Efficient indexing for Query By String text retrieval

Suman K. Ghosh* Lluís, Gómez, Dimosthenis Karatzas and Ernest Valveny
Computer Vision Center, Dept. Ciències de la Computació
Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain
Email: sghosh,ernest@cvc.uab.es

***Abstract*—This paper deals with Query By String word spotting in scene images. A hierarchical text segmentation algorithm based on text specific selective search is used to find text regions. These regions are indexed per character n-grams present in the text region. An attribute representation based on Pyramidal Histogram of Characters (PHOC) is used to compare text regions with the query text. For generation of the index a similar attribute space based Pyramidal Histogram of character n-grams is used. These attribute models are learned using linear SVMs over the Fisher Vector [1] representation of the images along with the PHOC labels of the corresponding strings.**

I. Introduction

The availability of handheld image capturing devices and the inclination to upload them in social media like Facebook, Flickr, Instagram etc has generated a great pool of on-line images with easy accessibility. In this context there has been a huge interest for searching relevant contents in these collections of images.

Methods which take image as input query(Query-By-Example) can exploit various state of the art image comparison mechanisms, but they restrict the usability, as the user needs to give an instance of the query as an input. On the other hand, approaches which take text as query to search for relevant information in images (Query-By-String) are more user friendly as they allow the user to type the keyword to search in a much more natural way. Practical applications of Query-By-String (QBS) methods, for example Google image search, relies on cues from meta tags or text available in the context of the image. The success of this approach is rather limited by the quality of the meta tags and the contextual text. On the other hand applications like Video Google [2] enables image search using image as a query, by finding visually similar regions in the database. Although this method exploits the visual content, this does not utilise the text present in the images, which carries important semantic information about the image. In this context Mishra *et al.* in [3], proposed text-to-image retrieval, where for an input query text the goal is to retrieve all possible images carrying the similar text.Another way to retrieve relevant content from image collections is to perform Query By String word spotting, where the aim is to extract all occurrences of the query word in large collections of images.

In this work we tackle both Query-By-String word spotting and text-to-image retrieval in a single framework. We use a task specific selective search strategy, where initial regions in the image are grouped by agglomerative clustering in a hierarchy where each node defines a possible word hypothesis. We generate an inverted index file to store a ranked list of these word hypothesis for a selected list of n-grams. At query time, given a query string, the indexing structure is accessed to retrieve all candidate hypothesis containing any of the n-grams in the string. These hypothesis are then ranked according to the text query.

II. Background

One approach for addressing the text-to-image retrieval problem is based on text localization, followed by text recognition. Once the text is recognized, the retrieval task becomes equivalent to that of text retrieval.

Existing methods treat the sub-problems of this pipeline namely detection [4], segmentation [5], and recognition [6] as separate challenges. More recently methods have been proposed to combine these sub problems into one end-to-end framework for scene text understanding. Initial works in end-to-end recognition *E.g.* [7], [8] are based on sliding window approaches to detect potential text locations followed by character or word recognizer based on features extracted from potential text locations. Traditional sliding window based approaches apart from their computational cost, do not fit well in the task of holistic word detection because the target object has no fixed aspect ratio.

As an alternative to the sliding window approach, other approaches *E.g.* [9], propose to extract candidate regions for text and then group them to construct word hypotheses.

Recently, the success of Convolutional Neural Networks in other computer vision tasks has also motivated exploring their application to character and word recognition. They commonly use a text/Background classifier followed by a character/bigram classifier, the outputs are then combined to give the recognized word. Example of such works are [10], [11]

Recent work in computer vision has approached the problem of object detection from a much more appealing perspective, in the form of selective search. A selective search strategy fits naturally in the text detection task because text can always be modelled as a perceptually significant group of similar atomic objects (characters or strokes) [5],[12]. Over and above in the specific problem of scene text detection, the use of bottom-up region agglomeration to generate candidate class-independent object locations has become a popular trend in computer vision in recent times [13],[14],[15], [16]. The main benefits are the reduction of the search space by providing a small set of high quality locations, thus allowing the use of more expensive and powerful recognition techniques, and the ability to naturally localise objects without a fixed aspect ratio.

*This work is partially supported by spanish project TIN2013-41751-P and a pre-doctoral research grant from UAB

In this paper we adopt a similar strategy to the selective search of Uijlings et al. [13] but we adapt it to the specificities of text regions: We use MSER instead of superpixels as the initial set of regions to be grouped. The MSER algorithm is fast to compute and the obtained regions are extensively used in recent state of the art methods for text detection, as text characters are by design regions with an extremal property of the intensity function over its outer boundary. We use a standard Single Linkage Clustering algorithm instead of the greedy agglomerative clustering proposed in [13], this is because we do not require regions to be adjacent in order to be merged, but instead we rely only in a distance metric combining similarity and proximity between regions to be grouped. Moreover, the design of our similarity measures between regions obey to the specific task of text detection. While we share with [13] the use of color and size similarity, we introduce two new similarity measures: stroke width similarity, and gradient magnitude similarity along the region border. Finally, we propose the use of a weighted distance metric to measure similarity between regions.

As far as text recognition is concerned, existing methods are based on detecting character candidate regions in the image followed by recognizing characters and grouping them using spatial and linguistic constraints. This leads to models that are well suited for recognition, but not so much for other tasks such as retrieval. In our work we build upon the work of Almazán [17], that explores a different strategy based on holistic word representation which captures semantic representation of the word images based on a Pyramidal Histogram of Characters (PHOC). This enables to address both problems recognition and retrieval, in a natural way.

Concerning scene image retrieval based on textual queries, Mishra *et al.* [3] propose a method to perform image retrieval using textual cues. They use a fixed vocabulary to create an inverted index. Their inverted index file contains scores for presence of a word in the concerned image. In this work we use a similar strategy but rather then using a fixed vocabulary index, we use a predefined set of character n-grams as index, thus our search space is not limited to any vocabulary.

In summary the contributions of this paper can be stated as : (1) we use a task specific selective search framework based on a hierarchical grouping of text components which inherently captures spatial constrains. (2) For indexing and final retrieval, we use a compact learned PHOC attribute based representation which encodes the probability of appearance of every character/ngram at different positions of the text region, this does not require a large set of annotated images for training as recent methods based on CNNs do; (3) Instead of using a fixed vocabulary, we index the images and candidate boxes by n-grams making search space independent of vocabulary; (4) We show results of query by string word spotting for full segmentation free scenario in scene text datasets.

III. Method Description

Our pipeline consists of two offline database preparation steps followed by a online retrieval step. The general pipeline is described in Figure III First a selective search strategy is used to detect candidate text regions in the scene image. These text regions are grouped using an agglomerative clustering algorithms giving us a hierarchy of text hypothesis. These text hypothesis are then indexed based on presence of n-grams in that region. An inverted index per n-gram is generated. The hierarchical grouping performed in first step is based on many different diversification strategies, this may end up with many overlapping text regions. We remove the overlapping candidate regions based on initial ranking performed for index generation. We remove text windows which overlaps by more then 90% of the area, this reduces the number of text regions to match significantly. At query time n-grams present in the query word are identified and then the regions corresponding to the n-grams are searched for presence of the query word. To generate the index and to match candidate word images with text strings an attribute based representations similar to that of Almazán *et al.* [17] is used.

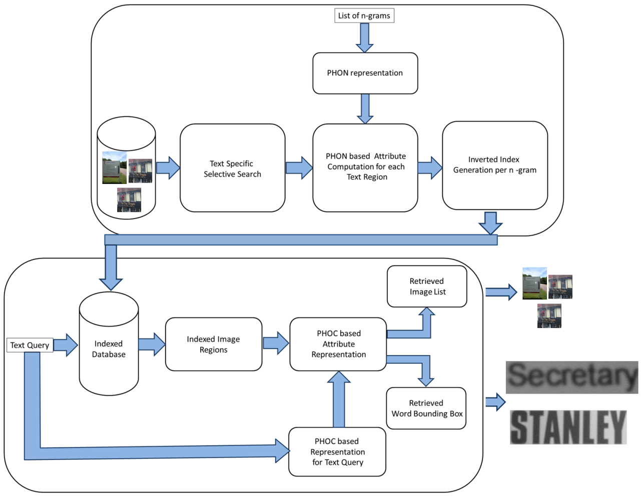
List of n-grams
PHON representation
Text Specific Selective Search
PHON based Attribute Computation for each Text Region
Inverted Index Generation per n -gram
Text Query
Indexed Database
Indexed Image Regions
PHOC based Attribute Representation
Retrieved Image List
Retrieved Word Bounding Box
PHOC based Representation for Text Query
Secretary
STANLEY

Fig. 1. Overview of the proposed pipeline

A. Selective Search based text hypothesis generation

Our text detection step builds upon the hierarchical method for text extraction proposed by Gomez *et al.* in [12] where text detection is posed as a search within a hierarchy produced by an agglomerative similarity clustering process over individual regions. The grouping process starts with a set of regions $\mathcal{R}_c$ extracted with the MSER algorithm. Initially each region $r \in \mathcal{R}_c$ starts in its own cluster and then the closest pair of clusters (A, B) is merged iteratively, using the single linkage criterion $(\min \{ \mathrm{d}(r_a, r_b) : r_a \in A,\, r_b \in B \})$, until all regions are clustered together $(C \equiv \mathcal{R}_c)$. The distance between two regions $\mathrm{d}(r_a, r_b)$ is defined as follows:

$$\mathrm{d}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{D} (w_i \cdot (a_i - b_i))^2 + \{(x_a - x_b)^2 + (y_a - y_b)^2\} \quad (1)$$

where we consider a feature space comprising five features $(D = 5)$. We use simple and low computational cost features allowing us to define the similarity between characters of a word or text line:

Size of the region. Characters in the same word usually have similar geometric appearance. We make use of the major axis of the fitting ellipse.

Intensity mean of the region. Characters belonging in the same word usually share the same colour. We calculate the mean intensity value of the pixels that belong to the region.

Intensity mean of the outer boundary. Same as before but for the pixels in the immediate outer boundary of the region.

Stroke width. Similar stroke width is expected for characters in the same word or text line, as they usually share the same font type. To determine the stroke width of a region we make here use of the Distance Transform as in [18].

Gradient magnitude mean on the border. The characters' contrast to their background is also expected to be similar by design within a word or text line. We calculate the mean of the gradient magnitude on the border of the region.

In addition to those similarity measures equation 1 includes a spatial term, the Euclidean distance between the x,y coordinates of the regions' centers. This way we restrict the groups of regions that are of interest to those that comprise of spatially close regions.

Using the above defined distance metric the SLC hierarchical clustering provides us with a dendrogram where each node represents a word hypothesis. Then we apply the binary classifier proposed in [12] for text/non-text discrimination in order to reduce the number of candidates. However, in our case we tune the acceptance threshold of this step in order to achieve a high detection recall.

Similarly as in [13] we assume that there is no single grouping strategy that is guaranteed to work well in all cases. Thus, our basic agglomerative process is extended with several diversification strategies in order to ensure the detection of the highest number of text regions in any case.

Diversification by use of different colour channels The MSER algorithm extracts regions from a single channel image, usually this is done in the gray scale version of the input image. However, there are many cases where text regions do not have enough contrast in a given single channel projection and thus we extract regions from the R,G,B, and gray channels separately.

Diversification by different initial regions. The MSER algorithm search for regions with high shape stability in the sequences of nested regions, bounded by a cut-off parameter δ, in the component tree. We extract regions using different values for the δ parameter in the MSER algorithm.

Diversification by complementary distance metrics We learn complementary optimal weights configurations in equation 1 using the same strategy as in [12].

Over-segmentation Since there is no guarantee that the nodes in our dendrograms match exactly with the word level group at some point, we further diversify the number of word hypothesis by assuming that all nodes contain more than one word and splitting them using different complementary threshold values over the average intra-region space.

B. *Attribute Computation*

For efficient word retrieval our aim is to learn a holistic representation which is discriminative yet low dimensional. Such a representation can be learned for word images and text strings using [17].

First, text strings are represented by a $d-$ dimensional binary embedding. This embedding – Pyramidal Histogram of Characters (PHOC) – encodes if a particular character appears in a particular spatial region of the string using a pyramidal decomposition making it more discriminative. The first level is just a basic histogram of characters encoding the presence or absence of a particular character in the string. Then, new levels are added where at each level of the pyramid the word is further split and a new histogram of characters is added for each new division to account for characters at different parts of the word. In particular Almazán *et al.* [17] used histograms at levels 2, 3, 4 and 5. In addition they also used a histogram of the 50 most popular bi-grams at level 2 thus resulting in a 604 dimensional word representations.

Then, this embedding is used as a source for learning character attributes from word images. Each word image is projected into a $d-$dimensional space (same dimension as the PHOC representation) where each dimension is a character attribute. In a way similar to the PHOC string representation each character attribute encodes the probability of appearance of a given character in a particular division of the image, using the same pyramidal decomposition as in the PHOC representation. Each attribute is independently learned using a SVM classifier on a Fisher Vector description of the word image, enriched with the x and y coordinates and the scale of the SIFT descriptor.

More formally, given a training image I and its associated text transcription, we can compute its Fisher Vector[1] representation $f(I)$, where $f(I)$ is a function of the form $f : I \rightarrow R^D$, where D is the dimension of the Fisher Vector representation. Now to project Fisher Vector representations into the PHOC attribute space, we learn an embedding function ϕ_I of the form $\phi_I : I \rightarrow R^d$ such that

$$\phi_I(I) = \mathbb{W}^{\mathrm{T}} f(I) \tag{2}$$

where W is a matrix with an SVM-based classifier for each attribute, that are learned using the PHOC labels obtained from the text transcription of all the training words.

At query time, text queries are encoded using the PHOC representation and word images are described with this attribute representation. Retrieval simply translates into finding the word candidates whose attribute representation is close to that of the query image. Almazán *et al.* [17] proposed an additional step consisting of learning a common subspace between strings and images as direct comparison between PHOCs and attribute representations is not well defined since PHOCs are binary, while the attribute scores are not. Thus, a final calibration step is added, using Canonical Correlation Analysis, that aims at maximizing the correlation among both representations.

This final calibration and dimensionality reduction step can be represented with an additional embedding function ψ represented as :$\psi_I : I \rightarrow R^{d'}$ and can be given as:

$$\psi_I(I) = U^T \phi_I(I) \tag{3}$$

being U the transformation matrix obtained with Canonical Correlation Analysis, in this work we have used this representation in the final retrieval step.

C. *n-gram Indexing*

For indexing, we have defined an alternative representation based on character n-grams based on a similar framework. Our starting hypothesis is that n-grams and their positions can be discriminative word features. Text regions can be effectively indexed per n-grams in an offline stage. Thus, the goal of this new representation is to identify the presence of a particular n-gram in a candidate image. To select the n-grams that are used to generate the index we refer to the study done in [19], where they analyzed a large corpus of English words for cryptographic studies. We consider 150 most popular bi-grams and 50 most popular tri-grams from this study which covers 99.21% of the total corpus. To include numeric fields the digits from 0-9 are also included in this representation. Then, this particular representation is obtained using 150 bi-grams and 50 tri-grams at level 2 of decomposition and using 10 digits at levels 2 and 3, thus resulting in a 450 dimensional representation. This representation is henceforth referred as PHON (Pyramidal Histogram of n-grams) in this article. Using the strategy described above, a similar embedding function is learned to project the Fisher Vector representation of text hypothesis into the PHON attribute space.

To generate an index over the text regions generated by the text detection step, n-grams are considered as strings and represented by the PHON representation introduced in section III-B. Each text region is also represented by the corresponding PHON based attribute representation using the Fisher Vector of the region. Both representations are then converted to a low dimensional common subspace using Canonical Correlation Analysis (CCA). Similarity between n-grams and segmented regions can be computed as a dot product between their corresponding representations in this space. For each n-gram, text regions are sorted in order of decreasing similarity and identifier for text regions are stored in inverted index files. For text-to-image retrieval purposes the image id for the corresponding text region is also recorded.

D. Retrieval

1) QBS Word Spotting: In word spotting, given a text string the aim is to extract all the occurrences of the string in the entire dataset. Given a query first all distinct n-grams of the query are found. Then, for each n-gram the inverted index is searched and top n text regions for each index are further considered for retrieval. Thus, for a query having k distinct n-grams from the list, $k \times n$ regions are searched. However many regions can be indexed by more than one such n-gram thus making the number of distinct regions to search much less for most cases. These detected candidates are then embedded to corresponding PHOC representation as described in III-B and compared through the cosine similarity with the PHOC representation of the query string. Finally, the candidate window list is ranked in order of decreasing similarity and non maximal suppression is performed to obtain the final relevance list.

2) Text-to-image Retrieval : In image retrieval, the goal is to find all images that may contain the query text. Similar to above, here the candidate regions for each n-gram occurring in the query is found and also the image id for the concerned region is extracted from the inverted index. All candidates are then evaluated by their corresponding PHOC representation and the highest match for each image is listed. Thus we get a list of image ids and their corresponding score for containing the text, which can be ranked in non-increasing order to get the final relevance list.

IV. Experimental Evaluation

We evaluate our method on the ICDAR2003 dataset [20] for the task of Query by string word spotting and the Street View Text (SVT) dataset[8] for text-to-image retrieval. The ICDAR03 dataset consists of 509 images, 258 for training and 251 for test, portraying scene text in various scenarios. The SVT dataset consists of a set of 349 annotated images, 100 for training and 249 for test, harvested from Google Street View. In our experiments on Query By String word spotting, we follow [17] and use the following standard protocol: a retrieval result is considered to be correct if the overlap over the union (OOU) of the bounding box with the ground truth is higher than 50% and the text matches with the query. All the unique words in ground truth are considered as queries. Then, after retrieval, extracted bounding boxes are ranked by cosine similarity with the query and Precision/Recall curves are generated. In our case, since we embed both images and strings into a common subspace, we are able to compute similarity measures between images and the text words. Finally, we use the cosine similarity score to apply non-maximal suppression and reduce the number of candidates.

In text-to-image retrieval the goal is to retrieve in a database of images, all the images that contain a given text word. A retrieved result is considered as correct if the image contains the query word. To evaluate our method in the SVT dataset, we follow [3]. All unique words of the test set of the SVT (a total of 427 query words) are considered as queries. We compute the similarity measure between the query word and all the candidates generated, and then, each image is scored using the maximum similarity score among its candidates. Following the state of the art methods, we also report Mean Average Precesion to evaluate the performance.

A. Evaluation of Alternative Selective Search Strategies

We have evaluated the combination of the different diversification strategies that have been drawn in III-A. It is obvious that increasing the number of candidates generated will result in higher text detection recall figures, and thus in principle better accuracy may be reached for the full end-to-end recognition. However, a complete search strategy, using all possible diversification strategies together, can become computationally expensive. We aim to find a trade-off between the amount of diversification and the obtained performance.

Among all the combinations that we have tested we keep two variants: the ”full” option which aim for maximizing the recall, and the ”efficient” with the aim of optimizing the time complexity for faster retrieval.

B. Baseline

To evaluate the effect of indexing we first evaluate the method with all text regions returned by the text detection step, i.e without use of any index. We take all nodes from the text detection step and rank them according to their similarity to qhe query. Table I shows the result in QBS word spotting. The performance for text-to-image retrieval is shown in Table II.

TABLE I. RESULT OF OUR BASELINE METHOD FOR QBS WORD SPOTTING TASK

	mAP	mean Recall	Total # of candidates
Proposed(full)	70.11	88.16	210750
Proposed(efficient)	68.89	87.16	11593

TABLE II. RESULTS OF OUR BASELINE METHOD COMPARED TO OTHER STATE OF THE ART METHOD FOR TEXT-TO-IMAGE RETRIEVAL

	f-score
Mishra *et al.* [3]	56.24
Jaderberg *et al.* [21]	86.3
Proposed (baseline-full)	60.91
Proposed(baseline-efficient)	57.08

The baseline results show that both *full* and *efficient* variants of selective search strategy achieve higher accuracy(mAP) than Mishra *et al.* [3] (56.2%), who introduced this particular task, while Jaderberg *et al.* [21] report an 86.3%.

It is important to note that, although our method is outperformed by [21], it is less demanding in terms of training both in the amount of training data and computational resources. Our method relies on an intermediate word representation that can be used in a natural way to recognize any word not previously seen in training. Finally, all steps of our pipeline could be easily parallelized, thus reducing the computational cost, especially the overhead introduced in the lexicon-based filtering.

In Query By String Word spotting, though is a popular approach in case of document image word spotting, there is no published result available for scene images to the best of our knowledge. No straightforward comparison is therefore possible.In the next section we will see how the use of n-gram indexing reduces the search space and time complexity significantly.

C. *Effect of indexing*

Table III shows our results with different choices of N for the word spotting task, where the N top candidates from the indexed list are considered for retrieval. In this case we have taken the full selective search strategy. We prefer full variant over efficient as using n-gram indexing the search space can be significantly reduced, so we go for maximum recall first and later reduce the search space in the indexing step.

The presented results shows that using n-gram indexing scheme, the search space is reduced significantly whereas the retrieval accuracy remains intact. For example when 5000 candidates for each n-gram are selected, we need to evaluate on average 9186 text regions in comparison to 210790 image regions in the case of full retrieval. However if we compare the performance then there is no significant difference.

V. CONCLUSION

A complete system for scene text retrieval in natural images is described. We apply our method in two different problems, QBS word spotting and text-to-image retrieval. We use a holistic word representation based on PHOC based attributes, that permits to tackle retrieval of any word as a simple nearest neighbour problem without explicitly recognizing it. An efficient inverted index file is generated for a set of n-grams from a set of candidate text boxes returned by a selective search text localization approach that generates multiple word hypotheses. For future work, we believe that we can further exploit this indexing scheme at different level of the text detection hierarchy and combine with retrieval step for faster retrieval.

TABLE III. RESULT OF FULL METHOD AFTER INDEXING

	mAP	Avg.# of candidate per query
Proposed(baseline)	**70.11**	**210790**
Proposed(N=1000)	66.82	1982
Proposed(N=2000)	68.86	3904
Proposed(N=3000)	69.23	5771
Proposed(N=5000)	**69.58**	**9186**

REFERENCES

[1] J. S. F. Perronnin and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010.

[2] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477. [Online]. Available: http://www.robots.ox.ac.uk/ vgg

[3] A. Mishra, K. Alahari, and C. Jawahar, "Image retrieval using textual cues," in *ICCV*, 2013.

[4] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, 2010.

[5] L. Gomez and D. Karatzas, "Multi-script text extraction from natural scenes," in *ICDAR*, 2013.

[6] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *Proc. ECCV*, 2012.

[7] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. ACCV*, 2010.

[8] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. ECCV*, 2010.

[9] O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid hmm maxout models," *arXiv preprint arXiv:1310.1811*, 2013.

[10] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *ECCV*, 2014.

[11] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. ICPR*, 2012.

[12] L. Gomez and D. Karatzas, "A fast hierarchical method for multi-script and arbitrary oriented scene text extraction," *arXiv preprint arXiv:1407.7504*, 2014.

[13] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, 2013.

[14] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *ICCV*, 2011.

[15] V. Yanulevskaya, J. Uijlings, and N. Sebe, "Learning to group objects," in *CVPR*, 2014.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[17] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," in *TPAMI*, 2014.

[18] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proc. ICIP*, 2011.

[19] [Online]. Available: "http://practicalcryptography.com/cryptanalysis/text-characterisation/monogram-bigram-and-trigram-frequency-counts/"

[20] L. P. Sosa, S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "Icdar 2003 robust reading competitions," in *In Proceedings of the Seventh International Conference on Document Analysis and Recognition*. IEEE Press, 2003, pp. 682–687.

[21] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *CoRR*, vol. abs/1412.1842, 2014.