An Approach to the Segmentation of Multi-page Document Flow Using Binary Classification

Onur Agin, Cagdas Ulas, Mehmet Ahat, Can Bekar
R&D and Special Projects Department
Yapi Kredi Bank, Kocaeli, Turkey
Email: {onur.agin, cagdas.ulas, mehmet.ahat, can.bekar}@yapikredi.com.tr

ABSTRACT

In this paper, we present a method for segmentation of document page flow applied to heterogeneous real bank documents. The approach is based on the content of images and it also incorporates font based features inside the documents. Our method involves a bag of visual words (BoVW) model on the designed image based feature descriptors and a novel approach to combine the consecutive pages of a document into a single feature vector that represents the transition between these pages. The transitions here could be represented by one of the two different classes: continuity of the same document or beginning of a new document. Using the transition feature vectors, we utilize three different binary classifiers to make predictions on the relationship between consecutive pages. Our initial results demonstrate that the proposed method can exhibit promising performance for document flow segmentation at this stage.

Keywords: Document Flow Segmentation, BoVW Model, Transition Feature Vector, Support Vector Machine, Random Decision Forest.

1. INTRODUCTION

A document flow is a list of successive scanned pages which are located in a production chain and represent several documents without explicit separation mark between them [1]. The problem of batch scanning of multiple documents commonly arises in many business application contexts where the separation of successive data is required. One of the main steps in batch scanning process is the segmentation of the resulting page flow into continuous sets of pages corresponding to the physical documents, defining a procedure also known as document separation or segmentation [2].

In many business applications, page flow separation is typically achieved by manually inserting separator pages or machine readable marks in the stream during scanning [3]. Nevertheless, the process of sorting and introducing separator sheets is time-consuming, costly and error-prone [4]. Hence, the development of an automatic separation system which is capable of segmenting a stream of documents without the need of any prior information on the number of pages or on document category is needed. This type of system should analyze the content of successive pages and point out the limit pages of document stream where each document may represent a set of successive well-ordered pages [1].

Although the development of a document flow segmentation system without any type of human intervention is a considerable commercial interest, comparatively little attention has been paid to this problem until recently by research community [2]. To the best of our knowledge, very few methods have been proposed to tackle this problem and to find out automatic solutions. Collins-Thompson et al. [5] proposed a document segmentation approach utilizing page similarity based on structural and textual similarity. In their work, they consider document separation as a constrained bottom-up clustering problem where each page is at first considered as a cluster, then progressively merging pairs of clusters using a single linkage criterion is performed. Nevertheless, their approach heavily relies on a quite good OCR performance and on correct localization of several layout features such as the location of page number inside the text pages. They also assume that pages in the same document contain a lot of similarities whereas in real-world scenario, it is very likely that the content of pages may possess very little similarities. Another proposed method by Meilender et al. [1] depends on Hidden Markov Model (HMM) within the framework of the multi-gram models and they propose an adaptation of the forward-backward algorithm to obtain the best segmentation for document streams. However, the
proposed method is tested on homogeneous documents and it is only applicable for invoice documents. Therefore, the performance of the method is open to questions and should be investigated on heterogeneous document collections.

In this paper, we propose a method for document flow separation where the task is to partition a flow of documents into multiple subsets of documents. Our approach incorporates both image based feature descriptors and font based features inside the text pages. We represent the structural difference between consecutive pages with a single feature vector representation which is given as the input to a binary classifier to predict on the relationship between consecutive pages. Our work differs from previous approach proposed by Daher et al. [6] in several ways: (1) We mainly deal with image based features for document pages whereas the work in [6] only focuses on textual content of the documents, thus, it relies on a pretty good OCR performance (2) Incorporating the BoVW model, we propose a new approach to determine the image based similarity measure between consecutive pages of document flow depending on Visual Words' matching.

The rest of this paper is organized as follows: In Section 2, the proposed document flow separation algorithm is presented. We provide a description of the experimental setup and demonstrate the classification results in Section 3, and finally we give concluding remarks in Section 4.

2. PROPOSED METHOD

The proposed method in this work comprises of following steps: the determination of image based features descriptors, the use of BoVW model and the utilization of Visual Words matching approach to obtain image based similarities, the representation of transition feature vector and the analysis of different binary classifiers. Fig.1 illustrates the main steps of the proposed approach in a flow chart. Following sections will provide the details of these steps.

2.1 Image Based Feature Descriptors

The document images in our database are binary (monochrome) images that the pixel values can take only one of two values (0,1), suffering from lack of information as compared to color and gray-scale images. Hence, specifically designed features should be found for this type of images. In this work, instead of using SIFT features [7] which are commonly used for local feature extraction in many computer vision applications; we represent image patches with a less number of easily extractable features.

First, we divide each training and test image into small square patches to determine a 60 × 40 image layout with totally 2400 image patches and then represent each patch with 4 feature descriptors based on structural variations in small regions of the image.

If $P_c$ denotes the width (column number) of the image patch; $P_r$ denotes the height (row number) of the patch and $w(i,j)$ represents the pixel value of the pixel at $t$th row and $f$th column, the feature descriptors of each patch are calculated as follows:

1. **Column Standard Deviation ($\sigma_c$)**

$$
\sigma_c = \sqrt{\text{Variance}(X_C)}
$$

where $X_C = [X_1X_2...X_{P_C}]$ and each $X_i = \sum_{j=1}^{P_R} w(j,i)$

2. **Row Standard Deviation ($\sigma_r$)**

$$
\sigma_r = \sqrt{\text{Variance}(X_R)}
$$

where $X_R = [X_1X_2...X_{P_R}]$ and each $X_i = \sum_{j=1}^{P_C} w(i,j)$.

3. **Patch Mean Value ($m_p$)**

$$
m_p = \frac{1}{P_RP_C} \left( \sum_{j=1}^{P_C} \sum_{i=1}^{P_R} w(i,j) \right)
$$
4. Pixel Transition Intensity ($t_s$)

\[
t_s = \frac{1}{P_R(P_C-1)} \left( \sum_{j=1}^{P_R} \sum_{i=1}^{P_C} \text{trans}(j, i) \right)
\]

where $\text{trans}(j, i)$ is defined as follows:

\[
\text{trans}(j, i) = \begin{cases} 
1, & \text{if } w(j, i - 1) = 1 \text{ and } w(j, i) = 0 \\
0, & \text{otherwise}
\end{cases}
\]

2.2 Bag of Visual Words (BoVW) Model

In computer vision, the BoVW model [8] can be applied to image classification and related tasks by treating image descriptors as words. A bag of visual words is a sparse vector representation of mostly occurrence counts or presence of the visual words from a vocabulary of local image features. A vocabulary (or codebook) of visual models is obtained by clustering local image descriptors extracted from training images, which is also described as vector quantization of image features into visual words. The vector quantization process is generally done by a hard or soft assignment (clustering) and a codebook of visual words is obtained. Visual words (codewords) are then defined as the centers of learned clusters. Each patch in a test image is mapped to a certain codeword whose cluster center is the closest to the feature vector of that patch.

In this study, we use $k$-means clustering [9] to determine the codebook of visual words. The number of cluster is empirically set as 4 due to the intuition that the structural variations inside the local patches of text document images are small. After obtaining the visual words, each train and test images are represented with a sequence of visual words in a $60 \times 40$ layout.

2.3 Visual Words Matching Approach

In this work, we propose a new metric in order to obtain the image based similarity measure between consecutive pages in a document flow. Our approach for calculating the similarity is based on the exact matching of visual words of layout images which are represented with a sequence of visual words. The approach can be described as in the following way:

Let $S$ is the set of available number of visual words, represented as $S = \{V_1, V_2, ..., V_{N_c}\}$ where $N_c$ is the total number of distinct visual words (codewords) in the vocabulary. For each $V_i \in S$, the similarity value $\gamma$ between consecutive pages can be calculated as in (5):

\[
\gamma(V_i(p_i, p_{i+1})) = \frac{H_{V_i}(p_i, p_{i+1}) - M_{V_i}(p_i, p_{i+1})}{TN_{V_i}(p_i) + 1}
\]

where $p_i$ and $p_{i+1}$ are denoted as the document pages at $i$th and $(i+1)$th position of the stream respectively, $H_{V_i}(p_i, p_{i+1})$ represents the number of hit positions where both $p_i$ and $p_{i+1}$ has the same visual word $V_i$, and thus referred to as exact matching, $M_{V_i}(p_i, p_{i+1})$ represents the number of miss positions where $V_i$ does not match for $p_i$ and $p_{i+1}$, Lastly $TN_{V_i}(p_i)$ stands for total number of positions inside the layout having $V_i$ for page $p_i$.

Note that the similarity value $\gamma(V_i(p_i, p_{i+1}))$ is not necessarily the same as $\gamma(V_i(p_{i+1}, p_i))$ due to the reason that the first value indicates how similar the layout of $p_i$ to the layout of $p_{i+1}$ whereas the latter indicates how similar the layout of $p_{i+1}$ to the layout of $p_i$.

2.4 Transition Feature Vector

A single feature vector which represents the relationship between two consecutive pages is constructed in this work. This single feature vector representation is also called as “transition feature vector” throughout the paper. In addition to image based similarity measures mentioned in Section 2.3, we utilize font based features from document pages so as to benefit from the information related to difference in textual content of successive pages. To extract these features, all page images have been OCRed with commercial OCR system of ABBYY FineReader Engine.

One of the font based features extracted from consecutive pages is the font type difference which is denoted as $FT(p_i, p_{i+1})$ for page $p_i$ and $p_{i+1}$. This feature is a binary feature that can be labelled as 0 or 1. “0” means that the major
font type of consecutive documents in a document flow is not same whereas “1” indicates that the major font type is same for these documents. Another font based feature is the font size difference which is represented as $FS(p_i, p_{i+1})$ for page $p_i$ and $p_{i+1}$. $FS(p_i, p_{i+1})$ can be computed as follows:

$$FS(p_i, p_{i+1}) = \frac{|t_{p_{i+1}} - t_{p_i}|}{t_{p_i} + t_{p_{i+1}}}$$

(6)

where $t_{p_i}$ and $t_{p_{i+1}}$ stand for average font size of the characters inside the document page $p_i$ and $p_{i+1}$, respectively.

Given the image based similarity values and font based differences between two consecutive pages, the transition feature vector $F_T$ is represented as in (7).

$$F_T = \left[ y\left(V_1(p_i, p_{i+1})\right), y\left(V_1(p_{i+1}, p_i)\right), \ldots, y\left(V_{N_c}(p_i, p_{i+1})\right), y\left(V_{N_c}(p_{i+1}, p_i)\right), FT(p_i, p_{i+1}), FS(p_i, p_{i+1}) \right]$$

(7)

### 2.5 Classification

The classification problem here involves two classes: whether the transition between consecutive pages indicates the continuity of the same document or beginning of a new document. The first class is denoted as (C) and the latter is denoted as (B) in Fig.1. The whole transition feature vector representing a document stream is obtained as in Section 2.4 and given as the input to a binary classifier in order to decide on the classes of available transitions. In this work, we have used three different types of classifier for binary classification: Support Vector Machine (SVM), Random Decision Forest (RDF) and Multilayer Perceptron (MLP). The performance evaluation of these classifiers will be demonstrated in detail in Section 3.
3. EXPERIMENTAL RESULTS

In this study, we use our own dataset to evaluate the performance of proposed document flow segmentation method. The whole dataset consists of 267 heterogeneous multi-page documents with a total number of 3268 document pages. The total number of transitions between document pages is 3001. Most of the images in the training and test dataset are contaminated with marginal (clutter) noise and salt & pepper noise during scanning, transmission or conversion to digital form. Any noise reduction procedure is not performed on documents and thus, the proposed document flow segmentation method here is robust to noise in heterogeneous document collections.

As mentioned in Section 2.5, three different algorithms for classification analysis of the binary classification problem are compared in this study. For each algorithm, the data set is partitioned into training and test subsets in the ratio of 50%-50%. 100 such partitions are generated randomly for the experiments. On each partition, the compared algorithms are trained and tested respectively. The final performance of each algorithm on a data set is the average of the results over the 100 partitions. For SVM classifier, we use the RBF kernel, \( k(x_i, x_j) = \exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \) and LIBSVM tool [10] is used for RBF-SVM implementation. The performance of RBF-SVM is mainly affected by the kernel parameters, such as \( \sigma \) and the regularization parameter, \( C \). For this reason, 5-fold cross-validation (CV) is conducted for both parameter tuning and generalization capability estimation. Since CV is time consuming, only a subset of 40% randomly selected training samples is used. A grid search heuristic is performed to select the best \((\sigma, C)\) parameters from \( \sigma = 2^{-15}, 2^{-13}, ..., 2^3 \) and \( C = 2^{-5}, 2^{-3}, ..., 2^3 \). Hence, there are 100 groups of parameters. For each group, a 5-fold CV is conducted. The group with the highest CV accuracy value is used to build an SVM on the whole training dataset and to predict on the test dataset.

For MLP classifier, the number of hidden layers is empirically set to be 2 and 3 hidden neurons per hidden layer are used. 20% percent of the training data is randomly selected to be used as validation set. For RDF classifier, the optimal number of tree is estimated according to the out-of-bag (OOB) classification error metric. The OOB classification error versus the number of decision trees (up to 500) using the whole data set is shown in Fig. 2. The number of tree satisfying the lowest OOB error is determined to be used in all training phases of the partitions.

The performance evaluation of the proposed document segmentation method on three different classification algorithms depends on the following important criteria: Precision, Recall, Accuracy, \( F_1 \) score and Area under Curve (AUC). These measures are calculated as provided in [11].

![Figure 2. Out-of-bag classification error versus the number of ensemble trees. Red triangular marker represents the number of tree satisfying the lowest OOB error.](image)

The average performance values for each classification method are provided in Table 1. The best results in the table are highlighted with bold fonts. Result in Table 1 demonstrates that RDF classifier outperforms other classification methods on our own data set in terms of all performance evaluation criteria except recall value. This classifier achieves an average
classification accuracy of 87.2 % and precision value of 90.1 %, which are initially promising results for this kind of difficult problem in document flow segmentation on heterogeneous collections. The best recall value is achieved by SVM classifier with 89.1 %, which indicates that this classifier is more successful in classifying correctly the positive examples (transitions) that denote the continuity of the same document. However, in this document flow separation task, the accurate detection of the page positions where a new subset of document begins inside the document flow is more significant. Therefore, RDF classifier should be preferred to obtain an effective document flow segmentation system since this classifier achieves slightly both higher accuracy and $F_1$ score ($F_1$ score = 0.888).

Table 1. Average performance values for each classifier (AUC: Area under Curve)

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>$F_1$ score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.8753</td>
<td>0.8909</td>
<td>0.8625</td>
<td>0.8829</td>
<td>0.8570</td>
</tr>
<tr>
<td>RDF</td>
<td>0.9014</td>
<td>0.8768</td>
<td>0.8724</td>
<td>0.8888</td>
<td>0.8716</td>
</tr>
<tr>
<td>MLP</td>
<td>0.8733</td>
<td>0.8756</td>
<td>0.8542</td>
<td>0.8747</td>
<td>0.8501</td>
</tr>
</tbody>
</table>

The receiver operating characteristics (ROC) curve for each classification method averaged over 100 runs (partitions) is shown in Fig.3. The area under the ROC curve values are also provided in Table 1. The AUC is frequently used in both evaluating and estimating the predictive modelling accuracy of different classification techniques. The results in Table 1 show that RDF classifier achieves the best AUC value (0.87). This result is also consistent with the results illustrated in Fig.3.

Although initial results on document flow separation in this study is quite promising, we believe that adding additional features based on the textual content of the documents can significantly improve the performance of our proposed model. For this reason, a good future direction of this work can be incorporating textual features extracted from document pages and utilizing an ensemble classifier technique which may involve effective combination of RDF and SVM classifiers.
4. CONCLUSION

In this paper, we have described a method for automatic segmentation of document flow applied to a heterogeneous document collection from our own dataset. In our method, we incorporate BoVW model using a set of features based on structural variations in local image patches and also font based features inside the text documents. Using the features of each individual page of a document, we propose a novel approach to combine the successive pages of it into a single feature representation, also named as “transition feature vector”, and effectively formulate the document segmentation as a binary classification problem. We then evaluate and demonstrate the performance of our proposed model using three different classifiers. The results show that without incorporation of any information about textual contents of the documents, the proposed model can not only achieve quite good performance for document flow segmentation but also exhibit robustness to noise in text documents.

ACKNOWLEDGMENT

This work was partially supported by the Scientific and Technological Research Council of Turkey under Grant 3120918 and by Yapi Kredi Bank under Grant 62609.

REFERENCES