Geometry-Aware Scene Text Detection with Instance Transformation Network

Fangfang Wang[1*] Liming Zhao[1*] Xi Li[1†] Xinchao Wang[2] Dacheng Tao[3]
[1]College of Computer Science, Zhejiang University, China
[2]Department of Computer Science, Stevens Institute of Technology, USA
[3]SIT, FEIT, University of Sydney, Australia

Abstract

Localizing text in the wild is challenging in the situations of complicated geometric layout of the targets like random orientation and large aspect ratio. In this paper, we propose a geometry-aware modeling approach tailored for scene text representation with an end-to-end learning scheme. In our approach, a novel Instance Transformation Network (ITN) is presented to learn the geometry-aware representation encoding the unique geometric configurations of scene text instances with in-network transformation embedding, resulting in a robust and elegant framework to detect words or text lines at one pass. An end-to-end multi-task learning strategy with transformation regression, text/non-text classification and coordinates regression is adopted in the ITN. Experiments on the benchmark datasets demonstrate the effectiveness of the proposed approach in detecting scene text in various geometric configurations.

1. Introduction

As an important and challenging problem in computer vision, scene text detection aims at accurately localizing text regions within an image of natural scene, and has a wide range of applications such as image retrieval, scene parsing, and blind-navigation. Despite the advanced techniques in documental and digital text detection, scene text detection is still challenging since text in the wild often appears to be in the forms of complicated geometric layout like random orientation and large aspect ratio.

In recent literature, various convolutional neural network (CNN) based methods [20, 9, 30, 32, 26, 22] have been proposed to detect scene text. Most of these works are built on successful generic object detection frameworks, such as proposal based two-stage frameworks like Faster-RCNN [24] and end-to-end one stage detectors like SSD [19], taking words or text lines as a special case of object.

Figure 1. Demonstration of regular representation and geometry-aware representation. For clear illustration of physical location, we use input images to represent feature maps. The first row shows the fixed receptive fields on feature maps of regular representations (yellow and red dotted grids). The second row shows the adaptive receptive fields of geometry-aware representations.

Different from generic objects, scene text usually possesses some particular geometric configurations such as large aspect ratio, random orientation, ranging scale, etc. To this point, proposal based detection frameworks use extensive proposals with limited geometric configurations to recall text, which face an enormous search space and lack generalization ability. One-stage end-to-end detectors usually generate feature maps through a fully convolutional neural network and produce representations through standard convolution with a fixed receptive field on the feature maps for all text instances. Such modeling is usually not suitable for scene text since it is incapable of well encoding their drastically varying geometric distributions. Therefore, how to effectively perform adaptive geometry-aware modeling for scene text representation with an end-to-end learning scheme is a key issue to solve.

Motivated by the above observations, we propose an adaptive geometry-aware representation learning scheme customized for scene text and incorporate it into a novel end-to-end network, Instance Transformation Network (ITN), to effectively detect multi-scale, multi-oriented and multi-lingual scene text. As illustrated in Figure 1, a regular representation is generated by standard 7×7 convolution operation with fixed receptive field, while a geometry-

*Authors contributed equally, {fangfangliana,zhaoliming}@zju.edu.cn
†Corresponding author, xilizju@zju.edu.cn

aware representation is convolved from a receptive field which is adaptive to the spatial layout of text area on the feature map. The generation of geometry-aware representation is realized by warping the regular convolutional sampling grid so that its receptive field can suitably cover the text region. The warping procedure is carried out under the guidance of the rigid geometric transformation from the regular sampling grid to the adaptive sampling grid. To learn adaptive geometry-aware representation, specific transformations are estimated for specific text instances and embedded in a convolutional layer in our Instance Transformation Network to perform instance-level modeling for scene text, which explores the unique geometric configurations for each instance. The ITN incorporates geometry-aware representation learning with an in-network transformation embedding module and perform joint optimization among instance transformation regression, text/non-text classification and coordinate regression, resulting in a robust and elegant framework to detect scene text at one pass.

The ITN is advantageous as follows. First, our network combines adaptive representation and concise one-pass structure by performing geometry-aware modeling for scene text. Second, the geometry-aware representation encodes the instance-level geometric configurations of scene text, which benefits both classification and regression tasks. Third, our network directly outputs tight word-level or line-level detections without complicated post-processing procedures like clustering, merging, or connecting of segments.

The contributions of this work are threefold:

- We introduce a geometry-aware representation customized for scene text which encodes the unique geometric configurations of text instances, allowing accurate scene text modeling in a one-pass structure.
- We present an in-network transformation embedding module which can be easily incorporated into other convolutional neural networks to generate adaptive representations for scene text.
- We propose an end-to-end Instance Transformation Network for scene text detection with geometry-aware representation learning. The proposed network is able to effectively detect scene text in various scenarios (e.g., multi-scale, multi-orientation, and multi-language) without complicated post-processing.

2. Related Work

Scene text detection has been widely studied in the past few years. In this section, we review related works of traditional methods and deep learning based methods on scene text detection, and then we review most related works on learning transformation in convolutional neural networks.

Traditional methods Traditional methods on scene text detection are mainly bottom-up approaches which focus on stroke or character-level structures. The two mainstreams of them are connected-components based approaches and sliding-window based methods. Connected-components based methods like Maximally Stable Extremal Regions (MSER) [23], Stroke Width Transform (SWT) [3] and their extensions [10, 11, 37, 38, 33] extract stroke or character candidates by filtering pixels according to low-level features (e.g.color, intensity, gradient). This method is impressively fast but the following negative suppression and text line construction require complex post-processing procedures. Sliding window based approaches [29, 31, 14, 28] densely shift a scanning window through locations and scales of an image to detect character candidates. Then windows are classified into text and non-text with pre-trained classifiers, which is computationally expensive.

Deep learning based methods In recent years, the mainstream of scene text detection has altered from character-level bottom-up methods to CNN-based detection systems [18, 9, 12, 5, 8, 40, 7] which consider words or text lines as a specially case of objects. Methods based on segmentation networks like FCN [21] usually generate text-salient maps first and use geometric techniques to calculate coordinates [36, 39]. Approaches in this manner translate noisy prediction maps to detections with several post processing steps. Methods [22] built on a proposal based detection system (e.g., Faster-RCNN) seek to incorporate orientation in the proposal stage and represent text with Region-of-Interest (RoI) pooling features [4]. Other approaches [30, 32, 26] detect text segments with a one-stage RPN [24] or SSD [19]-style detector, then utilize hand-crafted steps like connecting, clustering and merging to acquire words or text lines. Compared with previous approaches using regular representations, our method focuses on learning geometry-aware representations for scene text in a one-pass network.

Transformation learning networks The idea of learning spatial transformation in a network of our method is similar to STN [13]. STN performs global transformation on the entire feature map, while we apply instance-level transformations for multi-target detection. Inspired by recent work Deformable Convolutional Networks [2], we embed transformations in a convolutional layer to produce transformed representation. Notably, unlike unsupervised general transformation adopted in [2], the deformation of feature sampling grid in the ITN is constrained by rigid geometric transformation learnt under supervision, which is tailored for scene text detection.

3. Geometry-aware Scene Text Detection

The fundamental difficulty in detecting scene text lies in its drastically changeable geometry configurations including scale, orientation and aspect ratio. Mainstream regu-

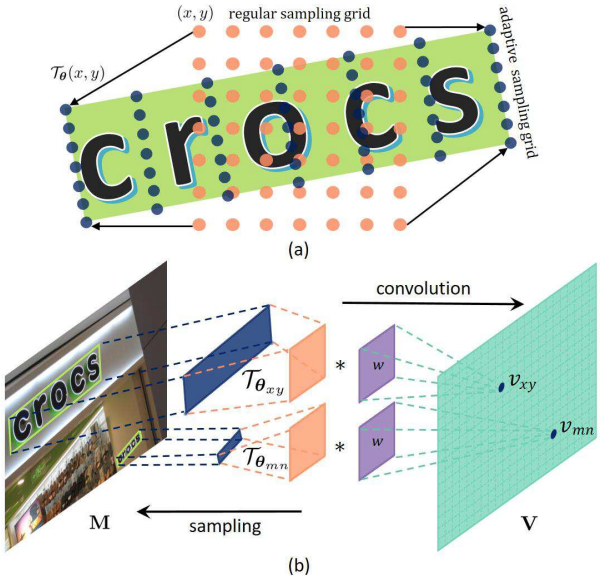

Figure 2. Illustration of geometry-aware representation. (a) shows the warping of the sampling grid: the orange dots show a regular 7×7 sampling grid and the dark blue dots are the warped adaptive sampling grid. (b) illustrates the generation of output feature map $\mathbf{V}$ from input feature map $\mathbf{M}$ (visualized as the input image). The orange and blue patches are simplified form of regular and adaptive sampling grids.

lar representation learnt in standard CNN models is incapable of well encoding the unique geometric distributions of scene text. So in this work, we propose an Instance Transformation Network (ITN) to learn geometry-aware representation tailored for scene text in an end-to-end network.

In this section, we present details of our method. Specifically, we first introduce the geometry-aware representation. Then we illustrate the framework of the ITN including the in-network transformation embedding module. The optimization of the ITN is introduced at last.

3.1. Geometry-aware Representation

Given an input feature map $\mathbf{M}$ generated by a backbone fully convolutional neural network, we aim to produce an output feature map $\mathbf{V}$ for the text classification and coordinate regression tasks with a convolutional layer. The classification and regression tasks are performed on the feature vector $\mathbf{v}_{xy}$ at the pixel location (x, y) in $\mathbf{V}$, convolved by $\mathbf{w}$ of kernel size $(2k+1) \times (2k+1)$ (e.g., 7×7) on $\mathbf{M}$. A standard convolution operation is defined as:

$$\mathbf{v}_{xy} = \sum_{p=-k}^{k} \sum_{q=-k}^{k} \mathbf{w}(p, q)\mathbf{M}(x+p, y+q). \quad (1)$$

Such feature $\mathbf{v}_{xy}$ is sampled from a fixed square patch of size $(2k+1) \times (2k+1)$ in the input feature map $\mathbf{M}$ for all locations. Thus for all text instances in an input image, their representations share the same shape of receptive field. Features learnt with this strategy can hardly be both intact and clean when the target is of large aspect ratio and inclined. To address this problem, we seek to guide the sampling of features with a transformation to generate geometry-aware representation for text instances in the input image.

For the choice of the particular transformation, we adopt affine transformation. We observe that scene texts are often rigid rectangle targets in actual world and their deformations in a picture obey projective transformation. A projective transformation matrix is composed of two parts, an affine matrix and a projection vector. In practice, directly learning a projective transformation in a network is hard since the deformation is sensitive to the parameters of projection vector. So we choose the affine transformation to encode the essential characteristics of the text deformation (e.g., rotation, translation, scale and shear), which is easier to learn and sufficient to approximate the real case.

We estimate an affine transformation $\mathcal{T}_{\boldsymbol{\theta}_{xy}}$ parameterized by $\boldsymbol{\theta}_{xy}$ at pixel location (x, y) in $\mathbf{V}$ and embed it in the feature sampling stage to adaptively fit the current receptive field to the surrounding text instance area. Particularly, we do not care about the transformation if the current location falls into no instance area. The transformation embedment is realized by warping the regular sampling grid to an adaptive sampling grid under the guidance of $\mathcal{T}_{\boldsymbol{\theta}_{xy}}$ through pixel-to-pixel alignment:

$$\mathbf{v}_{xy} = \sum_{p=-k}^{k} \sum_{q=-k}^{k} \mathbf{w}(p, q)\mathbf{M}(\mathcal{T}_{\boldsymbol{\theta}_{xy}}(x+p, y+q)), \quad (2)$$

where $\mathcal{T}_{\boldsymbol{\theta}}(u, v) = (u', v')$, and

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \mathtt{A}_{\boldsymbol{\theta}} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad (3)$$

in which $\mathtt{A}_{\boldsymbol{\theta}}$ is a 2D affine transformation matrix parameterized by a 6-dimensional vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$. Particularly, transformations are estimated for all the pixel locations in $\mathbf{V}$, forming a 3-order tensor $\boldsymbol{\Theta}$ with the same spatial resolution such that $\boldsymbol{\theta}_{xy} = \boldsymbol{\Theta}(x, y)$ is the transformation parameters for location (x, y).

The procedure of warping a regular sampling grid to an adaptive sampling grid is shown in Figure 2 (a). Compared with the standard sampling strategy in Eq. 1, the convolved feature $\mathbf{v}_{xy}$ in Eq. 2 is extracted from a region which is able to cover the whole text instance without introducing redundant background, and thus is more suitable for the tasks of text classification and regression. The process of generating output feature map $\mathbf{V}$ from input feature map $\mathbf{M}$ is illustrated in Figure 2 (b).

We compute $\mathbf{M}(u', v')$ on a real valued location (u', v') via bilinear interpolation which allows back-propagation through both transformation matrix and input feature map.

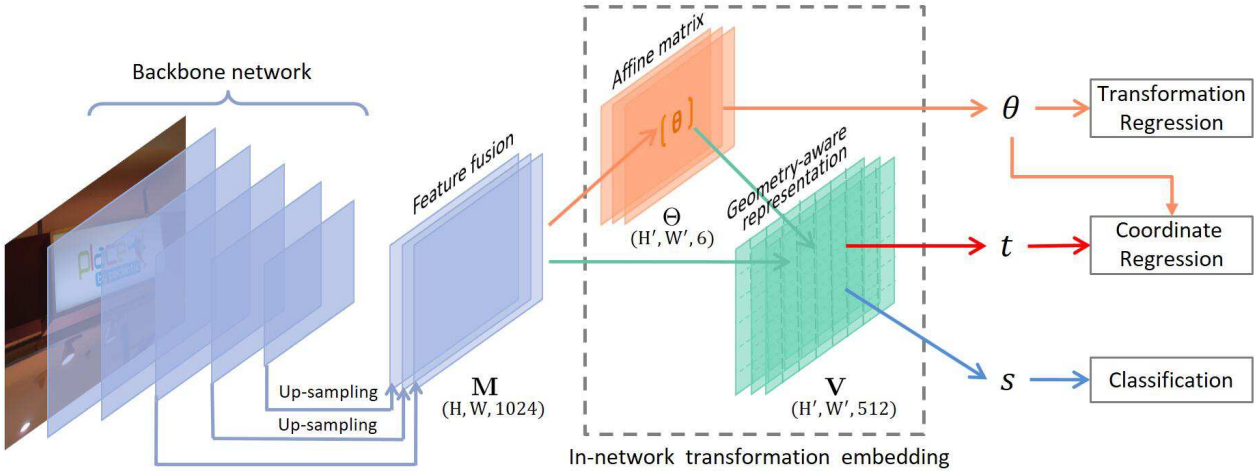

Figure 3. The architecture of the ITN. The ITN consists of three parts: convolutional feature extraction and fusion layers shown in the left where feature maps of three different scales are fused; in-network transformation embedding module shown in the gray dashed box where instance-level affine transformations are predicted and embedded; and multi-task learning shown in the right where classification, transformation regression and coordinate regression are jointly optimized.

The gradients with respect to $\mathbf{\Theta}$ and $\mathbf{M}$ are defined in a similar way as [13] for end-to-end training.

An important property of the predicted transformations is that they are *translational variant*. For pixel locations falling into different instance regions, their estimated transformations are different to adapt to their corresponding instances. For pixel locations who fall into the same instance, their predicted transformations are expected to result in the same sampling grid covering this particular instance. That is to say, affine transformations estimated at different locations for the same instance will share parameters with respect to rotation, scale, and shear and only differ in translation. Despite the changes in translational parameters, we consider the predicted transformations as instance-level transformations in this work.

3.2. Instance Transformation Network

The ITN is an end-to-end detection network which takes in an input image $\mathbf{I}$ and output word-level or line-level quadrilateral detections $\mathbf{D}$. Each detection contains a quadrilateral $\boldsymbol{d}$ represented by four clockwise corners (starting from the left-top vertex) in the form of $(d_{1x}, d_{1y}, d_{2x}, d_{2y}, d_{3x}, d_{3y}, d_{4x}, d_{4y})$ and its confidence score s. As depicted in Figure 3, the ITN includes mainly three parts: convolutional feature extraction and fusion, in-network transformation embedding introduced in Section 3.1 and multi-task learning.

Feature maps from different layers of a fully convolutional network are fused into a feature map $\mathbf{M}$ to detect targets in different scales. Then geometry-aware representation is produced by an in-network transformation embedding module. In this module, the parameters of affine transformations $\mathbf{\Theta}$ are generated by adding a branch of a 3×3 convolutional layer over the input feature map $\mathbf{M}$. The output of this branch has the same spatial resolution as the output feature map $\mathbf{V}$. Predicted transformations and the input feature map $\mathbf{M}$ are then fed into the convolution operation (7×7 kernel size) defined in Eq. 2.

Then the ITN performs multi-task learning which parallels three branches: text/non-text classification, coordinate regression and transformation regression. Taking one pixel location as an example, geometry-aware representation is used for classification, and the estimated transformation is simultaneously regressed. The coordinate regression demands both the representation and transformation. The bounding-box F of the adaptive sampling grid is transformed from the bounding-box E (a fixed 7×7 square in ITN) of the regular sampling grid under the guidance of the transformation. F is taken as a coarse estimation of the detection, and offset between F and the ground-truth is the target for coordinate regression. The three tasks are all trained under supervision.

The network produces three outputs for each pixel location in feature map $\mathbf{V}$, including: (a) the predicted probability s of the feature extracted at the current location being a "text"; (b) the predicted parameters of an affine transformation matrix $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$; and (c) the predicted coordinate offset $\boldsymbol{t} = (t_{1x}, t_{1y}, t_{2x}, t_{2y}, t_{3x}, t_{3y}, t_{4x}, t_{4y})$. In the test stage, the detections can be recovered from the network outputs:

$$\boldsymbol{d} = \mathcal{T}_{\boldsymbol{\theta}}(E) + \boldsymbol{t}, \tag{4}$$

here we abuse the notation of $\mathcal{T}_{\boldsymbol{\theta}}(E)$ to indicate iterative functioning on the four vertices of E. We apply a Soft-NMS [1] step with the linear rescoring function and no thresholding to decay the scores of overlapped detections to generate final detections $\mathbf{D}$.

3.3. Optimization

Training targets We divide pixels in the feature map $\mathbf{V}$ into three categories: positive pixels who lie in the region of a ground-truth instance, negative pixels who are excluded by all the ground-truth instances, and at last, silent pixels who fall in the boundary area. In the training stage, positive pixels are involved in all classification and regression tasks, while negative pixels are only involved in the classification task. Silent pixels contribute none to all the tasks.

For the *text/non-text classification* task, the label s^* at a positive pixel location, negative pixel location and silent pixel location is set to 1 (text), 0 (non-text), and -1 (ignore), respectively.

For the *transformation regression* task, the ground-truth affine matrix at each positive pixel location is calculated by projecting the bounding-box E of the regular sampling grid to the approximated parallelogram of its targeting ground-truth bounding-box G which is already re-scaled to the feature map $\mathbf{V}$ by solving the linear least-squares.

Additionally, we observe that the feature extracted from a part of a whole word or a text line is still a valid "text" due to the natural gaps between words or characters, which makes the boundaries of a word or a text line difficult to determine. To this point, we note that it is not trivial to let the detector see the whole boundaries of the target by including some background context into the effective receptive field. To generate representations with context, we expand the ground-truth text area with a scale factor of 1.2 to compute the target affine transformation parameter $\boldsymbol{\theta}^*$.

For the *coordinate regression* task, we use the learnt transformation $\mathcal{T}_{\boldsymbol{\theta}}$ at each positive pixel location and E to generate a bounding-box $F = \mathcal{T}_{\boldsymbol{\theta}}(E)$ in the feature map $\mathbf{V}$ as a coarse estimation of G. Then F is refined through coordinate regression using the geometry-aware representation. The regression target at each positive pixel location is defined as $\boldsymbol{t}^* = G - F$ where G and F are in the form of an 8-dimensional vector containing four clock-wise vertices.

The offset regression strategy is also adopted in proposal based methods like Fast-RCNN [4] and Faster-RCNN [24], which regress the offsets between the ground-truth bounding-boxes and the proposal coordinates. The bounding-box F and the region it covers in our method are analogous to their proposal and region of interest (RoI). The difference is that in proposal based methods, their RoIs from the proposal generation stage remain unchanged through the regression and thus RoI features are extracted from fixed regions in the feature map. But in the ITN, F and its constrained sampling grid evolves with the regressions, producing dynamically adaptive representation which benefit both classification and coordinate regression.

Loss function The ITN adopts a multi-task learning strategy. The overall loss function consists of three parts: text/non-text classification, coordinate regression, and transformation regression:

$$\begin{aligned} L(\{s_i\},\{\boldsymbol{t}_i\},\{\boldsymbol{\theta}_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(s_i, s_i^*) \\ + \frac{\lambda_1}{N_{coor}}\sum_i [s_i^* = 1] L_{coor}(\boldsymbol{t}_i, \boldsymbol{t}_i^*) \\ + \frac{\lambda_2}{N_{trans}}\sum_i [s_i^* = 1] L_{trans}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^*), \end{aligned} \tag{5}$$

where i enumerates the location indices in the final feature map (here we use i to replace (x, y) for simplicity), and $[\cdot]$ is the indicator function. L_{cls}, L_{coor} and L_{trans} are loss functions for text/non-text classification, coordinate regression and transformation regression, respectively. These three terms are normalized by N_{cls}, N_{coor} and N_{trans}, where N_{cls} is the number of positive and negative pixels and $N_{coor} = N_{trans}$ is the number of positive samples. λ_1 and λ_2 are the balancing weights, in which λ_1 is set to 0.1 and λ_2 is set to 0.01 empirically. s_i^*, $\boldsymbol{t}_i^*$ and $\boldsymbol{\theta}_i^*$ are the supervision for the three tasks.

Similar to the RPN [24], L_{cls} is a two-class softmax loss for classification task. For coordinate regression and transformation regression, we use $L_{coor}(\boldsymbol{t}_i, \boldsymbol{t}_i^*) = R(\boldsymbol{t}_i - \boldsymbol{t}_i^*)$ and $L_{trans}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^*) = R(\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*)$ where R is the smooth-L_1 loss function defined in [4].

4. Experiments

4.1. Implementation Details

In the proposed method, the layers in feature extraction stage are initialized with the backbone models (ResNet-50 [6] and VGG-16 [27]) pretrained on ImageNet [25]. New layers are initialized by random weights with Gaussian distribution of 0 mean and 0.01 standard deviation except the convolutional layer for matrix prediction which is initialized to produce an identity matrix at each spatial location. Each input image is resized such that its shorter side is 600 pixels similar to [24] during both training and test stage. The ITN is trained end-to-end by using the standard stochastic gradient descent (SGD) with back-propagation with a momentum of 0.9 and weight decay of 0.0005. We adopt the "step" policy in Caffe [15] to adjust learning rate. The base learning rate is 10^{-3} and decays by a weight of 0.1 every $50k$ iterations. Training a model of the maximum iteration $100k$ with a batch size of 1 takes less than a day. The ITN is implemented in Caffe framework [15] and run on a server with 3.0GHz CPU and Titan X GPU.

Pixel labeling The convolutional feature map $\mathbf{V}$ is scaled by a factor of $1/h$ against the input image $\mathbf{I}$ as the convolution stride accumulates. So we associate each pixel location in the $\mathbf{V}$ to a grid B of size $h \times h$ on the input image ($h = 8$ in our case). We define the intersection ratio of a grid B

with a ground-truth bounding-box G' as $|B \cap G'|/|B|$. We define a positive pixel as whose associated grid has intersection ratio with a ground-truth instance over 0.7 or whose grid has the highest intersection ratio with a ground-truth bounding-box to make sure that both large and small instances can be recalled. Negative pixels are defined as those whose highest intersection ratio with all instances is lower than 0.3 among the rest of the pixels. Other pixels are categorized as silent pixels.

Data augmentation An online data augmentation strategy is adopted in the training stage of our method. Training images are randomly rotated by $[-15, 15]$ degrees and scaled by a factor of $[0.8, 1.2]$. We abandon the shifted sample when the new ground-truths are out of the frame and use the original image instead.

Hard negative mining Considering that negative pixels are usually much more than positive pixels in the feature map, we utilize online hard negative mining similar to [26] to benefit the classification task. The largest ratio between negative pixels and positive pixels is set to 3:1 and the rest of the pixels are ignored.

Feature fusion To tackle with multi-scale targets, we fuse feature maps of different scales. For ResNet-50, we take *res_5c*, *res_4f* and *res_3d*, which are the final outputs of the last three blocks in ResNet-50. Specifically, *res_5c* and *res_4f* are up-sampled to the scale of *res_3d* ($1/8$ of the input image), then the three feature maps are concatenated. One additional convolutional layer is added following the concatenated feature maps to smooth the feature space. Similarly, we fuse the outputs of *conv5_3* and *conv4_3* for VGG-16 (scale is also $1/8$ of the input image).

4.2. Datasets

We evaluate the in-network transformation embedding module and the ITN on standard multi-oriented scene text detection benchmarks: MSRA-TD500 [35] and ICDAR2015 [17].

ICDAR2015 ICDAR2015 is an incidental scene text dataset which contains 1000 training images and 500 test images. This dataset is very challenging because the images are collected by wearable cameras without the users taking any specific prior action. So texts in the images appear in random scale, orientation, location, viewpoint and blurring. The annotations of ICDAR2015 are provided as quadrilateral bounding-boxes represented by 8 coordinates of four clock-wise corners. Word-level detections are required in evaluation stage.

MSRA-TD500 MSRA-TD500 contains 300 training images and 200 test images of multi-oriented texts. It is a multi-lingual dataset including English and Chinese. Unlike ICDAR2015, the annotations of MSRA-TD500 are at line-level which are represented by aligned horizontal rectangles and their orientations. We adopt the same standard evaluation metric as the ICDAR challenges for MSRA-TD500.

Table 1. Results on MSRA-TD500

Method	Precision	Recall	F-measure
Kang *et al.* [16]	71	62	66
Yao *et al.* [35]	63	63	60
Yin *et al.* [37]	81	63	74
Yin *et al.* [38]	71	61	65
Zhang *et al.* [39]	83	67	74
Ma *et al.* [22]	82	68	74
Yao *et al.* [36]	77	**75**	76
Shi *et al.* [26]	86	70	77
Zhou *et al.* [40]	87	67	76
Baseline_VGG16	58.8	55.3	57.0
Baseline_ResNet50	82.1	68.7	74.8
ITN_VGG16	80.3	65.6	72.2
ITN_ResNet50	**90.3**	72.3	**80.3**

4.3. Experimental Results

4.3.1 Detecting Oriented Multi-Lingual Text Lines

We evaluate our in-network transformation embedding module and the ITN in detecting oriented multi-lingual text lines on MSRA-TD500. The MSRA-TD500 only contains 300 training images which is too small to train our model. Following a standard solution widely adopted in the community [39, 36, 26], we mix the training set of MSRA-TD500 with other additional data. We use the training and test set of HUST-TR400 [34], which collects 400 images similar to MSRA-TD500 in scale and appearance. Detections with a score higher than 0.9 are taken as final results.

Our baseline network is built by replacing the in-network transformation embedding module in ITN with a standard convolutional layer of same configurations. Regular representations for text/non-text classification and coordinate regression are generated by 7×7 regular sampling grids. Other experimental conditions are the same with ITN in the baseline network.

The comparison results in terms of precision, recall and F-measure of the proposed method, baseline and other state-of-the-art methods are listed in Table 1. Our ITN based on ResNet-50 (ITN_ResNet50) outperforms all the other methods in precision and F-measure, and ranks second in recall. Moreover, promotions of 5.5% and 15.2% in F-measure are achieved by the ITNs with respect to their baseline networks (ResNet-50 and VGG-16) respectively. The experimental results demonstrate the effectiveness of the in-network transformation embedding module and the robustness of our geometry-aware representation. The high precision is mainly benefited by the geometry-aware representation which excludes redundant background noises. Figure 4 shows some detection results and how the geometry-aware representation adapts to each instance. We observe that the

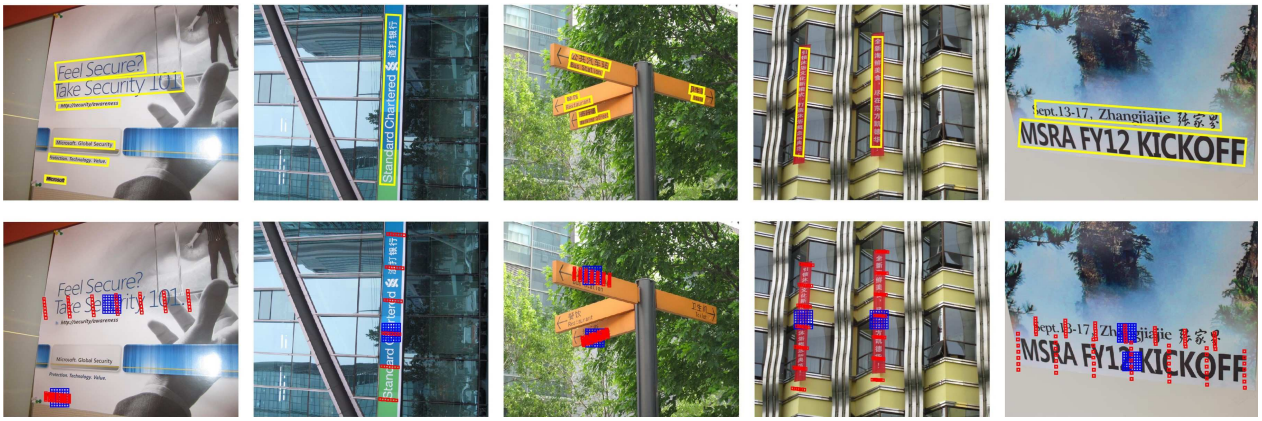

Figure 4. Examples on MSRA-TD500. The first row shows detection results (yellow bounding-boxes) of the ITN. The second row shows the regular sampling grids (blue) and the adaptive sampling grids (red) on the feature maps (visualized as input images). For clarity, we draw two pairs of sampling grid examples on one image at most.

Figure 5. Examples on ICDAR2015. Detection results (yellow bounding-boxes) and zoom-in small windows (blue boxes at the image corners) are shown in each image.

ITN is able to detect oriented text in long lines. The reason behind this phenomenon is that the embedment of transformation allows the model to see the global representation of a text line so that boundaries of long instances can be accurately determined.

4.3.2 Detecting Oriented English Words

To further demonstrate the effectiveness of our in-network transformation embedding module and the ITN in detecting oriented English words, we evaluate our baseline network and the ITN on ICDAR2015 Incidental Text. Targets in this dataset are commonly small and thus the positive pixels in the feature map is much less than MSRA-TD500. To ensure effective recall, we keep all the detections classified as "text" as the final detections. For fair comparison, we evaluate our performance with the official submission server.

As listed in Table 2, our method (ITN_VGG16) outperforms other methods, and the ITNs achieves improvements

Table 2. Results on ICDAR 2015 Incidental Text

Method	Precision	Recall	F-measure
MCLAB_FCN [39]	70.8	43.0	53.6
CTPN [30]	51.6	74.2	60.9
Yao *et al.* [36]	72.3	58.7	64.8
DMPN [20]	68.22	73.23	70.64
Shi *et al.* [26]	73.1	**76.8**	75.0
Zhou *et al.* [40]	83.6	73.5	78.2
Baseline_ResNet50	76.2	70.9	73.4
Baseline_VGG16	78.1	73.7	75.8
ITN_ResNet50	81.3	71.6	76.1
ITN_VGG16	**85.7**	74.1	**79.5**

of 2.7% and 3.7% in F-measure against the baselines based on ResNet-50 and VGG-16 respectively. As shown in Figure 5, our ITN is able to effectively detect English words in challenging situations like complex environment, skewed viewpoint and low resolution. Notably, our ITN generate tight quadrilateral detections which is much more accurate

(a) ITN-general (b) ITN w/o transformation regression (c) ITN

Figure 6. Example of the learnt transformations. Transformed sampling grids (red) and corresponding regular sampling grids (blue) on feature maps (visualized as input images) of (a) ITN-general, (b) ITN w/o transformation regression and (c) ITN.

than rigid rectangles especially in skewed viewpoints. This is because the combination of affine transformation and coordinate regression can closely approximate the real case.

4.4. Discussion

Comparison with other frameworks Compared with proposal based two-stage frameworks like [24], our one-stage ITN framework is impressively concise and does not need extensive proposals with discrete configurations of scale, rotation angle and aspect ratio. In a sense, our transformation embedding module incorporates RoI feature pooling into a convolutional layer, while the feature extraction and coordinate regression in our ITN are mutually calibrated in a dynamic way during training instead of fixed RoIs as we discussed in subsection 3.3. The comparison of experimental results between Faster-RCNN based method [22] and our ITN on MSRA-TD500 proves the robustness of our framework (74% vs. 80.3% in F-measure). Compared with other one-stage frameworks like RPN [24] or SSD [19], our framework produces geometry-aware representation rather than regular representation. The competitive methods [26, 20] are based on SSD [19] framework to detect words or segments. In comparison, our ITN achieves favorable results without pre-defined anchors or complicated post-processing.

Ablation study We empirically compare different options on the transformation in our ITN framework (ResNet-50 based): (a) *ITN-general*: using a general transformation [2], which makes the transformation supervision infeasible; (b) *ITN w/o transformation regression*: training an ITN without supervision on transformation regression.

The F-measure scores of baseline, ITN-general, ITN w/o transformation regression, and ITN on MSRA-TD500 are 74.8%, 78.2%, 79.0% and 80.3% respectively.

Both ITN w/o transformation regression and ITN-general achieve better results than baseline, which further demonstrates the effectiveness of our geometry-aware representation learning framework. The performance of ITN w/o transformation regression is better than ITN-general mainly due to the geometry constraint on transformation, which makes the transformation learning easier. The ITN trained with supervision on transformation regression outperforms both ITN-general and ITN w/o transformation regression. With the help of such supervision, the model is more likely to find a better solution during training, while the transformation learning may fall into different local optimal solutions without supervision, as shown in Figure 6.

5. Conclusion

In this paper, we have presented a novel end-to-end ITN to effectively detect scene text in the forms of complicated geometric layout. An adaptive geometry-aware representation learning scheme incorporated in the ITN has been proposed to encode the unique geometric configurations of scene text instances. The experimental results on standard benchmarks demonstrate that ITN is able to effectively detect multi-scale, multi-oriented and multi-lingual words or text lines at one pass. [1]

[1] **Acknowledgements.** This work is supported in part by the National Natural Science Foundation of China under Grant U1509206 and Grant 61472353, in part by the National Basic Research Program of China under Grant 2015CB352302. Xi Li is supported by the Alibaba-Zhejiang University Joint Institute of Frontier Technologies. Dacheng Tao is supported by the grants: ARC FL-170100117, DP-180103424, DP-140102164, and LP-150100671.

References

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Improving object detection with one line of code. In *ICCV*, 2017.

[2] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017.

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

[4] R. B. Girshick. Fast R-CNN. In *ICCV*, 2015.

[5] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In *ICCV*, 2017.

[8] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural network for scene text detection. *IEEE Trans. Image Processing*, 25(6):2529–2541, 2016.

[9] W. He, X. Zhang, F. Yin, and C. Liu. Deep direct regression for multi-oriented scene text detection. In *ICCV*, 2017.

[10] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *ICCV*, 2013.

[11] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced MSER trees. In *ECCV*, 2014.

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[14] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *ECCV*, 2014.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, 2014.

[16] L. Kang, Y. Li, and D. S. Doermann. Orientation robust text line detection in natural images. In *CVPR*, 2014.

[17] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. ICDAR 2015 competition on robust reading. In *ICDAR*, 2015.

[18] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, 2017.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.

[20] Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *CVPR*, 2017.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[22] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *CoRR*, abs/1703.01086, 2017.

[23] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, 22(10):761–767, 2004.

[24] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[26] B. Shi, X. Bai, and S. J. Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, 2017.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[28] S. Tian, S. Lu, B. Su, and C. L. Tan. Scene text recognition using co-occurrence of histogram of oriented gradients. In *ICDAR*, 2013.

[29] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan. Text flow: A unified text detection system in natural scene images. In *ICCV*, 2015.

[30] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 2016.

[31] K. Wang, B. Babenko, and S. J. Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

[32] D. Xiang, Q. Guo, and Y. Xia. Robust text detection with vertically-regressed proposal network. In *ECCV*, 2016.

[33] I. Z. Yalniz, D. Gray, and R. Manmatha. Efficient exploration of text regions in natural scene images using adaptive image sampling. In *ECCV*, 2016.

[34] C. Yao, X. Bai, and W. Liu. A unified framework for multi-oriented text detection and recognition. *IEEE Trans. Image Processing*, 23(11):4737–4749, 2014.

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

[36] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. *CoRR*, abs/1606.09002, 2016.

[37] X. Yin, W. Pei, J. Zhang, and H. Hao. Multi-orientation scene text detection with adaptive clustering. *PAMI*, 37(9):1930–1937, 2015.

[38] X. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. *PAMI*, 36(5):970–983, 2014.

[39] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In *CVPR*, 2016.

[40] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: An efficient and accurate scene text detector. In *CVPR*, 2017.