Text/Non-text Image Classification in the Wild with Convolutional Neural Networks

Article *in* Pattern Recognition · December 2016

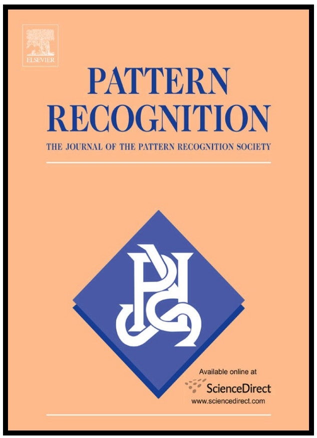
DOI: 10.1016/j.patcog.2016.12.005

Xiang Bai
Huazhong University of Science and Technology
184 PUBLICATIONS **6,267** CITATIONS
SEE PROFILE

Baoguang Shi
Huazhong University of Science and Technology
25 PUBLICATIONS **1,043** CITATIONS
SEE PROFILE

Zhang Chengquan
Shanghai Jiao Tong University
9 PUBLICATIONS **307** CITATIONS
SEE PROFILE

Li Qi
The Third Research Institute of the Ministry of Public Security, China
42 PUBLICATIONS **517** CITATIONS
SEE PROFILE



www.elsevier.com/locate/pr

Text/Non-text Image Classification in the Wild with Convolutional Neural Networks

Xiang Bai, Baoguang Shi, Chengquan Zhang, Xuan Cai, Li Qi

PII: S0031-3203(16)30392-2
DOI: http://dx.doi.org/10.1016/j.patcog.2016.12.005
Reference: PR5977

To appear in: *Pattern Recognition*

Received date: 13 March 2016
Revised date: 5 December 2016
Accepted date: 8 December 2016

Cite this article as: Xiang Bai, Baoguang Shi, Chengquan Zhang, Xuan Cai an Li Qi, Text/Non-text Image Classification in the Wild with Convolutional Neura Networks, *Pattern Recognition,* http://dx.doi.org/10.1016/j.patcog.2016.12.005

Text/Non-text Image Classification in the Wild with Convolutional Neural Networks

Xiang Bai[a], Baoguang Shi[a], Chengquan Zhang[a], Xuan Cai[b], Li Qi[b,*]

[a] *School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China 430074*
[b] *The Third Research Institute of the Ministry of Public Security, Shanghai, China*

Abstract

Text in natural images is an important source of information, which can be utilized for many real-world applications. This work focuses on a new problem: distinguishing images that contain text from a large volume of natural images. To address this problem, we propose a novel convolutional neural network variant, called Multi-scale Spatial Partition Network (MSP-Net). The network classifies images that contain text or not, by predicting text existence in all image blocks, which are spatial partitions at multiple scales on an input image. The whole image is classified as a text image (an image containing text) as long as one of the blocks is predicted to contain text. The network classifies images very efficiently by predicting all blocks simultaneously in a single forward propagation. Through experimental evaluations and comparisons on public datasets, we demonstrate the effectiveness and robustness of the proposed method.

Keywords: Natural images, Text/non-text image classification, Convolutional neural network, Multi-scale spatial partition

1. Introduction

Scene text is an important source of information that is helpful for many real-world applications, including image retrieval, human-computer interaction,

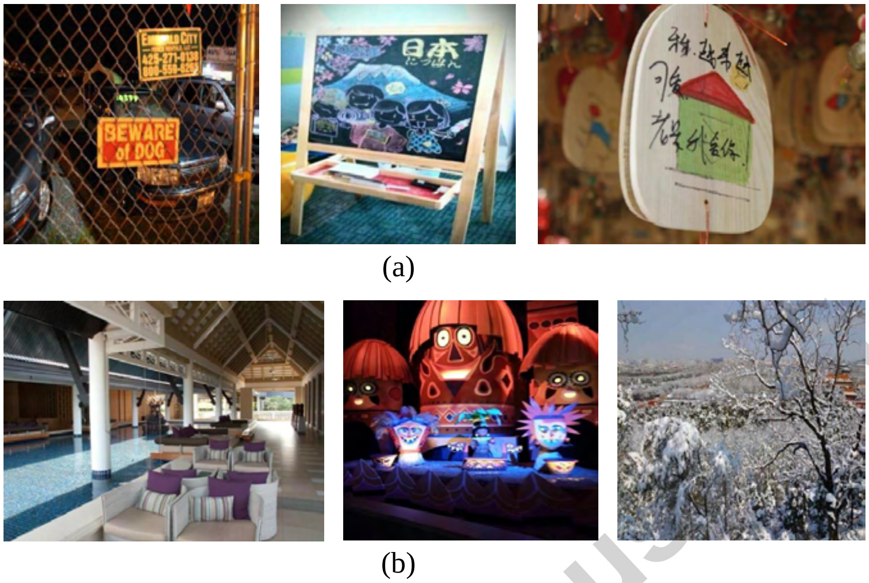
[*] Corresponding author

Email addresses: xbai@hust.edu.cn (Xiang Bai), shibaoguang@gmail.com (Baoguang Shi), zchengquan@gmail.com (Chengquan Zhang), caixuanfire@126.com (Xuan Cai), quick.qi@foxmail.com (Li Qi)

blind assistance system, transportation navigation, etc. Therefore, scene text reading, which includes text detection and recognition, has attracted much attention in the community [1, 2, 3]. However, typically, in a large volume of natural images and video data, only a small portion contains text. In our estimation on an image dataset collected from social networks, only 10%-15% of the images contain text. Directly applying scene text reading algorithms for mining textual information tends to be inefficient, as most of the existing text reading algorithms are time-consuming. To precisely localize text in an image, algorithms like [4, 5, 6, 7, 8] typically require searching a large set of text-line or character candidates, or dense image patches. The search would be meaningless if an image contains no text at all. Therefore, an efficient preprocessing algorithm that quickly distinguishes whether an image contains text or not is desirable, which can be utilized as an essential stage of the systems for text reading or script identification [9].

In this work, we address a relatively new problem: text/non-text image classification in the wild. The image that contains text is identified as text image (or text positive image), regardless of the scale or location of text in it. Whereas, the image that does not contain any text is named as non-text image (or text negative image). In this paper, we adopt the pair of *text image* and *non-text image* to distinguish two types of natural images. We define *text image* as an image that contains text, regardless of its scale or location, and *non-text image* as an image that contains no text at all. Although some previous works have already addressed the text/non-text image classification problem, their focus is mainly on video frames [10, 11], document images [12], or handwriting images [13, 14]. However, we focus on the discrimination of text/non-text natural images, which has been seldom studied.

Unlike scene text detection, text/non-text image classification neither requires finding precise text locations, nor recognizing text contents. Instead, computational efficiency is important. A text/non-text image classification algorithm should classify a large amount of images in a short period of time, while achieving high precision and recall.



(a)

(b)

Figure 1: Examples of text/non-text images. (a) Text images contain at least one piece of scene text, regardless of the scales and locations; (b) Non-text images contain no text at all.

We argue that the proposed problem is challenging in four aspects. First, scene text exhibits large variations in font, scale, color, orientation, illumination, and language type. The examples shown in Fig. 1 demonstrate some of the variations. Second, difficult to distinguish scene text with other background objects, such as windows, grass, and fences, which are similar to text. Third, the locations of scene text are not known in advance. It may appear at any position in an image. Last, a text/non-text image classification algorithm should work efficiently enough to process a large amount of data in a reasonable period of time.

Essentially, text/non-text image classification is a binary classification problem. A straight-forward solution is to fine-tune some well-trained image classifiers, such as the Convolutional Neural Network (CNN) model proposed in [15]. However, due to the above-mentioned challenges, general image classification algorithms may not work well for this problem. In particular, conventional CNN models do not explicitly handle large scale and location variations exhibited in scene text.

In this paper, we propose a novel variant of CNN, named Multi-scale Spatial Partition Network (MSP-Net), which is specially designed for the problem of text/non-text image classification. The main idea is to classify all image *blocks*, which are regions produced by multi-scale spatial partition on an input image. If at least one of the blocks is classified as text block, the whole image is recognized as a text image, otherwise a non-text image. Since blocks have various sizes and positions so the proposed block level classification scheme allows us to detect text at multiple scales and locations. Moreover, as a by-product, the proposed MSP-Net predicts coarse locations and scales of text.

MSP-Net can be evaluated and trained efficiently. During testing, all blocks of an image are classified simultaneously in a single network forward propagation. Plus our optimized GPU implementation, the proposed network classifies text/non-text images very efficiently. MSP-Net is end-to-end trainable, because every layer of it can back-propagate error differentials. It can be easily trained using images and corresponding block-level annotations.

The contributions of this paper are summarized as the following: (1) We propose a new scheme for text/non-text image classification based on block-level classification, rather than whole image-level classification; (2) We propose a novel variant of CNN, called MSP-Net, which efficiently classifies text/non-text images, and is robust to the large variations on scale, location and language type of scene text; (3) As a by-product, we show that MSP-Net is also capable of coarsely localizing scene text.

The rest of this paper is organized as followed. In Sec. 2 we review related work. In Sec. 3, we describe and explain the architecture of MSP-Net. Experimental evaluations, comparisons with other methods, and discussions are presented in Sec. 4. Sec. 5 concludes our work.

2. Related work

Scene text reading. Scene text reading has been extensively studied in recent years. Scene text detection and scene text recognition are two major topics in this area. Most of the previous works focus on scene text detection and recognition [4, 5, 7, 16, 17, 18]. As mentioned, text/non-text image classification can be handled by a scene text detection algorithm. Epshtein *et al.* in [19] utilized the stroke width transform to seek candidate character components. Neumann and Matas in [20] extracted maximally stable extremal regions (MSERs) as candidate character regions to set up a novel and robust pipeline for text localization in real-world images. Different from the use of single character or stroke, Zhang *et al.* exploited the symmetry property of character groups to directly extract text-line candidates. However, most of them designed for precise localizing text, which requires a lot of time to search and filter text/character candidates. Whereas, text/non-text image classification aims at finding if a natural image contains text or not.

Image classification. In term of the essence, text image discrimination is a sub task of image classification. The existing methods can be summarized into three categories: feature encoding based methods, deep learning based methods, and hybrid methods. The framework of Bag of Words (BoW) is a typical feature encoding based method. The local descriptors such as HOG [21], SIFT [22], LBP [23], etc. of regions of interesting (ROIs) are extracted, and aggregated by some feature encoding methods such as vector of locally aggregated descriptors (VLAD) [24], locality-constrained linear coding (LLC) [25]. After then, one image can be represented by a compact and discriminative vector, which are effective in image classification or retrieval. Recently, convolutional neural networks have achieved high performance of image classification. Thanks to the CNN equipped with many convolutional layers, rectified units, sampling layers, fully-connected layers,etc., the network can learn features and do image classification in an end-to-end manner. The learned CNN features have demonstrated the effectiveness and robustness for image classification [26], object detection [27], contour detection [28], etc. However, most of existing CNN models require a fixed-size input image. He *et al.* [29] proposed SPP-net model to generate a fixed-length representation regardless of image size/scale. In our approach, we also take the advantage of spatial pyramid pooling to generate fixed-length representations for image blocks.

Text/non-text image classification. There are several works that address the problem of text image discrimination in document images or video data, but most of them aren't suitable for natural images. In [30], Alessi *et al.* proposed a method to detect the potential text blocks of document image and set a threshold value to distinguish text and non-text documents. Vidya *et al.* [31] proposed a system to classify the text and non-text regions in handwritten documents, which can't deal with natural images either. To our knowledge, our previous work [32] first proposed a suitable method that is the combination of three mature techniques including: MSERs, BoW, and CNN for text/non-text image classification. We also released a large dataset which can be a benchmark for evaluating algorithms of text/non-text image classification. Another important related work is the method proposed in [10], Shivakumara *et al.* first proposed a method for video text frame classification based on fixed-size block partition. The text block can indicate the coarse position of text. Inspired by this idea, our work proposes multi-scale spatial partition for natural text/non-text image classification, due to the large variation of text scale and location in natural scenes. Unlike the simple features adopted in [10], we adopt the convolutional neural network to make the block-level prediction in a end-to-end manner by moving the multi-scale spatial partition operation from image space to feature map. The multi-scale spatial partition plays the same role of ROI layer designed in fast R-CNN [33], which can extract the CNN features for each region in an efficient way. Furthermore, one image block classified as text block should consider the scale and area together in our method, so that the text block in our method can also predict the position and scale of text at a coarse level.

3. The Proposed Methodology

3.1. Overview

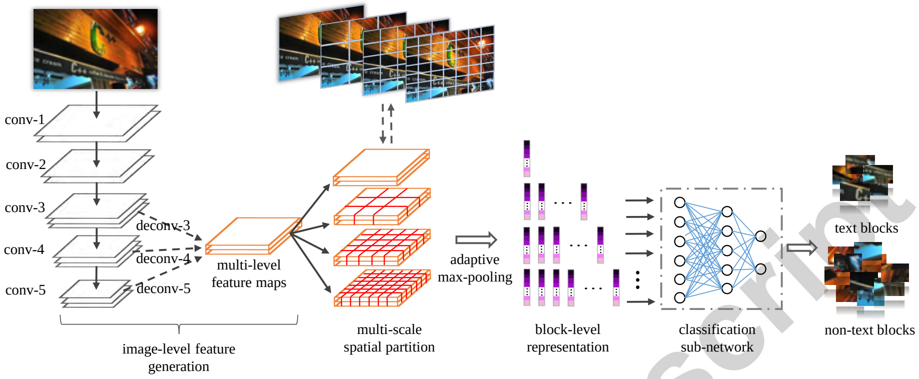


Figure 2: The overall architecture of MSP-Net.

As introduced in Sec. 1, our starting point is to classify text/non-text image through the examining images at a block level. However, different from the hand-crafted feature used for pre-partition image blocks in [10], our method combines spatial partition, feature extraction and text/non-text block classification into a single network (MSP-Net). The MSP-Net consists of 4 major parts: image-level feature generation, multi-scale spatial partition, block-level representation generation and text/non-text block classification sub-network. The overall structure of MSP-Net is illustrated in Fig. 2, which only requires the whole image as an input and examines all the image blocks in an end-to-end manner.

Given an input image, the network outputs block-level classification results in a single forward propagation. Inside the network, first, an image is fed into the convolutional layers, whose structure is derived from the VGG-16 CNN structure [26], to generate a hierarchy of feature maps. Feature maps are then upsampled to the same size by deconvolutional layers, and concatenated in depth, resulting in a representation that comprises equally sized feature maps. Next, the maps are spatially partitioned into blocks of different sizes. The adaptive max-pooling layer that equals to a spatial pyramid pooling layer [29] with only one pyramid level is applied to each block, producing feature vectors of the same length. Following the pooling, feature vector for each block is fed into the fully-connected layers which make the binary classification for that block. The final classification of the whole image is the logical OR of the individual block classification, *i.e.*, as long as one block is classified as containing text, the image is considered text image, otherwise non-text image.

3.2. Image-level feature generation

Recently, feature maps from different convolution layers are combined to make pixel-level prediction tasks successfully [34, 35, 36], as they carry rich and hierarchical information. When implementing, all images are scaled to have a fixed height (500 pixels in our case), keeping their aspect ratios. The feature generation part of MSP-Net consists of five convolutional layers that are derived from the VGG-16 model [26], which has achieved superior performance on image classification. Given the scaled input images, the convolutional layers produce a hierarchy of feature maps, where the map sizes produced by different layers vary. Three deconvolutional layers are respectively connected to the third, fourth and fifth convolutional layers (abbreviated as conv-3, conv-4, and conv-5). Via deconvolution, the maps are upsampled to the same size. The feature representation is then the concatenation in depth of these upsampled maps, which is a hierarchical representation of the whole image.

In a CNN, each convolutional layer has a particular receptive field size [37], indicating the size of image region which every node on the feature maps is path-connected to. Smaller receptive field sizes lead to finer feature granularity, while larger sizes lead to coarser granularity. In our network settings, the receptive field sizes of conv-3 is 40, which favors lower-level and local features. For conv-5, the size is 192, which enables it to describe higher-level global context. As shown in Fig. 3, feature maps (which are upsampled) of conv-3 have higher sensitivities to text strokes and edges, while feature maps of conv-4 and conv-5 favor the whole text regions.

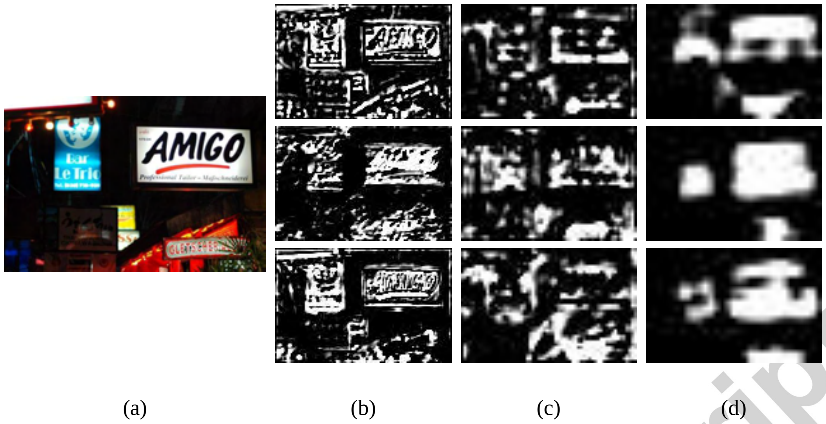


Figure 3: Feature maps of different layers. (a) is an input image, (b), (c) and (d) are feature maps randomly selected from conv-3, conv-4 and conv-5, respectively.

The deconvolutional layers perform strided convolution on feature maps [35]. They upsample input maps with ratios that are roughly the deconvolution strides. With proper strides, we make output feature maps to have identical width and height, so that they can be concatenated in depth.

3.3. Multi-scale spatial partition

Similar to the ROI pooling layer designed for fast feature extraction for each proposal in Fast R-CNN [33], we move the operation of multi-scale spatial partition from image-level space to feature-level space, in order to efficiently obtain the features of each image block. In the partition step, the generated feature maps are spatially partitioned into blocks with respect to several block sizes. We use block sizes of $\frac{w}{N} \times \frac{h}{N}$, where w, h are the width and height of the input feature maps, and N is an integer. Each block size uniformly partitions the maps into N^2 equally sized blocks. Mathematically, the partition is formulated by:

$$F^{ij}(x,y) = F(x + i\frac{w}{N}, y + j\frac{h}{N}), \begin{cases} 0 \leq x < \dfrac{w}{N} \\ 0 \leq y < \dfrac{h}{N}, \end{cases} \tag{1}$$

where $F(x, y)$ denotes the generated feature maps, F^{ij} denotes the block at row j, column i (i, j are indexs of row and column, both of them start from 0 to $N - 1$).

Following [29, 33], each block on the feature maps is associated to a region on the input image:

$$I^{ij}(x,y) = I(x + i\frac{W}{N}, y + j\frac{H}{N}), \begin{cases} 0 \leq x < \dfrac{W}{N} \\ 0 \leq y < \dfrac{H}{N}, \end{cases} \tag{2}$$

where $I(x, y)$ is an input image whose size is $W \times H$. We let the feature block describe its corresponding image region. Although this results in redundant description, since the receptive field for the feature block would be larger than the region we define, this simplifies our formulations, and works well in practice [29, 33]. Furthermore, we perform multi-scale spatial partition by choosing different values for N (e.g., 1, 3, 5 and 7), resulting in feature blocks of different sizes. The feature blocks describe local image regions of different sizes, and they are all used for the following adaptive pooling.

In a neural network, all operations need to back propagate error differentials. The back-propagation of the multi-scale spatial partition operation is formulated by:

$$\delta L/\delta F(x,y) = \sum_{N} \delta L/\delta F^{ij}_{i=\lfloor \frac{x}{N} \rfloor, j=\lfloor \frac{y}{N} \rfloor}(x - i*N, y - j*N), \begin{cases} 0 \leq x < w \\ 0 \leq y < h, \end{cases} \tag{3}$$

where L denotes the loss, the back-propagation on multi-scale spatial partition operation is the sum of back-propagation of each feature block $\delta L/\delta F^{ij}$.

3.4. Block-level representation and classification

Since multiple scale values are used in multi-scale spatial partition (we use 4 scales to partition feature maps into 1×1, 3×3, 5×5 and 7×7 feature blocks, respectively.), the output feature blocks represent corresponding image blocks are of different sizes, which are illustrated in Fig. 2. Hence, we normalize the representation of each image block into the same size for feeding it into the classification sub-network. In order to generate fixed-length feature representation, an adaptive max-pooling layer is adopted. As one scale of spatial partition illustrated in Fig. 4, a block is equally divided into $N_s \times N_s$ sub-blocks ($N_s \times N_s$ denotes the bock number partitioned under the s-th scale, $s = 1$ and $N_s \times N_s = 3 \times 3$ here), in a similar way which an image is divided into blocks. Then, max-pooling operation is applied to every block to generate a feature vector, whose length is N_{map}, which is the depth of the feature map. Last, feature vectors generated from all blocks are concatenated into one block, whose length is then $N_s^2 N_{\text{map}}$.

The spatial partition in a block is similar to the partition on feature maps, described in Sec. 3.3. However, the purpose of dividing blocks into sub-blocks is to capture the spatial relationships within a block, in order to improve the discrimination power of the resulting block-level representation. Essentially, the sub-network that generates block-level representation is a special case of the spatial pyramid pooling layer used in SPP-Net [29]. The spatial pyramid pooling layer consists of several pyramid level of pooling layers, where each pooling layer is adaptive layer that outputs fixed-size feature by divided the feature map into fixed-size bins. In fact, our spatial partition operation is equal to 1 pyramid level of spatial pyramid pooling layer whose partition bin number is $N_s \times N_s$.

After feature extraction for all blocks of an image, we classify the blocks using a single classification sub-network. The classification sub-network is a part of MSP-Net, which consists of three fully-connected layers. Since fixed-length representation of each image block is generated by adaptive max-pooling, all feature vectors can be fed into the classification sub-network in the form of batch processing to make the text/non-text block classification.

Besides, the numbers of dimensions for all block descriptors are the same, so the classification sub-network accepts blocks of arbitrary sizes. Recall that other parts of the network, namely convolutional layers, deconvolutional layers, and spatial partition layers, also accepts arbitrarily-sized input maps. Consequently, MSP-Net classifies input images of arbitrary sizes. This property allows us

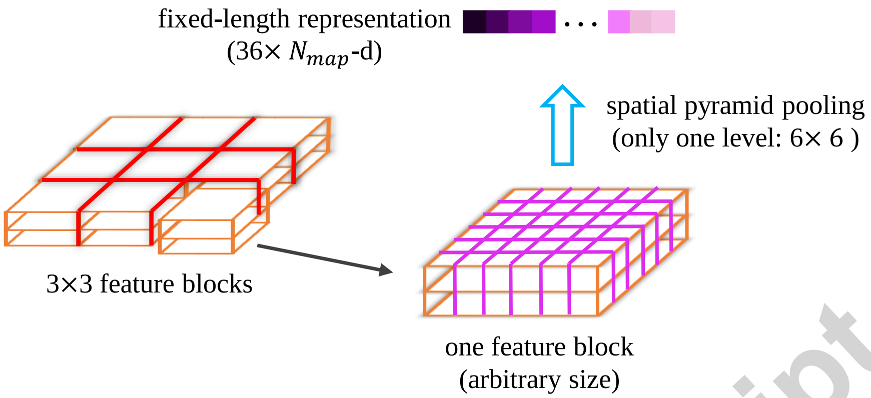


Figure 4: Block-level feature generation.

to directly feed original images into the network during testing, without any cropping or resizing that may cause loss of information.

3.5. Network training

Ground truth. The image blocks that are defined as text blocks must meet two constraint conditions: text area and scale. We use $r1$ denotes the text occupy ratio in one image block, and the height ratio of text lines to the image block represented as $r2$. In our experiments, the value of $r1$ must be over 0.05, as well as $r2$ must be over 0.5.As the dataset not only provides the image-level label but bounding boxes of text lines, we can easily infer the ground truth of all image blocks. As one example illustrated in Fig. 5, the yellow bounding boxes in Fig. 5(b) are the ground truth of text lines, which indicate the text area and scale (or height) of text lines. Therefore, each image block generated by multi-scale spatial partition in Fig. 5(c)∼Fig. 5(f) is defined as positive if it meets two constraints above, otherwise as negative. Besides, if an image block is classified as text block, it not only means the whole image should be considered as text image, but also indicates the coarse position and scale of text.

Loss definition. Due to the binary class output of MSP-Net, we use the cross-entropy loss function as the objective function. Suppose a training image I

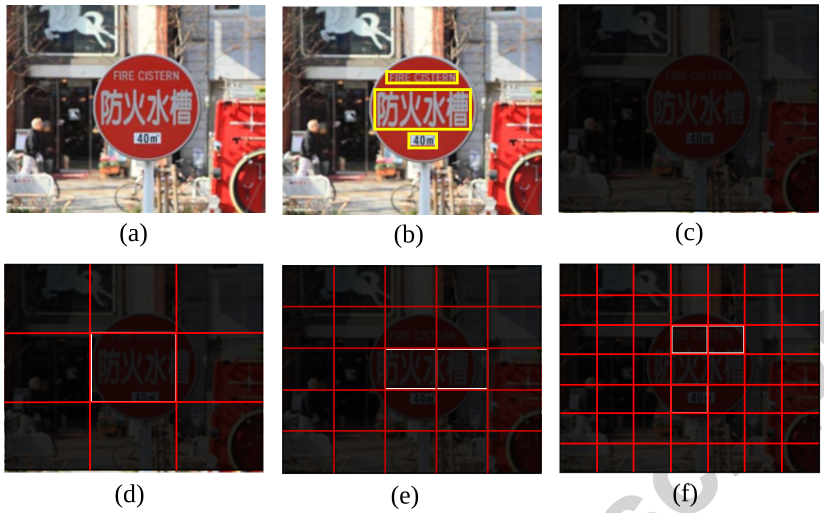


Figure 5: Ground truth of image blocks with different scales. (a) is a natural text image, yellow bounding boxes in (b) show the text lines. Image is partitioned with multiple scales of 1 × 1, 3 × 3, 5 × 5, 7 × 7 in (c), (d), (e), and (f), respectively. The white blocks mean positive and the black blocks are negative.

is partitioned with N image blocks, whose labels are denoted by $\{l_i\}_i^N$. The objective is to minimize the sum cross-entropy loss of all image blocks:

$$L = -\sum_{i=1}^{N}(l_i \log p_i + (1 - l_i)\log(1 - p_i)), \tag{4}$$

where p_i is the probability of i-th image block classified as text block,l_i is the label of i-th image block.

We use the VGG-16 model which is pre-trained on ImageNet [15] to initialize the 5 convolutional stages (first 13 convolutional layers) of MSP-Net. Then, stochastic gradient descent (SGD) is adopted to jointly optimize whole parameters by the back-propagation algorithm. Since the number of text blocks is much smaller than the one of non-text blocks, we use the class-balancing weight as a simple way to offset this imbalance between text/non-text block. Thus, we replace the equation (4) with the following formulation:

$$L = -\sum_{i=1}^{N}(\lambda l_i \log p_i + (1-\lambda)(1-l_i)\log(1-p_i)), \tag{5}$$

where λ denotes the class-balancing weight, whose value is 2/3 in the training stage.

4. Experiments

In this section, we first evaluate the proposed method on several public benchmarks including the TextDis benchmark [32], the ICDAR2003 dataset [38] and Hua's dataset [39]. Then we compare our method with some existing methods, which are either text/non-text image classification methods or general image classification methods. Last, in the discussion part, we evaluate the effects of some parameters in our design.

4.1. Datasets

TextDis benchmark. This dataset is introduced in [32], which contains 7302 text images and 8000 non-text images. The benchmark randomly selects 2000 images for each class to build the testing dataset, and the remaining images are used for training. To our knowledge, this dataset is the first dataset for the discrimination of text and non-text natural image. Due to the large variation in the fonts, scales, colors, languages and orientations of text in the image, this dataset is quite challenging. Precision, recall and F-Measure are used as the evaluation protocol for measuring the results of different algorithms.

ICDAR2003 dataset. 251 camera images are collected and released for evaluating scene text detection methods. Since all images are taken from natural scene, there is still large variation in the fonts, scales and colors of text. The most significant differences from TextDis lie in that the language of text is English only and the orientation of text is horizontal or nearly horizontal.

Hua's dataset. This dataset is a small video text detection benchmark, which contains 42 text frames and 3 non-text frames. Different from natural images, text appearing in text frames usually has regular formats including fonts, scales and positions.

4.2. Implementation details

Table 1: The details of MSP-Net. Each convolutional stage has 2 or 3 convolutional layers. 'k','s' and 'p' mean kernel size, stride, and padding size in convolutional layers. And 'ws' means the window size of pooling layer.

Layers	**Configurations**
conv-1	2×{#map:64, k:3×3, s:1, p:1}
maxpooling	ws:2 × 2, s:2
conv-2	2×{#map:128, k:3×3, s:1, p:1}
maxpooling	ws:2 × 2, s:2
conv-3	3×{#map:256, k:3×3, s:1, p:1}
maxpooling	ws:2 × 2, s:2
conv-4	3×{#map:512, k:3×3, s:1, p:1}
maxpooling	ws:2 × 2, s:2
conv-5	3×{#map:512, k:3×3, s:1, p:1}
deconv-3	#map:128, k:1 × 1, s:1
deconv-4	#map:256, k:4 × 4, s:2
deconv-5	#map:256, k:8 × 8, s:4
muti-scale spatial partition	#bin:{1 × 1, 3 × 3, 5 × 5, 7 × 7}
adaptive max-pooling	#bin:6 × 6
fc-1	#unit:4096
fc-2	#unit:4096
output	#uint:2

Architecture details. The details of our proposed network (MSP-Net) are listed in Table 1. The first 5 convolutional stages are derived from VGG-16 model, feature maps from conv-3, conv-4 and conv-5 are followed with up-sampling layers which are replaced by deconvolutional layers with different strides to make the feature maps have the same size. The multi-scale spatial partition with 4 scales (e.g. 1 × 1, 3 × 3, 5 × 5, 7 × 7) are adopted in the feature map space to efficiently generate features for 84 image blocks. After the spatial pyramid pooling layer with only one level (i.e. 6×6), the feature size of each block is $(128 + 256 + 256) \times 6 \times 6$. Finally, 84 feature blocks together form a team input to the classification sub-network for the final text/non-text block classification. The classification sub-network consist of three fully-connected layers. Naturally, if at least one block is classified as text block, the whole image is treated as text image.

Data preparation. We apply rotation and flipping operations to each training image, and randomly crop 10 image regions with the same aspect ratio for data argumentation. After that, all training image regions are resized to fixed height (500 pixels). Since 4 different scales are used in the layer of multi-scale spatial partition, the heights of image blocks in 4 partition scales correspond to 500, 167, 100 and 71. Due to $r2$(the minimal height ratio of text line in image block) is set to 0.5, one image block regarded as text block must meet the minimal height values: 250, 83, 50, and 10 for 4 partition scales.

Training details. We use stochastic gradient descent(SGD) to fine-tune the MSP-Net whose details are listed in 1 with following parameters: mini-batch size is 1 (due to multi-scale spatial partition, the number of image blocks is 84), learning rate is 1e-6 (divided 10 after each 50K iterations), momentum value is 0.9, and weight decay is 0.0002. Training takes about 10 hours for a single GPU (NVIDIA GTX TitanX). In testing phase, an input image is also resized to the fixed height and fed into the trained network to output 84 block-level prediction results. Furthermore, the MSP-Net is trained on TextDis benchmark, then tested on all datasets.

4.3. Comparison methods

Locality-constrained Linear Coding (LLC). LLC [25] is a useful coding method for image classification. In our paper, we extract dense sift features of 3 different scales (e.g.,8×8, 16×16, 24×24), and the size of codebook clustered by k-means is set to 2048. Besides, the spatial pyramid matching is replaced by global max-pooling, which still achieves a comparable result.

Spatial Pyramid Pooling Network (SPP-Net). The spatial pyramid pooling layer proposed in [29] can generate fixed-size and hierarchical features for image or region in arbitrary sizes, which achieves a quite competitive performance on object detection and recognition. In our comparison experiments, the SPP-Net adopts the same convolutional stages as our proposed method, and the pyramid levels are in 3 scales (e.g., 1×1, 3×3, 5×5). However, the output of SPP-Net is the image-level classification, which is different from our method.

CNN Coding. In our previous work [32], we proposed a method that combines maximally stable extremal region (MSER), convolutional neural network (CNN) and bag of words (BoW) for text image discrimination. This work utilizes the MSER to extract text candidates and feeds them into a trained CNN model to generate visual features, then all features are aggregated by BoW to obtain the final representation for natural image. All the same parameters in [32] are used for this comparison experiment.

In the above methods for the comparison, LLC and SPP-Net only use the information of image label, while the method of CNN Coding uses both image label and text-line bounding box information to classify an image. Therefore, the comparison between MSP-Net and CNN coding is more fair and representative.

4.4. Experiments results

4.4.1. Experiments on TextDis benchmark

In Table 2, the quantitative classification results of different methods on TextDis benchmark are listed. The proposed method (MSP-Net) outperforms CNN Coding by 3.9% in precision, 5.1% in recall and 4.5% in F-measure. And the speed of MSP-Net is more than 3 times faster than CNN Coding. The comparison results between MSP-Net and SPP-Net show that it is hard to achieve satisfied performance, if we directly use the existing framework of convolutional

Table 2: The results of different comparison methods. The metrics including precision, recall, F-measure and time cost are presented.

Methods	Precision	Recall	F-Measure	Time Cost
LLC	0.839	0.774	0.805	0.30s
SPP-Net	0.841	0.839	0.840	0.16s
CNN Coding	0.898	0.903	0.901	0.46s
MSP-Net	0.937	0.954	0.946	0.13s

network to do text/non-text image classification. In order to intuitively illustrate the advanced performance of MSP-Net, we also plot the precision-recall curves of different methods. Note that the MSP-Net can only output the confidence of image block identified as text block, so we use the maximum confidence value of all image blocks to approximate the score of the whole image that is classified as a text image. The curve of MSP-Net in Fig. 6 shows that our method keeps rather high precision even at the range of high recall.

In addition, an important advantage of our proposed method is that text blocks can indicate the coarse position and scale of text appeared in text image. In order to better display this advantage, we keep all pixels of text blocks and remove all non-text blocks. As shown in Fig. 7, text images are successfully classified and their candidate text blocks highlighted with red bounding boxes in the second row are kept. Meanwhile, the majority of text in text images is kept, and the scale (or height) of text line is comparable to the height of block which it belongs to. Different from other comparison methods which obtain only the image-level confidence of text image, our method can provide richer and more helpful information for scene text reading system.

4.4.2. Experiments on ICDAR2003 dataset

ICDAR2003 dataset is a publicly available scene text dataset whose text is focused. We test our proposed method on ICDAR2003 to show that it works well on focused text images. In order to acquire intuitive and fair comparison results of the methods proposed in [10, 11], we use the classification rate and the average processing time (APT) as the metrics.

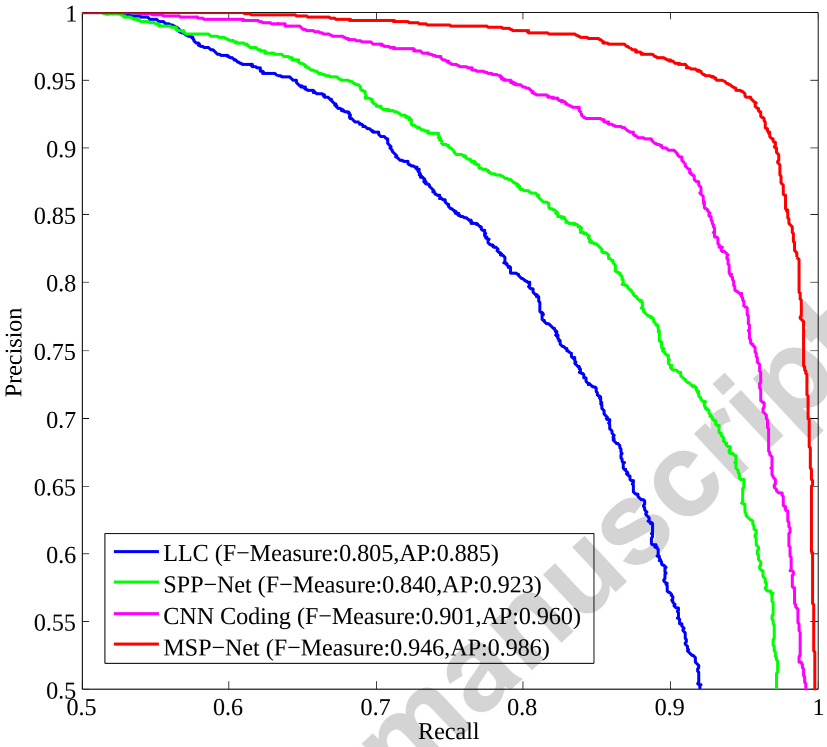
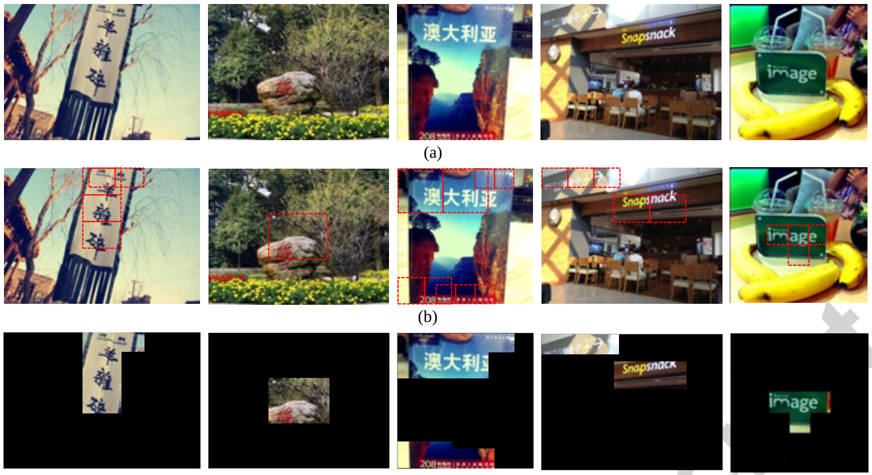


Figure 6: The precision-recall curves of comparison methods.

The results of different methods are list in Table. 3, which show that our method outperforms the video text frame classification methods [10, 11]. What's more, the average processing time of MSP-Net is much less. Some examples of 375 ICDAR2003 dataset are shown in Fig. 8.

4.4.3. Experiments on Hua's dataset

To discuss the generalization of our proposed method in video frames, we test it on Hua's dataset. The same metrics used in Sec. 4.4.2 are utilized to evaluate the performances of different methods. The results in Table. 4 show 380 that our method has obtained the highest classification results. What's more, the average processing time (APT) for each frame is quite faster than the other two methods [11, 10] which are specially designed for text frame classification.



(a)

(b)

(c)

Figure 7: Classification results of TextDis benchmark. (a) are some samples of text images from TextDis benchmark, red bounding boxes in (b) mean the text blocks detected by MSP-Net, (c) keeps all pixels of text blocks.

Table 3: Classification rates of proposed methods and existing methods on ICDAR2003.

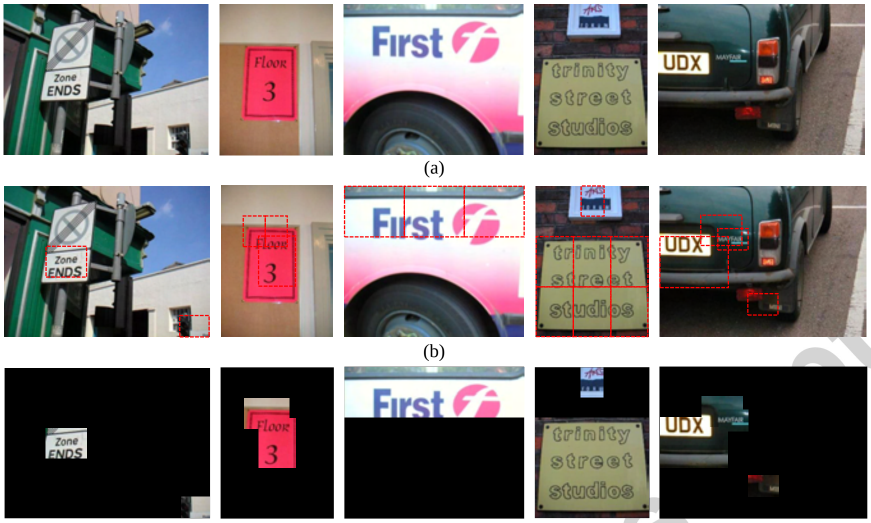
Methods	Text(%)	Error(%)	APT
Proposed method	89.2	10.8	0.132s
Shivakumara et al. [11]	80.97	19.03	1.23s
Shivakumara et al. [10]	81.12	18.88	N.A

In Fig. 9, we show some results of our method tested on Hua's dataset. Most text in Hua's dataset is in the form of caption, which is easily captured, for example video frames at the first, second and third column of Fig. 9. Besides, some scene text in video frames can also be well captured by our proposed method, like video frames in the fourth and fifth columns of Fig. 9.

4.5. Discussion

4.5.1. Effect of feature combination

In our proposed method, features from different convolutional layers are concatenated after up-sampling to generate richer and more hierarchical features. In order to discuss the effect of different groups of feature concatenation,



(a)

(b)

(c)

Figure 8: Classification results of ICDAR2003 dataset. (a) are some samples of text images from ICDAR2003 dataset, red bounding boxes in (b) mean the text blocks detected by MSP-Net, (c) keeps all pixels of text blocks.

we adjust the feature maps from different convolutonal layers and keep other settings of the network. Table 5 list three settings of feature concatenation and performance on the TextDis benchmark. From the listed results, the comparison between Variant-1 and Variant-2 (or MSP-Net) also demonstrates that different feature maps that represent information with different levels can be concatenated to form rich and hierarchical representation for text/non-text image. More feature maps from different convolutional stages are concatenated, the final performance would be enhanced. Since the size of feature map at conv-1 and conv-2 stages is large, which would need more memory and consuming time for feature concatenation, we don't use feature maps from these two convolutional stages.

4.5.2. Effect of multiple scale for spatial partition

Since the large variance of natural text, especially the scale and area, we demonstrate the importance of multi-scale spatial partition through the com-

Table 4: Classification rates of proposed methods and existing methods on Hua's dataset.

Methods	Text(%)	Non-text(%)	APT
Proposed method	100	100	0.127s
Shivakumara et al. [11]	97.62	100	1.05s
Shivakumara et al. [10]	75.54	24.46	2.04s

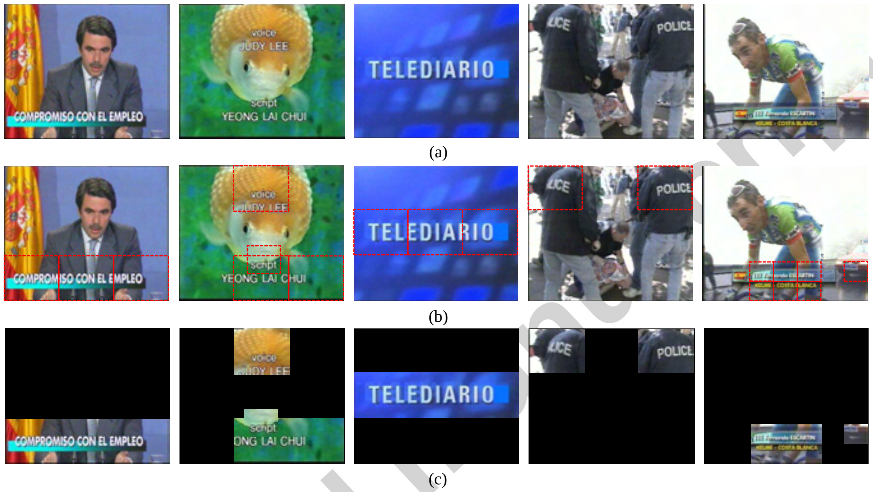


Figure 9: Classification results of Hua's dataset. (a) are some samples of text images from Hua's dataset, red bounding boxes in (b) mean the text blocks detected by MSP-Net, (c) keeps all pixels of text blocks.

parison experiments with several groups of single-layer spatial partition. In practice, we only change the layer of multi-scale spatial partition with different numbers and scales, keeping the same configuration of other layers. In Tab. 6, the result of multi-scale spatial partition outperforms any single spatial partition method. Although the result of single-layer with 7×7 achieves considerable results, the multi-scale partition has obvious improvement. According to the comparison results, we can demonstrate that convolutional neural network can learn richer and more discriminative features for text block discrimination if the range of text scale is proper.

Table 5: Results of different settings of feature combination. Variant-1 only uses the feature maps from 5-th convolutional stages and Variant-2 combines the feature maps from 4-th and 5-th stages.

Variants	Settings	Precision	Recall	F-Measure	Time Cost
Variant-1	conv-5	0.915	0.890	0.905	0.106s
Variant-2	conv4 + conv5	0.924	0.945	0.936	0.118s
MSP-Net	conv-3 + conv-4 + conv-5	0.937	0.954	0.946	0.130s

Table 6: Effect of multiple scale for spatial partition.

Scale	Precision	Recall	F-Measure
1×1	0.825	0.819	0.822
3×3	0.870	0.864	0.867
5×5	0.892	0.921	0.906
7×7	0.931	0.914	0.922
$1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7$	0.937	0.954	0.946

4.5.3. Comparing with text detection methods

Table 7: Classifying text/non-text images on TextDis benchmark with different text detection methods.

Methods	Precision	Recall	F-Measure
MSP-Net	0.937	0.954	0.946
Zhang et al. [40]	0.754	0.979	0.851
Yao et al. [6]	0.808	0.902	0.853
Neumann et al. [20]	0.525	0.984	0.685

In this section, we compare MSP-Net with some existing natural text detection methods on classifying text/non-text image, which shows the effectiveness and efficiency of our proposed method. Similar with the classification mechanism of MSP-Net, text detection methods classify one natural image as text image as long as one text line on it is detected. The results of different text detection methods on TextDis benchmark are listed in Tab. 7. The MSP-Net obtain the highest accuracy as well as the least time.

Table 8: Time cost between **Only Text Detection** and **MSP-Net + Text Detection** on TextDis benchmark.

Methods	Only Text Detection	MSP-Net + Text Detection
Zhang et al. [40]	2.10s	0.85s
Yao et al. [6]	5.00s	2.10s
Neumann et al. [20]	0.94s	0.46s

Besides, we find a interesting phenomenan that the time cost of text detection would be largely decreased if we use the MSP-Net to eliminate the non-text images before. In the Tab. 8, we find the speeds of text detection methods on TextDis benchmark are about more than doubled.

4.6. Limitations of the proposed method

While our proposed method outperforms other compared methods, there still exists some failure cases. Text in difficult natural conditions would get wrong classification using our proposed method. For example, text in Fig. 10(a) is in the condition of low illumination, while text in Fig. 10(b) are exposed. And some regular curves, bricks or windows in Fig. 10(c),Fig. 10(d) are similar to text, and would make false positive results. Due to the rigid spatial partition , the majority of text is kept after text/non-text block classification, but sometimes the remaining text is fragile if some text blocks are misclassified, shown in Fig. 10(e)(f). In other way, the proposed method is based on the framework of convolutional neural network, and therefore its time cost is limited to GPU.

5. Conclusion

In this paper, we have proposed a novel architecture of convolutional neural network (named MSP-Net) for text/non-text image classification. The MSP-Net takes input as a whole image and outputs block-level classification results in an end-to-end manner. The results on several datasets have demonstrated the

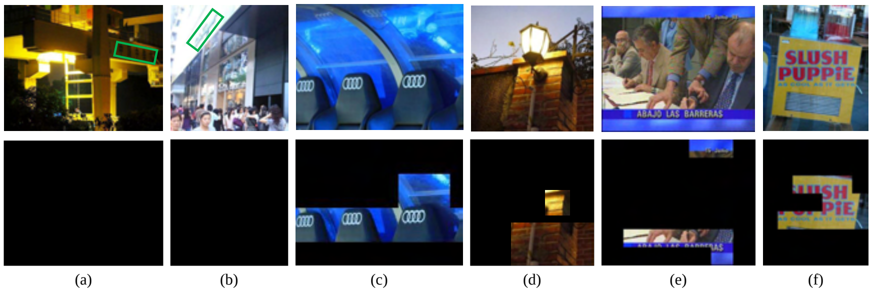


Figure 10: Some failure cases. (a),(b) are text images in difficult conditions. Some curves or objects in (c),(d) are similar to text. Some true text blocks in (e) and (f) are eliminated, which make the remaining text is fragile.

robustness and effectiveness of our proposed method. Besides, one image block classified as text block can also coarsely indicate the scale and position of text, which is helpful to scene text reading. The combination of text/non-text image classification with scene text reading system for mining scene text semantics from the large scale images/videos on the Internet is worthy of exploration in our future work.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China (NSFC) No. 61222308 and No. 61573160, and in part by Program for New Century Excellent Talents in University (No. NCET-12-0217).

References

[1] Y. Zhu, C. Yao, X. Bai, Scene text detection and recognition: recent advances and future trends, Frontiers of Computer Science 10 (1) (2016) 19–36.

[2] Y. Y. Tang, S.-W. Lee, C. Y. Suen, Automatic document processing: a survey, Pattern Recognition 29 (12) (1996) 1931–1952.

[3] M. Khayyat, L. Lam, C. Y. Suen, Learning-based word spotting system for arabic handwritten documents, Pattern Recognition 47 (3) (2014) 1021–1030.

[4] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, International Journal of Computer Vision 116 (1) (2016) 1–20.

[5] Z. Zhang, W. Shen, C. Yao, X. Bai, Symmetry-based text line detection in natural scenes, in: Proc. of CVPR, 2015, pp. 2558–2567.

[6] C. Yao, X. Bai, W. Liu, A unified framework for multioriented text detection and recognition, IEEE Transactions on Image Processing 23 (11) (2014) 4737–4749.

[7] X. C. Yin, X. Yin, K. Huang, H. Hao, Robust text detection in natural scene images, IEEE Transactions on PAMI 36 (5) (2014) 970–983.

[8] H. Hase, T. Shinokawa, M. Yoneda, C. Y. Suen, Character string extraction from color documents, Pattern Recognition 34 (7) (2001) 1349–1365.

[9] B. Shi, X. Bai, C. Yao, Script identification in the wild via discriminative convolutional neural network, Pattern Recognition 52 (2016) 448–458.

[10] P. Shivakumara, A. Dutta, T. Q. Phan, C. L. Tan, U. Pal, A novel mutual nearest neighbor based symmetry for text frame classification in video, Pattern Recognition 44 (8) (2011) 1671–1683.

[11] N. Sharma, P. Shivakumara, U. Pal, M. Blumenstein, C. L. Tan, Piecewise linearity based method for text frame classification in video, Pattern Recognition 48 (3) (2015) 862–881.

[12] E. Indermuhle, H. Bunke, F. Shafait, T. Breuel, Text versus non-text distinction in online handwritten documents, in: Proc. of SAC, 2010, pp. 3–7.

[13] A. Delaye, C.-L. Liu, Text/non-text classification in online handwritten documents with conditional random fields, Pattern Recognition 321 (2012) 514–521.

[14] A. Delaye, C. Liu, Contextual text/non-text stroke classification in online handwritten notes with conditional random fields, Pattern Recognition 47 (3) (2014) 959–968.

[15] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[16] P. R. Cavalin, R. Sabourin, C. Y. Suen, A. S. Britto Jr, Evaluation of incremental learning algorithms for hmm in the recognition of alphanumeric characters, Pattern Recognition 42 (12) (2009) 3241–3253.

[17] X. Bai, C. Yao, W. Liu, Strokelets: A learned multi-scale mid-level representation for scene text recognition, IEEE Transactions on Image Processing 25 (6) (2016) 2789–2802.

[18] X.-X. Niu, C. Y. Suen, A novel hybrid cnn–svm classifier for recognizing handwritten digits, Pattern Recognition 45 (4) (2012) 1318–1325.

[19] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proc. of CVPR, IEEE, 2010, pp. 2963–2970.

[20] L. Neumann, J. Matas, A method for text localization and recognition in real-world images, in: Proc. of ACCV, 2010, pp. 770–783.

[21] N. Dalal, B. Triggs, Histograms of oriented gradients for human detecgtion, in: Proc. of CVPR, Vol. 1, 2005, pp. 886–893.

[22] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[23] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition 29 (1) (1996) 51–59.

[24] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proc. of CVPR, 2010, pp. 3304–3311.

[25] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proc. of CVPR, 2010, pp. 3360–3367.

[26] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. of ICLR, 2015.

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

[28] W. Shen, X. Wang, Y. Wang, X. Bai, Z. Zhang, Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection, in: Proc. of CVPR, 2015, pp. 3982–3991.

[29] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proc. of ECCV, 2014, pp. 346–361.

[30] N. G. Alessi, S. Battiato, G. Gallo, M. Mancuso, F. Stanco, Automatic discrimination of text images, in: Proc. of SPIE, 2003, pp. 351–359.

[31] V. Vidya, T. R. Indhu, V. K. Bhadran, Classification of handwritten document image into text and non-text regions, in: Proc. of ICSIP, 2012, pp. 103–112.

[32] C. Zhang, C. Yao, B. Shi, X. Bai, Automatic discrimination of text and non-text natural images, in: Proc. of ICDAR, 2015, pp. 886–890.

[33] R. Girshick, Fast r-cnn, in: Proc. of ICCV, 2015, pp. 1440–1448.

[34] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Hypercolumns for object segmentation and fine-grained localization, in: Proc. of CVPR, 2015, pp. 447–456.

[35] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. of CVPR, 2015, pp. 3431–3440.

[36] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proc. of ICCV, 2015, pp. 1395–1403.

[37] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[38] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, Icdar 2003 robust reading competitions, in: Proc. of ICDAR, 2003, p. 682.

[39] X.-S. Hua, L. Wenyin, H.-J. Zhang, An automatic performance evaluation protocol for video text detection algorithms, IEEE Transactions on Circuits and Systems for Video Technology 14 (4) (2004) 498–507.

[40] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: Proc. of CVPR, 2016.