layout class under already learned criteria. Another highlight is that adding new layout class to the already trained chopping space can be done efficiently by projecting new training instances into the space and adding necessary chops to the space whenever two layout classes are intensively mixed up in the old space. Unlike the previous system, which has to store all positive quadrilateral cluster parameters for retrieval usage, the new system needs to store only the chop parameters, which is unrelated to the number of training samples. Our experiments demonstrate that it it can also handle situations when training sample for a layout class is quite sparse, say, less than 10.

The present research focuses on textual document images, so patterns like stamps, tables, figures, etc. are ignored. Our future work will consider heterogeneous feature types and accordingly a combination of different types of classifiers aimed at different feature types.

References

[1] M. Huang, D. DeMenthon, D. Doermann, L. Golebiowski, “Document Ranking by Layout Relevance”, Proc. ICDAR, pp. 362–366, Aug. 2005.

[2] F. Fleuret, ”Fast binary Feature Selection with Conditional Mutual INformation”, Jounal of Machine Learning Research pp.1531-1555, Nov. 2004

[3] F.Fleuret, G.Blanchard, ”Pattern Recogntion from One Example by Chopping”, NIPS 2005

[4] S.Jaeger, G. Zhu, and D. Doermann, ”DOCLIB: A Software Library for Document Processing”, In Proceedings of Document Recognition and Retrieval XIII(SPIE), San Jose, CA, USA, Jan. 2006.