Strokelets: A Learned Multi-Scale Mid-Level Representation for Scene Text Recognition

Xiang Bai, *Senior Member, IEEE*, Cong Yao, and Wenyu Liu, *Senior Member, IEEE*

***Abstract*— In this paper, we are concerned with the problem of automatic scene text recognition, which involves localizing and reading characters in natural images. We investigate this problem from the perspective of representation and propose a novel multi-scale representation, which leads to accurate, robust character identification and recognition. This representation consists of a set of mid-level primitives, termed strokelets, which capture the underlying substructures of characters at different granularities. The Strokelets possess four distinctive advantages: 1) usability: automatically learned from character level annotations; 2) robustness: insensitive to interference factors; 3) generality: applicable to variant languages; and 4) expressivity: effective at describing characters. Extensive experiments on standard benchmarks verify the advantages of the strokelets and demonstrate the effectiveness of the text recognition algorithm built upon the strokelets. Moreover, we show the method to incorporate the strokelets to improve the performance of scene text detection.**

***Index Terms*— Scene text recognition, scene text detection, mid-level representation, multi-scale representation, natural images.**

I. Introduction

WRITING, considered as a hallmark of civilization [1], is one of the greatest inventions of humanity. Text, as the consequence of writing, has played an irreplaceably important role in almost every aspect of human life, from ancient times to nowadays. Due to its huge significance and utility, text is nearly ubiquitous, especially in modern urban environments. For example, posters, product tags, licence plates, electronic signs, guideposts and billboards, all contain text, probably in different forms.

The rich and precise semantics embodied in text are usually complementary to low level cues (e.g., color, texture and edge) and high level concepts (e.g., object, scene and event), thus can be very beneficial to a variety of applications, for instance, image understanding [2], video indexing [3], product search [4], target geo-location [5], robot navigation [6] and industrial automation [7]. Moreover, the popularization of cameras (including surveillance cameras, consumer cameras, smartphone cameras, etc.) and development of the Internet (including both the traditional Internet and mobile Internet) are leading to continuous, rapid growth of images and videos containing text, which opens doors for new opportunities and possibilities. Therefore, automatic text detection and recognition, providing a means to access textual information in images and videos, have become active research topics in computer vision.

However, localizing and reading text in natural scenes are extremely difficult for computers. Though considerable progresses have been achieved in recent years [8]–[16], detecting and recognizing text in uncontrolled environments are still open problems. Various interference factors, such as variation (e.g., changes in character size, color and font), distortion, noise, blur, non-uniform illumination, local distractor (e.g., non-text objects) and complex background, all may pose big challenges [14], [17].

How to tackle these challenges? We believe representation is the key and core component of the whole solution. Excellent representations should be able to effectively describe the characteristics of characters in natural scenes and meanwhile to robustly overcome the impacts of interference factors.

In this work, we are concerned with the problem of text recognition in natural images (a.k.a. scene text recognition) and propose a novel multi-scale representation. This representation consists of a set of mid-level primitives, termed as *strokelets*, each of which under ideal conditions represents a specific stroke shape. As a multi-scale representation, strokelets capture the substructures of characters at different granularities.

In particular, strokelets possess four distinctive advantages over conventional representations, which are called the "URGE" properties:

- *Usability:* automatically learned from character level bounding boxes, not requiring heavy supervision or detailed annotations.
- *Robustness:* insensitive to various interference factors, endowing the text recognition system based on strokelets with the ability to deal with real-world complexity.
- *Generality:* applicable to characters of different languages, as long as sufficient training examples are available.
- *Expressivity:* effective at describing the properties of characters in natural scenes, bringing high recognition accuracy.

A subset of learned strokelets and several character recognition examples by a system operating on those strokelets are demonstrated in Fig. 1. Strokelets, as a universal representation

Manuscript received May 3, 2015; revised December 14, 2015 and February 14, 2016; accepted April 3, 2016. Date of publication April 15, 2016; date of current version April 29, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61222308, Grant 61572207, and Grant 61573160, and in part by the Open Project Program of the State Key Laboratory of Digital Publishing Technology under Grant F2016001. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter Tay. *(Corresponding author: Cong Yao.)*

The authors are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: xbai@hust.edu.cn; yaocong2010@gmail.com; liuwy@hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2555080

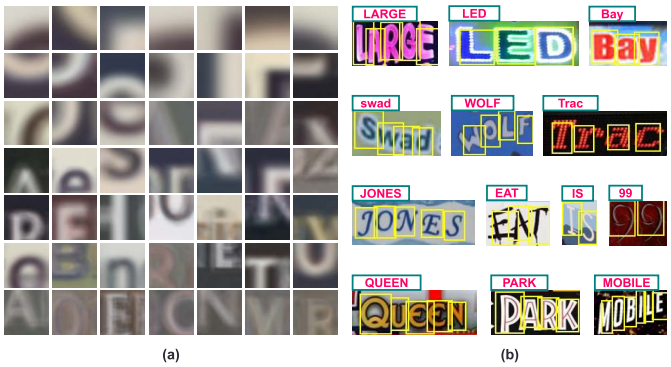

Fig. 1. Illustration of strokelets and character recognition. (a) Strokelets learned on IIIT 5K-Word [8]. Strokelets capture the structural characteristics of characters at multiple scales, ranging from local primitives, like bar, arc and corner (top), to whole characters (bottom). (b) Character recognition examples. Strokelets produce accurate character identification and recognition.

for characters, faithfully seize the representative parts of characters at multiple scales; and characters in different fonts, scales, colors, and layouts can be successfully localized and read, even with the presence of noise and local distractor.

Character identification,[1] the process of hunting each individual character and estimating the position and extent of these characters, is a critical stage in text recognition, as it constitutes the basis of subsequent feature computation, character classification and error correction. In this sense, the quality of character identification largely determines the accuracy of text recognition. However, this stage is very prone to failures, since numerous factors, such as noise, blur, shadow, unusual layout, local distractor and connected characters, may result in errors.

To address these issues, several approaches were proposed, which employed adaptive binarization [14], [18], connected component extraction [12], [13] or sliding window based character detection [8], [10], [19], [20]. These methods work well in certain cases, but are still far from producing all satisfactory results. Binarization is sensitive to noise, non-uniform illumination and local distractor. Connected component extraction is unable to handle broken strokes and connected characters, while sliding window based character detection cannot handle significant variation in character aspect ratio and may produce plenty of false alarms.

The learned strokelets memorize the relative positions and dimensions of characters in the training phase, which can be used to estimate these attributes of characters in test images at runtime. Taking advantage of this property, we propose an alternative strategy for character identification. Different from the aforementioned approaches, the proposed strategy accomplishes the task of character identification via multi-scale strokelet detection and Hough voting [21]. This strategy provides more accurate character localization, produces fewer false alarms, and meanwhile is more robust to interference factors, such as font variation, noise and non-uniform illumination.

[1]We intentionally avoid the term "character detection" as certain algorithms (such as [14], [18]) utilize binarization to seek character candidates.

Moreover, detection activations of strokelets compose a histogram feature, similar to Bag of Words [22] and Bag of Parts [23], which provides extra discriminative power for character classification. Based on strokelets, we devise an effective algorithm for scene text recognition, which achieves higher recognition rate than existing systems.

In our previous work [24], we have presented the main idea of learning strokelets from training data and applied them to scene text recognition. This paper extends that article with the following modifications and contributions: (1) An improved scheme for character identification (Sec. IV); (2) A scene text detection system based on strokelets (Sec. V); (3) More technical details, experimental comparisons and quantitative analyses.

To evaluate the proposed representation as well as the text detection and recognition algorithms, we have conducted extensive experiments on standard benchmarks for scene text detection and recognition, including the challenging public datasets ICDAR 2003 [25], SVT [26] and IIIT 5K-Word [8]. The experiments verify the advantages of strokelets and demonstrate the effectiveness of the proposed text detection and recognition algorithms.

In summary, the major contributions of this paper are: (1) This is the first work that introduces discriminatively trained mid-level elements into the area of scene text detection and recognition; (2) We propose a novel multi-scale representation (named as strokelets), which captures the substructures of characters at different granularities, produces robust character identification, and yields accurate character classification; (3) Based on strokelets, we construct scene text detection and recognition algorithms, which achieve state-of-the-art or highly competitive performances on various standard benchmarks.

The rest of the paper is structured as follows. Sec. II briefly reviews related works in this field. In Sec. III, we describe in detail the procedure of strokelet generation. The text recognition and detection algorithms based on strokelets are presented in Sec. IV and Sec. V, respectively. Sec. VI provides experimental results, comparisons and analyses. Conclusion remarks and future works are given in Sec. VII.

II. Related Work

In recent years, the computer vision community and document analysis community have witnessed a surge in research efforts in the area of text detection and recognition in natural scenes. Consequently, a rich body of novel and inspiring works have emerged [9], [10], [12], [14], [27]–[33].

Neumann and Matas [27] and Epshtein *et al.* [9] proposed powerful text detectors based on MSER (Maximally Stable Extremal Region) and SWT (Stroke Width Transform), respectively, which popularized connected component based methods and inspired a lot of subsequent works [11], [34]–[37].

Wang *et al.* [10], [26] used HOG (Histograms of Oriented Gradients) templates [38] to match character instances in test images with training examples. Neumann and Matas [12] extracted extremal regions as building blocks to localize and recognize characters. Weinman *et al.* [14] proposed to integrate character segmentation and recognition.

Rodriguez-Serrano and Perronnin [31] explored a new way for text recognition, in which label embedding was utilized to directly perform matching between strings and images, bypassing pre- or post-processing operations.

Part based methods [19], [29], [39] have been very popular in this field. Shi *et al.* [19] described a part-based model, employing DPM (Deformable Part Model) [40] and CRF (Conditional Random Field) [41], for scene text recognition. However, the structure of character models and parts of each character class were manually designed and labeled. In [29], Yildirim *et al.* developed a part-based algorithm which adopted multi-class Hough Forest to detect and recognize characters in natural images. Neumann and Matas [39] introduced an approach combining the advantages of sliding window and connected component methods, in which character parts (strokes) are modelled by oriented bar filters. The parts of [19], [29], and [39] are essentially single-scale representation, though the multi-scale scanning strategy was adopted. In contrast, the proposed representation is automatically inferred from training data and represents character parts at multiple scales.

The proposed representation is mainly inspired by the renewed trend of automatically learning mid-level representation for detection and recognition [42]–[44]. Singh *et al.* [42] presented a discriminative clustering approach for discovering mid-level patches. In their work, a set of representative patch clusters are automatically learned from a large image database for scene classification. Lim *et al.* [43] proposed a novel approach to learn local edge-based mid-level features, called sketch tokens, by clustering patches of human generated contours. In this paper, we learn a set of multi-scale mid-level part prototypes to represent characters. Activations of such part prototypes compose a histogram feature, akin to Bag of Words [22] and Bag of Parts [23].

The great success of deep learning methods in various computer vision tasks [45]–[49] has enlightened researchers in the area of scene text detection and recognition. Coates *et al.* [50] and Wang *et al.* [51] used CNN (Convolutional Neural Network) with unsupervised pre-training for text detection and character recognition. Bissacco *et al.* [52] built a system using a DNN (Deep Neural Network) running on HOG features, which is able to read characters in uncontrolled conditions. Jaderberg *et al.* [53] proposed a new CNN architecture, which allows feature sharing for character detection, character classification and bigram classification. Similar to these deep learning methods, our work also adopts the idea of learning representation from data, but it explores a different way and learns multi-scale part prototypes, rather than single-scale global templates.

The presented work is complementary to a line of research efforts on error correction [8], [13], [28]. Novikova *et al.* [13] proposed a unified probabilistic framework, which utilized Weighted Finite-State Transducers [54] to simultaneously introduce language prior and enforce attribute consistency within hypotheses. Mishra *et al.* [28] constructed a CRF model to impose both bottom-up (i.e. character detections) and top-down (i.e. language statistics) cues. In [8], Mishra *et al.* extended this model by inducing higher order language priors. These methods were built upon existing modules for character identification (e.g. MSER extraction or sliding window) and description (e.g. HOG templates). According to [8], replacing such modules with those based on strokelets, these methods could attain better performance.

III. Strokelet Generation

Given a set of training images $S = \{(I_i, B_i)\}_{i=1}^{n}$ containing characters, where I_i is an image and B_i is a set of bounding boxes specifying the location and extent of the characters in the image I_i, the goal of strokelet generation is to learn a set of universal part prototypes Ω from the training set S. The part prototypes should be able to capture the essential sub-structures of characters and be distinctive from local background and against each other.

As S only provides character level annotations, the part prototypes should be automatically discovered. The newly developed discriminative clustering algorithm proposed by Singh *et al.* [42] meets the requirements well, since it learns visual primitives that are both representative and discriminative from large image collections in an unsupervised manner. In this paper, we adopt this algorithm to learn the strokelet set Ω from S.

Given a "discovery" image set $\mathcal{D}$ and a "natural world" image set $\mathcal{N}$, the algorithm of Singh *et al.* [42] aims at discovering a set of representative patch clusters that are discriminative against other clusters in $\mathcal{D}$, as well as the rest of visual world modelled by $\mathcal{N}$. The algorithm is an iterative procedure which alternates between two phases: clustering and training.

The output of the algorithm is a set of top-ranked patch clusters K and a set of classifiers C. Each cluster K_j corresponds to a classifier C_j that can detect patches similar to those in the cluster K_j in novel images. For more details, please refer to [42]. In this work, the clusters K are learned part prototypes of characters and the classifiers C will serve as part detectors at runtime.

The algorithm of Singh *et al.* [42] was originally designed for discovering discriminative patches from generic natural images. To adopt it to learn part prototypes (strokelets) for characters, we made the following customizations:

- The regions within the bounding boxes B constitute the discovery set $\mathcal{D}$ as we aim to discover discriminative parts for characters. The remaining regions of the training images are taken as the natural world set $\mathcal{N}$.
- To learn multi-scale parts for characters, the training examples (patches) are randomly drawn from the discovery set $\mathcal{D}$. The scales of these patches (following [42], we also use square patches, i.e. the width w and height h are equal and $w = h = s$) are random and proportional to the scale of the bounding box bb. The scale of a specific patch is $s = r \cdot max(w(bb), h(bb))$. The ratio r is a random variable in the interval $[a, b]$ and $0 < a \leq b \leq 1$. a and b control the scale of the learned strokelets. If $a = b$, single-scale strokelets will be generated.
- To make the learned strokelets robust to interference factors from local background, we also randomly draw

Algorithm 1 Procedure of Strokelet Generation

Require: Training set S, interval $[a, b]$, strokelet count Γ

$\{\mathcal{D}, \mathcal{N}\} \Leftarrow construct(S)$ ▷ Construct Discovery set $\mathcal{D}$ and Natural World set $\mathcal{N}$ from S

$\mathcal{D} \Rightarrow \{D_1, D_2\};\ \mathcal{N} \Rightarrow \{N_1, N_2\}$ ▷ Split $\mathcal{D}$ and $\mathcal{N}$ into equal sized disjoint subsets

$R \Leftarrow random\_sample(D_1, [a, b])$ ▷ Sample patches with scale ratio randomly drawn from $[a, b]$

$K \Leftarrow cluster(R, \lambda\Gamma)$ ▷ Cluster sampled patches, the initial cluster number is set to $\lambda\Gamma$ $(\lambda > 1)$

repeat ▷ Iterate until convergence

  for all i such that $size(K[i]) \geq \tau$ **do** ▷ Maintain clusters with enough members, τ is a predefined threshold

    $C_{new}[i] \Leftarrow train(K[i], N_1)$ ▷ Train classifier for each cluster

    $K_{new}[i] \Leftarrow detect\_top(C[i], D_2, q)$ ▷ Find top q new members in the other discovery subset

  end for

  $K \Leftarrow K_{new};\ C \Leftarrow C_{new}$ ▷ Update clusters and classifiers

  $swap(D_1, D_2);\ swap(N_1, N_2)$ ▷ Swap the two subsets

until converged

$A[i] \Leftarrow score(K[i])\ \forall i$ ▷ Compute score for each cluster, see [42] for details

$\Omega \Leftarrow select\_top(K, C, A, \Gamma)$ ▷ Sort according to scores and select top Γ clusters and classifiers

return Ω

examples (patches) from the natural world set $\mathcal{N}$ at different scales.

- At the initial clustering stage, each patch p_k from the discover set is represented by a scale and location augmented descriptor, which is the concatenation of the appearance descriptor $d(p_k)$, the relative scale r and the normalized coordinates (x_{p_k}, y_{p_k}), following [55]. This forces the patches in each cluster to be compact in configuration space.
- The SVM (Support Vector Machine) classifier used in [42] was replaced by Random Forest [56] because the latter can achieve similarly high accuracy as SVM and directly gives probabilities, which are more intuitive and interpretable.
- The size of the patch descriptors (HOG [38]) is 3×3 (rather than 8×8) cells as they are sufficient for describing character parts.

The whole procedure for learning strokelets is summarized in Algorithm 1. The learned strokelet set can be expressed as $\Omega = \{(K_j, C_j)\}_{j=1}^{\Gamma}$, where K and C are the discovered part prototypes and corresponding classifiers respectively, and Γ is the size of the strokelet set. For each cluster K_j, the following information is stored: The set of all the members (patches) M_j, their offset vectors to character center V_j, and the average width $\bar{w}_j$ and height $\bar{h}_j$ of the parent rectangles, from which the members M_j originate. V_j, $\bar{w}_j$ and $\bar{h}_j$[2] will be used to estimate the location and extent of characters in the character identification stage (see Sec. IV-A).

Fig. 2 depicts the strokelets (classifiers not shown) learned on the IIIT 5K-Word dataset [8]. As can be seen, strokelets, as a universal multi-scale representation, describe part prototypes of characters at different granularities, ranging from simple micro-structures to entire characters. Moreover, they are able to capture the parts that are common across different character classes (see the top rows of Fig. 2 (b)) as well as those unique to certain character classes (see the bottom row of Fig. 2 (b)).

[2]We assume that V_j, $\bar{w}_j$ and $\bar{h}_j$ have been normalized with respect to the members M_j.

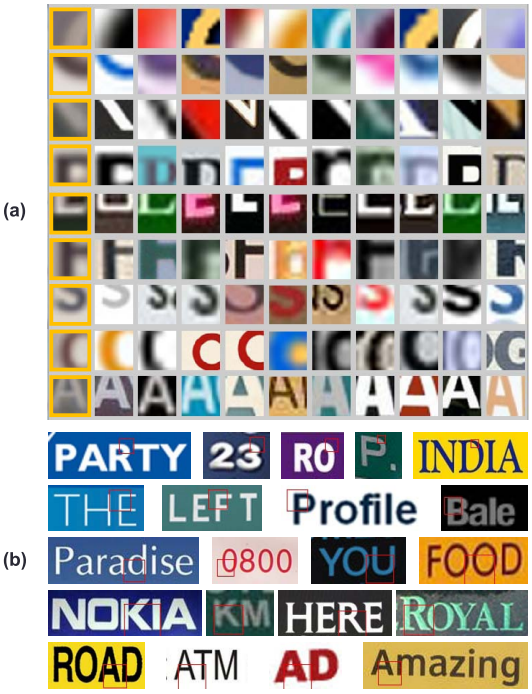

Fig. 2. Learned strokelets on the IIIT 5K-Word dataset [8]. (a) Each row illustrates a cluster of part instances that constitute a strokelet. The images in the first column (orange rectangle) are the average of all the instances of that strokelet. The rest are top-ranked part instances. (b) Discovered part instances in original images. The learned part prototypes are tightly clustered in both appearance and configuration space.

In principle, strokelets are an over-complete representation, but this is not guaranteed in reality, because of the greedy pursuit strategy in strokelet generation and the limited diversity in training data. However, the learned strokelets are sufficient for the task of text recognition and work well in practice (see Sec. VI).

Strokelets are by construction detectable primitives, as they are generated via discriminative learning. Moreover, the learned strokelets are tightly clustered in both appearance and configuration space (see Fig. 2 (b)). These properties make

strokelets closely analogous to poselets [57], [58]. However, different from poselets, which are obtained using manually labeled data (part regions and keypoints), strokelets are automatically learned using character level annotations.

IV. Recognition Algorithm

The algorithmic pipeline for scene text recognition is fairly straightforward: Character candidates are first sought from the image via a voting based scheme for character identification (Sec. IV-A); these candidates are then described by a histogram feature based on strokelets and a holistic descriptor (Sec. IV-B); and character classification is applied to assign the most probable class label to each character (Sec. IV-C). Optionally, the inferred word is replaced by the most similar item in a given dictionary, following [10], [28].

The algorithm described above is quite effective, even though without sophisticated approaches to error correction [8], [13]. We believe better performance could be achieved if such error correction methods are incorporated.

A. Character Identification

As stated in Sec. I, character identification is a key stage in scene text recognition. However, binarization based methods [14], [18] are sensitive to noise, blur and non-uniform illumination; connected component based methods [12], [13] are unable to handle connected characters and broken strokes; and sliding window based character detection based methods [8], [26] usually produce a lot of false alarms, mainly due to varying aspect ratios of characters and background clutters.

In our previous work [24], we proposed a novel scheme to seek character candidates, via multi-scale strokelet detection and voting. The scheme shares the idea of estimating character centers through voting with the work of Yildirim *et al.* [29]. The work in [29] is essentially a patch based method, which does not explicitly infer character parts, but simply learns the mapping relations (multi-class Hough Forests) between local patches and character center; besides, it only performs voting at single scale, though multi-scale scanning is used during character detection. In contrast, the strategy in [24] casts votes from multiple scales.

Firstly, the original word image (Fig. 3 (a)) is resized to a standard height (64 pixels in this paper) with aspect ratio kept unchanged; since strokelets are naturally multi-scale representation, a multi-scale sliding-window paradigm is performed to detect strokelets (Fig. 3 (b)); a Hough map (Fig. 3 (c)) is then generated by casting and accumulating the votes from the strokelets activations, similar to [59] and [21]; finally, the centers of the character candidates are found by seeking maxima in the Hough map using Mean Shift [60] and the extents of these candidates are determined by computing the weighted average of the attributes of the clusters (average width $\bar{w}_j$ and height $\bar{h}_j$), which have been stored in the training phase (Sec. III).

The number of character centers is estimated before the Mean Shift process. Specifically, the character number is calculated using the aspect ratio of the image. In the training phase, we partition the aspect ratios (width over height) of the training images into 24 discrete bins (e.g., 0.5, 1, 1.5...), and compute the average word count for each aspect ratio bin. In the testing phase, the initial number of character center is set to 1.5 times of the average word count of the nearest aspect ratio bin. After the Mean Shift process, overlapping character candidates are merged if the overlap ratio (intersection over union) between them is larger than 0.5.

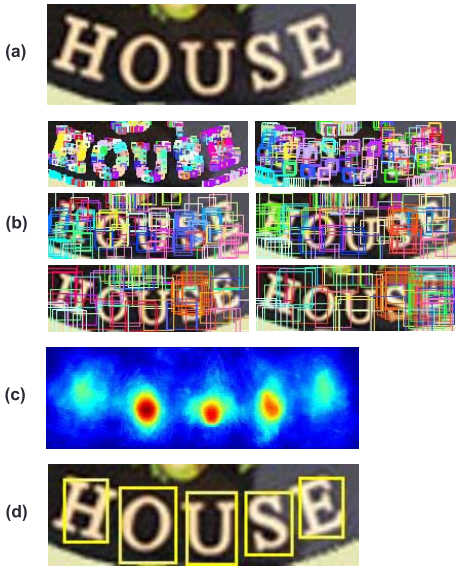

Fig. 3. Character identification. (a) Original image. (b) Detections of strokelets at different scales. Activations of different types of strokelets are marked in different colors. For better visualization, the images are rescaled and non-maximum suppression is applied to the activation windows. (c) Hough map. (d) Identified characters. Different from [28], non-maximum suppression for false alarm removal is not a tough task in our work, as multi-scale strokelet detection and voting generate high-quality Hough maps.

It has been proven in [24] that the above described scheme provides accurate character identification and is robust to various interference factors. However, it may produce spurious characters in certain situations. For example, it may merge two adjacent characters into one (note the image in the first column of Fig. 5 (c)) or split a character into two separate pieces (note the images in the second and third column of Fig. 5 (c)). Such spurious characters will give rise to errors in both character localization and classification, thus bringing down the final recognition rate.

Upon investigation, we figured out that the reason for such wrongly identified candidates stems from Hough voting. In the original Hough voting step, each strokelet activation blindly votes for character centers, totally ignoring the influence of other strokelet activations, which makes it impossible to distinguish between true votes (votes that are close to character centers) and spurious votes (votes that deviate from character centers), as shown in Fig. 4.

An intuitive strategy for solving this problem is to exploit the interactions between different strokelet activations, since such high order relationships (e.g. pairwise co-occurrence) conduce to disambiguation in Hough voting. However, this strategy would cause combinatorial explosion at runtime, because there are typically hundreds of types of strokelets in the strokelet set Ω and millions of strokelet activations in a

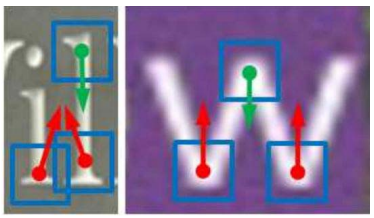

Fig. 4. True and spurious votes in Hough voting. Blue rectangles: strokelet activations; Green arrows: true votes; Red arrows: spurious votes.

single image. Therefore, we introduce in this paper a simple yet effective scheme for character identification.

In [24], each strokelet activation gives its vote according to the offset vector set of the corresponding strokelet type, V_j. V_j consists of a group of offset vectors, i.e., $V_j = \{v_t\}_{t=1}^{o}$, where o indicates the number of the offset vectors. In the scheme of [24], all the offset vectors in V_j are treated equally and no weight is assigned to each offset vector v_t.

However, in the voting process a portion of the offset vectors contribute positively (i.e. producing true votes) while others might contribute negatively (i.e. producing spurious votes). To enhance the effect of the offset vectors that contribute positively and suppress that of the offset vectors that contribute negatively, a weight ω_t is assigned to each offset vector v_t. ω_t is determined as follows: We perform multi-scale strokelet detection and voting in all the training images and calculate for each offset vector the ratio of true votes. This ratio reflects the relative contribution of each offset vector v_t and thus serves as the weight ω_t. The offset vector set becomes:

$$V_j = \{v_t, \omega_t\}_{t=1}^{o}. \tag{1}$$

The procedure of computing ω_t is as follows: We perform multi-scale strokelet detection and voting in all the training images and calculate for each offset vector the ratio of true votes. For each offset vector v_t, we count the total number of votes by v_t (denoted as P_t), and the number of true votes (denoted as Q_t). If the distance between the endpoint of a offset vector and a character center is less than a quarter of the smaller dimension of the character, it is considered as a true vote. The weight ω_t is defined as: $\omega_t = Q_t / P_t$.

In the testing phase, the improved scheme for character identification performs weighted Hough voting, which introduces negligible additional computational cost, compared to the original scheme. The improved scheme produces higher-quality Hough maps and thus better character identification results, as shown in Fig. 5 (e). It is able to avoid merging thin characters into one spurious candidate or splitting wide characters into two.

The experiments in Sec. VI confirm that the improved scheme for character identification provides more accurate and reliable character localization, leading to much higher recognition performance. Moreover, the modification we made to the original Hough voting is general, so it can be adopted to improve other Hough voting based object detection methods [21], [61].

The procedure for character extent (width and height) estimation is the same as in [24]. For a character candidate α,

Fig. 5. Improved scheme for character identification. (a) Original image. (b) Hough map produced by the scheme in [24]. (c) Characters identified by the scheme in [24]. (d) Hough map produced by the improved scheme. (e) Characters identified by the improved scheme.

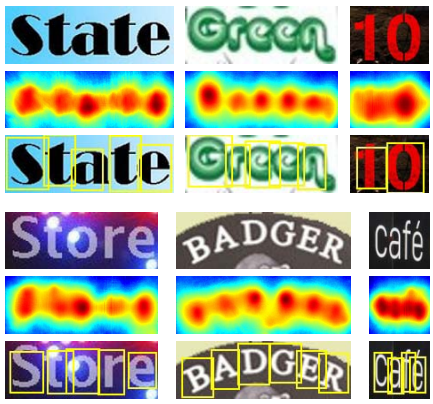

Fig. 6. Examples of character identification. The improved scheme for character identification is able to hunt characters of different fonts, sizes, colors and layouts with the presence of noise, non-uniform illumination and local distractor.

assume a set of strokelet detections $\{d_l(\alpha)\}_{l=1}^{m}$ have contributed to it, then the width and height of α are calculated as:

$$w(\alpha) = \frac{\sum_{l=1}^{m} \rho(d_l) \cdot w(d_l) \cdot \bar{w}_{d_l}}{\sum_{l=1}^{m} \rho(d_l)}, \tag{2}$$

$$h(\alpha) = \frac{\sum_{l=1}^{m} \rho(d_l) \cdot h(d_l) \cdot \bar{h}_{d_l}}{\sum_{l=1}^{m} \rho(d_l)}, \tag{3}$$

where $\rho(d_l)$ is the detection score of d_l, $w(d_l)$ and $h(d_l)$ stand for the width and height of d_l, and $\bar{w}_{d_l}$ and $\bar{h}_{d_l}$ denote the average width and height of the cluster corresponding to d_l, respectively.

Several examples of character identification by the proposed scheme are shown in Fig. 6. By adopting discriminative training and multi-scale voting, the proposed scheme gives precise estimation of character center and extent, and is capable of handling issues like noise, non-uniform illumination, varying aspect ratios and broken strokes.

B. Character Description

Based on detection activations of strokelets, we introduce a histogram feature called Bag of Strokelets, in addition to the traditional feature HOG [38].

1) Bag of Strokelets: For each identified character candidate, all the strokelets that have voted for it are sought via back-projection. A histogram feature is formed by binning the strokelets. Strokelets of all scales (see Fig. 3 (b)) are assembled together. Each strokelet contributes to the histogram feature according to its detection score. To incorporate spatial information, the Spatial Pyramid strategy [62] (1×1 and 2×2 grids) is also adopted. Among all pooling methods, we found max-pooling [63] gave the highest accuracy, so max-pooling is employed in all the experiments in this paper.

2) HOG: Following [13], [28], we also adopt the HOG descriptor (the version proposed in [40]) to describe characters. A template with 5×7 cells is constructed for each character candidate.

The Bag of Strokelets feature is complementary to HOG, as it conveys information from different levels and is robust to font variation, subtle deformation and partial occlusion. We will evaluate the effectiveness of these two types of features and compare their contributions to recognition accuracy in Sec. VI-B6.

C. Character Classification

In this paper, we consider English letters (52 classes) and Arabic numbers (10 classes), i.e. the alphabet $\Phi = \{a, \ldots, z; A, \ldots, Z; 0, \ldots, 9\}$ and $|\Phi| = 62$. To handle invalid characters (e.g. punctuations and background components), we also introduce a special class, so there are 63 classes in total.

We train 63 character recognizers (binary classifiers), one for each character class, in a one-vs-all manner. Random Forest [56] is adopted as the strong classifier because of its high performance and efficiency. Training examples are harvested by applying the strokelets to the images in the training set and compare the identified rectangles with the ground truth annotations. At runtime, the character candidates are classified by the trained recognizers; for each character, the class label with the highest probability is assigned as the recognition consequence.

V. Detection Algorithm

It has become an emerging trend in computer vision to adopt representation trained for one task to accomplish other tasks [49], [64]. Donahue *et al.* [64] directly used Krizhevsky's CNN model [46], which was trained on the dataset of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [65], [66], as a feature extractor to perform various recognition tasks, yielding excellent recognition performance. Girshick *et al.* [49] utilized this CNN model to construct a complete system for generic object detection, achieving significant boost in detection accuracy. So here raises an interesting question that if strokelets, as a representation originally learned for text recognition, can act as a feature extractor in other related tasks. We answer this question by constructing a powerful text detection system, based on strokelets.

The proposed text detection system is built upon the algorithm in [67]. Specifically, the Bag of Strokelets feature is used as a descriptor for image regions that may contain text strings (words or lines), similar to T-HOG [68]. A region classification module (again a Random Forest classifier) is trained on the Bag of Strokelets feature using positive and negative examples collected by the system of [67], as shown in Fig. 7. This region classification module is then plugged into the system of [67] to identify and remove non-text regions, before the final non-maximum suppression and thresholding processes. In both training and testing, the regions are rescaled to a standard height (64 pixels), before feature extraction with strokelets. This text detection system has proven to be effective and robust (see Sec. VI-B8).

Fig. 7. Samples of regions for training. (a) Text regions. (b) Non-text regions.

VI. Experiments

We have evaluated the proposed representation, the text detection and recognition algorithms based on strokelets, and compared them with other competing methods, including the leading algorithms in this field. All the experiments were conducted on a regular PC (2.8GHz 8-core CPU, 16G RAM and Windows 64-bit OS).

For all the Random Forest classifiers, 200 trees were used. The windows for strokelet detection were sampled at 12 scales. $a = 0.2$ and $b = 1.0$ for all the experiments unless specifically stated.

A. Datasets

In this section, we introduce in detail the benchmark datasets used for evaluation in this paper.

1) IIIT 5K-Word: The IIIT 5K-Word dataset [8] is the largest and most challenging benchmark in this field to date. This database includes 5000 images with text in both natural scenes and born-digital images. It is challenging because of the variation in font, color, size, layout and the presence of noise, blur, distortion and varying illumination. 2000 images are used for training and 3000 images for testing. This dataset comes with three types of lexicons (small, medium, and large) for each test image.

2) ICDAR 2003: The ICDAR 2003 Robust Word Recognition Competition [25] was held to track the advances in word recognition in natural images. This dataset is widely used in the community to evaluate algorithms for text recognition in cropped images. Following previous works [8], [13], [19], we skipped the words with two or fewer characters, as well as those with non-alphanumeric characters.

3) ICDAR 2011: The ICDAR 2011 dataset [69] is an extension to the dataset used for the text locating competitions of ICDAR 2003 [25] and 2005 [70]. The previous evaluation protocol is replaced by the evaluation method proposed by

TABLE I

IMPACT OF STROKELET COUNT Γ TO WORD RECOGNITION RATE ON THE IIIT 5K-WORD DATASET

Γ	100	200	300	400	500	600	700
Proposed	80.6	83.0	83.3	84.1	**85.6**	84.2	83.9
Strokelets (original) [24]	71.4	75.9	78.1	77.1	**80.2**	79.0	78.3

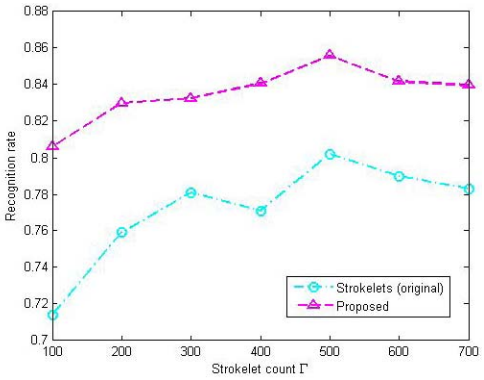

Fig. 8. Impact of strokelet count Γ to recognition rate on the IIIT 5K-Word dataset.

Wolf and Jolion [71], as the latter is able to handle the cases of one-to-many and many-to-many matches. We evaluated and compared the performances of different text detection methods on this dataset.

4) SVT: The Street View Text (SVT) dataset [10], [26] is a collection of outdoor images with scene text of high variability. This dataset can be used for both cropped word recognition and full image word detection and recognition. We adopted the SVT-WORD subset, which contains 647 word images, to evaluate the proposed algorithm.

For fair comparison, the lexicons for the ICDAR 2003 and SVT dataset provided in [10] were also used in this work.

B. Experimental Results and Discussions

1) Impact of Strokelet Count Γ: Strokelet count Γ is a key parameter as it determines the number of learned strokelets. We first investigated the impact of strokelet count to word recognition rate on the IIIT 5K-Word dataset, since it is the largest dataset for character recognition. We believe that the assessment and conclusion acquired on this dataset are more credible than those on small-sized datasets.

As can be seen from Table I and Fig. 8, the recognition rate increases with strokelet count Γ upto a certain point and then slightly decreases. The highest accuracy was achieved with $\Gamma = 500$. In all the following experiments except the experiments in Sec. VI-C, strokelet count Γ is fixed at 500.

The proposed method consistently outperforms the original version [24] by a large margin (more than 5% improvement on average in recognition rate). This demonstrates the effectiveness of the proposed scheme for character identification, since the only difference between the proposed method and [24] is the scheme for character identification.

2) Recognition Results on IIIT 5K-Word: We learned a set of strokelets on the IIIT 5K-Word dataset and evaluated the proposed algorithm on it. The performances (word level recognition rates) of the proposed algorithm and other recently published works are illustrated in Table II. In general, the proposed algorithm outperforms all the conventional methods, but lags behind those based on deep learning [72], [73]. Note that these deep learning based systems utilized tremendous amount of extra data for training, while other methods (including the proposed algorithm) only used the training examples from the IIIT 5K-Word dataset.

TABLE II

PERFORMANCES OF DIFFERENT ALGORITHMS EVALUATED ON THE IIIT 5K-WORD DATASET

Lexicon	Small	Medium	Large
Proposed	85.6	72.7	40.9
Strokelets (original) [24]	80.2	69.3	38.3
Reading Text in the Wild with CNN [72]	**97.1**	**92.7**	-
Deep Structured Output Learning [73]	95.5	89.6	-
Label Embedding [31]	76.1	57.4	-
Higher Order [8](with edit distance)	68.25	55.50	28
Higher Order [8](without edit distance)	64.10	53.16	**44.30**
Pairwise CRF [28](with edit distance)	66	57.5	24.25
Pairwise CRF [28](without edit distance)	55.50	51.25	20.25
ABBYY9.0 [74]	24.33	-	-

With small lexicon, the proposed algorithm achieves a recognition accuracy of 85.6%, which is 9.5% higher than that of the closest competitor Label Embedding [31] (76.1%); with medium lexicon, the improvement (15.3%) is even more notable; with large lexicon, the proposed algorithm is comparable to Higher Order without edit distance, but behind it (40.9% vs 44.3%). This is reasonable as the large lexicon is independent from IIIT 5K-Word[3] and Higher Order [8] without edit distance incorporated statistical language model for error correction. The comparison between the proposed approach and Higher Order with edit distance is much fairer, where the improvement (from 28% to 40.9%) is also very significant.

The proposed method substantially improves upon the original version of strokelets in [24]. The gains in recognition rate under three settings are 5.4%, 3.4% and 2.6%, respectively.

The IIIT 5K-Word dataset is the largest and most challenging benchmark in this field. The comparisons above demonstrate that the proposed representation and text recognition method are both effective and robust. Moreover, the proposed method can be integrated with those of [8] and [28], which will create a more powerful system for scene text recognition.

3) Recognition Results on ICDAR 2003 and SVT: We also applied the learned strokelets to the test images of the ICDAR 2003 and SVT dataset. Fig. 9 shows several character recognition examples on these two datasets. The strokelets trained on the IIIT 5K-Word dataset generalize well to novel images from other databases. The proposed algorithm is able to handle challenges like font variation, scale change, blur, non-uniform illumination and partial occlusion.

[3]This means the large lexicon does not necessarily contain the ground truth words.

TABLE III
PERFORMANCES OF DIFFERENT ALGORITHMS EVALUATED ON THE ICDAR 2003 AND SVT DATASET

Dataset	ICDAR 2003(FULL)	ICDAR 2003(50)	SVT
Proposed	82.64	90.27	80.99
Strokelets (original) [24]	80.33	88.48	75.89
Reading Text in the Wild with CNN [72]	**98.6**	**98.7**	**95.4**
Deep Structured Output Learning [73]	97.0	97.8	93.2
Deep Features [53]	91.5	96.2	86.1
PhotoOCR [52]	-	-	90.39
Discriminative Feature Pooling [20]	76	88	80
CNN [51]	84	90	70
Whole [30]	-	89.69	77.28
TSM+CRF [19]	79.30	87.44	73.51
TSM+PLEX [19]	70.47	80.70	69.51
Multi-Class Hough Forests [29]	-	85.70	-
Large-Lexicon Attribute-Consistent [13]	82.8	-	72.9
Higher Order [8](with edit distance)	-	80.28	73.57
Higher Order [8](without edit distance)	-	72.01	68.00
Pairwise CRF [28](with edit distance)	-	81.78	73.26
Pairwise CRF [28](without edit distance)	-	69.90	62.28
SYNTH+PLEX [10]	62	76	57
ICDAR+PLEX [10]	57	72	56
ABBYY9.0 [74]	55	56	35

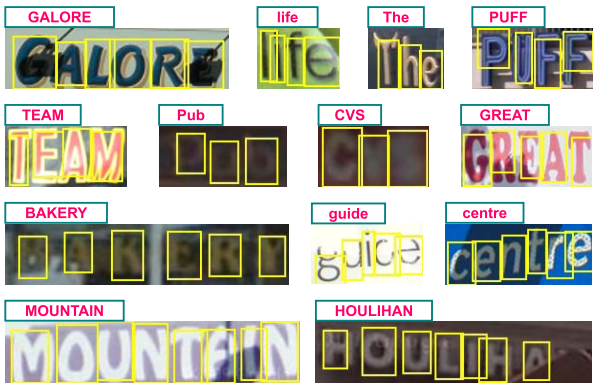

Fig. 9. Examples of character recognition on the ICDAR 2003 and SVT datasets. Though only trained on the IIIT 5K-Word dataset, the strokelets generalize well to the images from ICDAR 2003 and SVT.

The performances of the proposed algorithm as well as other competing methods on the ICDAR 2003 and SVT dataset are depicted in Table III. The proposed algorithm achieves recognition accuracy of 82.64%, 90.27% and 80.99% on ICDAR 2003(FULL), ICDAR 2003(50) and SVT respectively, outperforming the competing methods of [8], [10], [13], [19], [28], and [29], but still behind those in [30] and [51–[53].

CNN based methods [30], [51]–[53], [72], [73], [75] generally achieve much higher performance than other methods, including the proposed algorithm in this paper. However, the excellent performances of these methods depend heavily on large volume of training data. For example, the method in [51] used 12k training images, while the PhotoOCR engine [52] was trained with 2.2 million labelled examples, which are not publicly available. The algorithm in [53] mined about 14k words and 71k characters from Flickr. In contrast, our algorithm is trained using a far smaller training set (2000 images from IIIT 5K-Word). The system in [30] obtains good results on these two datasets, but cannot handle words out of the given dictionaries. Compared to these methods, the proposed algorithm requires less training data and has a broader scope of application.

Though there is an obvious gap between the performance of our approach and deep learning based methods, we still think it novel and valuable, since: (1) If trained with the same training set, the gap in performance will not be so obvious. (2) Deep learning based methods have achieved excellent performance on scene text recognition, but this does not indicate that the research of conventional approaches is insignificant. On the contrary, conventional approaches may provide insightful and inspiring ideas, which can be complementary to the deep learning based methods. (3) This work explores an alternative way for automatically learning multi-scale prototypes of characters. The model is capable of capturing the structural information of characters at different granularities (from local structures to, to strokes and even to whole characters), which produces robust character localization and yields accurate character classification. In contrast, most existing deep learning based methods only perform coarse level (whole characters or words) classification. We argue that exploiting information from different levels is beneficial for character localization and recognition. (4) The main idea of Strokelets is actually quite generic, thus it can be readily combined with deep learning techniques. For instance, we could replace the Random Forest classifiers with CNN classifiers and/or employ deep features instead of HOG features. We believe that in this way the quality of learned

TABLE IV

CHARACTER IDENTIFICATION ACCURACIES OF DIFFERENT ALGORITHMS EVALUATED ON THE IIIT 5K-WORD DATASET

Algorithm	Precision@Recall=78%	Precision@EER
Proposed	**51**	**69**
Strokelets (original) [24]	45	64
Higher Order [8]	17	35

Strokelets and the final recognition performance would be significantly enhanced. This direction will be explored in a future work.

It is worth mentioning that the proposed algorithm is superior to [19], which employed manually designed character models and detailed part annotations. This proves that in character recognition automatically learned part prototypes could work better than those defined and labeled by human.

The performance gains achieved by the proposed method are mainly due to two reasons: (1) Compared to other approaches, strokelets produce more accurate and robust character identification; (2) The proposed Bag of Strokelets feature offers extra discriminative power, further boosting the recognition rate.

4) Effectiveness of Strokelets in Character Identification: We validated the excellent ability of strokelets in character identification on the IIIT 5K-Word dataset. Character identification performance is measured by precision and recall, following the Higher Order algorithm in [8]. Similar to the evaluation protocol of the PASCAL VOC object detection task [76], detections are considered true or false positives based on the overlap ratio (intersection over union) between the predicted rectangles and the ground truth annotations. The character identification accuracy of strokelets and that of Higher Order [8] are shown in Table IV. As can be observed, the original version of strokelets in [24] obtains precision of 45% at recall=78% and precision of 64% at EER (Equal Error Rate), far surpassing [8] (precision≈17% at recall=78% and precision≈35% at EER).[4]

Note that the proposed scheme for character identification is more effective and further enhances the character identification accuracy (precision=51% at recall=78% and precision=69% at EER).

5) Effectiveness of Strokelets in Character Classification: We also validated the excellent ability of strokelets in character classification on the Chars74K [77], ICDAR-CHAR and SVT-CHAR [10] datasets. Different from previous methods, which resized the character images into a canonical size (e.g. 24×24) before feature computation and character classification, we rescale the character images to a standard height (64 pixels) with aspect ratio unchanged and performed character identification and classification in these images, assuming a single character in each image.

The character classification accuracies (case insensitive) of different algorithms are shown in Table V. The original version of strokelets in [24] obtains accuracies of 60%, 67% and 64% on Chars74K [77], ICDAR-CHAR and SVT-CHAR [10], respectively, outperforming the conventional character classification methods such as NATIVE+FERNS, SYNTH+FERNS [10] and HOG+NN [26]. Since the improved scheme for character identification produces more precise estimation of character location and extent, the proposed algorithm in this paper yields higher character classification performance compared to the previous version [24], achieving accuracies of 62%, 69% and 71% on Chars74K, ICDAR-CHAR and SVT-CHAR, respectively. The work in [19] used a different evaluation protocol, thus is not directly comparable.

[4]These two precision values are read from graph in [8], thus are not precise.

TABLE V

CHARACTER CLASSIFICATION ACCURACIES OF DIFFERENT ALGORITHMS EVALUATED ON THE CHARS74K, ICDAR-CHAR AND SVT-CHAR DATASETS

Algorithm	Chars74K	ICDAR-CHAR	SVT-CHAR
Proposed	**62**	**69**	**71**
Strokelets (original) [24]	60	67	64
NATIVE+FERNS [10]	54	64	-
SYNTH+FERNS [10]	47	52	-
HOG+NN [26]	58	52	-
MKL [77]	55	-	-

TABLE VI

PERFORMANCES OF DIFFERENT TYPES OF FEATURES

Feature	Bag of Strokelets	HOG	Bag of Strokelets+HOG
Accuracy(%)	80.7	83.2	**85.6**

TABLE VII

ADVANTAGE OF MULTI-SCALE REPRESENTATION

Scale(a=b)	0.2	0.3	0.4	0.5	0.6	0.7	multi-scale
Accuracy(%)	64.5	75.6	78.4	80.3	80.9	80.5	**85.6**

6) Contributions of Different Types of Features: In addition, we evaluated the effectiveness of the Bag of Strokelets feature and compared it with HOG. We tested three types of features: Bag of Strokelets, HOG, and their concatenation (Bag of Strokelets+HOG). The recognition rates of these three types of features on the IIIT 5K-Word dataset are shown in Table VI. The conventional feature HOG is quite informative, achieving a recognition rate of 83.2%, while that of Bag of Strokelets is 80.7%. These two types of features are indeed complementary. Their combination leads to higher performance (85.6%).

7) Advantage of Multi-Scale Representation: To verify the advantage of multi-scale representation over single-scale representation, we also trained several sets of single-scale strokelets with different scales on the IIIT 5K-Word dataset. The recognition rates of those strokelets as well as multi-scale strokelets are shown in Table VII.

As can be observed, even single-scale strokelets perform fairly well on this challenging benchmark, while multi-scale strokelets bring further improvement. Multi-scale representation, being able to capture the characteristics of characters at different granularities and thus convey more information, performs much better than single-scale representations.

TABLE VIII
PERFORMANCES OF DIFFERENT TEXT DETECTION METHODS EVALUATED ON THE ICDAR 2011 DATASET

Algorithm	Precision	Recall	F-measure
Proposed	85.6	68.8	76.3
Baseline [67]	82.2	65.7	73.0
Huang *et al.* [78]	**88**	**71**	**78**
Yin *et al.* [37]	86.3	68.3	76.2
Neumann *et al.* [79]	85.4	67.5	75.4
Koo *et al.* [35]	81.4	68.7	74.5
Yi *et al.* [80]	67.2	58.1	62.3

TABLE IX
PERFORMANCES OF RANDOM FOREST AND SVM EVALUATED ON THE IIIT 5K-WORD DATASET

Lexicon	Small	Medium	Large
Proposed (RF)	85.6	72.7	40.9
Proposed (SVM)	78.9	66.2	31.5

8) Effectiveness of Strokelets in Text Detection: We tested the proposed text detection system on the ICDAR 2011 dataset [69] and compared it with the baseline method [67] as well as other competing algorithms. The performance of the proposed system as well as that of other methods on this benchmark are shown in Table VIII.

Compared to the baseline method [67], the proposed system achieves significantly enhanced performance (improvements of 3.4%, 3.1% and 3.3% in precision, recall and F-measure, respectively). Note that we applied the strokelets based classification module before non-maximum suppression and thresholding, so the proposed text detection system can obtain improvements in both precision and recall. This comparison demonstrates that strokelets can serve as an effective feature extractor for scene text detection.

The performance of the proposed text detection system outperforms the competing algorithms [37], [79], but is inferior to the current state-of-the-art method of Huang and Tang [78], which incorporated a CNN model trained on tremendous synthesized examples.

9) Choice of Strong Classifier in Character Classification: In the previous experiments, we chose Random Forest as the strong classifier for character classification, mainly because it is easy to implement and leads to high performance and efficiency. In this section, we validate the choice of strong classifier in character classification.

In particular, we have conducted a group of experiments to compare the performance of SVM and Random Forest on the task of scene text recognition. We replaced the Random Forest classifiers (character recognizers) with SVM classifiers, with other parameters kept unchanged. The SVM classifiers are trained and evaluated on the training set and testing set of the IIIT-5K Word dataset, respectively. The settings of the SVM classifiers are as follows: OpenCV 3.0 implementation, RBF kernel, 5 fold cross-validation for parameter search and other parameters set as default.

The performances of SVM as well as Random Forest on the IIIT-5K Word dataset are shown in Table IX. As can be seen, on this specific task Random Forest achieves better results than SVM, in three settings of the IIIT-5K Word benchmark. The main reason is possibly that in this task the training examples are highly imbalanced (e.g. the number of 'a' is much larger than that of 'z') and Random Forest is able to automatically handle this issue, while SVM requires that the class weights being elaborately tuned.

In summary, we think that in general Random Forest is not necessarily better than SVM, but the former is more suitable for this specific task, mainly due to its performance, efficiency and usability.

Fig. 10. Learned strokelets ($\Gamma = 100$) on different languages. (a) Chinese. Original images are from [81]. (b) Korean. Original images are from [82]. (c) Russian. Original images are harvested from the Internet.

C. Generality of Strokelets

The previous qualitative and quantitative results have confirmed the *usability*, *robustness* and *expressivity* properties of strokelets. To verify the *generality* property, we demonstrate three sets of strokelets learned on different languages in Fig. 10.

As we can see, the learned strokelets faithfully reflect the characteristics of the corresponding languages. For example, the strokelets learned on Chinese capture the rich horizontal and vertical structures, while those on Korean additionally highlight the arc structures. Strokelets can be readily applied to other languages, without further tweaking or customization. The only requirement is sufficient training examples. Moreover, in order to cope with multilingual scenarios, we could learn a hybrid set of strokelets on multiple languages.

D. Limitations of Proposed Algorithm

As demonstrated in the above examples and evaluations, the proposed algorithm works fairly well in various real-world situations, it is however far from perfect. It would make mistakes under certain conditions, as depicted in Fig. 11.

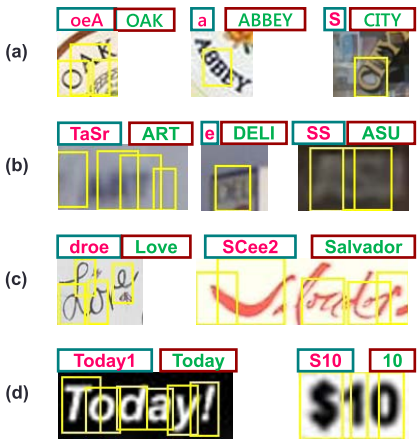

Fig. 11. Typical failure cases of the proposed algorithm. (a) Extreme tilt. (b) Heavy blur. (c) Scribbling. (d) Characters out of the alphabet. Magenta characters: recognition results; Green characters: ground truth.

The proposed algorithm is robust to rotation to some degree, but cannot handle extremely tilted words (Fig. 11 (a)). Heavy blur (Fig. 11 (b)) and scribbling (Fig. 11 (c)) pose major problems to both character identification and classification. Moreover, characters that are out of the given alphabet (e.g., '!' and '$' in Fig. 11 (d)) can be successfully detected by the proposed algorithm, but cannot be correctly recognized.

Another shortcoming of the proposed algorithm lies in processing efficiency. The average speed of the algorithm on the IIIT 5K-Word dataset (cropped images) is about 1fps,[5] which is insufficient to support time-sensitive applications, such as instant translation. As a reference, the mean processing time of PhotoOCR [52] is 600ms per full image. However, it is possible to deliver low-latency services, if we implement the proposed algorithm in high-performance computing clusters (just like PhotoOCR).

VII. Conclusions and Future Work

We have introduced strokelets, a novel multi-scale representation for characters in natural scenes. Strokelets are automatically learned merely from character level annotations and are able to capture the underlying substructures of characters at different granularities. Moreover, strokelets provide an alternative way to accurately identify individual characters and compose a histogram feature to effectively describe characters. The scene text detection and recognition algorithms based on strokelets are both effective and robust. Extensive experiments on standard benchmarks verify the advantages of strokelets and demonstrate the effectiveness of the proposed text detection and recognition algorithms.

Actually, the idea of learning part prototypes from training data in an unsupervised manner is quite general, and thus can be readily extended to other object classes (e.g. cars, persons, and horses) or problems [83]–[86]. It would be interesting to employ this idea to learn a universal representation for multiple object classes. We will devote ourselves to the development of unified systems for multi-class object recognition in the future.

[5]8 threads were used to accelerate the process of multi-scale strokelet detection and voting.

References

[1] H. Webster, *World History*. Boston, MA, USA: Heath, 1921.
[2] L. V. Remias, *A Real-Time Image Understanding System for an Autonomous Mobile Robot*. Washington, DC, USA: Storming Media, 1996.
[3] H. Yang and C. Meinel, "Content based lecture video retrieval using speech and video text information," *IEEE Trans. Learn. technol.*, vol. 7, no. 2, pp. 142–154, Apr. 2014.
[4] J. Feng, "Mobile product search with bag of hash bits and boundary reranking," in *Proc. CVPR*, 2012, pp. 3005–3012.
[5] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo-location using a fixed-wing miniature air vehicle," *J. Intell. Robot. Syst.*, vol. 47, no. 4, pp. 361–382, 2006.
[6] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.
[7] Y. K. Ham, M. S. Kang, H. K. Chung, R.-H. Park, and G.-T. Park, "Recognition of raised characters for automatic classification of rubber tires," *Opt. Eng.*, vol. 34, no. 1, pp. 102–109, 1995.
[8] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *Proc. BMVC*, 2012, pp. 1–11.
[9] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, 2010, pp. 2963–2970.
[10] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. ICCV*, 2011, pp. 1457–1464.
[11] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. CVPR*, 2012, pp. 1083–1090.
[12] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. CVPR*, 2012, pp. 3538–3545.
[13] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *Proc. ECCV*, 2012, pp. 752–765.
[14] J. J. Weinman, Z. Butler, D. Knoll, and J. Feild, "Toward integrated scene text reading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 375–387, Feb. 2013.
[15] C. Yi and Y. Tian, "Scene text recognition in mobile applications by character descriptor and structure configuration," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2972–2982, Jul. 2014.
[16] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.
[17] C. Yao, X. Zhang, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Rotation-invariant features for multi-oriented text detection in natural images," *PLoS One*, vol. 8, no. 8, p. e70173, 2013.
[18] A. Mishra, K. Alahari, and C. V. Jawahar, "An MRF model for binarization of natural scene text," in *Proc. ICDAR*, 2011, pp. 11–16.
[19] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *Proc. CVPR*, 2013, pp. 2961–2968.
[20] C. Y. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu, "Region-based discriminative feature pooling for scene text recognition," in *Proc. CVPR*, 2014, pp. 4050–4057.
[21] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *Proc. CVPR*, 2009, pp. 1022–1029.
[22] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. CVPR*, 2005, pp. 524–531.
[23] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. CVPR*, 2013, pp. 923–930.
[24] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proc. CVPR*, 2014, pp. 4042–4049.
[25] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. ICDAR*, 2003, pp. 682–687.
[26] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. ECCV*, 2010, pp. 591–604.
[27] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. ACCV*, 2010, pp. 770–783.
[28] A. Mishra, K. Alahari, and C. V. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Proc. CVPR*, 2012, pp. 2687–2694.
[29] G. Yildirim, R. Achanta, and S. Süsstrunk, "Text recognition in natural images using multiclass Hough forests," in *Proc. VISAPP*, 2013, pp. 737–741.

[30] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar, "Whole is greater than sum of parts: Recognizing scene text words," in *Proc. ICDAR*, 2013, pp. 398–402.

[31] J. A. Rodriguez-Serrano and F. Perronnin, "Label embedding for text recognition," in *Proc. BMVC*, 2013.

[32] R. Wang, N. Sang, and C. Gao, "Text detection approach based on confidence map and context information," *Neurocomputing*, vol. 157, pp. 153–165, Jun. 2015.

[33] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontier Comput. Sci.*, vol. 10, no. 1, pp. 19–36, 2016.

[34] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proc. ICIP*, 2011, pp. 2609–2612.

[35] H. IlKoo and D. HoonKim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2296–2305, Jun. 2013.

[36] X.-C. Yin, X. Yin, K. Huang, and H.-E. Hao, "Accurate and robust text detection: A step-in for text retrieval in natural scene images," in *Proc. SIGIR*, 2013, pp. 1091–1092.

[37] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.

[38] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. 886–893.

[39] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proc. ICCV*, 2013, pp. 97–104.

[40] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[41] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2005, pp. 282–289.

[42] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. ECCV*, 2012, pp. 73–86.

[43] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proc. CVPR*, 2013, pp. 3158–3165.

[44] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *Proc. ICML*, 2013, pp. 846–854.

[45] Y. Le Cun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. NIPS*, 1990, pp. 396–404.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[47] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[48] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. CVPR*, 2014, pp. 1701–1708.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014, pp. 580–587.

[50] A. Coates *et al.*, "Text detection and character recognition in scene images with unsupervised feature learning," in *Proc. ICDAR*, 2011, pp. 440–445.

[51] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. ICPR*, 2012, pp. 3304–3308.

[52] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proc. ICCV*, 2013, pp. 785–792.

[53] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. ECCV*, 2014, pp. 512–528.

[54] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 69–88, 2013.

[55] S. McCann and D. G. Lowe, "Spatially local coding for object recognition," in *Proc. ACCV*, 2012, pp. 204–217.

[56] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[57] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *Proc. ICCV*, 2009, pp. 1365–1372.

[58] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *Proc. ECCV*, 2010, pp. 168–181.

[59] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 259–289, 2008.

[60] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[61] C. Yao, X. Bai, W. Liu, and L. J. Latecki, "Human detection using learned part alphabet and pose dictionary," in *Proc. ECCV*, 2014, pp. 251–266.

[62] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006, pp. 2169–2178.

[63] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. CVPR*, 2009, pp. 1794–1801.

[64] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. ICML*, 2014, pp. 647–655.

[65] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[66] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. (2012). *ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012)*. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2012/

[67] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.

[68] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "T-HOG: An effective gradient-based descriptor for single line text regions," *Pattern Recognit.*, vol. 46, no. 3, pp. 1078–1090, 2013.

[69] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *Proc. ICDAR*, 2011, pp. 1491–1496.

[70] S. M. Lucas, "ICDAR 2005 text locating competition results," in *Proc. ICDAR*, 2005, pp. 80–84.

[71] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *Int. J. Document Anal. Recognit.*, vol. 8, no. 4, pp. 280–296, 2006.

[72] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. (2014). "Reading text in the wild with convolutional neural networks." [Online]. Available: http://arxiv.org/abs/1412.1842

[73] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep structured output learning for unconstrained text recognition," in *Proc. ICLR*, 2015, pp. 1–10.

[74] *ABBYY FineReader 9.0*, acessed on Apr. 15, 2016. [Online]. Available: http://www.abbyy.com/

[75] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proc. NIPS Deep Learn. Workshop*, 2014, pp. 1–10.

[76] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[77] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proc. VISAPP*, 2009, pp. 273–280.

[78] W. Huang, Y. Qu, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," in *Proc. ECCV*, 2014, pp. 497–511.

[79] L. Neumann and J. Matas, "On combining multiple segmentations in scene text recognition," in *Proc. ICDAR*, 2013, pp. 523–527.

[80] C. Yi and Y. Tian, "Text detection in natural scene images by stroke Gabor words," in *Proc. ICDAR*, 2011, pp. 171–177.

[81] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011.

[82] S. Lee, M. S. Cho, K. Jung, and J. H. Kim, "Scene text extraction with edge constraint and text collinearity," in *Proc. ICPR*, 2010, pp. 3983–3986.

[83] B. Shi, X. Bai, and C. Yao, "Script identification in the wild via discriminative convolutional neural network," *Pattern Recognit.*, vol. 52, pp. 448–458, Apr. 2016.

[84] G.-S. Xia, J. Delon, and Y. Gousseau, "Accurate junction detection and characterization in natural images," *Int. J. Comput. Vis.*, vol. 106, no. 1, pp. 31–56, Jan. 2014.

[85] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian, "Robust feature matching for remote sensing image registration via locally linear transforming," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6469–6481, Dec. 2015.
[86] X. Bai, S. Bai, Z. Zhu, and L. J. Latecki, "3D shape matching via two layer coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2361–2373, Dec. 2015.

Xiang Bai received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications, HUST, where he is also the Vice Director of the National Center of Anti-Counterfeiting Technology. His research interests include object recognition, shape analysis, scene text recognition, and intelligent systems.

Cong Yao received the B.S. and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. He was a Visiting Research Scholar with Temple University, Philadelphia, PA, USA, in 2013. His research has focused on computer vision and machine learning, in particular, the area of text detection and recognition in natural images.

Wenyu Liu received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is currently a Professor and the Associate Dean of the School of Electronic Information and Communications with HUST. His current research areas include sensor network, multimedia information processing, and computer vision. He is a Senior Member of the IEEE System, Man and Cybernetics Society.