Tex-Lex: Automated Generation of Texture Lexicons using Images from the World Wide Web

Demetrios Gerogiannis Christophoros Nikou
Department of Computer Science
University of Ioannina
45110 Ioannina, Greece
{dgerogia,cnikou}@cs.uoi.gr

***Abstract*—A method for automatic creation of a semantic texture database is introduced, which exploits the cumulative knowledge that exists in the image tags on the World Wide Web. In the first step of the method, a number of images are retrieved from the Web using the text search option provided by search engines by querying simple notions (e.g. sky, grass water, etc.). These images are segmented into a number of predefined regions using standard clustering and each region is described by a set of image features. The descriptors of the extracted regions of the whole set of images are compared based on the Bhattacharyya distance and the ones that are more similar are considered to be entries of a dictionary associated with the initial keyword used for the query. Moreover, the corresponding regions are parts of the visual lexicon describing the keyword. Also, an already existing lexicon may be iteratively updated by new features that may not match the existing dictionary entries but they are represented over a significant number of query results. Early results on common keywords representing landscapes indicate that the method is promising and may be extended to describe composite structures and objects.**

***Index Terms*—Computer vision, automatic image annotation, texture description, dictionary learning, image retrieval.**

I. Introduction

During the last decade, the World Wide Web has revolutionized many aspects of the research in computer vision and image analysis including image retrieval, object detection and image annotation. More recently, a plethora of images stored in many photo-sharing web sites are continuously being attributed several tags by many individuals. This semantic knowledge could be exploited within an automatic scheme to provide semantic ontologies for various contextual entities. At a second step, this information could be embedded in an automatic image annotation scheme. Image annotation may be regarded as a preprocessing or assisting module in image retrieval tasks, whose goal is to detect images that match a specific example whether this is an image or a keyword. In the earlier stages of image retrieval, the input of the process was plain text, while since the last few years the web search engines provide the ability to use an image as a query entry. For a comprehensive review on automatic image annotation we address the reader to [1].

Several techniques have been proposed to address the problem of image annotation which may be grouped into three main categories. The oldest family of methods solves the annotation problem by letting users to manually associate images with textual information (tags). However, such an approach is not efficient for a large amount of images, which is the current case in the Web. A second approach is content based image retrieval [2], [3], where low level context features, like shape, color and texture, are matched with semantic concepts used by humans to interpret an image. In that case, the query item is an image. Finally, a third category consists of the automatic image annotation methods, where the goal is to simultaneously exploit the advantages of the first two categories using both semantic learning and annotation [4]. In brief, the overall scheme of this category's methods involves the extraction of features and the exploitation of machine learning algorithms, for example support vector machines (SVM) [5], [6], [7], [8], or artificial neural networks [9], [10], [11] to learn the parameters of the feature space. In all cases, it is assumed that a set of already labeled images is available. For example, in the seminal method presented in [4], the objective of generating a number of images for a given object class is accomplished through web search and removal of irrelevant (lowly ranked) images using metadata features and text. Then, the remaining images are used to train a classifier to improve the ranking.

In the proposed work, we try to establish a primal framework that automatically associates visual information with a text query. The novel characteristic of our method is that it exploits the hidden knowledge in the manually annotated images (image tags) of the World Wide Web to visually model a notion described by a single word or a phrase. More specifically, at first, the application programming interface (API) of a web image search engine is employed in order to get a collection of images based on some text query. In the following step, an image segmentation method is used to segment the collected images and then textural features are extracted from the resulting image segments. These features are used to produce a similarity matrix between all the regions of the segmented images. Then, a spectral clustering algorithm [12] is performed in order to group the obtained features. Finally, for each cluster, a measurable quantity is computed, which is called the support, that describes the relation of the cluster with the text query. The support of a cluster is analogous to the number of members in the cluster. Moreover, based on the support, clusters with small number of members are eliminated. The result of this pruning process is a visual model associated with the given text query provided at the beginning

978-1-4673-5807-1/13/$31.00 ©2013 IEEE

of the process. The proposed method differs significantly with respect to [4] which also uses metadata from the users. In [4], the main output is *a set of images* satisfying a query. The output of the work presented herein is *a set of features* representing a semantic term associated with a visual category. To the best of our knowledge, it is the first time such an approach is presented.

The remaining of the paper is organized as follows. Section II introduces the proposed algorithm, while in section III we present experiments conducted to investigate the behavior of our method to simple standard queries. Finally, in IV section we conclude our work and give some remarks concerning our future research directions.

II. FROM KEYWORDS TO TEXTURE DESCRIPTION

In order to automatically extract features to model a human concept through a query we benefit from the rich meta-information associated with images in the Web, that is, we exploit the ability of modern search engines to create collections of images related to a query string. Then, features are extracted from the returned images and the most common of them become members of a dictionary. This implies a clustering step which could be accomplished either by the standard but generally powerful k-means algorithm or by more sophisticated algorithms like spectral clustering [12] which is the choice in this work. To decrease the computational complexity, each initial image in the collection is summarized by its representative features. Thus, each image is segmented either by simple griding of the image plane or by using a superpixel pre-segmentation [13] which is much more accurate but more time consuming. This decision depends on the application at hand. Then, in each region, the corresponding features are extracted and finally, a clustering process groups all the similar features in the image, which is summarized by the features associated with the cluster centers. The medoid vector is generally a good choice for representing the features of the cluster centers.

The final step consists in clustering the representatives of the whole image collection which provides the set of features $\mathcal{L}_i = \{\mathcal{L}_i^j\}$ representing the i^{th} query, where the index j indicates the j^{th} feature vector representing the i^{th} query. We call the set $\mathcal{L} = \{\mathcal{L}_i\}$ the lexicon or dictionary and we refer to its members $\{\mathcal{L}_i^j\}$ as the codewords.

Note that the resulted cluster representatives are not necessarily the ones that would best describe the query string in a given image, as irrelevant images may exist in the initial image collection. For example, one could use the query *water* and apart from images depicting regions with water, an image with the water molecule could be returned by the search engine. Thus, there is a need to eliminate irrelevant data from the results. This may be achieved by measuring a quantity that could define the saliency of a feature. Having in mind that these irrelevant features may have a small repeatability, the saliency of a lexicon codeword $\mathcal{L}_i^j$ is defined by:

$$\mathrm{sal}(\mathcal{L}_i^j) = \frac{\text{\# features represented by the } j^{th} \text{ lexicon entry}}{\text{\# features of the } i^{th} \text{ category in the lexicon}}.$$

Clusters with fewer members have smaller saliency and consequently they may be pruned from the final model. This can be determined by examining the distance between the saliency value of a codeword and the mean saliency and comparing it with a predefined threshold $T > 0$. A codeword is eliminated from the lexicon if:

$$|\mathrm{sal}(\mathcal{L}_i^j) - \mathrm{mean}\{\mathrm{sal}(\mathcal{L}_i^j)\}| > T. \quad (1)$$

The overall lexicon creation process is presented in Algorithm 1.

Algorithm 1 Automatic Visual Feature Extraction

```
• input: A text query (keyword) and a threshold $T$.
• output: A collection of visual features (lexicon $\mathcal{L}$).
    – Perform an image search in the Web using the query
      and get a collection on $M$ images.
    – For each image $I_j | j = 1, \ldots, M$,
        ∗ Segment the image into regions $\{R_i^j\}_{i=1}^K$ and represent
          each region by its representative vector.
        ∗ Cluster the region representatives.
        ∗ Summarize image $I_j$ by the centers of the clusters.
    – Cluster the representatives of the whole image collection
      using spectral clustering. Let the set of the cluster
      centers be $\mathcal{L}_i | i = 1, \ldots, N$.
    – Prune codeword $\mathcal{L}_i^j$ of the $i^{th}$ keyword if eq. (1) is
      not satisfied.
```

In order to annotate an image, we need to extract the visual information in a similar manner to the one we used in the lexicon creation. This process presegments the new image to be annotated into a large number of regions and the region representatives are extracted. Each region representative is compared with each entry of the lexicon and the category corresponding to the entry with the higher similarity is selected as the annotation category of the region. For coherence, the same similarity measure with the one used in Algorithm 1 should be employed. In principle, this may be the Euclidean distance but other distances, such as the Mahalanobis distance may also be considered. The overall annotation procedure is presented in Algorithm 2.

Algorithm 2 Image annotation based on lexicon entries

```
• input: An image $I$, a lexicon $\mathcal{L}$.
• output: Annotation of image $I$.
    – Summarize image $I$ by $\{R_i\}_{i=1}^K$ regions through a
      rough oversegmentation.
    – For each region $R_i$
        ∗ Extract a representative visual feature $\mathcal{F}_i$.
        ∗ For each entry $\mathcal{L}_k^j$ in $\mathcal{L}$
            · $D_i^k(j) = \mathrm{distance}(\mathcal{F}_i, \mathcal{L}_k^j)$
        ∗ $k^* = \arg\min\{D_i^j(k)\}$
        ∗ Categorize $R_i$ as $\mathcal{L}_{k^*}$
```

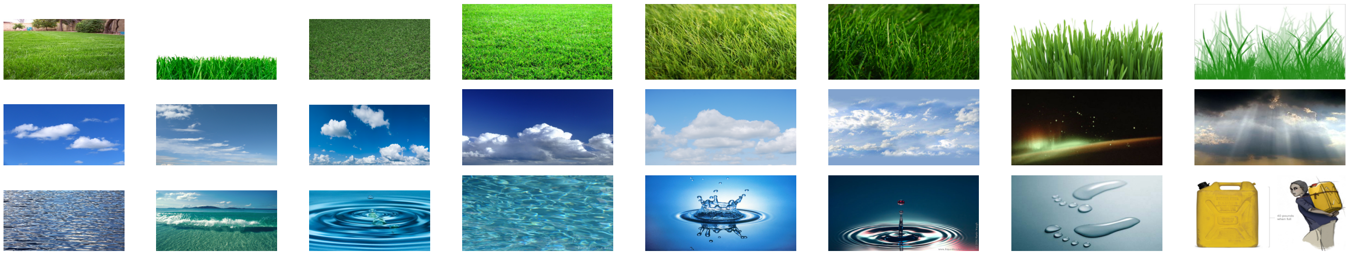


Fig. 1: Some representative images returned by the Google Image Search Engine [14] used to create the lexicon for the categories *grass* (top row), *sky* (middle row) and *water* (bottom row).

III. EXPERIMENTAL RESULTS

In order to apply Algorithm 1, image search based on text queries was performed using the widely employed Google API [14]. The corresponding API call takes as input an alphanumeric value (query string) and returns a collection of images pertained to the query string. In this early study, for our image query string, we used common words representing wide sceneries like *grass*, *sky* and *water* and 40 images per keyword were retained. Representative images returned by the Google image search engine are shown in figure 1. These are 8 images out of the 40 first results used in the experiments. Notice that although all the images contain the queried term, *uncommon* results may appear as it may be observed in the last two images of the category *water*. Also, results containing both entities (*grass* and *sky* in the top row) as well as results containing other related notions (*storm* and *sky* in the middle row) introduce an additional difficulty to the system.

Image regions were obtained by creating a grid over the image frame. The grid dimensions were proportional to the image size in order to tackle the different sizes of the images in the collection. For each grid cell, we computed the corresponding Lab color space histogram and we represented a grid cell by its mean vector and a diagonal covariance matrix. We preferred the Lab space because it is designed to approximate human visual perception. More sophisticated representations could be employed both for the feature space (such as the integration of SIFT features [15]) and its modeling (e.g. Gaussian mixture models [16]). It is true that the richer the visual information represented by the features is, the more accurate the description of the initial concept becomes. Thus, one could also employee textural information at a superpixel level [13] or exploit the output of Gabor filters. However, we opted for simpler representations in this primal study in order to confirm the truth of concept. Finally, spectral clustering [12] was employed to group the features and the Bhattacharyya distance [17] was used to compute the similarity between a a pair of feature descriptors. Corresponding lexicon entries for these notions which were extracted from the web search using Algorithm 1 are depicted in figure 2. These are representative grid cells for each category.

In Algorithm 1, a crucial step is the determination of the number of clusters (lexicon entries) for each semantic category. The eigenvalues of the normalized Laplacian matrix are sorted and the index of the eigenvalue that differs significantly from its successor determines the number of clusters for each notion (eigenvalue gap). The position of the eigenvalue gaps for the three notions employed in this study are shown in figure 3 where it can be seen that there are clear abrupt changes in the diagram of the sorted eigenvalues as the first eigenvalues are very small for all of the three categories.

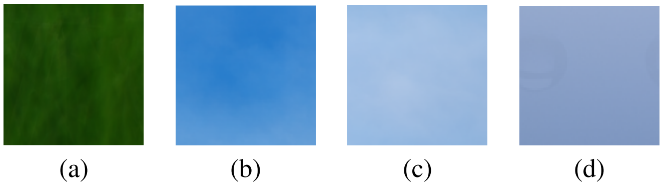
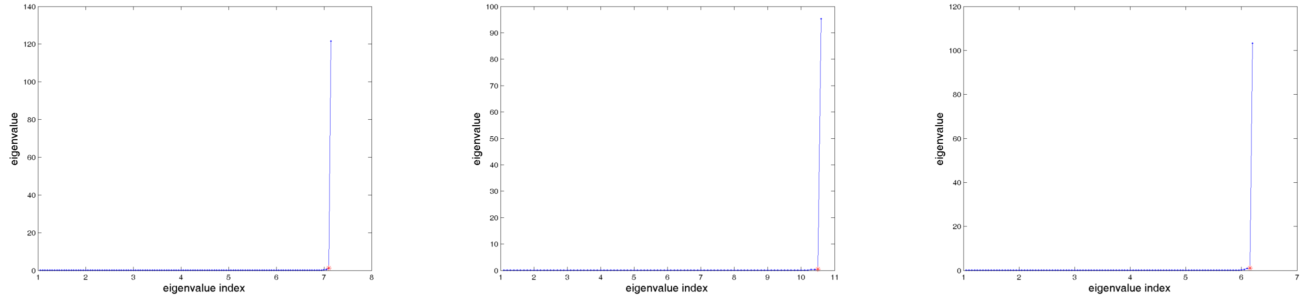


Fig. 2: Lexicon entries provided by Algorithm 1 for the categories (a) *grass*, (b) - (c) *sky* and (d) *water*.

To evaluate the efficiency of the proposed method, we used the Microsoft Object Class Recognition (v2) database (MSRC) [18] to perform automatic image annotation. The database consists of 591 images depicting items of 23 object classes in total. The database contains also the pixel-wise labeled ground truth images. Most of the images of the database have a resolution of 320×213 pixels. Figure 4 shows some representative images of the database. Please note that the purpose of our evaluation experiments is to verify the efficiency of the automatic lexicon creation method, and not the annotation process itself. Thus, in order to eliminate any erroneous result that may occurred due to the segmentation of the image, we opted for using the ground truth as the input regions of algorithm. Moreover, we utilized only the regions that depict either *grass* or *sky* or *water*. Thus, any annotation error is due to the fact that the proposing method failed to model accurately the corresponding notion.

The MSRC database has a strong object recognition orientation as there are many categories like *sheep* or *building* which may be conceptually divided into simpler entities. For example, a *sheep* consists of *wool*. Therefore, we decided to use only three out of the 23 classes for our evaluation experiments. These categories are the notions of *grass*, *sky* and *water*. Hence, we employed 328 out of 591 images of the database, which contain regions annotated as *grass*, *sky* or em water.

Each image of the MSRC database was segmented and



grass *sky* *water*

Fig. 3: The eigenvalue gap of the normalized Laplacian matrix determines the number of clusters for each category.

Fig. 4: Some representative images of the MSRC image database [18] used in our experiments.

automatically annotated, according to Algorithm 2 based on the dictionaries provided by Algorithm 1. Since the ground truth category for each pixel is known, we were able to decide wether the annotation provided by our method compared to the real one is true by calculating the percentage of misclassified pixels:

$$\text{Annotation Error} = \frac{\#\text{falsely identified pixels}}{\#\text{pixels}}. \tag{2}$$

Table I demonstrates the classification results of our experiments, where it can be observed that the proposed scheme presents very accurate results for the case of *grass-sky* categorization. Nevertheless, the very high accuracy is partly due to the fact that the notions *grass* and *sky* are relatively distinct, meaning that using descriptors that are not sophisticated efficient models may be designed. However, if we conduct the same experiment using also the notion of *water*, the annotation error increases. This is due to the fact that the categories *water* and *sky* bear a visual resemblance, and thus, more complex descriptors are needed to describe them such as textons capturing the ripples of the water. Moreover, apart from information on texture, position information could also be employed. For example, in most of the cases, the pixels representing *sky* are expected to appear at the top parts of the images, while pixels representing *water* would more likely be at the bottom parts of an image. One may observe the visual similarity between grid cells of these categories in figure 2(b)-(d). Figure 5 illustrates another example of confusion due to reflection.

TABLE I: Annotation error percentages over the MSRC (v2) database [18] for the categories *grass*, *sky* and *water*.

	Annotation Error	
Categories	Proposed scheme	Proposed scheme (no pruning)
grass - sky	0.86%	9.67%
grass - water	0.94%	14.93%
water - sky	26.18%	53.71%
grass - sky - water	15.06%	32.89%

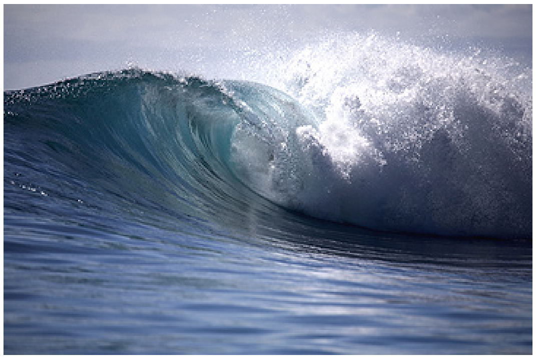


Fig. 5: A representative image depicting sea water with the sky reflecting into it. Plain color descriptors cannot discriminate between water and sky pixels.

Furthermore, we need to mention the importance of the pruning process, during the creation of the lexicon. To that end, we conducted the same experiments without pruning irrelevant features. The corresponding experimental results are also presented in Table I (second column). One may observe that the accuracy falls significantly. This is due to the fact that some images, collected during the query for the word *grass*, contain regions depicting a sky, as it can be seen in figure 6.

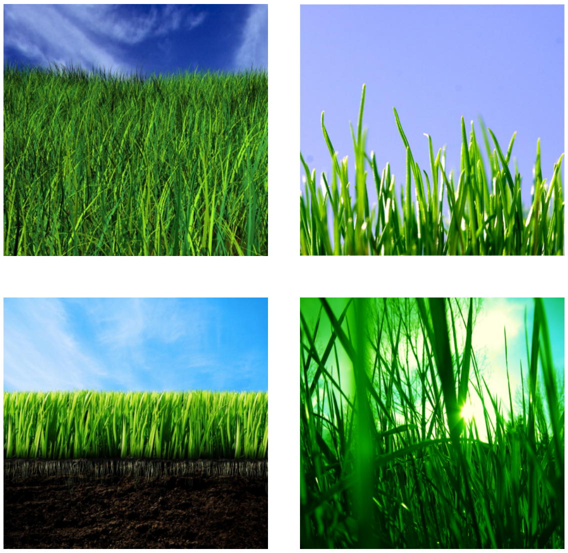


Fig. 6: Images resulted from a ”grass” query, which contain regions depicting sky.

Let us note that the results presented in Table I are obtained for a value of the pruning threshold $T = 0.19$ in Algorithm 1, which provides the maximum value of the F-measure (0.916) as it shown in figure 7. In the same figure, we may observe the consistency of the method as the threshold T varies between 0.1 and 0.4. The vertical errorbars indicate the standard deviation of the F-measure for 20 realizations of the learning phase. This dispersion is due to different initializations of the spectral clustering algorithm by the k-means algorithm.

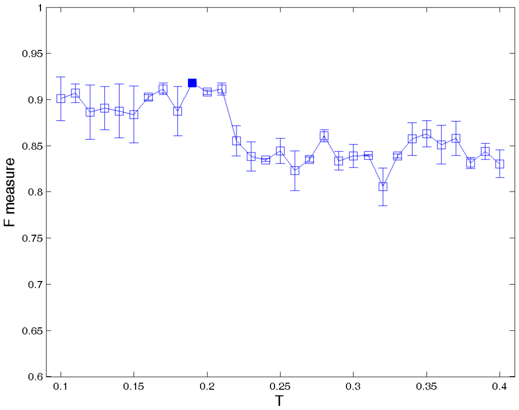


Fig. 7: The F-measure as a function of the pruning threshold.

Let us finally mention that the average execution time for annotating each image on an Intel Dual Core at 2.50GHz with 2GB RAM was 0.85 sec.

IV. Conclusion

A methodology for the automated creation of a texture lexicon was presented. The main motivation was to exploit the information in the World Wide Web and extract visual features that could be associated with words in an automatic manner. The resulted lexicon, may assist an automatic image annotation algorithm and therefore an image retrieval task. In this paper, a preliminary version of the method was introduced. The method may employ more sophisticated texture descriptors and image models to improve its efficiency. On the other hand, simpler classification algorithms than the spectral clustering applied here could be sufficient for certain applications. These are issues of ongoing research.

References

[1] D. Zhang, M. M. Islam, and G. Lu, “A review on automatic image annotation techniques,” *Pattern Recognition*, vol. 45, pp. 346–362, 2012.

[2] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: image segmentation using expectation-maximization and its application to image querying,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, 2002.

[4] F. Schroff, A. Criminisi, and A. Zisserman, “Harvesting image databases from the web,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 754–766, 2011.

[5] R. Shi, H. Feng, T. S. Chua, and C. H. Lee, “An adaptive image content representation and segmentation approach to automatic image annotation,” in *International Conference on Image and Video Retrieval (ICIVR)*, Dublin, Ireland, 2004, pp. 545–554.

[6] C. Cusano, G. Ciocca, and R. Schettini, “Image annotation using SVM,” in *Proceedings of SPIE Internet Imaging V*, vol. 5304, 2004, pp. 330–338.

[7] K. S. Goh, E. Y. Chang, and B. Li, “Using one-class and two-class SVMs for multiclass image annotation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 10, pp. 1333–1346, 2005.

[8] X. Qi and Y. Han, “Incorporating multiple SVMs for automatic image annotation,” *Pattern Recognition*, vol. 40, no. 2, pp. 728–741, 2007.

[9] S. B. Park, J. W. Lee, and S. K. Kim, “Content-based image classification using a neural network,” *Pattern Recognition Letters*, vol. 25, no. 3, pp. 287–300, 2004.

[10] S. Kim, S. Park, and M. Kim, “Image classification into object/non-object classes,” in *International Conference on Image and Video Retrieval (ICIVR)*, Dublin, Ireland, 2004, pp. 393–400.

[11] F. D. Frate, F. Pacifici, G. Schiavon, and C. Solimini, “Use of neural networks for automatic classification from high-resolution images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 800–809, 2007.

[12] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” in *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2001, pp. 849–856.

[13] X. Ren and J. Malik, “Learning a classification model for segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2003, pp. 10–17.

[14] Google, https://developers.google.com/image-search/, June 2012.

[15] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] G. Sfikas, C. Nikou, N. Galatsanos, and C. Heinrich, “Majorization-minimization mixture model determination in image segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, USA, 2011, pp. 2169–2176.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] Microsoft Research Cambridge, MSRC Object Category Image Database (v2), February 2013.