Text versus non-Text Distinction in Online Handwritten Documents

Emanuel Indermühle Horst Bunke

Institute of Computer Science and Applied Mathematics
University of Bern, CH-3012 Bern, Switzerland
{eindermu,bunke}@iam.unibe.ch

Faisal Shafait Thomas Breuel

Image Understanding and Pattern Recognition Research
German Research Center for AI (DFKI), D-67663 Kaiserslautern, Germany
{faisal,tmb}@iupr.net

ABSTRACT

The aim of this paper is to explore how well the task of text vs. non-text distinction can be solved in online handwritten documents using only offline information. Two systems are introduced. The first system generates a document segmentation first. For this purpose, four methods originally developed for machine printed documents are compared: x-y cut, morphological closing, Voronoi segmentation, and whitespace analysis. A state-of-the art classifier then distinguishes between text and non-text zones. The second system follows a bottom-up approach that classifies connected components. Experiments are performed on a new dataset of online handwritten documents containing different content types in arbitrary arrangements. The best system assigns 94.3% of the pixels to the correct class.

Categories and Subject Descriptors

I.7.5 [**Document and Text Processing**]: Document Capture—*Document analysis*

General Terms

Algorithms

Keywords

Document Zone Classification, Document Segmentation, Online Handwritten Documents

1. INTRODUCTION

The distinction of different content types like text, drawings, formulas, or tables is one of the key tasks in the analysis of documents. It allows one to assign content of one particular type to the corresponding, specialized recognition systems, which is inevitable to fully "understand" each type of information contained in a complex document [12]. In the domain of online handwritten documents this topic attracts increasing attention, as documents produced on a Tablet PC or with Anoto technology typically include diverse contents, such as text, graphics, formulas and tables. Since the most frequent content type is text, it is a good start to distinguish between text and non-text ink in a document first.

Two approaches can be applied to distinguish text from non-text in documents. Starting with an entire document, under the top-down strategy the document is segmented into meaningful document zones. Then the zones can be classified into text or non-text [9, 20]. The bottom-up strategy, on the other hand, performs classification on small, naturally given parts of a document e.g. pixels, connected components, or individual strokes in online documents [2, 8, 18]. A clustering algorithm may follow to group small entities into larger, meaningful segments.

In the literature, both approaches have been applied to machine printed documents. Top-down methods are prevailing if the document structure can be analyzed rather easily [9] (as in scientific papers or newspapers, for example). Pixel classification is preferred where the structure is difficult to recognize [2] (as in magazines where text and images may be mixed rather irregularly). In the field of online handwritten document analysis, the distinction of text and non-text is accomplished with a bottom-up approach in [8, 13] where single strokes, as the smallest entities, are classified. In [17] the top down strategy is applied.

In our work the target is to distinguish text from non-text content in online handwritten documents. However, in the current paper we report only on the use of methods that rely exclusively on offline information. We intend to expand these methods in the near future by additionally using online information.

The first system presented in this paper implements the top-down approach with document segmentation methods originally developed for machine printed documents [15]. The segmentation methods are applied to the images of the documents, i.e. to the offline version of given online documents. Then, a support vector machine (SVM) decides if a resulting document zone consist of text or non-text ink. This system is compared with a system implementing the bottom-up strategy, where connected components, as the smallest entities, are classified with the same classifier.

The rest of the paper is organized as follows. In Section 2 the dataset on which the experiments have been performed is introduced. Section 3 gives an overview of the two systems and describes the segmentation methods used in the top-down approach.

SAC'10 March 22-26, 2010, Sierre, Switzerland.

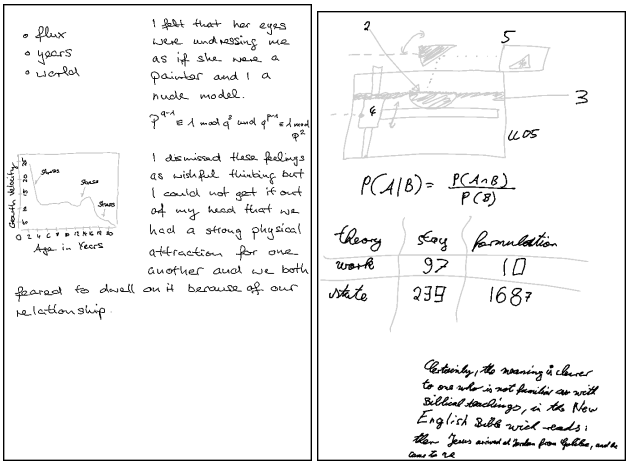

Figure 1: Two sample documents from the dataset. Text ink is black, non-text ink is gray.

Section 4 introduces the classifier and the features extracted from the document zones. In Section 5, the experimental setup and the results are shown. Finally, conclusions are drawn in Section 6.

2. DATASET

The methods proposed in this paper are trained and tested on a set of online handwritten documents which have recently been collected at the University of Bern, and in the near future they will become available to the public. The dataset consists of 1,000 documents produced by 200 writers. The documents contain text in textblocks, lists, tables, and diagrams, as well as non-text in drawings and diagrams. About 72% of all strokes belong to text. Examples of these documents can be seen in Figure 1. In the database generation process, the writers compiled each individual document by randomly copying parts from the following sources:

- 200 diagrams (circuits, URL diagrams, charts, flow charts, chemical formulas, etc.) obtained from Wikimedia Commons[1]. In contrast to drawings, diagrams contain text labels.
- 200 mathematical formulas obtained from Wikipedia[2]
- 200 drawings obtained from Wikimedia Commons[1]
- random sentences from the Brown corpus [6]
- tables containing numbers and random nouns from the Brown corpus [6]
- list of random nouns from the Brown corpus [6]

No further constraint were imposed on the writers. Therefore, some of the documents are quite challenging regarding the proper discrimination of text and non-text.

The data format used to store the documents is InkML[3]. The digital ink, which is the main part of the document, is stored in terms of groups of successive vectors consisting of X-, and Y coordinates, time, and pressure value. Additional to this information, the annotation of the ink is stored in the same files, which is structured hierarchically as illustrated in Figure 2.

[1]Wikimedia Commons: `http://commons.wikimedia.org`
[2]Wikipedia: `http://en.wikipedia.org`
[3]InkML is a format proposed by a working group of the W3C. Currently a draft is published at `http://www.w3.org/TR/InkML/`

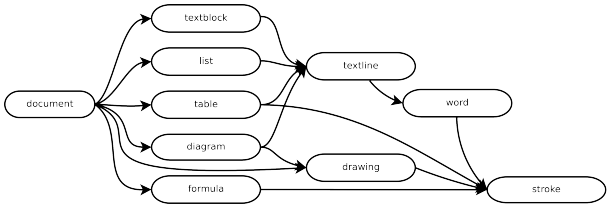

Figure 2: Hierarchical annotation of the online handwritten documents in the dataset.

The aim of this paper is to explore how well the task of text vs. non-text distinction can be solved using only offline information. Therefore the digital ink is transformed to the offline image format. Binary images are the input to the systems. For all images, the size is chosen to be 1000 pixels for the longer side. This results in a resolution of about 84 dots per inch (dpi).

3. SYSTEMS

Two systems are proposed and compared to each other in this paper. The first one, referred to as *segment classification*, follows the top-down approach, while the second one, *connected component classification*, falls into the category of the bottom-up strategies.

3.1 Segment Classification

In the first system, the input images are segmented into document zones. The zones are then classified into zones containing text and zones containing non-text by an SVM, which has been trained on the segments of the *segmentation ground truth* images.

Four different segmentation methods are compared. The selection was inspired by [15] where these methods are applied to machine printed documents. Two of the methods proposed in [15] are ignored, since they are based on constraints that are not met by the documents considered in the current paper. One of the methods was simplified for the same reason.

In this work, source code of the OCRopus [4] project was adopted for the implementation of the segmentation methods.

3.1.1 X-Y Cut

The recursive x-y cut (RXYC) algorithm by Nagy et al. [11] is a tree-based top-down algorithm. Starting with the entire document as the root, the image is split into two or more parts, which become the children. This is repeated recursively with every node until the image can not be split any further. The leaf nodes represent the final segmentation. To split an image, the horizontal and vertical projection histograms v_y and v_x are calculated. The valleys in the histograms are defined as the values smaller than the noise removal thresholds t_x^n and t_y^n, which are linearly scaled to the corresponding dimension of the image. The values in these valleys are set to zero. If a continuous valley in v_x or v_y is larger than the thresholds t_x or t_y, respectively, the image is split vertically or horizontally at the center of this valley. This procedure is stopped when no image at a leaf position in the tree can be split any further under the given thresholds.

The parameters to validate are t_x^n and t_y^n, which lead to more segments if they are larger, and t_x and t_y which lead to more segments if they are smaller.

3.1.2 Morphological Smearing

On the binary images, the morphological operation *closing* was applied. This operation consists of the morphological operation *dilation* followed by *erosion* as they are defined in [7]. Basically, the background of the new image is defined as the union of the

regions covered by a structuring element when it is translated to every position in the image and it is not touching any foreground pixel of the original image. The structuring element is a rectangle with width c_x and height c_y. Every connected component of the resulting image becomes part of the final segmentation.

The parameters to validate are the dimensions of the structuring element, which will result in more segments if they are smaller.

3.1.3 Voronoi Diagram based Algorithm

The Voronoi diagram based algorithm was introduced by Kise et al. in [10]. In its first step, the contour of the foreground is sampled with a sampling rate equal to r_s. Noise is then removed using a noise removal threshold t_n. For each sample point a Voronoi region is generated resulting in a Voronoi-diagram covering the entire document. Then, adjacent Voronoi regions v_1 and v_2 of this diagram are merged as long as one of the following criteria is satisfied:

- their separating edge crosses foreground
- $d(v_1, v_2)/T_{d1} < 1$
- $d(v_1, v_2)/T_{d2} + a_r(v_1, v_2)/t_a < 1$

where $d(v_1, v_2)$ is the minimum distance between the foreground pixels of the Voronoi regions v_1 and v_2. The inter-character gap T_{d1} is defined by the first maximum in the nearest neighbor histogram of the connected components. The inter-line gap T_{d2} is derived from the second maximum in the nearest neighbor histogram by adding a margin control factor f_m (we refer to [10] for more details). The area ratio a_r is defined as:

$$a_r(v_1, v_2) = \frac{max(a(g_1), a(g_2))}{min(a(g_1), a(g_2))} \tag{1}$$

where $g_1 \in v_1$ and $g_2 \in v_2$ are the connected components with the smallest distance and $a(g_i)$ is the area of g_i.

The free parameters that have to be validated are the sampling rate r_s, the noise removal threshold t_n, the margin control factor f_m, which controls the inter-line gap, and the area ratio threshold t_a.

3.1.4 Whitespace Analysis

This algorithm, which was introduced by Baird [1], analyzes the background of a document image. It starts by finding a set of largest white rectangles covering the entire background of the document. This is done by an algorithm proposed by Breuel [3]. Then the set of rectangles is sorted according to a weight $w_1(c)$ of the rectangle c. Originally $w_1(c)$ is defined as:

$$w_1(c) = area(c) * W\left(\left|log_2\left(\frac{height(c)}{width(c)}\right)\right|\right) \tag{2}$$

where $W(x)$ is a weighting function which has been experimentally determined in [1]. The following approximation was proposed by Shafait [15]:

$$W(x) = \begin{cases} 0.5 & \text{if } x < 3 \\ 1.5 & \text{if } 3 \leq x < 5 \\ 1 & \text{if } x \geq 5 \end{cases} \tag{3}$$

However, as the whitespace in handwritten documents is not as equally distributed as in printed documents, this weighting function may not be suited here. Therefore, three other weights to sort the rectangles are evaluated:

$$w_2(c) = area(c) \tag{4}$$

$$w_3(c) = width(c)^2 \tag{5}$$

$$w_4(c) = height(c)^2 \tag{6}$$

In descending order, using any of the weighting functions, the white rectangles are sequentially plotted onto the output image, which is initialized with black pixels. This process is terminated as soon as the following stopping rule is satisfied:

$$w_i(c_j) - f_w \frac{j}{m} \leq t_s \tag{7}$$

where c_j is the last rectangle added to the mask, f_w is a factor to weight the influence of the number of segments added, m is the total amount of rectangles, and t_s is a stopping threshold. Finally the segmentation is defined by the black regions left in the output image. The free parameters to validate are f_w, t_s and the weights $w_1, \ldots, w_4$.

3.2 Connected Component Classification

The *connected component classification* systems is characterized by performing a connected component analysis in order to segment a given input document. The connected component analysis is performed in a 8-point neighborhood and results in regions with connected black pixels. For classification, an SVM is used again. It is trained on the labelled connected components of the training set. Note that the connected component analysis has no free parameters.

4. CLASSIFIER

4.1 Support Vector Machine

One of the most popular classification methods is the support vector machine (SVM)[14, 16, 19]. The key idea is to find a hyperplane that separates the data into two classes with a maximal margin. Such a hyperplane can only be found if the data is linearly separable. If linear separability is not fulfilled, a weaker definition of the margin, the soft margin, can be used. In this case, the optimal hyperplane is the one that minimizes the sum of errors and maximizes the margin at the same time. This optimization problem is usually solved by quadratic programming. In order to improve the classification of non-linearly separable data, an explicit mapping to a higher-dimension feature space can be performed, or instead, a kernel function can be applied.

In our experimental evaluation we make use of an SVM with RBF-kernel $\kappa(x, x') = \exp\left(-\gamma||x - x'||^2\right)$, where $\gamma > 0$. Hence, besides the weighting parameter C, which controls whether the maximization of the margin or the minimization of the error is more important, the metaparameter γ has to be tuned. The experiments were performed using the libsvm [5] implementation.

4.2 Feature Extraction

Different feature sets to classify document zones in printed documents have been investigated by Keysers et al. in [9]. A well performing set of run-length histograms and connected component statistics have been proposed, which can be extracted very fast. The features used in the current paper for the classification of the document zone in the first system and for the connected components in the second system are, in fact, similar to those of [9].

The feature set consists of run-length histograms of black and white pixels along the horizontal, vertical, and the two diagonal directions. Each histogram uses eight bins, counting runs of length $\leq 1, 3, 7, 15, 31, 63, 127$, and ≥ 128. Additionally, histograms with the same bins as mentioned before are created from the width and height distributions of the connected components within the document zone. Moreover, a two dimensional 64-bin histogram of the joint distribution of widths and heights, as well as a histogram of the nearest neighbor distances of the connected components are calculated. All in all, 152 numerical features are extracted.

Segmenter	Parameter	Range	Best
X-Y Cut	t_x^n	$10 * \{0, \ldots, 6\}$	10,10,10,0
	t_x^n	$10 * \{0, \ldots, 3\}$	10
	t_x^c	$2 * \{0, \ldots, 15\}$	2
	t_x^c	$2 * \{0, \ldots, 15\}$	2
Smearing	c_x	$5 * \{0, \ldots, 20\}$	10,0,0,5
	c_y	$5 * \{0, \ldots, 20\}$	10,5,5,5
Voronoi	r_s	$1 + 5 * \{0, \ldots, 8\}$	1
	t_n	$5 * \{0, \ldots, 7\}$	5,5,35,10
	f_m	0.2	0.2
	t_a	$1 + 25 * \{0, \ldots, 10\}$	1
Whitespace	t_s	$1 + 125 * \{0, \ldots, 40\}$	1
	f_w	$1 + 5 * \{0, \ldots, 10\}$	1
	w	*Baird*, area, w, h	width

Table 1: Parameter validation of the four segmentation methods. If four values are present in the *Best* column then they vary for the four cross validation runs.

The part of the feature set created from connected component statistics is meaningless when extracted from a single connected component, as it would be the case in the second system. Therefore these features are omitted in that system.

5. EXPERIMENTS

5.1 Setup

For all experiments, four-fold cross validation is performed in order to reduce the risk of a biased dataset division. Four disjoint, equally sized subsets $ss_0, \ldots, ss_3$ of the dataset are created. In run i ($i = 0, \ldots, 3$), the training set is defined as $ss_{(0+i)} \cup ss_{(1+i \mod 4)}$, the validation set as $ss_{(2+i \mod 4)}$, and the test set as $ss_{(3+i \mod 4)}$.

For the first system, the SVM is trained on the *segmentation ground truth* of the training set. In a grid search on the validation set, the best parameter combination of C and γ in the range of $\{2^{2*j} \mid j = -12, \ldots, 12\}$ is identified. The SVM with this parameter set is then applied on the test set. The classification rates of the four segmentation methods are maximized over all parameter combinations on the validation set as well. The parameter combinations that lead to the best performance on the validation set (and are applied on the independent test set) are shown in Table 1.

In the second system the SVM-classifier is trained on the connected components of the training set. The parameters C and γ are validated in the same range as for the first system. Using the optimal parameter values, the connected components of the test set are classified to get the final result.

In both systems the recognition rate is calculated as the fraction of pixels within a document that have been correctly assigned to the text or the non-text class. The mean value is computed over all documents evaluated.

5.2 Results

Figure 3 shows the recognition rates that were achieved when classifying the document zones returned by the four segmentation methods. With a recognition rate above 0.92, the *Voronoi segmenation* and *morphological smearing* methods are significantly better than the *X-Y cut* and the *whitespace analysis*. (This is statistically backed using a dependent t-test for paired samples with a significance level of $\alpha = 0.05$.) The reason for the difference might be that the latter two methods can correctly segment documents in Manhattan layout only, which is not given for the dataset used here.

The parameter sets of the different segmentation methods that lead to the best recognition rate also lead to an over-segmentation.

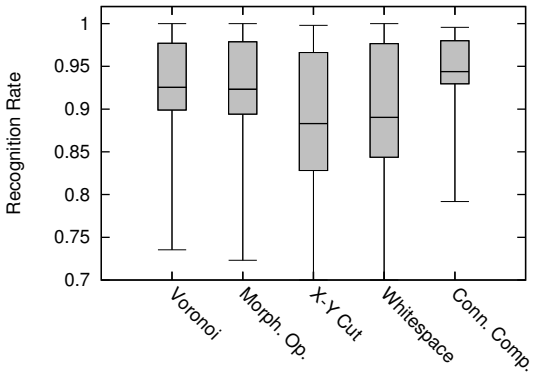

Figure 3: Recognition rates achieved with the five different methods.

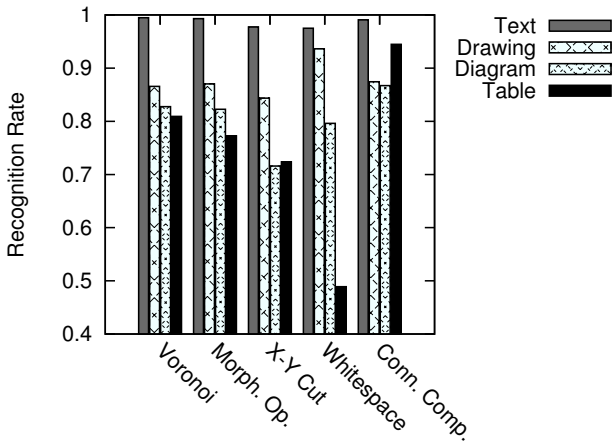

Figure 4: Recognition rates of the distinction between text and non-text considering only pixels of text blocks, drawing, diagram, or tables.

It seams easier to correctly classify small segments than to generate a segmentation with large document zones containing only one content type. This finding confirms the results of the *connected component classification* system, where small segments are produced by default. With a recognition rate of 0.9437 it performs significantly better than all other methods.

Analyzing the results on the different content types (see Figure 4), the advantage of the *connected component classification* system can be explained. On tables and diagrams, its recognition rate is higher than that of other methods. It seems that the lines occurring in the tables lead to a wrong classification. The *connected component classification* can classify each text part in a table individually, while the segmentation methods can not split the tables apart. The problem with diagrams might be the text labels. If they belong to the same segment as the rest of the diagram, they will be misclassified.

6. CONCLUSION

The distinction of text and non-text is an important step to tran-

scribe a scanned or online recorded document into an electronic office document (e.g. ODF). In this paper, we compare two systems to solve this problem using only methods originally developed for offline handwritten and machine printed documents. The first system is implementing the top-down approach. It segments the documents into zones which are then classified by an SVM. Four different segmentation methods are compared to each other. In the second system, a bottom-up strategy is implemented where connected components are classified by an SVM. The dataset on which the experiments have been performed is a collection of online handwritten documents containing text, drawings, diagrams, tables, lists, and formulas.

In the result section we demonstrate that the distinction of text and non-text content in handwritten document is possible. Both approaches reach good recognition rates. However, with 94.3% of all pixels correctly assigned, the bottom-up strategy outperforms the top-down approach. The use of connected components as document zones reduces the risk to mix text and non-text content, which inevitably results in misclassified pixels.

In the near future we intend to extend the proposed methods by using online information. In addition, other bottom-up approaches will be applied which use strokes or individual pixels as their smallest entities. The dataset introduced in this paper opens possibilities for further investigations in the field of document analysis. The detection of different content types or the reconstruction of tables and lists might be other interesting subjects for future work.

7. ACKNOWLEDGMENTS

This work has been supported by the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). We thank the developers of OCRopus[4] for making available their software to the public.

8. REFERENCES

[1] H. S. Baird. Background structure in document images. In H. Bunke, P. Wang, and H. S. Baird, editors, *Document Image Analysis*, pages 17–34. World Scientific, Singapore, 1994.

[2] H. S. Baird, M. A. Moll, and C. An. Document image content inventories. In *Proc. 14th Conf. on Document Recognition and Retrieval*, page 296, 2007.

[3] T. M. Breuel. Two geometric algorithms for layout analysis. In *Proceedings of the Workshop on Document Analysis Systems*, pages 188–199, Princeton, NJ, USA, 2002.

[4] T. M. Breuel. The OCRopus open source OCR system. In *Proceedings IS&T/SPIE 20th Annual Symposium 2008*, 2008.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] W. Francis and H. Kucera. *Manual of information to accompany a standard sample of present-day edited American English for use with digital computers*. Department of Linguistics, Brown University, 1979.

[7] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):532–550, 1987.

[8] A. K. Jain, A. M. Namboodiri, and J. Subrahmonia. Structure in on-line documents. In *In Proc. 6th Int. Conf. on Document Analysis and Recognition*, pages 844–848, 2001.

[4]OCRopus: `http://code.google.com/p/ocropus/`

[9] D. Keysers, F. Shafait, and T. M. Breuel. Document image zone classification – a simple high-performance approach. In *Proc. 2nd Int. Conf. on Computer Vision Theory and Applications*, pages 44—51, 2007.

[10] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, June 1998.

[11] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, 1992.

[12] O. Okun, D. Doermann, and M. Pietikäinen. Page segmentation and zone classification: the state of the art. Technical report, University of Maryland, Language and Media Processing Laboratory, 1999.

[13] S. Rossignol, D. Willems, A. Neumann, and L. Vuurpijl. Mode detection and incremental recognition. In *In Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 597–602, 2004.

[14] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[15] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):941–954, Jun 2008.

[16] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[17] M. Shilman, P. Liang, and P. Viola. Learning non-generative grammatical models for document analysis. In *Porc. of 10th Int. Conf. on Computer Vision*, volume 2, pages 962–969, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[18] C. Strouthopoulos and N. Papamarkos. Text identification for document image analysis using a neural network. *Image and Vision Computing*, 16:879–896, 1998.

[19] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.

[20] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, 1982.