This CVPR2014 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

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

Cong Yao Xiang Bai* Baoguang Shi Wenyu Liu
Department of Electronics and Information Engineering, Huazhong University of Science and Technology
yaocong2010@gmail.com, xbai@hust.edu.cn, chn.edward@gmail.com, liuwy@hust.edu.cn

Abstract

Driven by the wide range of applications, scene text detection and recognition have become active research topics in computer vision. Though extensively studied, localizing and reading text in uncontrolled environments remain extremely challenging, due to various interference factors. In this paper, we propose a novel multi-scale representation for scene text recognition. This representation consists of a set of detectable primitives, termed as strokelets, which capture the essential substructures of characters at different granularities. Strokelets possess four distinctive advantages: (1) Usability: automatically learned from bounding box labels; (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 strokelets and demonstrate the effectiveness of the proposed algorithm for text recognition.

1. Introduction

As an important carrier of human thoughts and emotions, text plays a crucial role in our daily lives. It is almost ubiquitous, especially in modern urban environments. For example, product tags, license plates, guideposts and billboards, all contain text. The rich information embedded in text can be very beneficial, but the rapid growth of text data has made it prohibitive to process, interpret and apply it manually. Consequently, automatic text detection and recognition have become an irresistible general trend.

However, spotting and reading text in natural scenes are extremely difficult for computers. Though considerable progress has been achieved in recent years [8, 29, 35, 23, 25, 20, 34], detecting and recognizing text in uncontrolled environments are still open problems in computer vision. Various interference factors, such as variation, distortion, noise, blur, non-uniform illumination, local distractor and complex background, all may pose major challenges [36, 34].

To tackle these challenges, representation is lying in the middle and core. Excellent representations should be able to effectively describe the characteristics of characters in natural images and meanwhile be robust to interference factors.

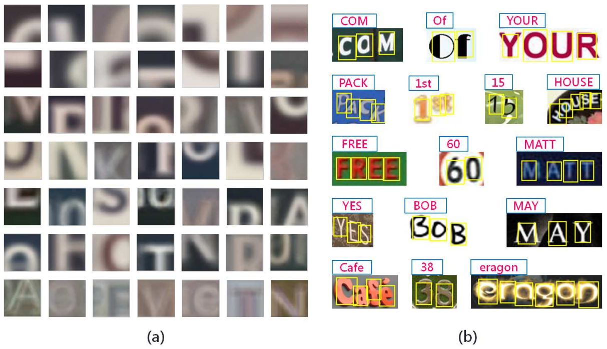

Figure 1. Illustration of strokelets and character recognition. (a) Strokelets learned on IIIT 5K-Word [20]. Strokelets capture the structural characteristics of characters at multiple scales, ranging from local primitives, like bar, arc and corner (top), to whole characters (bottom). (b) Character recognition examples. Strokelets produce accurate character identification and recognition.

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

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

- ***U**sability*: automatically learned from bounding box labels, not requiring detailed annotations.
- ***R**obustness*: insensitive to interference factors, endowing the system with the ability to deal with real-world complexity.
- ***G**enerality*: applicable to variant languages, as long as sufficient training examples are available.
- ***E**xpressivity*: effective at describing 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 for characters, faithfully seize the representative parts of characters at multiple scales; and characters in different fonts, scales, colors, anc e

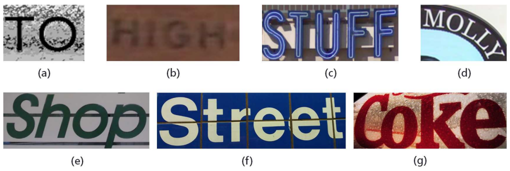

Figure 2. Typical issues encountered in character identification. (a) Noise. (b) Blur. (c) Shadow. (d) Unusual layout. (e) Local distractor. (f) Broken strokes. (g) Connected characters.

successfully localized and read, even with the presence of noise, blur and 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, for instance, noise, blur, shadow, unusual layout, local distractor and connected characters (Fig. 2), might result in errors. To address these issues, several approaches were proposed, which employed adaptive binarization [19, 34], connected component extraction [23, 25] or direct character detection [29, 20, 27]. These methods work well in certain cases, but are still far from producing all satisfactory results. For example, connected component extraction is unable to handle broken strokes and connected characters, while direct character detection 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 predict these attributes of characters in test images at runtime. Therefore, an alternative method for character identification is introduced, leading to more accurate and robust character candidate identification. Moreover, detection activations of strokelets compose a histogram representation, similar to Bag of Words [9] and Bag of Parts [12], which provides extra discriminative power. Based on strokelets, we devise an effective algorithm for scene text recognition, which achieves much higher recognition rate than existing systems.

To evaluate the effectiveness and robustness of the proposed representation and text recognition algorithm, we have conducted extensive experiments on standard benchmarks for scene text recognition, including the challenging public datasets ICDAR 2003 [17], SVT [30] and IIIT 5K-Word [20]. The experiments verify the advantages of strokelets and demonstrate that the proposed algorithm outperforms the state-of-the-art methods in the literature.

To deliver reproducible research, we will make the source code publicly available, and hope it would be useful to other researchers.

2. Related Work

There has been a rich body of works concerning text recognition in natural images in recent years [30, 29, 23, 21, 37, 34]. Wang *et al.* [30, 29] used HOG templates [6] to match character instances in test images with training examples. Neumann *et al.* [23] extracted connected components (extremal regions) as building blocks to localize and recognize characters. Weinman *et al.* [34] proposed to integrate character segmentation and recognition.

Part based methods [27, 37, 32] have been very popular in this field. Shi *et al.* [27] described a part-based model, employing DPM [10] and CRF, for scene text recognition. However, the structure of character models and parts of each character class were manually designed and labeled. In [37], Yildirim *et al.* developed a part-based algorithm which adopted multi-class Hough Forest to detect and recognize characters in natural images. Neumann *et al.* [24] 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 [27, 37, 24] 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 [28, 16, 33]. Singh *et al.* [28] 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.* [16] 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 part prototypes to represent characters. Activations of such part prototypes compose a histogram feature, which is akin to Bag of Words [9] and Bag of Parts [12].

The presented work is complementary to a line of research efforts on error correction [25, 21, 20], which we briefly review here. Novikova *et al.* [25] proposed a unified probabilistic framework, which utilized Weighted Finite-State Transducers [22] to simultaneously introduce language prior and enforce attribute consistency within hypotheses. Mishra *et al.* [21] constructed a CRF model to impose both bottom-up (i.e. character detections) and top-down (i.e. language statistics) cues. In [20], 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). Replacing such modules with those based on strokelets, these methods could attain better performance.

[1] We intentionally avoid the term "character detection" as certain algorithms (such as [19, 34]) utilize binarization to seek character candidates.

3. Methodology

In this section, we describe the procedure for strokelet generation and present the algorithm for text recognition.

3.1. Strokelet Generation

Given a set of training images containing scene text $S = \{(I_i, B_i)\}_{i=1}^n$, 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 S. The part prototypes should be able to capture the essential substructures of characters and be distinctive from local background and against each other.

As S only provides bounding box level annotations for each character, the part prototypes should be automatically discovered. The newly developed discriminative clustering algorithm proposed by Singh *et al.* [28] 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.* [28] aims at discovering a set of representative patch clusters that are discriminative against other clusters in $\mathcal{D}$, as well as the rest 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 K_j in novel images. These classifiers will serve as part detectors at runtime. For more details, please refer to [28].

The algorithm of Singh *et al.* [28] 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 rest 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 [28], 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 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 [18]. This forces the patches in each cluster to be compact in configuration space.
- The SVM classifier used in [28] was replaced by Random Forest [4] 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 [6]) 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 object centroid 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 objects in the character identification stage (see Sec. 3.2.1).

Fig. 3 depicts the strokelets (classifiers not shown) learned on the IIIT 5K-Word dataset [20]. As can be seen, strokelets, as a universal representation, express 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. 3 (b)) as well as those unique to certain character classes (see the bottom row of Fig. 3 (b)).

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. 4).

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. 3 (b)). These properties make strokelets closely analogous to poselets [3, 2].

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

Algorithm 1 Algorithm for strokelet generation

```
Require: Training set $S$, interval $[a, b]$, strokelet count $\Gamma$
{D, N}
$\{\mathcal{D}, \mathcal{N}\} \Leftarrow construct(S)$    ▷ 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, $\tau$ 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 [28] for details
$\Omega \Leftarrow select\_top(K, C, A, \Gamma)$    ▷ Sort according to scores and select top $\Gamma$ clusters and classifiers
return $\Omega$
```

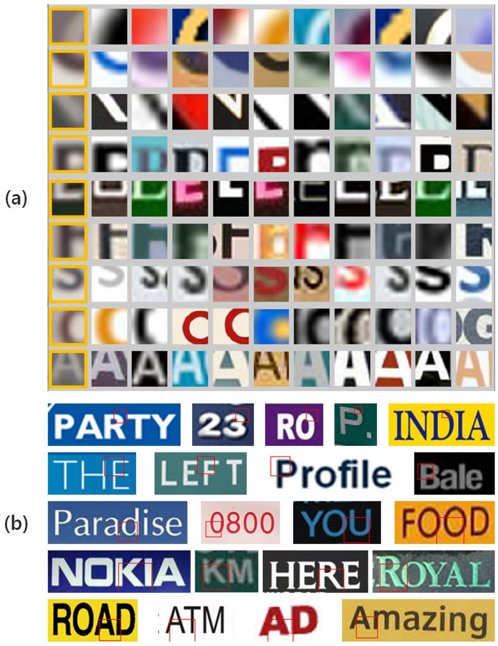

Figure 3. Learned strokelets on the IIIT 5K-Word dataset [20]. (a) Each row illustrates a cluster of part instances that constitute a strokelet. The images in the first column (orange rectangle) are the average of all the instances of that strokelet. The rest are top-ranked part instances. (b) Discovered part instances in original images. The learned part prototypes are tightly clustered in both appearance and configuration space.

However, different from poselets, which are obtained using manually labeled data (part regions and keypoints), strokelets are automatically learned using bounding box level annotations.

3.2. 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. 3.2.1); these candidates are then described by a histogram feature based on strokelets and a holistic descriptor (Sec. 3.2.2); and character classification is applied to assign the most probable class label to each character (Sec. 3.2.3). Optionally, the inferred word is replaced by the most similar item in a given dictionary, following [29, 21].

The algorithm described above is quite effective, even though without sophisticated approaches to error correction [25, 20]. We believe better performance could be achieved if such error correction methods are incorporated.

3.2.1 Character Identification

As stated in Sec. 1, character identification is a key stage in scene text recognition. However, binarization based methods [19, 34] are sensitive to noise, blur and non-uniform illumination; connected component based methods [23, 25] are unable to handle connected characters and broken strokes; and direct character detection based methods [30, 20] usually produce a lot of false alarms. In this paper, we propose a voting scheme to seek characters, based on multi-scale strokelet detection.

This scheme shares the idea of estimating character centers through voting with the work of Yildirim *et al.* [37]. However, the work in [37] 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), while the proposed strategy casts votes from multiple scales.

Firstly, the original image (Fig. 4 (a)) is resized to a standard height (64 pixels in this paper) with aspect ratio kept

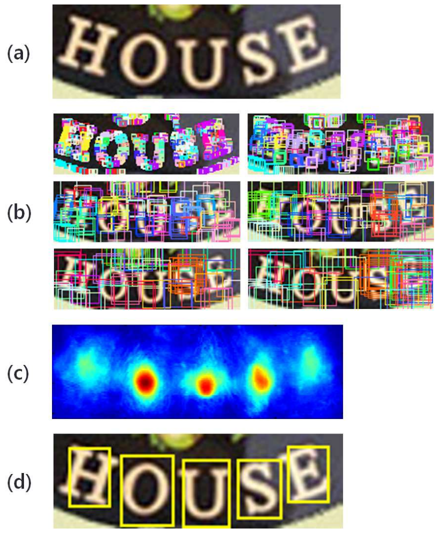

Figure 4. Character identification. (a) Original image. (b) Detections of strokelets at different scales. Activations of different types of strokelets are marked in different colors. For better visualization, the images are rescaled and non-maximum suppression is applied to the activation windows. (c) Hough map. (d) Identified characters. Different from [21], 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.

unchanged; since strokelets are naturally multi-scale representation, a multi-scale sliding-window paradigm is performed to detect strokelets (Fig. 4 (b)); a Hough map (Fig. 4 (c)) is then generated by casting and accumulating the votes from the detected strokelets, similar to [15]; finally, the centers of the character candidates are found by seeking maxima in the Hough map using Mean Shift [5] 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. 3.1).

For a character candidate α, 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:

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

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

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. 5. By adopting discriminative training and multi-scale voting, the proposed scheme is capable of handling issues like noise, blur, local distractor and connected characters.

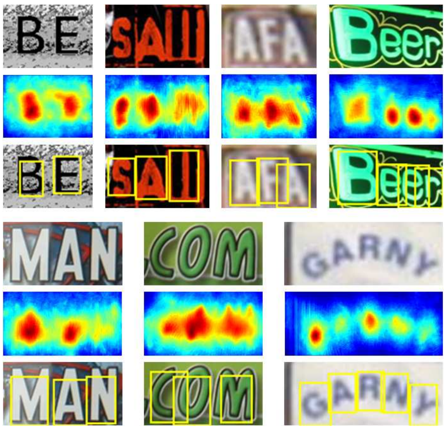

Figure 5. Examples of character identification. The scheme for character identification is able to hunt characters of different fonts, sizes, colors and layouts with the presence of noise, blur and local distractor.

3.2.2 Character Description

It has been widely accepted in the community that informative features promise high performance. Based on detection activations of strokelets, we introduce a histogram feature called Bag of Strokelets, in addition to the traditional feature HOG [6].

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. 4 (b)) are assembled together. Each strokelet contributes to the histogram feature according to its detection score. To incorporate spatial information, the Spatial Pyramid strategy [13] (1×1 and 2×2 grids) is also adopted.

HOG. Following [21, 25], we also adopt the HOG descriptor (the version proposed in [10]) 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. 4.

3.2.3 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, partial of valid characters, 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 [4] 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

Γ	100	200	300	400	500	600	700
Accuracy(%)	71.4	75.9	78.1	77.1	**80.2**	79.0	78.3

Table 1. Impact of strokelet count Γ on the IIIT 5K-Word dataset.

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.

4. Experiments

We have evaluated the proposed representation and text recognition algorithm on several standard benchmarks, and compared it to 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.

4.1. Datasets

IIIT 5K-Word. The IIIT 5K-Word dataset [20] 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.

ICDAR 2003. The ICDAR 2003 Robust Word Recognition Competition [17] 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 [25, 20, 27], we skipped the words with two or fewer characters, as well as those with non-alphanumeric characters.

SVT. The Street View Text (SVT) dataset [30, 29] is a collection of outdoor images with scene texts 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, as we focus on word recognition in cropped images.

For fair comparison, the lexicons for the ICDAR 2003 and SVT dataset provided in [29] are also used in this work.

4.2. Experimental Results

Strokelet count Γ is a key parameter as it determines the number of learned strokelets. We first investigated the impact of strokelet count on the IIIT 5K-Word dataset. As can be seen from Tab. 1, 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 last one, strokelet count Γ is fixed at 500.

Lexicon	Small	Medium	Large
Proposed	**80.2**	**69.3**	38.3
Higher Order [20](with edit distance)	68.25	55.50	28
Higher Order [20](without edit distance)	64.10	53.16	**44.30**
Pairwise CRF [21](with edit distance)	66	57.5	24.25
Pairwise CRF [21](without edit distance)	55.50	51.25	20.25
ABBYY9.0 [1]	24.33	-	-

Table 2. Performances of different algorithms evaluated on the IIIT 5K-Word dataset.

We learned a set of strokelets on the IIIT 5K-Word dataset and evaluated the proposed algorithm on it. The performances of the proposed algorithm and other recently published works are illustrated in Tab. 2. In general, the proposed algorithm outperforms all the competing methods. With small lexicon, the proposed algorithm achieves a recognition accuracy of 80.2%, which is 12% higher than that of the closest competitor Higher Order [20] without edit distance (68.25%); with medium lexicon, the improvement (13.8%) is even more notable; with large lexicon, the proposed algorithm is comparable to Higher Order without edit distance, but behind it. This is reasonable as the large lexicon is independent from IIIT 5K-Word[3] and Higher Order [20] 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 38.3%) is also very significant.

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 [20] and [21], which will create a more powerful system for scene text recognition.

We also applied the learned strokelets to the test images of the ICDAR 2003 and SVT dataset. Fig. 6 shows several character recognition examples on these two datasets. The strokelets generalize well to novel images from other databases. The proposed algorithm is able to handle challenges like font variation, scale change, noise, connected characters, non-uniform illumination and partial occlusion.

The performances of the proposed algorithm as well as other competing methods on the ICDAR 2003 and SVT dataset are depicted in Tab. 3. The proposed algo-

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

Dataset	ICDAR 2003(FULL)	ICDAR 2003(50)	SVT
Proposed	80.33	88.48	75.89
CNN [31]	**84**	**90**	70
Whole [11]	-	89.69	**77.28**
TSM+CRF [27]	79.30	87.44	73.51
TSM+PLEX [27]	70.47	80.70	69.51
Multi-Class Hough Forests [37]	-	85.70	-
Large-Lexicon Attribute-Consistent [25]	82.8	-	72.9
Higher Order [20](with edit distance)	-	80.28	73.57
Higher Order [20](without edit distance)	-	72.01	68.00
Pairwise CRF [21](with edit distance)	-	81.78	73.26
Pairwise CRF [21](without edit distance)	-	69.90	62.28
SYNTH+PLEX [29]	62	76	57
ICDAR+PLEX [29]	57	72	56
ABBYY9.0 [1]	55	56	35

Table 3. Performances of different algorithms evaluated on the ICDAR 2003 and SVT dataset.

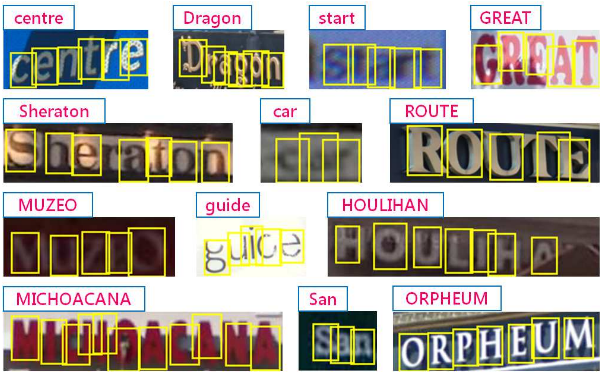

Figure 6. Examples of character recognition on the ICDAR 2003 and SVT dataset. Though only trained on the IIIT 5K-Word dataset, the strokelets generalize well to the images from ICDAR 2003 and SVT.

rithm achieves recognition accuracy of 80.33%, 88.48% and 75.89% on ICDAR 2003(FULL), ICDAR 2003(50) and SVT respectively, outperforming the competing methods of [29, 25, 21, 20, 37, 27], but still behind those in [31, 11]. Note that the amount of training data used in [31] is far more than that of our algorithm, while [11] cannot handle words out of the given dictionary. Compared to these methods, the proposed algorithm requires less training examples and has a broader scope of application.

It is worth mentioning that the proposed algorithm is superior to the best performer [27], which employed manually designed character models and detailed part annotations. This proves automatically learned parts can 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.

Feature	HOG	Bag of Strokelets	HOG+Bag of Strokelets
Accuracy(%)	78.6	73.7	**80.2**

Table 4. Performances of different types of features.

Scale(a=b)	0.2	0.3	0.4	0.5	0.6	0.7	**multi-scale**
Accuracy(%)	59.9	71.9	74.1	74.4	74.8	74.3	**80.2**

Table 5. Advantage of multi-scale representation.

We validated the excellent ability of strokelets in character identification. For character localization, strokelets obtain precision of 45% at recall=78% and precision of 64% at EER on IIIT 5K-Word, far surpassing [20] (precision≈17% at recall=78% and precision≈35% at EER). For character classification, strokelets obtain 60%, 67% and 64% (case insensitive) on Chars74K [7], ICDAR-CHAR and SVT-CHAR [29], respectively, outperforming [21, 29]. The work in [27] used a different evaluation protocol, thus is not directly comparable.

We also evaluated the effectiveness of the Bag of Strokelets feature and compared it with HOG. The recognition rates of different types of features on the IIIT 5K-Word dataset are shown in Tab. 4. The conventional feature HOG is quite informative, achieving a recognition rate of 78.6%, while that of Bag of Strokelets is 73.7%. These two types of features are complementary. Their combination leads to higher performance (80.2%).

To verify the advantage of multi-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 Tab. 5. 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 convey more information, performs much better than single-scale representations.

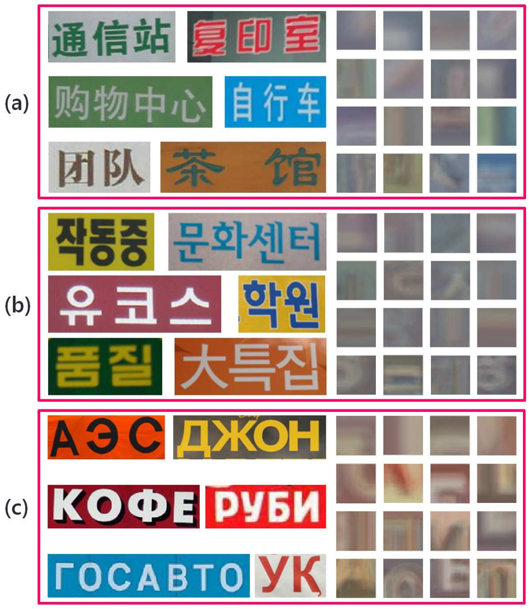

Figure 7. Learned strokelets ($\Gamma = 100$) on different languages. (a) Chinese. Original images are from [26]. (b) Korean. Original images are from [14]. (c) Russian. Original images are harvested from the Internet.

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. 7. 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. In order to cope with multilingual scenarios, we could learn a hybrid set of strokelets on multiple languages.

5. Conclusions and Future Work

We have introduced strokelets, a novel presentation automatically learned from bounding box labels, for the purpose of capturing the underlying substructures of characters at different granularities. Strokelets provide an alternative way to accurately identify individual characters and compose a histogram feature to effectively describe characters in natural scenes. The scene text recognition algorithm based on strokelets is both effective and robust. Extensive experiments on standard benchmarks verify the advantages of strokelets and demonstrate that the proposed algorithm consistently outperforms the current state-of-the-art approaches in the literature.

In this paper, we only demonstrated the strengths of strokelets on the task of text recognition in cropped images. The idea is actually quite general and can be employed to perform both text detection and recognition in full images. This is an ongoing work. Furthermore, we could extend the applicability of this idea by learning multi-scale prototypes for other object classes (e.g. cars, persons, and faces) and using them to detect and recognize such object classes.

Acknowledgement

This work was primarily supported by National Natural Science Foundation of China (NSFC) (No. 61222308), and in part by NSFC (No. 61173120), Program for New Century Excellent Talents in University (No. NCET-12-0217) and Fundamental Research Funds for the Central Universities (No. HUST 2013TS115).

References

[1] ABBYY FineReader 9.0. http://www.abbyy.com/. 6, 7

[2] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proc. ECCV*, 2010. 3

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. ICCV*, 2009. 3

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 3, 5

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. PAMI*, 17(8):790–799, 1995. 5

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 2, 3, 5

[7] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proc. of VISAPP*, 2009. 7

[8] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. of CVPR*, 2010. 1

[9] L. Fei-Fei and P. Perona. A bayesian heirarcical model for learning natural scene categories. In *Proc. of CVPR*, 2005. 2

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. PAMI*, 32(9):1627–1645, 2010. 2, 5

[11] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In *Proc. of ICDAR*, 2013. 7

[12] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proc. of CVPR*, 2013. 2

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of CVPR*, 2006. 5

[14] S. H. Lee, K. J. M. S. Cho, and J. H. Kim. Scene text extraction with edge constraint and text collineairty link. In *Proc. of ICPR*, 2010. 8

[15] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008. 5

[16] J. J. Lim, C. L. Zitnick, and P. Dollar. Sketch tokens: A learned mid-level representation for contour and object detection. In *Proc. of CVPR*, 2013. 2

[17] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *Proc. of ICDAR*, 2003. 2, 6

[18] S. McCann and D. G. Lowe. Spatially local coding for object recognition. In *Proc. ACCV*, 2012. 3

[19] A. Mishra, K. Alahari, and C. V. Jawahar. An MRF model for binarization of natural scene text. In *Proc. of ICDAR*, 2011. 2, 4

[20] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *Proc. of BMVC*, 2012. 1, 2, 3, 4, 6, 7

[21] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Proc. of CVPR*, 2012. 2, 4, 5, 6, 7

[22] M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, 2013. 2

[23] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. of CVPR*, 2012. 1, 2, 4

[24] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *Proc. of ICCV*, 2013. 2

[25] T. Novikova, , O. Barinova, P. Kohli, and V. Lempitsky. Large-lexicon attribute-consistent text recognition in natural images. In *Proc. of ECCV*, 2012. 1, 2, 4, 5, 6, 7

[26] Y. Pan, X. Hou, and C. Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. Image Processing*, 20(3):800–813, 2011. 8

[27] 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. of CVPR*, 2013. 2, 6, 7

[28] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *Proc. ECCV*, 2012. 2, 3, 4

[29] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proc. of ICCV*, 2011. 1, 2, 4, 6, 7

[30] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. of ECCV*, 2010. 2, 4, 6

[31] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. of ICPR*, 2012. 7

[32] X. Wang, X. Bai, X. Yang, W. Liu, and L. J. Latecki. Maximal cliques that satisfy hard constraints with application to deformable object model learning. In *Proc. of NIPS*, 2011. 2

[33] X. Wang, B. Wang, X. Bai, and Z. T. W. Liu. Max-margin multiple-instance dictionary learning. In *Proc. of ICML*, 2013. 2

[34] J. J. Weinman, Z. Butler, D. Knoll, and J. Feild. Toward integrated scene text reading. *IEEE Trans. PAMI*, 2013. 1, 2, 4

[35] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. of CVPR*, 2012. 1

[36] 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*, 8(8), 2013. 1

[37] G. Yildirim, R. Achanta, and S. Susstrunk. Text recognition in natural images using multiclass hough forests. In *Proc. of VISAPP*, 2013. 2, 4, 7