Enhancing Energy Minimization Framework for Scene Text Recognition with Top-Down Cues

Anand Mishra[1]*, Karteek Alahari[2], C. V. Jawahar[1]

[1] *IIIT Hyderabad* [2] *Inria*

Abstract

Recognizing scene text is a challenging problem, even more so than the recognition of scanned documents. This problem has gained significant attention from the computer vision community in recent years, and several methods based on energy minimization frameworks and deep learning approaches have been proposed. In this work, we focus on the energy minimization framework and propose a model that exploits both bottom-up and top-down cues for recognizing cropped words extracted from street images. The bottom-up cues are derived from individual character detections from an image. We build a conditional random field model on these detections to jointly model the strength of the detections and the interactions between them. These interactions are top-down cues obtained from a lexicon-based prior, i.e., language statistics. The optimal word represented by the text image is obtained by minimizing the energy function corresponding to the random field model. We evaluate our proposed algorithm extensively on a number of cropped scene text benchmark datasets, namely Street View Text, ICDAR 2003, 2011 and 2013 datasets, and IIIT 5K-word, and show better performance than comparable methods. We perform a rigorous analysis of all the steps in our approach and analyze the results. We also show that state-of-the-art convolutional neural network features can be integrated in our framework to further improve the recognition performance.

Keywords: Scene text understanding, text recognition, lexicon priors, character recognition, random field models.

1. Introduction

The problem of understanding scenes semantically has been one of the challenging goals in computer vision for many decades. It has gained considerable attention over the past few years, in particular, in the context of street scenes [1, 2, 3]. This problem has manifested itself in various forms, namely, object detection [4, 5], object recognition and segmentation [6, 7]. There have also been significant attempts at addressing all these tasks jointly [2, 8, 9]. Although these approaches interpret most of the scene successfully, regions containing text are overlooked. As an example, consider an image of a typical street scene taken from Google Street View in Fig. 1. One of the first things we notice in this scene is the sign board and the text it contains. However, popular recognition methods ignore the text, and identify other objects such as car, person, tree, and regions such as road, sky. The importance of text in images is also highlighted in the experimental study conducted by Judd *et al.* [10]. They found that viewers fixate on text when shown images containing text and other objects. This is further evidence that text recognition forms a useful component in understanding scenes.

In addition to being an important component of scene understanding, scene text recognition has many potential applications, such as image retrieval, auto navigation, scene text to speech systems, developing apps for visually impaired people [13, 14]. Our method for solving this task is inspired by the many advancements made in the object detection and recognition problems [4, 5, 7, 15]. We present a framework for recognizing text that exploits bottom-up and top-down cues. The bottom-up cues are derived from individual character detections from an image. Naturally, these windows contain true as well as false positive detections of characters. We build a conditional random field (CRF) model [16] on these detections to determine not only the true positive detections, but also the word they represent jointly. We impose top-down cues obtained from a lexicon-based prior, i.e., language statistics, on the model. In addition to disambiguating between characters, this prior also helps us in recognizing words.

The first contribution of this work is a joint framework with seamless integration of multiple cues—individual character detections and their spatial arrangements, pairwise lexicon priors, and higher-order priors—into a CRF framework which can be optimized effectively. The proposed method performs significantly better than other related

[1] Center for Visual Information Technology, IIIT Hyderabad, India.

[2] LEAR team, Inria Grenoble Rhone-Alpes, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.

*Corresponding author –
Email: anand.mishra@research.iiit.ac.in
Phone: +91-40-6653 1000 Fax: +91-40-6653 1413

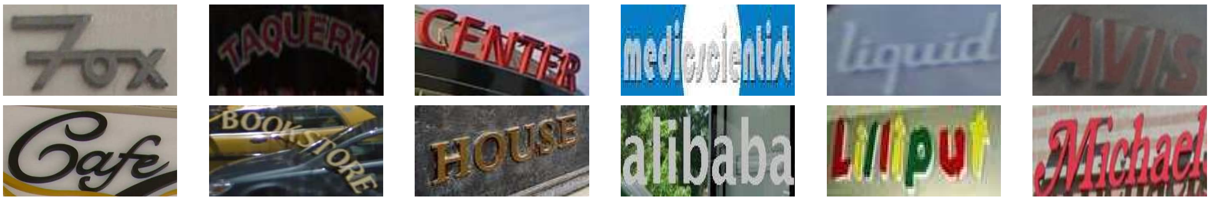


Figure 2 Challenges in scene text recognition. A few sample images from the SVT and IIIT 5K-word datasets are shown to highlight the variation in view point, orientation, non-uniform background, non-standard font styles and also issues such as occlusion, noise, and inconsistent lighting. Standard OCRs perform poorly on these datasets (as seen in Table 1 and [11, 12]).

Figure 1 A typical street scene image taken from Google Street View. It contains very prominent sign boards with text on the building and its windows. It also contains objects such as car, person, tree, and regions such as road, sky. Many scene understanding methods recognize these objects and regions in the image successfully, but overlook the text on the sign board, which contains rich, useful information. The goal of this work is to address this gap in understanding scenes.

energy minimization based methods for scene text recognition. Our second contribution is devising a cropped word recognition framework which is applicable not only to closed vocabulary text recognition (where a small lexicon containing the ground truth word is provided with each image), but also to a more general setting of the problem, i.e., open vocabulary scene text recognition (where the ground truth word may or may not belong to a generic large lexicon or the English dictionary). The third contribution is comprehensive experimental evaluation, in contrast to many recent works, which either consider a subset of benchmark datasets or are limited to the closed vocabulary setting. We evaluate on a number of cropped word datasets (ICDAR 2003, 2011 and 2013 [17], SVT [18], and IIIT 5K-word [19]) and show results in closed and open vocabulary settings. Additionally, we analyzed the effectiveness of individual components of the framework, the influence of parameter settings, and the use of convolutional neural network (CNN) based features [20].

The remainder of the paper is organized as follows. In Section 2 we discuss related work. Section 3 describes our scene text recognition model and its components. We then present the evaluation protocols and the datasets used in experimental analysis in Section 4. Comparison with related approaches is shown in Section 5, along with implementation details. We then make concluding remarks in Section 6.

2. Related Work

The task of understanding scene text has gained a huge interest for more than a decade [11, 12, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 20, 31]. It is closely related to the problem of Optical Character Recognition (OCR), which has a long history in the computer vision and pattern recognition communities [32]. However, the success of OCR systems is largely restricted to text from scanned documents. Scene text exhibits a large variability in appearance, as shown in Fig. 2, and can prove to be challenging even for the state-of-the-art OCR methods (see Table 1 and [11, 12]). The problems in this context are: (1) text localization, (2) cropped word recognition, and (3) isolated character recognition. They have been tackled either individually [21, 27, 33], or jointly [11, 20, 23, 29]. This paper focuses on addressing the cropped word recognition problem. In other words, given an image region (e.g., in the form of a bounding box) containing text, the task is to recognize this content. The core components of a typical cropped word recognition framework are: localize the characters, recognize them, and use statistical language models to compose the characters into words. Our framework builds on these components, but differs from previous work in several ways. In the following, we review the prior art and highlight these differences. The reader is encouraged to refer to [34] for a more comprehensive survey of scene text recognition methods.

A popular technique for localizing characters in an OCR system is to binarize the image and determine the potential character locations based on connected components [35]. Such techniques have also been adapted for scene text recognition [12], although with limited success. This is mainly because obtaining a clean binary output for scene text images is often challenging; see Fig. 3 for examples. An alternative approach is proposed in [36] using gradient information to find potential character locations. More recently, Yao *et al.* [31] proposed a mid-level feature based technique to localize characters in scene text. We follow an alternative strategy and cast the character localization problem as an object detection task, where characters are the *objects*. We then define an energy function on all the potential characters.

One of the earliest works on large-scale natural scene character recognition was presented in [27]. This work develops a multiple kernel learning approach using a set of shape-based features. Recent work [11, 37] has improved over this with histogram of gradient features [15]. We perform an extensive analysis on features, classifiers, and propose methods to improve character recognition further, for example, by augmenting the training set. In addition to this, we show that the state-of-the-art CNN features [20] can be successfully integrated with our word recognition framework to further boost its performance.

A study on human reading psychology shows that our reading improves significantly with prior knowledge of the language [38]. Motivated by such studies, OCR systems have used, often in post-processing steps [35, 39], statistical language models like n-grams to improve their performance. Bigrams or trigrams have also been used in the context of scene text recognition as a post-processing step, e.g., [40]. A few other works [41, 42, 43] integrate character recognition and linguistic knowledge to deal with recognition errors. For example, [41] computes n-gram probabilities from more than 100 million characters and uses a Viterbi algorithm to find the correct word. The method in [43], developed in the same year as our CVPR 2012 work [37], builds a graph on potential character locations and uses n-gram scores to constrain the inference algorithm to predict the word. In contrast, our approach uses a novel location-specific prior (cf. (6)).

The word recognition problem has been looked at in two contexts— with [11, 25, 37, 44, 45] and without [22, 19, 46] the use of an image-specific lexicon. In the case of image-specific lexicon-driven word recognition, also known as the closed vocabulary setting, a list of words is available for every scene text image. The task of recognizing the word now reduces to that of finding the best match from this list. This is relevant in many applications, e.g., recognizing text in a grocery store, where a list of grocery items can serve as a lexicon. Wang *et al.* [44] adapted a multi-layer neural network for this scenario. In [11], each word in the lexicon is matched to the detected set of character windows, and the one with the highest score is reported as the predicted word. In one of our previous works [45], we compared features computed on the entire scene text image and those generated from synthetic font renderings of lexicon words with a novel weighted dynamic time warping (wDTW) approach to recognize words. In [25] Rodriguez-Serrano and Perronnin proposed to embed word labels and word images into a common Euclidean space, wherein the text recognition task is posed as a retrieval problem to find the closest word label for a given word image. While all these approaches are interesting, their success is largely restricted to the closed vocabulary setting and cannot be easily extended to the more general cases, for instance, when image-specific lexicon is unavailable. Weinman *et al.* [22] proposed a method to address this issue, although with a strong assumption of known character boundaries, which are not trivial to obtain with high precision on the datasets we use. The work in [46] generalizes their previous approach by relaxing the character-boundary requirement. It is, however, evaluated only on "roughly fronto-parallel" images of signs, which are less challenging than the scene text images used in our work.

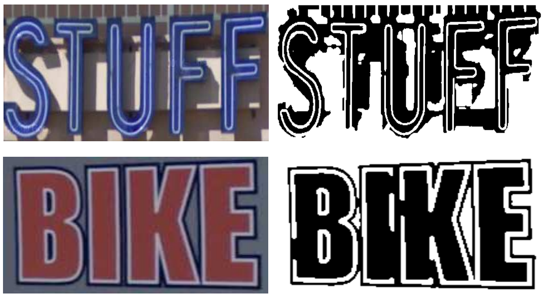


Figure 3 Binarization results obtained with one of the state-of-the-art methods [47] are shown for two sample images. We observed similar poor performance on most of the images in scene text datasets, and hence do not use binarization in our framework.

Our work belongs to the class of word recognition methods which build on individual character localization, similar to methods such as [12, 48]. In this framework, the potential characters are localized, then a graph is constructed from these locations, and then the problem of recognizing the word is formulated as finding an optimal path in this graph [49] or inferring from an ensemble of HMMs [48]. Our approach shows a seamless integration of higher order language priors into the graph (in the form of a CRF model), and uses more effective modern computer vision features, thus making it clearly different from previous works.

Since the publication of our original work in CVPR 2012 [37] and BMVC 2012 [19] papers, several approaches for scene text understanding (e.g., text localization [50, 29, 51, 52], word recognition [20, 23, 30, 31, 53, 51] and text-to-image retrieval [13, 51, 54, 55]) have been proposed. Notably, there has been an increasing interest in exploring deep convolutional network based methods for scene text tasks (see [20, 30, 44, 51, 52] for example). These approaches are very effective in general, but the deep convolutional network, which is at the core of these approaches, lacks the capability to elegantly handle structured output data. To understand this with the help of an example, let us consider the problem of estimating human pose [56, 57], where the task is to predict the locations of human body joints such as head, shoulders, elbows and wrists. These locations are constrained by human body kinematics and in essence form a structured output. To deal with such structured output data, state-of-the-art deep learning algorithms include an additional regression step [56] or a graphical model [57], thus showing that these techniques are complementary to the deep learning philosophy. Similar to human pose, text is structured output data [58]. To better handle this structured data, we develop our energy minimization framework [19, 37] with the motivation of building a complementary approach, which can further benefit methods built on the deep learning paradigm. Indeed, we see that combining the two frameworks further improves text recognition results (Section 5).

3. The Recognition Model

We propose a conditional random field (CRF) model for recognizing words. The CRF is defined over a set of N random variables $x = \{x_i | i \in \mathcal{V}\}$, where $\mathcal{V} = \{1, 2, \ldots, N\}$. Each random variable x_i denotes a potential character in the word, and can take a label from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\} \cup \epsilon$, which is the set of English characters, digits and a null label ϵ to discard false character detections. The most likely word represented by the set of characters x is found by minimizing the energy function, $E : \mathcal{L}^n \rightarrow \mathbb{R}$, corresponding to the random field. The energy function E can be written as sum of potential functions:

$$E(x) = \sum_{c \in \mathcal{C}} \psi_c(x_c), \tag{1}$$

where $\mathcal{C} \subset \mathcal{P}(\mathcal{V})$, with $\mathcal{P}(\mathcal{V})$ denoting the powerset of $\mathcal{V}$. Each x_c defines a set of random variables included in subset c, referred to as a clique. The function ψ_c defines a constraint (potential) on the corresponding clique c. We use unary, pairwise and higher order potentials in this work, and define them in Section 3.2. The set of potential characters is obtained by the character detection step discussed in Section 3.1. The neighbourhood relations among characters, modelled as pairwise and higher order potentials, are based on the spatial arrangement of characters in the word image.

In the following we show an example energy function composed of unary, pairwise and higher order (of clique size three) terms on a sample word with four characters. For a word to be recognized as "OPEN" the following energy function should be the minimum.

$$\begin{aligned} \psi(O,P,E,N) = {} & \psi_1(O) + \psi_1(P) + \psi_1(E) + \psi_1(N) \\ & + \psi_2(O,P) + \psi_2(P,E) + \psi_2(E,N) \\ & + \psi_3(O,P,E) + \psi_3(P,E,N). \end{aligned}$$

The third order terms $\psi_3(O,P,E)$ and $\psi_3(P,E,N)$ are decomposed as follows.

$$\begin{aligned} \psi_3(O,P,E) = {} & \psi_1^a(OPE) + \psi_2^a(OPE,O) \\ & + \psi_2^a(OPE,P) + \psi_2^a(OPE,E). \end{aligned}$$

$$\begin{aligned} \psi_3(P,E,N) = {} & \psi_1^a(PEN) + \psi_2^a(PEN,P) \\ & + \psi_2^a(PEN,E) + \psi_2^a(PEN,N). \end{aligned}$$

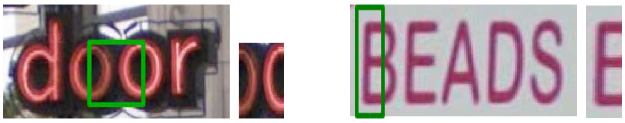


Figure 4 Typical challenges in character detection. (a) Inter-character confusion: A window containing parts of the two *o*'s is falsely detected as *x*. (b) Intra-character confusion: A window containing a part of the character B is recognized as E.

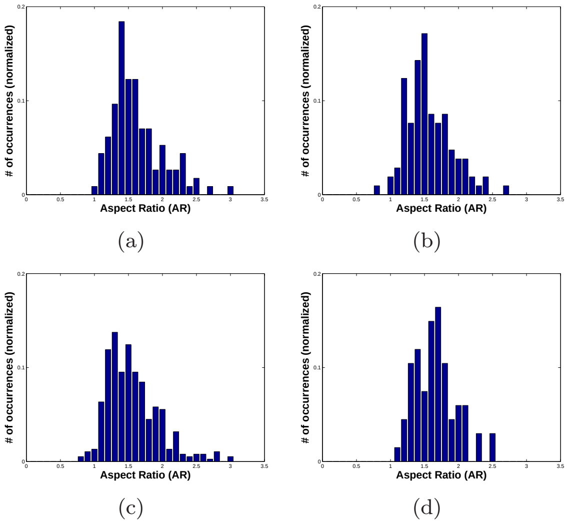
3.1. Character Detection

The first step in our approach is to detect potential locations of characters in a word image. In this work we use a sliding window based approach for detecting characters, but other methods, e.g., [31], can also be used instead.

Sliding window detection. This technique has been very successful for tasks such as, face [59] and pedestrian [15] detection, and also for recognizing handwritten words using HMM based methods [60]. Although character detection in scene images is similar to such problems, it has its unique challenges. Firstly, there is the issue of dealing with many categories (63 in all) jointly. Secondly, there is a large amount of inter-character and intra-character confusion, as illustrated in Fig. 4. When a window contains parts of two characters next to each other, it may have a very similar appearance to another character. In Fig. 4(a), the window containing parts of the characters '*o*' can be confused with '*x*'. Furthermore, a part of one character can have the same appearance as that of another. In Fig. 4(b), a part of the character 'B' can be confused with 'E'. We build a robust character classifier and adopt an additional pruning stage to overcome these issues.

The problem of classifying natural scene characters typically suffers from the lack of training data, e.g., [27] uses only 15 samples per class. It is not trivial to model the large variations in characters using only a few examples. To address this, we add more examples to the training set by applying small affine transformations [61, 62] to the original character images. We further enrich the training set by adding many non-character negative examples, i.e., from the background. With this strategy, we achieve a significant boost in character classification accuracy (see Table 3).

We consider windows at multiple scales and spatial locations. The location of the ith window, d_i, is given by its center and size. The set $\mathcal{K} = \{c_1, c_2, \ldots, c_k\}$, denotes label set. Note that $k = 63$ for the set of English characters, digits and a background class (null label) in our work. Let ϕ_i denote the features extracted from a window location d_i. Given the window d_i, we compute the likelihood,



(a) (b)

(c) (d)

Figure 5 Distribution of aspect ratios of few digits and characters: (a) 0 (b) 2 (c) B (d) Y. The aspect ratios are computed on character from the IIIT-5K word training set.

$p(c_j|\phi_i)$, of it taking a label c_j for all the classes in $\mathcal{K}$. In our implementation, we used explicit feature representation [63] of histogram of gradient (HOG) features [15] for ϕ_i, and the likelihoods p are (normalized) scores from a one vs rest multi-class support vector machine (SVM). Implementation details of the training procedure are provided in Section 5.1.

This basic sliding window detection approach produces many potential character windows, but not all of them are useful for recognizing words. We discard some of the weak detection windows using the following pruning method.

Pruning windows. For every potential character window, we compute a score based on: (i) SVM classifier confidence, and (ii) a measure of the aspect ratio of the character detected and the aspect ratio learnt for that character from training data. The intuition behind this score is that, a strong character window candidate should have a high classifier confidence score, and must fall within some range of the sizes observed in the training data. In order to define the aspect ratio measure, we observed the distribution of aspect ratios of characters from the IIIT-5K word training set. A few examples of these distributions are shown in Fig. 5. Since they follow a Gaussian distribution, we chose this score accordingly. For a window d_i with an aspect ratio a_i, let c_j denote the character with the best classifier confidence value given by S_{ij}. The mean aspect ratio for the character c_j computed from training data is denoted by μ_{a_j}. We define a goodness score (GS) for the window d_i as:

$$\text{GS}(d_i) = S_{ij} \exp\left(-\frac{(\mu_{a_j} - a_i)^2}{2\sigma_{a_j}^2}\right), \tag{2}$$

where σ_{a_j} is the variance of the aspect ratio for character c_j in the training data. A low goodness score indicates a weak detection, which is then removed from the set of candidate character windows.

We then apply character-specific non-maximum suppression (NMS), similar to other sliding window detection methods [5], to address the issue of multiple overlapping detections for each instance of a character. In other words, for every character class, we select detections which have a high confidence score, and do not overlap significantly with any of the other stronger detections of the same character class. We perform NMS after aspect ratio pruning to avoid wide windows with many characters suppressing weaker single character windows they overlap with. The pruning and NMS steps are performed conservatively, to discard only the obvious false detections. The remaining false positives are modelled in an energy minimization framework with language priors and other cues, as discussed below.

3.2. Graph Construction and Energy Formulation

We solve the problem of minimizing the energy function (1) on a corresponding graph, where each random variable is represented as a node in the graph. We begin by ordering the character windows based on their horizontal location in the image, and add one node each for every window sequentially from left to right. The nodes are then connected by edges. Since it is not natural for a window on the extreme left to be strongly related to another window on the extreme right, we only connect windows which are close to each other. The intuition behind close-proximity windows is that they could represent detections of two separate characters. As we will see later, the edges are used to encode the language model as top-down cues. Such pairwise language priors alone may not be sufficient in some cases, for example, when an image-specific lexicon is unavailable. Thus, we also integrate higher order language priors in the form of n-grams computed from the English dictionary by adding an auxiliary node connecting a set of n character detection nodes.

Each (non-auxiliary) node in the graph takes one label from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\} \cup \epsilon$. Recall that each l_u is an English character or digit, and the null label ϵ is used to discard false windows that represent background or parts of characters. The cost associated with this label assignment is known as the unary cost. The cost for two neighbouring nodes taking labels l_u and l_v is known as the pairwise cost. This cost is computed from bigram scores of character pairs in the English dictionary or an image-specific lexicon. The auxiliary nodes in the graph take labels from the extended label set $\mathcal{L}_e$. Each element of $\mathcal{L}_e$ represents one of the n-grams present in the dictionary and an additional label to assign a constant (high) cost to all n-grams that are not in the dictionary. The proposed model is illustrated in Fig. 6, where we show a CRF of order four as an example. Once the graph is constructed, we compute its corresponding cost functions as follows.

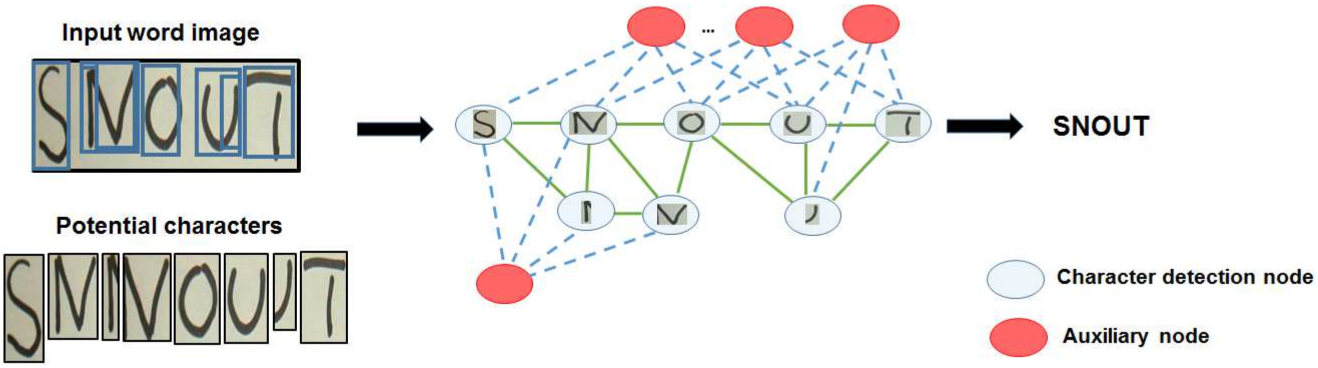


Figure 6 The proposed model illustrated as a graph. Given a word image (shown on the left), we evaluate character detectors and obtain potential character windows, which are then represented in a graph. These nodes are connected with edges based on their spatial positioning. Each node can take a label from the label set containing English characters, digits, and a null label (to suppress false detections). To integrate language models, i.e., n-grams, into the graph, we add auxiliary nodes (shown in red), which constrain several character windows together (sets of 4 characters in this example). Auxiliary nodes take labels from a label set containing all valid English n-grams and an additional label to enforce high cost for an invalid n-gram.

3.2.1. Unary cost

The unary cost of a node taking a character label is determined by the SVM confidence scores. The unary term ψ_1, which denotes the cost of a node x_i taking label l_u, is defined as:

$$\psi_1(x_i = l_u) = 1 - p(l_u|x_i), \tag{3}$$

where $p(l_u|x_i)$ is the SVM score of character class l_u for node x_i, normalized with Platt's method [64]. The cost of x_i taking the null label ϵ is given by:

$$\psi_1(x_i = \epsilon) = \max_u p(l_u|x_i) \exp\left(-\frac{(\mu_{a_u} - a_i)^2}{\sigma^2_{a_u}}\right), \tag{4}$$

where a_i is the aspect ratio of the window corresponding to node x_i, μ_{a_u} and σ_{a_u} are the mean and variance of the aspect ratio respectively of the character l_u, computed from the training data. The intuition behind this cost function is that, for taking a character label, the detected window should have a high classifier confidence and its aspect ratio should agree with that of the corresponding character in the training data.

3.2.2. Pairwise cost

The pairwise cost of two neighbouring nodes x_i and x_j taking a pair of labels l_u and l_v respectively is determined by the cost of their joint occurrence in the dictionary. This cost ψ_2 is given by:

$$\psi_2(x_i = l_u, x_j = l_v) = \lambda_l \exp(-\beta p(l_u, l_v)), \tag{5}$$

where $p(l_u, l_v)$ is the score determining the likelihood of the pair l_u and l_v occurring together in the dictionary. The parameters λ_l and β are set empirically as $\lambda_l = 2$ and $\beta = 50$ in all our experiments. The score $p(l_u, l_v)$ is commonly computed from joint occurrences of characters in the lexicon [41, 42, 43, 65]. This prior is effective when the lexicon size is small, but it is less so as the lexicon increases in size. Furthermore, it fails to capture the location-specific information of pairs of characters. As a toy example, consider a lexicon with only two words CVPR and ICPR. Here, the character pair (P,R) is more likely to occur at the end of the word, but a standard bigram prior model does not incorporate this location-specific information.

To overcome the lack of location-specific information, we devise a node-specific pairwise cost by adapting [66] to the scene text recognition problem. We divide a given word image into T parts, where T is an estimate of the number of characters in the image. This estimate T is given by the image width divided by the average character window width, with the average computed over all the detected characters in the image. To determine the pairwise cost involving windows in the t th image part, we define a region of interest (ROI) which includes the two adjacent parts $t-1$, $t+1$, in addition to t. With this, we do a ROI based search in the lexicon. In other words, we consider all the character pairs involving characters in locations $t-1$, t and $t+1$ in all the lexicon words to compute the likelihood of a pair occurring together. Note that the extreme cases (involving the leftmost and rightmost character in the lexicon word) are treated appropriately by considering only one of the two pairs.

This pairwise cost using the node-specific prior is given by:

$$\psi_2(x_i = l_u, x_j = l_v) = \begin{cases} 0 & \text{if } (l_u, l_v) \in \text{ROI}, \\ \lambda_l & \text{otherwise.} \end{cases} \tag{6}$$

We evaluated our approach with both the pairwise terms (5) and (6), and found that the node-specific prior (6) achieves better performance. The cost of nodes x_i and x_j taking label l_u and ϵ respectively is defined as:

$$\psi_2(x_i = l_u, x_j = \epsilon) = \lambda_o \exp(-\beta(1 - O(x_i, x_j))^2), \tag{7}$$

where $O(x_i, x_j)$ is the overlap fraction between windows corresponding to the nodes x_i and x_j. The pairwise cost $\psi_2(x_i = \epsilon, x_j = l_u)$ is defined similarly. The parameters are set empirically as $\lambda_{\text{o}} = 2$ and $\beta = 50$ in our experiments. This cost ensures that when two character windows overlap significantly, only one of them are assigned a character/digit label in order to avoid parts of characters being labelled.

3.2.3. Higher order cost

Let us consider a CRF of order $n = 3$ as an example to understand this cost. An auxiliary node corresponding to every clique of size 3 is added to represent this third order cost in the graph. The higher order cost is then decomposed into unary and pairwise terms with respect to this node, similar to [67]. Each auxiliary node in the graph takes one of the labels from the extended label set $\{L_1, L_2, \ldots, L_M\} \cup L_{M+1}$, where labels $L_1 \ldots L_M$ represent all the trigrams in the dictionary. The additional label L_{M+1} denotes all those trigrams which are absent in the dictionary. The unary cost ψ_1^a for an auxiliary variable y_i taking label L_m is:

$$\psi_1^a(y_i = L_m) = \lambda_{\text{a}} \exp(-\beta P(L_m)), \tag{8}$$

where λ_{a} is a constant. We set $\lambda_{\text{a}} = 5$ empirically, in all our experiments, unless stated otherwise. The parameter β controls penalty between dictionary and non-dictionary n-grams, and is empirically set to 50. The score $P(L_m)$ denotes the likelihood of trigram L_m in the English, and is further described in Section 3.2.4. The pairwise cost between the auxiliary node y_i taking a label $L_m = l_u l_v l_w$ and the left-most non-auxiliary node in the clique, x_i, taking a label l_r is given by:

$$\psi_2^a(y_i = L_m, x_i = l_r) = \begin{cases} 0 & \text{if } r = u \\ 0 & \text{if } l_r = \epsilon \\ \lambda_{\text{b}} & \text{otherwise,} \end{cases} \tag{9}$$

where λ_{b} penalizes a disagreement between the auxiliary and non-auxiliary nodes, and is empirically set to 1. The other two pairwise terms for the second and third nodes are defined similarly. Note that when one or more x_i's take null label, the corresponding pairwise term(s) between x_i(s) and the auxiliary node are set to 0.

3.2.4. Computing language priors

We compute n-gram based priors from the lexicon (or dictionary) and then adapt standard techniques for smoothing these scores [41, 68, 69] to the open and closed vocabulary cases.

Our method uses the score denoting the likelihood of joint occurrence of pair of labels l_u and l_v represented as $P(l_u, l_v)$, triplets of labels l_u, l_v and l_w denoted by $P(l_u, l_v, l_w)$ and even higher order (e.g., fourth order). Let $C(l_u)$ denote the number of occurrences of l_u, $C(l_u, l_v)$ be the number of joint occurrences of l_u and l_v next to each other, and similarly $C(l_u, l_v, l_w)$ is the number of joint occurrences of all three labels l_u, l_v, l_w next to each other. The smoothed scores [68] $P(l_u, l_v)$ and $P(l_u, l_v, l_w)$ are now:

$$P(l_u, l_v) = \begin{cases} 0.4 & \text{if } l_u, l_v \text{ are digits,} \\ \frac{C(l_u,l_v)}{C(l_v)} & \text{if } C(l_u, l_v) > 0, \\ \alpha_{l_u} P(l_v) & \text{otherwise,} \end{cases} \tag{10}$$

$$P(l_u, l_v, l_w) = \begin{cases} 0.4 & \text{if } l_u, l_v, l_w \text{ are digits,} \\ \frac{C(l_u,l_v,l_w)}{C(l_v,l_w)} & \text{if } C(l_u, l_v, l_w) > 0, \\ \alpha_{l_u} P(l_v, l_w) & \text{else if } C(l_u, l_v) > 0, \\ \alpha_{l_u,l_v} P(l_w) & \text{otherwise,} \end{cases} \tag{11}$$

Image-specific lexicons (small or medium) are used in the closed vocabulary setting, while in the open vocabulary case we use a lexicon containing half a million words (henceforth referred to as large lexicon) provided by [22] to compute these scores. The parameters α_{l_u} and α_{l_u,l_v} are learnt on the large lexicon using SRILM toolbox.[3] They determine the low score values for n-grams not present in the lexicon. We assign a constant value (0.4) when the labels are digits, which do not occur in the large lexicon.

3.2.5. Inference

Having computed the unary, pairwise and higher order terms, we use the sequential tree-reweighted message passing (TRW-S) algorithm [70] to minimize the energy function. The TRW-S algorithm maximizes a concave lower bound of the energy. It begins by considering a set of trees from the random field, and computes probability distributions over each tree. These distributions are then used to reweight the messages being passed during loopy belief propagation [71] on each tree. The algorithm terminates when the lower bound cannot be increased further, or the maximum number of iterations has been reached.

In summary, given an image containing a word, we: (i) locate the potential characters in it with a character detection scheme, (ii) define a random field over all these potential characters, (iii) compute the language priors and integrate them into the random field model, and then (iv) infer the most likely word by minimizing the energy function corresponding to the random field.

4. Datasets and Evaluation Protocols

Several public benchmark datasets for scene text understanding have been released in recent years. ICDAR [17] and Street View Text (SVT) [18] datasets are two of the initial datasets for this problem. They both contain data for text localization, cropped word recognition and isolated character recognition tasks. In this paper we use the cropped word recognition part from these datasets.

[3] Available at: `http://www.speech.sri.com/projects/srilm/`

Table 1 Our IIIT 5K-word dataset contains a few less challenging (Easy) and many very challenging (Hard) images. To present analysis of the dataset, we manually divided the words in the training and test sets into *easy* and *hard* categories based on their visual appearance. The recognition accuracy of a state-of-the-art commercial OCR – ABBYY9.0 – for this dataset is shown in the last column. Here we also show the total number of characters, whose annotations are also provided, in the dataset.

	Training Set		
	#words	#characters	ABBYY9.0(%)
Easy	658	-	44.98
Hard	1342	-	16.57
Total	2000	9658	20.25

	Test Set		
	#words	#characters	ABBYY9.0(%)
Easy	734	-	44.96
Hard	2266	-	5.00
Total	3000	15269	14.60

Although these datasets have served well in building interest in the scene text understanding problem, they are limited by their size of a few hundred images. To address this issue, we introduced the IIIT 5K-word dataset [19], containing a diverse set of 5000 words. Here, we provide details of all these datasets and the evaluation protocol.

SVT. The street view text (SVT) dataset contains images taken from Google Street View. As noted in [72], most of the images come from business signage and exhibit a high degree of variability in appearance and resolution. The dataset is divided into SVT-spot and SVT-word, meant for the tasks of locating and recognizing words respectively. We use the SVT-word dataset, which contains 647 word images.

Our basic unit of recognition is a character, which needs to be localized before classification. Failing to detect characters will result in poorer word recognition, making it a critical component of our framework. To quantitatively measure the accuracy of the character detection module, we created ground truth data for characters in the SVT-word dataset. This ground truth dataset contains around 4000 characters of 52 classes, and is referred to as as SVT-char, which is available for download [73].

ICDAR 2003 dataset. The ICDAR 2003 dataset was originally created for text detection, cropped character classification, cropped and full image word recognition, and other tasks in document analysis [17]. We used the part corresponding to the cropped word recognition called robust word recognition. Following the protocol of [11], we ignore words with less than two characters or with non-alphanumeric characters, which results in 859 words overall. For subsequent discussion we refer to this dataset as ICDAR(50) for the image-specific lexicon-driven case (closed vocabulary), and ICDAR 2003 when this lexicon is unavailable (open vocabulary case).

Table 2 Analysis of the IIIT 5K-word dataset. We show the percentage of non-dictionary words (Non-dict.), including digits, and the percentage of words containing only digits (Digits) in the first two rows. We also show the percentage of words that are composed from valid English trigrams (Dict. 3-grams), four-grams (Dict. 4-grams) and five-grams (Dict. 5-grams) in the last three rows. These statistics are computed using the large lexicon.

	IIIT 5K train	IIIT 5K test
Non-dict. words	23.65	22.03
Digits	11.05	7.97
Dict. 3-grams	90.27	88.05
Dict. 4-grams	81.40	79.27
Dict. 5-grams	68.92	62.48

ICDAR 2011/2013 datasets. These datasets were introduced as part of the ICDAR robust reading competitions [74, 75]. They contain 1189 and 1095 word images respectively. We show case-sensitive open vocabulary results on both these datasets. Also, following the ICDAR competition evaluation protocol, we do not exclude words containing special characters (such as &, :), and report results on the entire dataset.

IIIT 5K-word dataset. The IIIT 5K-word dataset [19, 73] contains both scene text and born-digital images. Born-digital images—category of images which has gained interest in ICDAR 2011 competitions [74]—are inherently low-resolution, made for online transmission, and have a variety of font sizes and styles. This dataset is not only much larger than SVT and the ICDAR datasets, but also more challenging. All the images were harvested through Google image search. Query words like billboard, signboard, house number, house name plate, movie poster were used to collect images. The text in the images was manually annotated with bounding boxes and their corresponding ground truth words. The IIIT 5K-word dataset contains in all 1120 scene images and 5000 word images. We split it into a training set of 380 scene images and 2000 word images, and a test set of 740 scene images and 3000 word images. To analyze the difficulty of the IIIT 5K-word dataset, we manually divided the words in the training and test sets into *easy* and *hard* categories based on their visual appearance. An annotation team consisting of three people have done three independent splits. Each word is then tagged as either being easy or hard by taking a majority vote. This split is available on our project page [73]. Table 1 shows these splits in detail. We observe that a commercial OCR performs poorly on both the train and test splits. Furthermore, to evaluate components like character detection and recognition, we also provide annotated character bounding boxes. It should be noted that around 22% of the words in this dataset are not in the English dictionary, e.g., proper nouns, house numbers, alphanumeric words. This makes this dataset suitable for open vocabulary cropped word recognition. We show an analysis of dictionary and non-dictionary words in Table 2.

Evaluation protocol. We evaluate the word recognition accuracy in two settings: closed and open vocabulary. Following previous work [11, 53, 19], we evaluate case-insensitive word recognition on SVT, ICDAR 2003, IIIT 5K-word, and case-sensitive word recognition on ICDAR 2011 and ICDAR 2013. For the closed vocabulary recognition case, we perform a minimum edit distance correction, since the ground truth word belongs to the image-specific lexicon. On the other hand, in the case of open vocabulary recognition, where the ground truth word may or may not belong to the large lexicon, we do not perform edit distance based correction. We perform many of our analyses on the IIIT 5K-word dataset, unless otherwise stated, since it is the largest dataset for this task, and also comes with character bounding box annotations.

5. Experiments

Given an image region containing text, cropped from a street scene, our task is to recognize the word it contains. In the process, we develop several components (such as a character recognizer) and also evaluate them to justify our choices. The proposed method is evaluated in two settings, namely, closed vocabulary (with an image-specific lexicon) and open vocabulary (using an English dictionary for the language model). We compare our results with the best-performing recent methods for these two cases. For baseline comparisons we choose commercial OCR namely ABBYY [78] and a public implementation of a recent method [79] in combination with an open source OCR.

5.1. Character Classifier

We use the training sets of ICDAR 2003 character [17] and Chars74K [27] datasets to train the character classifiers. This training set is augmented with 48 × 48 patches harvested from scene images, with buildings, sky, road and cars, which do not contain text, as additional negative training examples. We then apply affine transformations to all the character images, resize them to 48 × 48, and compute HOG features. Three variations (13, 31 and 36-dimensional) of HOG were analyzed (see Table 3). We then use an explicit feature map [63] and the χ^2 kernel to learn the SVM classifier. The SVM parameters are estimated by cross-validating on a validation set. The explicit feature map not only allows a significant reduction in classification time, compared to non-linear kernels like RBF, but also achieves a good performance.

The two main differences from our previous work [37] in the design of the character classifier are: (i) enriching the training set, and (ii) using an explicit feature map and a linear kernel (instead of RBF). Table 3 compares our character classification performance with [11, 27, 37, 76, 77, 43] on several test sets. We achieve at least 4% improvement over our previous work (RBF [37]) on all the datasets, and also perform better than [11, 27]. We are also comparable to a few other recent methods [43, 76], which show a limited evaluation on the ICDAR 2003 dataset. Following an evaluation insensitive to case (as done in a few benchmarks, e.g., [20, 53], we obtain 77% on ICDAR 2003, 75% on SVT-char, 79% on Chars74K, and 75% on IIIT 5K-word. It should be noted that feature learning methods based on convolutional neural networks, e.g., [77, 20], show an excellent performance. This inspired us to integrate them into our framework. We used publicly available features [20]. This will be further discussed in Section 5.3. We could not compare with other related recent methods [30, 23] since they did not report isolated character classification accuracy.

In terms of computation time, linear SVMs trained with HOG-13 features outperform others, but since our main focus is on word recognition performance, we use the most accurate combination, i.e., linear SVMs with HOG-36. We observed that this smart selection of training data and features not only improves character recognition accuracy but also improves the second and third best predictions for characters.

5.2. Character Detection

Sliding window based character detection is an important component of our framework, since our random field model is defined on these detections. We use windows of aspect ratio ranging from 0.1 to 2.5 for sliding window and at every possible location of the sliding window, we evaluate a character classifier. This provides the likelihood of the window containing the respective character. We pruned some of the windows based on their aspect ratio, and then used the goodness measure (2) to discard the windows with a score less than 0.1 (refer Section 3.1). Character-specific NMS is done on the remaining windows with an overlap threshold of 40%, i.e., if two detections have more than 40% overlap and represent the same character class, we suppress the weaker detection. We evaluated the character detection results with the intersection over union measure and a threshold of 50%, following ICDAR 2003 [17] and PASCAL-VOC [80] evaluation protocol. Our sliding window approach achieves recall of 80% on the IIIT 5K-word dataset, significantly better than using a binarization scheme for detecting characters and also superior to techniques like MSER [81] and CSER [79] (see Table 7 and Section 5.4).

5.3. Word Recognition

Closed vocabulary recognition. The results of the proposed CRF model in closed vocabulary setting are presented

Table 3 Character classification accuracy (in %). A smart choice of features, training examples and classifier is key to improving character classification. We enrich the training set by including many affine transformed (AT) versions of the original training data from ICDAR and Chars74K (c74k). The three variants of our approach (H-13, H-31 and H-36) show noticeable improvement over several methods. The character classification results shown here are case sensitive (all rows except the last two). It is to be noted that [27] only uses 15 training samples per class. The last two rows show a case insensitive (CI) evaluation. ∗We do not evaluate the convolutional neural network classifier in [20] (CNN feat+classifier) on the c74K dataset, since the entire dataset was used to train the network.

Method	SVT	ICDAR	c74K	IIIT 5K	Time
Exempler SVM [76]	-	71	-	-	-
Elagouni *et al.* [43]	-	70	-	-	-
Coates *et al.* [77]	-	82	-	-	-
FERNS [11]	-	52	47	-	-
RBF [37]	62	62	64	61	3ms
MKL+RBF [27]	-	-	57	-	11ms
H-36+AT+Linear	69	73	68	66	2ms
H-31+AT+Linear	64	73	67	63	1.8ms
H-13+AT+Linear	65	72	66	64	0.8ms
H-36+AT+Linear (CI)	75	77	79	75	0.8ms
CNN feat+classifier [20] (CI)	83	86	∗	85	1ms

in Table 4. We compare our method with many recent works for this task. To compute the language priors we use lexicons provided by authors of [11] for SVT and ICDAR(50). The image-specific lexicon for every word in the IIIT 5K-word dataset was developed following the method described in [11]. These lexicons contain the ground truth word and a set of distractors obtained from randomly chosen words (from all the ground truth words in the dataset). We used a CRF with higher order term (n=4), and similar to other approaches, applied edit distance based correction after inference. The constant λ_{a} in (8) to 1, given the small size of the lexicon.

The gain in accuracy over our previous work [37], seen in Table 4, can be attributed to the higher order CRF and an improved character classifier. The character classifier uses: (i) enriched training data, and (ii) an explicit feature map, to achieve about 5% gain (see Section 5.1 for details). Other methods, in particular, our previous work on holistic word recognition [45], label embedding [25] achieve a reasonably good performance, but are restricted to the closed vocabulary setting, and their extension to more general settings, such as the open vocabulary case, is unclear. Methods published since our original work [37], such as [23, 53], also perform well. Very recently, methods based on convolutional neural networks [30, 20] have shown very impressive results for this problem. It should be noted that such methods are typically trained on much larger datasets, for example, 10M compared to 0.1M typically used in state-of-the-art methods, which are not publicly available [30]. Inspired by these successes, we use a CNN classifier [20] to recognize characters, instead of our SVM classifier based on HOG features (see Sec. 3.1). We show results with this CNN classifier on SVT, ICDAR 2003 and IIIT-5K word datasets in Table 4 and observe significant improvement in accuracy, showing its complementary nature to our energy based method. However, there remains a difference in performance between the deep feature based method [20] and [This work, CNN]. This is primarily due to use of CNN features for learning classifiers for individual character as well as bi-grams in [20]. In contrast, our method only uses the pre-trained character classifier provided by [20]. Nevertheless, the improvement observed over [This work, HOG] does show the complementary nature of the two approaches, and integrating the two further would be an interesting avenue for future research.

Open vocabulary recognition. In this setting we use a lexicon of 0.5 million words from [22] instead of image-specific lexicons to compute the language priors. Many character pairs are equally likely in such a large lexicon, thereby rendering pairwise priors is less effective than in the case of a small lexicon. We use priors of order four to address this (see also analysis on the CRF order in Section 5.4). Results on various datasets in this setting are shown in Table 5. We compare our method with recent work by Feild and Miller [26] on the ICDAR 2003 dataset, where our method with HOG features shows a comparable performance. Note that [26] additionally uses web-based corrections, unlike our method, where the results are obtained directly by performing inference on the higher order CRF model. On the ICDAR 2011 and 2013 datasets we compare our method with the top performers from the respective competitions. Our method outperforms the ICDAR 2011 robust reading competition winner (TH-OCR method) method by 17%. This performance is also better than a recently published work from 2014 by Weinman *et al.* [23]. On the ICDAR 2013 dataset, the proposed higher order model is significantly better than the baseline and is in the top-5 performers among the competition entries. The winner of this competition (PhotoOCR) uses a large

Table 4 Word recognition accuracy (in %): closed vocabulary setting. We present results of our proposed higher order model ("This work") with HOG as well as CNN features. See text for details.

Method	Accuracy
ICDAR 2003 (50) dataset	
Baseline (ABBYY) [78]	56.04
Baseline (CSER+tesseract) [79]	57.27
Novikova *et al.* [24]	82.80
Our Holistic recognition [45]	89.69
Deep learning approaches	
Wang *et al.* [44]	90.00
Deep features [20]	96.20
Other energy min. approaches	
PLEX [11]	72.00
Shi *et al.* [53]	87.04
Our variants:	
Pairwise CRF [37]	81.74
Higher order [This work, HOG]	84.07
Higher order [This work, CNN]	88.02
SVT-Word dataset	
Baseline(ABBYY) [78]	35.00
Baseline (CSER+tesseract) [79]	37.71
Novikova *et al.* [24]	72.90
Our Holistic recognition [45]	77.28
Deep learning approaches	
Wang *et al.* [44]	70.00
PhotoOCR [30]	90.39
Deep features [20]	86.10
Other energy min. approaches	
PICT [72]	59.00
PLEX [11]	57.00
Shi *et al.* [53]	73.51
Weinman *et al.* [23]	78.05
Our variants:	
Pairwise CRF [37]	73.26
Higher order [This work, HOG]	75.27
Higher order [This work, CNN]	78.21
IIIT 5K-Word (Small)	
Baseline(ABBYY) [78]	24.50
Baseline (CSER+tesseract) [79]	33.07
Rodriguez & Perronnin [25]	76.10
Strokelets [31]	80.20
Our variants:	
Pairwise CRF [37]	66.13
Higher order [This work, HOG]	71.80
Higher order [This work, CNN]	78.07

proprietary training dataset, which is unavailable publicly, making it infeasible to do a fair comparison. Other methods (NESP [82], MAPS [83], PLT [84]) use many preprocessing techniques, followed by off-the-self OCR. Such preprocessing techniques are highly dataset dependent and may not generalize easily to all the challenging datasets we use. Despite the lack of these preprocessing steps, our method shows a comparable performance. On the IIIT 5K-word dataset, which is large (three times the size of ICDAR 2013 dataset) and challenging, the only published result to our knowledge is Strokelets [31] from CVPR 2014. Our method performs 7% better than Strokelets. Using CNN features instead of HOG further improves our word recognition accuracy, as shown in Table 5.

The main focus of this work is on evaluating datasets containing scene text images or a mixture of scene text and born-digital images. Nevertheless, we also tested our method on the born-digital image dataset from the recent ICDAR 2013 competition. Our approach with pre-trained CNN features achieves 78% accuracy on this dataset, which is comparable to other top performers (80.40%, 80.26%, 79.40%), and lower than PhotoOCR (82%), the competition winner using an end-to-end deep learning approach.

To sum up, our proposed method performs well consistently on several popular scene text datasets. Fig. 7 shows the qualitative performance of the proposed method on a few sample images. The higher order CRF outperforms the unary and pairwise CRFs. This is intuitive due to the better expressiveness of the higher order potentials. One of the failure cases is shown in the last row in Fig. 7, where the higher order potential is computed from a lexicon which does not have sufficient examples to handle alphanumeric words.

5.4. Further Analysis

Lexicon size. The size of the lexicon plays an important role in the word recognition performance. With a small-size lexicon, we obtain strong language priors which help overcome inaccurate character detection and recognition in the closed vocabulary setting. A small lexicon provides much stronger priors than the large lexicon in this case, as the performance degrades with increase in the lexicon size. We show this behaviour on the IIIT 5K-word dataset in Table 6 with small (50), medium (1000) and large (0.5 million) lexicons. We also compare our results with a state-of-the-art methods [25, 31]. We observe that [25, 31] shows better recognition performance with the small lexicon, when we use HOG features, but as the size of the lexicon increases, our method outperforms [25].

Alternatives for character detection.. While our sliding window approach for character detection performs well in several scenarios, including text that is not aligned with the image axes to a small extent (e.g., rows 4 - 6 in Figure 7), there are other alternatives. In particular, we investigated the use of binarization, MSER [81], and CSER [49] algorithms. In the first experiment, we replaced our detection module with a binarization based character extraction scheme – either a traditional binarization technique [85] or a more recent random field based approach [47]. A connected component analysis was performed on the binarized images to obtain a set of potential character locations. We then defined the CRF on these characters and performed

Table 5 Word recognition accuracy (in %): open vocabulary setting. The results of our proposed higher order model ("This work") with HOG as well as CNN features are presented here. Since the network used here to compute CNN features, i.e. [20], is learnt on data from several sources (e.g., ICDAR 2013), we evaluated with CNN features only on ICDAR 2003 and IIIT-5K word datasets, as recommended by the authors. Note that we also compare with top performers (as given in [74, 75]) in the ICDAR 2011 and 2013 robust reading competitions. We follow standard protocols for evaluation – case sensitive on ICDAR 2011 and 2013 and case insensitive on ICDAR 2003 and IIIT 5K-Word.

Method	Accuracy
ICDAR 2003 dataset	
Baseline (ABBYY)	46.51
Baseline (CSER+tesseract) [79]	50.99
Feild and Miller [26]	62.76
Our variants	
Pairwise [37]	50.99
Higher order [This work, HOG]	63.02
Higher order [This work, CNN]	67.67
ICDAR 2011 dataset	
Baseline (ABBYY)	46.00
Baseline (CSER+tesseract) [79]	51.98
Weinman *et al.* [23]	57.70
Feild and Miller [26]	48.86
ICDAR'11 competition [74]	
TH-OCR System	41.20
KAIST AIPR System	35.60
Neumann's Method	33.11
Our variants	
Pairwise [37]	48.11
Higher order [This work, HOG]	58.03
ICDAR 2013 dataset	
Baseline (ABBYY)	45.30
Baseline (CSER+tesseract) [79]	50.26
ICDAR'13 competition [75]	
PhotoOCR [30]	82.83
NESP [82]	64.20
MAPS [83]	62.74
PLT [84]	62.37
PicRead [24]	57.99
POINEER [22, 23]	53.70
Field's Method [26]	47.95
TextSpotter [12, 29, 49]	26.85
Our variants	
Pairwise [37]	49.86
Higher order [This work, HOG]	60.18
IIIT 5K-Word	
Baseline (ABBYY)	14.60
Baseline (CSER+tesseract) [79]	25.00
Stroklets [31]	38.30
Our variants	
Pairwise [37]	32.00
Higher order [This work, HOG]	44.50
Higher order [This work, CNN]	46.73

Table 6 Studying the influence of the lexicon size – small (S), medium (M), large (L) – on the IIIT 5K-word dataset in the closed vocabulary setting.

Method	S	M	L
Rodriguez & Perronnin [25]	76.10	57.50	-
Strokelets [31]	80.20	69.30	38.30
Higher order [This work, HOG]	71.80	62.17	44.50
Higher order [This work, CNN]	78.07	70.13	46.73

inference to get the text contained in the image. These results are summarized in Table 7. We observe that binarization based methods perform poorly compared to our model using a sliding window detector, both in terms of character-level recall and word recognition. They fail in extracting characters in the presence of noise, blur or large foreground-background variations. MSER [81] or related algorithms (e.g., CSER [49]) may also help to deal with text that is not axis-oriented, but they are not necessarily ideal for character extraction compared to a sliding window method. To study this, we replaced our sliding window based character detection scheme with either one of these approaches. From Table 7 we observe that sliding window character extraction is marginally better than CSER and significantly better than MSER. One of the reasons for this is that the classifier used in the sliding window detector is trained on a large variety of character classes and is less prone to errors than the MSER equivalent. These results further justify our choice of sliding window based character detection, although the challenging problem of effectively dealing with text that is not axis-oriented remains an interesting task for the future.

Effect of pruning. We propose a pruning step to discard candidates based on a combination of character-specific aspect ratio and classification scores (2), instead of simply using extreme aspect ratio to discard character candidates. This pruning helps in removing many false positive windows, and thus improves recognition performance. We conducted an experiment to study the effect of pruning on the IIIT-5K dataset in the open vocabulary setting, and observed a gain of 4.23% (46.73% vs 42.50%) due to pruning.

CRF order. We varied the order of the CRF from two to six and obtained accuracy of 32%, 43%, 45%, 43%, 42% respectively on the IIIT 5K-word dataset in the open vocabulary setting. Increasing the CRF order beyond four forces a recognized word to be one from the dictionary, which leads to poor recognition performance for non-dictionary words, and thus deteriorates the overall accuracy. Empirically, the fourth order prior shows the best performance.

Limits of statistical language models. Statistical language models have been very useful in improving traditional OCR performance, but they are indeed limited [65, 86]. For instance, using a large weight for language prior potentials may bias the recognition towards the closest dictionary

Table 7 Character recall (C. recall) and recognition accuracy, with unary only (Unary), unary and pairwise (Pairwise) and the full higher order (H. order) models, (all in %), on the IIIT 5K-word dataset with various character extraction schemes (Char. method). See text for details.

Char. method	C. recall	Unary	Pairwise	H. order
Otsu [85]	56	17.07	20.20	24.87
MRF model [47]	62	20.10	22.97	28.03
MSER [81]	72	23.20	28.50	34.70
CSER [49] [79]	78	24.50	30.00	42.87
Sliding window	80	25.83	32.00	44.50

word. This is especially true when the character recognition part of the pipeline is weak. We study such impact of language models in this experiment. Our analysis on the IIIT 5K-word dataset suggests that many of the non-dictionary words are composed of valid English n-grams (see Table 2). However, there are few exceptions, e.g., words like 35KM, 21P, which are composed of digits and characters; see last row of Fig. 7. Using language models has an adverse effect on the recognition performance in such cases. This results in inferior recognition performance on non-dictionary words as compared to dictionary words, e.g. on IIIT-5K dataset our method achieves 51% and 24% word recognition accuracy on dictionary and non-dictionary words respectively.

6. Summary

This paper proposes an effective method to recognize scene text. Our model combines bottom-up cues from character detections and top-down cues from lexicon. We jointly infer the location of true characters and the word they represent as a whole. We evaluated our method extensively on several challenging street scene text datasets, namely SVT, ICDAR 2003/2011/2013, and IIIT 5K-word and showed that our approach significantly advances the energy minimization based approach for scene text recognition. In addition to presenting the word recognition results, we analyzed the different components of our pipeline, presenting their pros and cons. Finally, we showed that the energy minimization framework is complementary to the resurgence of convolutional neural network based techniques, which can help build better scene understanding systems.

Acknowledgements. We thank Jerod Weinman for providing the large lexicon. This work was partially supported by the Ministry of Communications and Information Technology, Government of India, New Delhi. Anand Mishra is supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD fellowship award.

References

[1] G. J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, in: ECCV, 2008.

[2] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, P. H. S. Torr, What, Where & How Many? Combining object detectors and CRFs, in: ECCV, 2010.

[3] A. Geiger, P. Lenz, R. Urtasun, Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite, in: CVPR, 2012.

[4] C. Desai, D. Ramanan, C. Fowlkes, Discriminative models for multi-class object layout, in: ICCV, 2009.

[5] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.

[6] A. Levin, Y. Weiss, Learning to Combine Bottom-Up and Top-Down Segmentation, Int'l J. Computer Vision 81 (1) (2009) 105–118.

[7] J. Shotton, J. Winn, C. Rother, A. Criminisi, Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, Int'l J. Computer Vision 81 (1) (2009) 2–23.

[8] S. Gould, T. Gao, D. Koller, Region-based segmentation and object detection, in: NIPS, 2009.

[9] J. Yao, S. Fidler, R. Urtasun, Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation, in: CVPR, 2012.

[10] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: ICCV, 2009.

[11] K. Wang, B. Babenko, S. Belongie, End-to-End Scene Text Recognition, in: ICCV, 2011.

[12] L. Neumann, J. Matas, A Method for Text Localization and Recognition in Real-World Images, in: ACCV, 2010.

[13] A. Mishra, K. Alahari, C. V. Jawahar, Image Retrieval using Textual Cues, in: ICCV, 2013.

[14] L. Neumann, J. Matas, A real-time scene text to speech system, in: ECCV workshops, 2012.

[15] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in: CVPR, 2005.

[16] J. Lafferty, A. McCallum, F. Pereira, Conditional Random Fields: Probabilistic models for segmenting and labelling sequence data, in: ICML, 2001.

[17] ICDAR 2003 datasets, `http://algoval.essex.ac.uk/icdar`.

[18] Street View Text dataset, `http://vision.ucsd.edu/~kai/svt`.

[19] A. Mishra, K. Alahari, C. V. Jawahar, Scene Text Recognition using Higher Order Language Priors, in: BMVC, 2012.

[20] M. Jaderberg, A. Vedaldi, A. Zisserman, Deep Features for Text Spotting, in: ECCV, 2014.

[21] B. Epshtein, E. Ofek, Y. Wexler, Detecting Text in Natural Scenes with Stroke Width Transform, in: CVPR, 2010.

[22] J. J. Weinman, E. G. Learned-Miller, A. R. Hanson, Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1733–1746.

[23] J. Weinman, Z. Butler, D. Knoll, J. Feild, Toward Integrated Scene Text Reading, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 375–387.

[24] T. Novikova, O. Barinova, P. Kohli, V. S. Lempitsky, Large-Lexicon Attribute-Consistent Text Recognition in Natural Images, in: ECCV, 2012.

[25] J. Rodriguez, F. Perronnin, Label embedding for text recognition, in: BMVC, 2013.

[26] J. L. Feild, E. G. Learned-Miller, Improving Open-Vocabulary Scene Text Recognition, in: ICDAR, 2013.

[27] T. E. de Campos, B. R. Babu, M. Varma, Character Recognition in Natural Images, in: VISAPP, 2009.

[28] D. Chen, J. M. Odobez, H. Bourlard, Text Segmentation and Recognition in Complex Background Based on Markov Random Field, in: ICPR, 2002.

[29] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: CVPR, 2012.

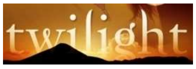
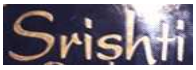
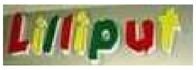
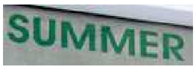
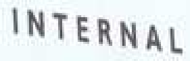
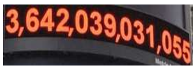
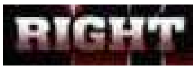
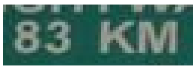
Test Image	Unary	Pairwise	Higher order(=4)
	TWI1IOHT	TWILIOHT	TWILIGHT
	SRISNTI	SRISNTI	SRISHTI
	LIIIIPUT	LIIIIPUT	LILLIPUT
	EUMMER	EUMMER	SUMMER
	IDTERNAL	IDTERNAL	INTERNAL
	364203903105S	3642039031055	3642039031055
	REGHT	REGHT	RIGHT
	83KM	BOKM	BOOM

Figure 7 Results of our higher order model on a few sample images. Characters in red represent incorrect recognition. The unary term alone, based on the SVM classifier, yields poor accuracy, and adding pairwise terms to it improves this. Due to their limited expressiveness, they do not correct all the errors. Higher order potentials capture larger context from the English language, and help address this issue. Note that our method also deals with non-dictionary words (e.g., second row) and non-horizontal text (sixth row). A typical failure case containing alphanumeric words is shown in the last row. (**Best viewed in colour**).

[30] A. Bissacco, M. Cummins, Y. Netzer, H. Neven, PhotoOCR: Reading Text in Uncontrolled Conditions, in: ICCV, 2013.

[31] C. Yao, X. Bai, B. Shi, W. Liu, Strokelets: A Learned Multi-scale Representation for Scene Text Recognition, in: CVPR, 2014.

[32] G. Nagy, Twenty Years of Document Image Analysis in PAMI, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 38–62.

[33] X. Chen, A. L. Yuille, Detecting and Reading Text in Natural Scenes, in: CVPR, 2004.

[34] Q. Ye, D. Doermann, Text Detection and Recognition in Imagery: A survey, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2014) 1–20.

[35] T. Hong, J. J. Hull, Visual Inter-Word Relations and their Use in OCR Postprocessing, in: ICDAR, 1995.

[36] P. Shivakumara, S. Bhowmick, B. Su, C. L. Tan, U. Pal, A New Gradient Based Character Segmentation Method for Video Text Recognition, in: ICDAR, 2011.

[37] A. Mishra, K. Alahari, C. V. Jawahar, Top-Down and Bottom-Up Cues for Scene Text Recognition, in: CVPR, 2012.

[38] K. Rayner, A. Pollatsek, The psychology of reading, Routledge, 1989.

[39] X. Tong, D. A. Evans, A statistical approach to automatic OCR error correction in context, in: Workshop on very large corpora, 1996.

[40] R. Beaufort, C. Mancas-Thillou, A Weighted Finite-State Framework for Correcting Errors in Natural Scene OCR, in: ICDAR, 2007.

[41] C. Thillou, S. Ferreira, B. Gosselin, An embedded application for degraded text recognition, EURASIP Journal on applied signal processing (2005) 2127–2135.

[42] K. Elagouni, C. Garcia, P. Sébillot, A comprehensive neural-based approach for text recognition in videos using natural language processing, in: ICMR, 2011.

[43] K. Elagouni, C. Garcia, F. Mamalet, P. Sébillot, Combining multi-scale character recognition and linguistic knowledge for natural scene text OCR, in: DAS, 2012.

[44] T. Wang, D. Wu, A. Coates, A. Ng, End-to-End Text Recognition with Convolutional Neural Networks, in: ICPR, 2012.

[45] V. Goel, A. Mishra, K. Alahari, C. V. Jawahar, Whole is Greater than Sum of Parts: Recognizing Scene Text Words, in: ICDAR, 2013.

[46] J. J. Weinman, E. G. Learned-Miller, A. R. Hanson, A discriminative semi-Markov model for robust Scene Text Recognition, in: ICPR, 2008.

[47] A. Mishra, K. Alahari, C. V. Jawahar, An MRF Model for Binarization of Natural Scene Text, in: ICDAR, 2011.

[48] N. R. Howe, S. Feng, R. Manmatha, Finding words in alphabet soup: Inference on freeform character recognition for historical scripts, Pattern Recognition 42 (12) (2009) 3338–3347.

[49] L. Neumann, J. Matas, On Combining Multiple Segmentations in Scene Text Recognition, in: ICDAR, 2013.

[50] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, V. S. Lempitsky, Image Binarization for End-to-End Text Understanding in Natural Images, in: ICDAR, 2013.

[51] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, CoRR abs/1412.1842.

[52] W. Huang, Y. Qiao, X. Tang, Robust scene text detection with convolution neural network induced mser trees, in: ECCV, 2014.

[53] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, Z. Zhang, Scene Text Recognition Using Part-Based Tree-Structured Character Detection, in: CVPR, 2013.

[54] J. Almazán, A. Gordo, A. Fornés, E. Valveny, Word spotting and recognition with embedded attributes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (12) (2014) 2552–2566.

[55] U. Roy, A. Mishra, K. Alahari, C. V. Jawahar, Scene text recognition and retrieval for large lexicons, in: ACCV, 2014.

[56] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, CVPR.

[57] J. J. Tompson, A. Jain, Y. Lecun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, in: NIPS, 2014.

[58] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Deep

structured output learning for unconstrained text recognition, CoRR abs/1412.5903.

[59] P. A. Viola, M. J. Jones, Robust Real-Time Face Detection, Int'l J. Computer Vision 57 (2) (2004) 137–154.

[60] A. Bianne-Bernard, F. Menasri, R. A. Mohamad, C. Mokbel, C. Kermorvant, L. Likforman-Sulem, Dynamic and contextual information in HMM modeling for handwritten word recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (10) (2011) 2066–2080.

[61] P. Simard, B. Victorri, Y. LeCun, J. S. Denker, Tangent Prop - A Formalism for Specifying Selected Invariances in an Adaptive Network, in: NIPS, 1991.

[62] M. Mozer, M. I. Jordan, T. Petsche, Improving the accuracy and speed of support vector machines, in: NIPS, 1997.

[63] A. Vedaldi, A. Zisserman, Efficient Additive Kernels via Explicit Feature Maps, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 480–492.

[64] J. C. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, in: Advances in Large Margin Classifiers, MIT Press, 1999.

[65] R. Smith, Limits on the application of frequency-based language models to OCR, in: ICDAR, 2011.

[66] E. M. Riseman, A. R. Hanson, A Contextual Postprocessing System for Error Correction Using Binary n-Grams, IEEE Trans. Comput. (1974) 480–493.

[67] C. Russell, L. Ladicky, P. Kohli, P. H. S. Torr, Exact and Approximate Inference in Associative Hierarchical Networks using Graph Cuts, in: UAI, 2010.

[68] S. M. Katz, Estimation of probabilities from sparse data for the language model component of a speech recognizer, in: IEEE Trans. Acoustics, Speech and Singal processing, Vol. ASSP-35, 1987, pp. 400–401.

[69] J. T. Goodman, A bit of progress in language modeling, Tech. rep., Microsoft Research (2001).

[70] V. Kolmogorov, Convergent Tree-Reweighted Message Passing for Energy Minimization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1568–1583.

[71] J. Pearl, Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference, Morgan Kauffman, 1988.

[72] K. Wang, S. Belongie, Word spotting in the wild, in: ECCV, 2010.

[73] http://cvit.iiit.ac.in/projects/SceneTextUnderstanding.

[74] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, P. P. Roy, ICDAR 2011 Robust Reading Competition - Challenge 1: Reading Text in Born-Digital Images (Web and Email), in: ICDAR, 2011.

[75] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazán, L. de las Heras, ICDAR 2013 robust reading competition, in: ICDAR, 2013.

[76] K. Sheshadri, S. K. Divvala, Exemplar Driven Character Recognition in the Wild, in: BMVC, 2012.

[77] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, A. Y. Ng, Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning, in: ICDAR, 2011.

[78] ABBYY Finereader 9.0, http://www.abbyy.com/.

[79] L. G. i Bigorda, D. Karatzas, Scene text recognition: No country for old men?, in: ACCV Workshops, 2014, pp. 157–168.

[80] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) challenge, Int'l J. Computer Vision 88 (2) (2010) 303–338.

[81] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: BMVC, 2002.

[82] D. Kumar, M. N. A. Prasad, A. G. Ramakrishnan, NESP: nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images, in: DRR, 2013.

[83] D. Kumar, M. N. A. Prasad, A. G. Ramakrishnan, MAPS: midline analysis and propagation of segmentation, in: ICVGIP, 2012.

[84] D. Kumar, A. G. Ramakrishnan, Power-law transformation for enhanced recognition of born-digital word images, in: SPCOM, 2012.

[85] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst., Man, Cybern. 9 (1) (1979) 62–66.

[86] A. Kornai, Language models: where are the bottlenecks?, AISB Quarterly 88 (1994) 36–40.