HAM: Hidden Anchor Mechanism for Scene Text Detection

Jie-Bo Hou, Xiaobin Zhu, Chang Liu, Kekai Sheng, Long-Huang Wu, Hongfa Wang, and Xu-Cheng Yin, *Senior Member, IEEE*

***Abstract*— Direct regression and anchor are the two mainly effective and prevailing mechanisms in the paradigm of scene text detection. However, the use of direct regression-based methods may be challenging during optimization without the help of anchors as references. Unfortunately, the anchor-based methods always suffer from the careful design of the anchors, degrading the robustness to complex scenes. To address the above-mentioned problems, we propose a novel hidden anchor mechanism (HAM) especially for scene text detection. The predictions of anchors are innovatively regarded as hidden layers, and the weighted sum of the predictions is integrated into a direct regression-based network. Hence, the architecture of our HAM still has the characteristic of simplicity as with direct regression-based methods. Moreover, it is easier to optimize anchors as references with this type of method than with direct regression-based methods. In this way, our network can take advantage of both direct regression and anchor mechanisms. In addition, we decouple three kinds of one-dimensional anchors from three-dimensional anchors, greatly reducing the number of anchors in text bounding box matching without performance degradation. We also propose a post-processing technique for long text detection, named iterative regression box (IRB), which takes a few additional computational costs and can be easily generalized to other methods. Experiments on several public datasets demonstrate that the proposed method achieves state-of-the-art performance. Code is available at https://github.com/hjbplayer/HAM.**

Manuscript received July 31, 2019; revised March 14, 2020, May 22, 2020, and June 30, 2020; accepted July 9, 2020. Date of current version July 22, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2019YFB1405900, in part by the Beijing Top Discipline for Artificial Intel-Ligence Science and Technology (University of Science and Technology Beijing), in part by the Fundamental Research Funds for the Central Universities under Grant FRF-AT-19-008 and Grant FRF-TP-18-060A1, in part by the Beijing Natural Science Foundation under Grant 4194084, and in part by the Fundamental Research Funds for the Central Universities and USTB-NTUT Joint Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo. *(Jie-Bo Hou and Xiaobin Zhu contributed equally to this work.) (Corresponding author: Xu-Cheng Yin.)*

Jie-Bo Hou, Xiaobin Zhu, and Chang Liu are with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: houjiebo@gmail.com; zhuxiaobin@ustb.edu.cn; lasercat@gmx.us).

Kekai Sheng is with the Youtu Laboratory, Tencent, Shanghai 200233, China (e-mail: saulsheng@tencent.com).

Long-Huang Wu and Hongfa Wang are with Tencent Technology (Shenzhen) Company Limited, Shenzhen 518057, China (e-mail: jerrylhwu@tencent.com; hongfawang@tencent.com).

Xu-Cheng Yin is with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, also with the Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China, and also with the USTB-EEasyTech Joint Laboratory of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China (e-mail: xuchengyin@ustb.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.3008863

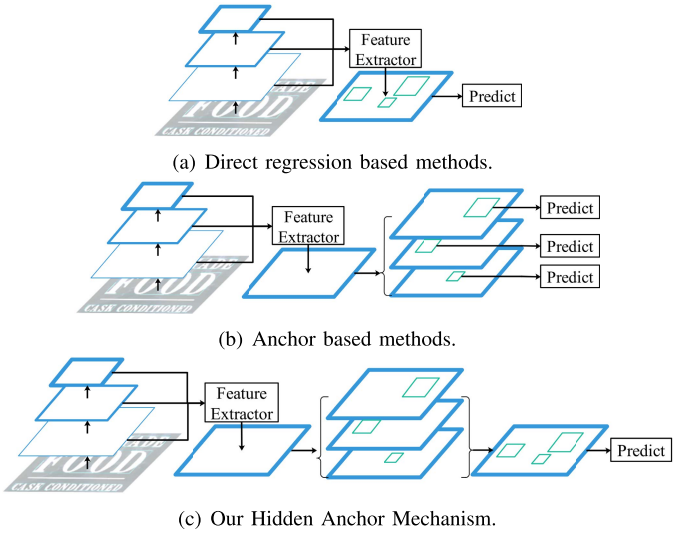
***Index Terms*— Scene text detection, multi-oriented text, multi-task, fully convolutional network.**

I. Introduction

SCENE text contains abundant valuable information and is ubiquitous, such as on traffic signs, billboards, and guideposts. Reading scene text is necessary for automatic driving, translation and other useful applications [1]–[5]. However, the complexity of backgrounds and variations in font, size, color, and orientation make scene text detection a challenging task.

With the development of deep learning, a series of scene text detection approaches have been proposed [6]–[12]. Direct regression [13]–[15] and anchor [16]–[18] are two popular types of mechanisms for scene text detection. For each positive sample/point, direct regression-based methods regress the bounding box directly, while anchor-based methods regress it with a suitable anchor as a reference. Direct regression-based methods have a simple pipeline and predict all the bounding boxes in one group of classifications and regressions. Anchor-based methods predict texts with different shapes in different groups of classifications and regressions. However, direct regression-based methods [13], [14] often fail in detecting long texts [18]. Although, the anchor-based methods can simplify the regression-based architecture and make it easier to be optimized [19]. However, the anchor-based methods often rely on carefully designing specific anchors for adapting to the variations of text bounding boxes. Generally, handcrafted anchors have difficulty in covering the variations in bounding boxes in different scenes. Notably, regression-based methods may fail in detecting long texts. LOMO [20] presents an iterative refinement module (IRM) that perceives the entire long text by iterative refinement based on the extracted feature blocks of the preliminary proposals. However, IRM incurs massive computational cost and cannot be directly extended to other methods.

In this paper, we propose a novel hidden anchor mechanism (HAM), which integrates the anchor mechanism into a direct regression-based network, as shown in Fig. 1. In this way, our HAM can well keep the simplicity of direct regression-based methods while assimilating the advantage of anchor-based methods. Different from traditional anchor mechanisms, we use a softmax loss for all the anchors instead of the binary cross-entropy loss for each anchor to predict the anchors' weights in our HAM. The predictions of the anchors



(a) Direct regression based methods.

(b) Anchor based methods.

(c) Our Hidden Anchor Mechanism.

Fig. 1. (a) Direct regression based methods predict all boxes of different shapes in one feature map directly. (b) Anchor based methods predict boxes of different shapes in different groups of feature maps. (c) Our Hidden Anchor Mechanism (HAM) integrates anchors into a direct regression based method, and combines the advantages of both (a) and (b). Anchors in our HAM are taken as hidden layers, and the outputs of HAM are similar to (a).

mainly contain two parts: classification and regression. The classification scores can be regarded as the weights for the anchors. We innovatively take the total predictions of the anchors as hidden layers. The sum of the anchor classification results (except the background) will be used for the final classification. The weighted sum of the anchor regression results will be used for final regression, as shown in Fig. 2. In addition, we propose a decoupled anchor mechanism (DAM) that explores three kinds of one-dimensional anchors, i.e., width anchors, height anchors, and angle anchors, instead of traditional three-dimensional anchors (ratio, scale, and angle anchors). For detecting long texts, we propose a post-processing method called iterative regression box (IRB), which can iteratively merge predicted bounding boxes for convergence. IRB can even predict long texts in which the predicted bounding boxes have not completely covered the long texts. Notably, our IRB takes much less time than IRM [20] and can be easily generalized to other methods.

In summary, our main contributions are four-fold:

- We propose an innovative framework for scene text detection. The proposed method achieves state-of-the-art performance on various benchmark datasets.
- We propose a HAM that integrates the advantages of anchors into direct regression-based methods. Moreover, our HAM can maintain the simplicity of direct regression-based methods.
- We propose a DAM that can adaptively accommodate to the variances of text bounding boxes and reduce the number of anchors in text bounding box matching.
- We propose a postprocessing method called IRB for long text detection. IRB takes a few additional computational costs and can be easily generalized to other methods.

The rest of the paper is organized as follows. Section II summarizes the related work. Section III elaborates our work. In Section IV, we demonstrate experimental results on several datasets. Finally, we conclude our work in Section V.

II. Related Work

Scene text detection has been actively studied in recent decades. Recently, with the progress of deep learning, many approaches have been proposed to address scene text detection. Here, we briefly introduce related studies of two categorizations for scene text detection: direct regression-based methods and anchor-based methods. In addition, we also introduce some other related methods in Section II-C.

A. Direct Regression-Based Methods

EAST [14] and DDR [13], [15] are two representative direct regression-based methods for scene text detection. Following the general design procedure of DenseBox [21], EAST has two branches, namely segmentation and direct regression. The segmentation branch is based on a fully convolutional network (FCN) [22], while EAST predicts text/no-text for each pixel. For each pixel, the regression branch predicts a rotated bounding box directly without anchor references or links. The main idea of DDR [13] is similar to that of EAST, but the design of the ground truth and loss functions are slightly different. PixelLink [23] is a promising work that deals only with segmentation. It predicts extra eight linking maps for post-processing steps to link the pixels to bounding boxes or boundaries. PixelLink can be regarded as a particular kind of direct regression-based method, in which the regression results are the linking maps, and their values are either one (link) or zero (not link). Direct regression-based methods have fewer post-processing steps than segmentation-based methods.

Direct regression-based methods are simple, but their ability to detect long texts is insufficient. Without anchors as references, the network could be difficult to train for regression [19]. For example, YOLO [24] is similar to the direct regression-based method, while YOLO9000 [19] and YOLO v3 [25] add anchors to improve the regression results.

B. Anchor-Based Methods

Anchor-based methods for scene text detection focus on designing proper anchor mechanisms. These methods are highly inspired by Faster R-CNN [26], [27], which designs different shapes of anchors to regress corresponding objects. However, the limited number of anchors cannot well cover all the sizes of texts and may drop complex scene texts. Moreover, region of interest (RoI) pooling is also not suitable for rotated scene texts. To overcome the limitation of anchors, CTPN [28] and SegLink [29] design anchors to cover a part of a word rather than the whole word and then group/link them to predict the final text boxes. CTPN designs vertical anchors to match parts of horizontal texts, while SegLink develops anchors with the angle for rotated texts. Inspired by SSD [30], TextBoxes [31] adopts anchors with more kinds of aspect ratios for horizontal texts for detecting arbitrarily

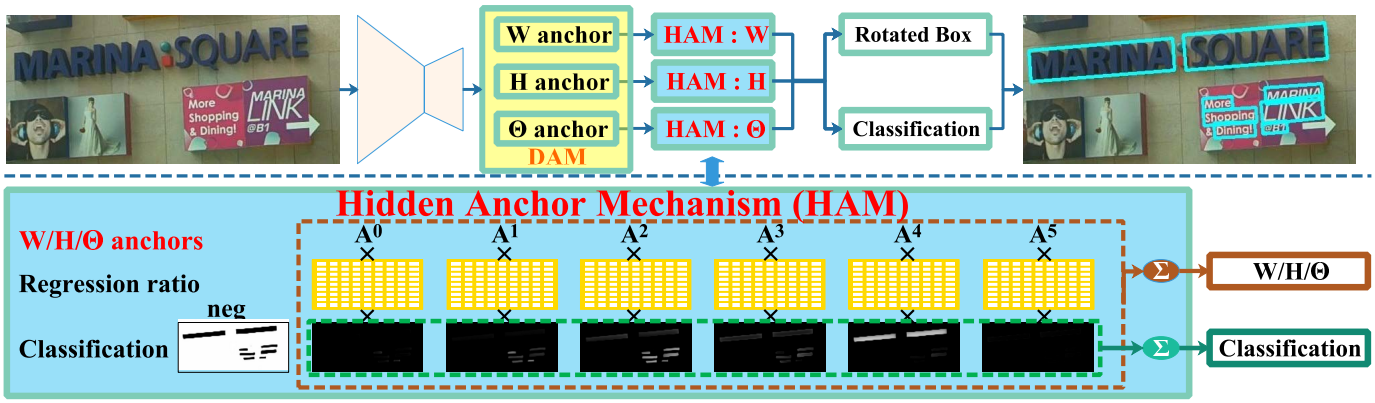


Fig. 2. Overview of our Hidden Anchor Mechanism (HAM). The A_0, A_1, A_2, A_3, A_4, and A_5 are the values from one-dimension anchors (W, H, and Θ represent width, height, and angle, respectively). The Decoupled Anchor Mechanism (DAM) predicts weights and ratios of width, height, and angle anchors. **DAM are shown in Fig. 3**. And the HAM uses the weighted sum of anchors' predictions to predict the final classification, the regression of width, height, and angle, respectively (width, height, and angle all have one own HAM). Finally, the predicted width, height, and angle are combined to rotated boxes.

oriented text. It sets the aspect ratios of the default boxes to $1, 2, 3, 5, 1/2, 1/3$, and $1/5$ and regresses the four points of arbitrarily oriented texts to the anchor boxes. RRPN [16] and DMPNet [32] modify rotated anchors to detect oriented texts. RRPN sets six different orientations ($-\frac{\pi}{6}$, 0,$\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$ and $\frac{2\pi}{3}$), three aspect ratios (1:2, 1:5 and 1:8) and three scales (8, 16 and 32). With the rotated anchors, RRPN can better match arbitrarily oriented texts than other methods. RRPN also adopts rotation RoI (RRoI) pooling for arbitrarily oriented text instead of RoI pooling. IncepText [18] utilizes an Inception-Text module combined with deformable position-sensitive RoI (PSRoI) pooling [33] instead of RoI pooling.

Anchor-based methods have to carefully design many appropriate anchors to match the rotated bounding boxes for scene texts. To solve this problem for complex object detection, YOLO9000 [19] adopts K-means clustering to cluster the anchors. RRPN designs a large number of anchors (54 anchors) for complex arbitrarily oriented texts. However, it is still challenging to detect texts in complex backgrounds and foregrounds in scene images.

C. Other Methods

Inspired by Mask R-CNN [34], some methods predict bounding boxes and instance segmentation and then utilize the segmentation results to refine the bounding boxes. Mask TextSpotter [35] uses end-to-end networks for text detection and recognition with character instance segmentation. FTSN [36] is inspired by instance-aware semantic segmentation [22], [34] and leverages the merits from accurate region proposal-based methods. Flexible segmentation-based methods can easily generate a mask for arbitrarily shaped text. There are also many other methods for text detection [37], [38]: some methods predict the corners of the texts [39], some predict curve text [40], [41], and some utilize the end-to-end network for simultaneous detection and recognition [35], [42].

III. The Proposed Method

The framework of our method is shown in Fig. 2. Inspired by EAST [14] and FOTS [42], we implement a direct regression-based scene text detector as a basic detector. SharpMask [44] and FPN [45] concatenate $1/4$, $1/8$, $1/16$ and $1/32$ feature maps. The size of the final feature map is $1/4$ of the input image. The DAM and HAM are integrated into our method. Specifically, the DAM uses feature maps to predict the classification and regression of width, height, and angle anchors. Our HAM uses the sum of the anchors' classifications except the background to predict the final text/nontext classifications and the weighted sum of the anchors' regression results to predict the final regression for all the texts. Finally, detection results are produced by thresholding and locality-aware non-maximum suppression (NMS) on the predicted bounding boxes. For a sanity check, we use both ResNet-50 [46] and DenseNet-169 [47] as backbones in our experiments.

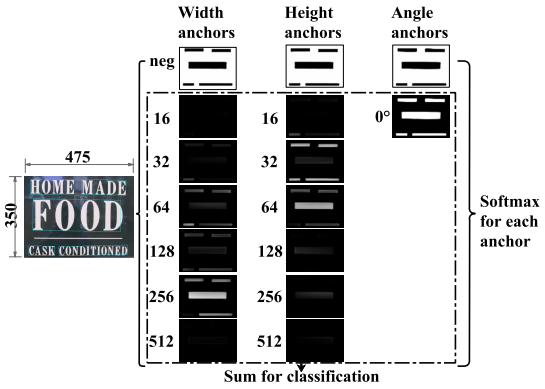


Fig. 3. An example of Decoupled Anchor Mechanism (DAM) (the regression ratios of value to anchor are not shown here). Width, height, and angle are predicted by their own anchor, respectively. The Classification results can be regarded as weights of anchors for the Hidden Anchor Mechanism (HAM), and **neg** means the negative samples (background).

A. Decoupled Anchor Mechanism

Traditional anchor-based methods always adopt a multiplicity of anchors with two dimensions (i.e., scale and ratio) as regression references. However, the angle is a necessary and important factor in arbitrarily oriented scene text detection.

(a) Rotated anchors (b) DeRPN (c) DAM

Fig. 4. Differences of Rotated anchors, DeRPN [43], and our Decoupled Anchor Mechanism (DAM). The anchors in red means the matched anchors.

Thus, RRPN [16] adopts anchors with three dimensions, i.e., scale, ratio, and angle (shown in Fig. 4 (a)). To treat variant object shapes, DeRPN [43] decouple two-dimensional anchors into two one-dimensional anchors (width and height; ratio and scale can be translated to width and height), as shown in Fig. 4 (b). However, DeRPN is not suitable for arbitrarily oriented text, so we propose a DAM that uses three kinds of one-dimensional anchors, i.e., width anchors, height anchors, and angle anchors, for arbitrarily oriented scene text detection, as shown in Fig. 4 (c), similar to AAM [48].

To be concrete, we use 6 scales for width anchors as a set: $W = \{16, 32, 64, 128, 256, 512\}$. In our method, the range of the i-th width anchor is $(W_i * 0.7,\ W_i/0.7)$. The parameter sets of the height anchor are identical to those of the width anchor. Notably, our implementation only uses one angle anchor: $\Theta = \{0°\}$. For comparison, if we utilize three angle anchors, our method employs only 15 one-dimensional anchors (e.g., 6 widths, 6 heights, and 3 angles) to represent 108 ($6 \times 6 \times 3$) three-dimensional anchors; e.g., RRPN uses 54 three-dimensional anchors (3 scales, 3 ratios, and 6 angles). In conclusion, our DAM can use fewer outputs to represent more anchors than other methods. In addition, the recombination method of decoupled anchor results in our work is totally different from the dimension recombination for dimension-decomposition mechanism in DeRPN [43]. Our DAM utilizes the proposed HAM to merge the predictions of anchors into the height, width, and angle for each pixel. A DeRPN selects the top-N width segments (W_N). For each width segment in W_N, DeRPN chooses the top-k height segments at the corresponding pixels, which will be formed into pairs (denoted by B_w). Similarly, for height segments, DeRPN tries to generate B_h. Adding angle anchors would make the DeRPN more complex to implement, not necessarily that the network would have more difficulty in finding the right boxes.

For classification, the traditional anchor mechanism, e.g., Faster R-CNN, assigns a binary class label to each anchor and assigns a positive label if the anchor/anchors with the highest intersection-over-union (IoU) overlap with a ground-truth box or have an IoU overlap greater than 0.7 with any ground-truth box. In our DAM, we take anchors as a multi-class classifier, assign multi-class labels to the anchors, and use the softmax loss for classification. Moreover, we regard the classification results as the weights of the anchors. An example of the predicted result is shown in Fig. 3. The corresponding label generation algorithm for width anchors is listed in Algorithm 1. In assigning the labels to the anchors, a bounding box will be abandoned for training if its width is in the range of $[W_i/0.7, W_{i+1}/0.7]$. We also set "DO NOT CARE" labels to the samples whose width/height is in the boundary of two width/height anchors to make the anchor "softer". More details are illustrated in Algorithm 1. The label generation procedure of the height anchors is identical to that of the width anchors.

Algorithm 1 Label Generation Procedure

Input: input Point P and width w
Parameters: range $R = 0.7$, boundary range $D = 0.1$, width anchors:$W = \{16, 32, 64, 128, 256, 512\}$, the number of width anchors $|W|$
Output: the label for the width anchor ("DO NOT CARE" label i_d, background label i_{bg})

if P is not a positive sample for classification **then**
  return i_{bg}
end if
if $w \geq W_{|W|-1} \cdot (1/R)$ or $w \leq W_0 \cdot R$ **then**
  return i_d
end if
for $i = 0; i < |W|; i$++ **do**
  if $w \geq W_i \cdot (R + D)$ and $w \leq W_i \cdot (1/R - D)$ **then**
    return i
  end if
  if $(w > W_i \cdot R$ and $w < W_i \cdot (R + D))$ or $(w > W_i \cdot (1/R - D)$ and $w < W_i \cdot (1/R))$ **then**
    return i_d
  end if
end for

Note that we only implement one angle anchor, since our work focuses on improving the regression of width and height in a direct regression-based method. In addition, the values of the angles are different from those of the width and height. The width and height values are in the range of $[0, \infty)$, so the weighted sum of the predicted widths or heights is still in the range of $[0, \infty)$. However, the range of the angle values is defined as $(-45°, 45°)$, and the angles values are periodic; i.e., $-45°$ is equal to $45°$. Therefore, the weighted sum of the anchors may exceed the range $(-45°, 45°)$. To limit the prediction values for angles, we utilize a sigmoid function for angles. If we employ three-angle anchors as $\Theta = \{-30°, 0°, 30°\}$, the values between anchors, e.g., $-15°$ and $15°$, will be unreachable for every single anchor alone the value range of Δ_{Θ_i} in Equation 3.

B. Hidden Anchor Mechanism

In traditional anchor mechanisms, the anchors' predictions will be used to predict region proposals [16], [26] or the final output layers [17], [30]. Hence, there exist K (the number of anchors) groups of predictions. However, our HAM uses the weighted sum of the anchors' predictions to predict the final output layers. It serves as a hidden layer and integrates anchor references into the direct regression-based method. Therefore, the output of the network is similar to that of EAST, and only one group with a total of six channels is computed: one channel for classification and five channels for regression.

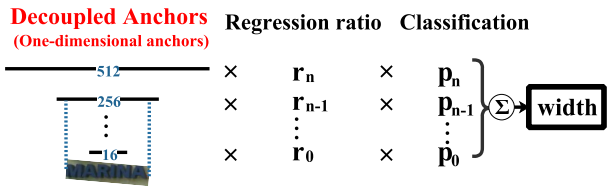


Fig. 5. An example of width Hidden Anchor Mechanism (HAM). Regression ratios are the ratios of width to decoupled anchors, and classifications are predicted weights of each anchor. HAM utilizes the weighted sum of these decoupled anchors to predict the width of the text (Equation 1).

To predict the weight of each anchor for our HAM, we explore a soft anchor mechanism. We use the softmax function and "DO NOT CARE" labels, which do not contribute to the loss for the classification, making the anchors "softer" than those from traditional anchor mechanisms. The softmax function can predict each anchor to be used as a weight, and "DO NOT CARE" labels can successfully exclude the bounding boxes whose scales are near the border of two neighboring anchors. We do not care which anchor is classified strictly, and the results of anchor classification are just used for reference. The sum of the classification results of the anchors except for the background are for the final classification, and the final regression is the weighted sum of the regression results of the anchors.

As described above, the anchor in our HAM can be taken as a multi-class classifier, and we assign multi-class labels to anchors and use the softmax loss for classification. The predicted classifications of the anchors are regarded as the weights of each anchor, rather than the probability of each anchor in the traditional method (with a binary class). The final classification results of our HAM are the sum of the anchor-predicted weights except for the background. The final regression results of our HAM are the weighted sum of each anchor's regression results. We do not use any extra loss function for the regression of width, height, or angle. The only loss function used for regression is similar to that of EAST [14]. Our HAM merges the predicted width and height as shown in Fig. 5, and this method is described in the following.

The predicted width, height and angle values of each pixel in the output layers can be formulated as

$$width = \sum_{i=0}^{|W|-1} p_{W_i} \cdot (r_{W_i} \cdot W_i), \tag{1}$$

$$height = \sum_{i=0}^{|H|-1} p_{H_i} \cdot (r_{H_i} \cdot H_i), \tag{2}$$

$$angle = \sum_{i=0}^{|\Theta|-1} p_{\Theta_i} \cdot (\Delta_{\Theta_i} + \Theta_i), \tag{3}$$

where W_i, H_i and Θ_i denote the i-th values of the width, height and angle anchor references, respectively; $|W|$, $|H|$ and $|\Theta|$ denote the numbers of width, height and angle anchors, respectively; p_{W_i}, p_{H_i} and p_{Θ_i} denote the predicted weights of the width, height and angle anchors, respectively; r_{W_i} and r_{H_i} are for the predicted ratios of the ground truth to the i-th anchor references; Δ_{Θ_i} is the offset value of the corresponding angle anchor; and Δ_{Θ_i} is in the range of $(-\frac{45^\circ}{|\Theta|}, \frac{45^\circ}{|\Theta|})$.

For each pixel, the predicted classification and the box part (in Fig. 2) can be formulated as

$$cls = \frac{\sum_{i=0}^{|W|-1} p_{W_i} + \sum_{i=0}^{|H|-1} p_{H_i}}{2}, \tag{4}$$

$$Box \begin{cases} top = AH_p \cdot height \\ bottom = height - top \\ left = AW_p \cdot width \\ right = width - left \end{cases}. \tag{5}$$

Each predicted box contains four values: the top, right, bottom, and left boundaries of the rectangle [14]. The four values are calculated by the width, height, AH_p and AW_p. Here, AH_p and AW_p represent the ratio of the top boundary to the height and the ratio of left boundary to the width, respectively.

C. Loss Functions

To distinguish very close text instances, a shrunk quadrangle is used to assign ground-truth text labels, and only the pixels in the shrunk polygon [14] of the original text are considered positive samples. For regression, we predict the angle, the axis-aligned bounding box (AABB), as in EAST [14]. AABB represents the 4 distances of each pixel to the top, right, bottom, and left boundaries of the rectangle. According to FOTS [42], we use online hard example mining (OHEM) [49] to solve the sample imbalance problem. We also use Instance-Balance [23], which regards a text box as an instance to balance the text boxes. For the i-th instance with $area = S_i$, the average area of all the positive pixels is B, and the area of all the negative pixels is S_{neg}. The instance weight is computed as $w_i = \frac{B}{S_i}$. The weights of the negative samples are set to 1 for L_{cls} and to $\frac{B_i}{S_{neg}}$ for L_a. We use w_x to represent the Instance-Balance weight of one pixel x. The loss for text/no-text classification and anchor classification is computed as

$$L_{cls} = \frac{1}{|\Omega_{cls}|} \sum_{x \in \Omega_{cls}} H(p_x, p_x^*) \cdot w_x, \tag{6}$$

$$L_a = \frac{1}{|\Omega_{cls}|} \sum_{x \in \Omega_{cls}} Softmax\_loss(S_x, S_x^*) \cdot w_x, \tag{7}$$

where Ω_{cls} denotes the set of selected points by OHEM in the classification score map, $|\Omega|$ is the number of selected points, $H(p_x, p_x^*)$ denotes the cross-entropy loss, p_x is the prediction and p_x^* is the label of point x. For the anchor loss, S_x is with the predictions at point x, and S_x^* is the label. If S_x^* is with "DO NOT CARE" label, the loss will be set to zero.

The regression loss contains two parts: L_{AABB} [50] for the AABB and L_θ [14] for the angle. We also use the Instance-Balance weight for the regression loss. The regression loss is only used in the positive area, and the loss in the negative area is set to 0. These components are calculated as

$$L_{AABB} = \frac{1}{|\Omega_{AABB}|} \sum_{x \in \Omega_{AABB}} IoU(R_x, R_x^*) \cdot w_x, \tag{8}$$

$$L_\theta = \frac{1}{|\Omega_\theta|} \sum_{x \in \Omega_\theta} (1 - cos(\theta_x - \theta_x^*)) \cdot w_x, \tag{9}$$

Fig. 6. An example of our post-processing method Iterative Regression Box (IRB) for long text. In the first row, Left: result of NMS. Middle: all the predicted dense boxes without NMS. Right: result of IRB. The iterative steps are in the second row, where cyan boxes are predicted by the points with maximum/minimum coordinates in the box of last step, and white boxes are the minimum bounding rectangle of cyan boxes. From left to right, we can witness the growth of predicted box for very long text with the increasing of iteration.

where Ω_{AABB} and Ω_θ are the selected samples of OHEM for AABB and the corresponding angle, respectively; $|\Omega_{AABB}|$ and $|\Omega_\theta|$ denote the numbers of Ω_{AABB} and Ω_θ, respectively; R_x represents the predicted AABB geometry; R_x^* is its corresponding ground truth; θ_x is the predicted angle; and θ_x^* is the ground truth. Finally, the overall loss is defined as

$$L = L_{cls} + L_{AABB} + \lambda_\theta \cdot L_\theta + \lambda_a \cdot L_a, \tag{10}$$

where λ_θ is empirically set to 10 and λ_a is set to 0.1 for the initial 100k iterations and will be set to 0 afterwards.

D. Iterative Regression Box

The outputs of our network are classification and regression results for every point in the images. We only take the points with the classification score above the threshold as the text point and then regress a bounding box for each point. Dense boxes exist because of the dense classification predictions. Consequently, we utilize locality-aware NMS to postprocessing the dense detected boxes. We find that our HAM may fail in detecting very long text, as shown in Fig. 6. However, when we remove the NMS step, we can figure out that the predicted dense boxes can cover all the text (the yellow boxes in the middle of the first row), but as a single box, most of them cannot cover the long text well.

To overcome this problem, we propose a post-processing method, named IRB. Different from the IRM in LOMO [20], which needs to be trained together with network, our IRB is just a post-processing moudule and can be easily extended to other methods. In addition, IRM takes about 35 ms/iteration, while our IRB only takes about 1 ms/iteration. In IRB, we first utilize locality-aware NMS. Then, in each box, we select points that have the maximum or minimum coordinates; moreover, the predicted score is above the threshold (to avoid the missed merging of two close neighboring text lines, the box is shrunk to the selected points). The predicted boxes are merged using the minimum bounding rectangle if their IoU is larger than IoU_m, the merged box is taken as a new input box for IRB, and the box is regressed iteratively until convergence or the maximum iteration step is reached. In Fig. 6, the IoUs of the white boxes in the second row, the fourth, and fifth columns are less than the stop IoU threshold (IoU_s), so the iteration stops. Details are described in Algorithm 2.

Algorithm 2 Iterative Regression Box

Input: input box: b_{in}
Parameters: maximum number of iterations: $iter_{max}$
merge IoU threshold: IoU_m
stop IoU threshold: IoU_s
Function: IoU of boxes b_1 and b_2: $IOU\{b_1, b_2\}$
Output: output box: b_{out}

$b_{pre} = b_{in}$, $iter_{cnt}$=0, IoU=0
while $iter_{cnt} < iter_{max}$ and $IoU < IoU_s$ **do**
  $iter_{cnt}$ += 1
  b_{shrunk} is a shrunk box of b_{pre}
  P_s = points in b_{shrunk} with max or min coordinates
  B_s = predicted boxes in the position of Ps
  $B \leftarrow \varnothing$, $B \leftarrow B \cup \{b_{pre}\}$
  for b in B_s **do**
    if $IOU(b, b_{pre}) > IoU_m$ **then**
      $B \leftarrow B \cup \{b\}$
    end if
  end for
  b_{out} = minareaRect(B)
  $IoU = IOU(b_{out}, b_{pre})$
  $b_{pre} = b_{out}$
end while
return b_{out}

IV. EXPERIMENTS

A. Datasets

We conduct experiments on four benchmark datasets: the 2013 International Conference on Document Analysis and Recognition (ICDAR2013) dataset [51], the ICDAR2015 dataset [52], the ICDAR2017 Multi-Lingual Text (MLT) datast [53], and the Microsoft Research Asia Text Detection 500 (MSRA-TD500) dataset [54].

1) ICDAR2013: It contains 229 training images (849 words) and 233 testing images (1, 095 words). The texts in this dataset are horizontal and annotated as rectangles in words.

2) ICDAR2015: It is the dataset of Challenge 4 of ICDAR2015 Robust Reading Competition for incidental scene text detection. It consists of 1, 000 training images (11, 886 words) and 500 testing samples (5, 230 words). The text regions are annotated by four vertices of the quadrangle.

TABLE I

ABLATION STUDY FOR THE HAM ON ICDAR 2015 WITH DIFFERENT BACKBONES AND TRAINING DATASETS. IC 13, IC 15, AND IC 17 REPRESENT ICDAR 2013, ICDAR 2015, AND ICDAR 2017 MLT, RESPECTIVELY

Methods	Recall	Precision	F-measure	Gap	Backbone	Training dataset	FPS	m p-value	t p-value
Basemodel	82.8	88.94	85.79		ResNet-50	IC 13+IC 15	11.1	0.449	0.125
HAM	84.01	88.93	86.40	0.61 ↑	ResNet-50	IC 13+IC 15	10.5		
Basemodel	85.36	88.34	86.82		ResNet-50	IC 13+IC 15+IC17	11.1	1.90E-4	3.08E-12
HAM	85.84	89.82	87.79	0.97 ↑	ResNet-50	IC 13+IC 15+IC17	10.5		
Basemodel	85.17	88.23	86.67		DenseNet-169	IC 13+IC 15	8.4	0.0261	5.48E-5
HAM	86.37	89.03	87.68	1.01 ↑	DenseNet-169	IC 13+IC 15	8.0		
Basemodel	86.42	89.03	87.71		DenseNet-169	IC 13+IC 15+IC17	8.4	1.52E-5	2.89E-17
HAM	87.77	90.69	89.21	1.50 ↑	DenseNet-169	IC 13+IC 15+IC17	8.0		

3) ICDAR2017 MLT: It is from the ICDAR 2017 Competition that focuses on the multi-oriented and multi-lingual scenarios. It contains 9 languages (i.e., Chinese, Japanese, Korean, English, French, Arabic, Italian, German, and Indian), representing 6 different scripts. It consists of 7, 200 training images (84, 868 cropped words), 2,000 validation images, and 9, 000 testing images (97, 619 cropped words).

4) MSRA-TD500: It is collected for detecting arbitrarily oriented long text lines from indoor and outdoor settings. It consists of 300 training images (1, 068 text lines) and 200 testing images (651 text lines) with text-line-level annotations.

Fig. 7. Images in the first row are detection results from Basemodel, and images in the second row are detection results from our HAM.

B. Implementation Details

ResNet-50 [46] and DenseNet-169 [47], pre-trained on ImageNet, are used as backbones. Adam [55] is used to optimize our network, and the initial learning rate is set to 0.0001. The learning rate will decay with by a factor of 0.94 after every 10k iterations.

Data augmentation is important for the robustness of deep neural networks. In our method, we first resize the training images with four ratios (0.5, 1, 1.5, 2), followed by random rotations in the range of $[-10°$ to $10°]$. For the first 100k iterations, one-third of the images are added an extra 90° and the rest are not added. Finally, 640×640 random samples are cropped to form a mini-batch. The mini-batch size is six in the ResNet-50 backbone and three in the DenseNet-169 backbone. As described in Section III-C, we use OHEM to balance the samples. During classification, there are 512 hard negative samples, 512 random negative samples, and all the positive samples are selected for each image. In regression, AABB and the angle select 128 hard positive samples and 128 random positive samples for training, respectively. The weight of the loss in Equation 10 λ_θ is empirically set to 10, and λ_a is set to 0.1 for the first 100k iterations and 0 afterwards.

In the testing phase, results are produced by a threshold of 0.9. The dense boxes exist because of the dense classification prediction . Consequently, we utilize locality-aware NMS as our post-processing step. We utilize IRB and set $iter_{max} = 10$, $IoU_m = 0.2$ and $IoU_s = 0.99$ if the longest side of a predicted box is longer than 256 pixels in ICDAR2013, ICDAR2015 and ICDAR2017 MLT and utilize IRB for all the predicted boxes on MSRA-TD500 since it is a line-level dataset. It takes approximately 1 ms/iteration with an E5-2650 V4 @ 2.2 GHz CPU. Notably, DAM and HAM are indivisible and interdependent in our work, so the **HAM** in our experiments/tables means utilizing both the DAM and HAM.

C. Ablation Study

1) Ablation Study for the HAM With Different Backbones and Training Datasets: To verify the effectiveness of our HAM, we conducted an ablation study on the ICDAR2015 dataset. Table I summarizes the quantitative experimental results, and the detected results are shown in Fig. 7.

Basemodel is a re-implemented direct regression scene text detector. Basemodel is inspired by EAST [14] and FOTS [42] and utilizes OHEM, Instance-Balance and data augmentation but without the proposed DAM and HAM.

HAM adds the proposed HAM and DAM to **Basemodel**. The DAM and HAM are highly integrated, so we could not perform ablation studies on them.

From Table I, we can conclude that our method outperforms the Basemodel with both ResNet-50 and DenseNet-169 as the backbone. The proposed HAM can greatly boost the scene text detection performance, and the gap between Basemodel and HAM increases with the addition of training data and the increasing strength of the backbone. HAM outperforms Basemodel to the greatest extent in terms of the F-measure (i.e., by 1.5 %). As shown in Fig. 7, we can determine that

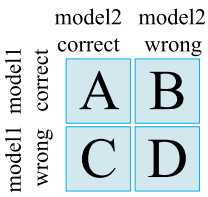


Fig. 8. A four-fold table of McNemar's Test [56].

TABLE II

ABLATION STUDY FOR OUR HAM ON ICDAR 2015. BACKBONE: DENSENET-169, TRAINING DATASETS: ICDAR 13+ICDAR 15

Methods	Recall	Precision	F-measure	Gap
Basemodel	85.17	88.23	86.67	
One-anchor-512	85.21	88.14	86.65	0.02 ↓
One-anchor-64	85.36	88.47	86.89	0.22 ↑
Average	85.89	88.31	87.08	0.41 ↑
HAM	86.37	89.03	87.68	1.01 ↑

HAM can regress the box better than Basemodel. In the second column, we can determine that the bad detected box may suppress a neighboring box.

2) Statistical Significance Results: We analyze the statistical significance of our HAM against the Basemodel. To the best of our knowledge, there are no scene text detection methods in deep learning paradigm have reported statistical significance results. To verify the statistical significance of our HAM against the Basemodel, we adopt McNemar's Test (M-test) [56] and T-test as criteria. We adopt M-test as it was used in scene text script identification task [57], and T-test used in object detection task [58], [59].

McNemar's Test (M-Test): In the scene text detection task, we can't directly judge correct or wrong of one text box. So we evaluate M-test by the F-measure in one image. Here, we define our Basemodel as model1, and our HAM as model2 as shown in Fig 8. We define F1: F-measure of model 1 and F2: F-measure of model 2. A, B, C, and D represent (1) A: F1=F2; (2) B: F1> F2; (3) C: F1< F2; (4) D: F1=F2=0. The corresponding part (A, B, C or D) adds the product of the ground-truth boxes number and the difference value of F-measure in one image (F2-F1).

T-Test: it can't be directly applied in the scene text detection task. So we calculate the p-value of T-test ("t p-value") by the F-measure of each image, and we also regard an image as N (the number of ground-truth boxes in the image) samples.

As shown in Table I, in most cases, the improvements of HAM are statistically significant at the 0.05 significance level in both the M-test and T-test. The p-values of our HAM with ResNet-50, trained on IC13+IC15+IC17 are 1.90E-4 and 3.08E-12 with M-test and t-test. Conclusively, our HAM can statistically improve the performance.

3) Ablation Study for Different Forms of the HAM: We conducted another ablation study for different forms of the HAM on the ICDAR2015 dataset. The experimental results are shown in Table II. The backbone here is DenseNet-169, and the training datasets are ICDAR2013 and ICDAR2015.

TABLE III

ABLATION STUDY OF TRAINING BATCH SIZE ON ICDAR 2015, TRAINING DATASETS:ICDAR 13+ICDAR 15+ICDAR 17 MLT

Methods	Recall	Precision	F-score
HAM-ResNet-50 (Batch 3)	84.25	88.96	86.54
HAM-ResNet-50 (Batch 6)	85.84	89.82	87.79
HAM-ResNet-50 (Batch 12)	86.03	89.80	87.87

(a) ResNet-50 (b) DeneseNet-169 (c) ResNet-50+IRB

Fig. 9. Results of our method with/without IRB. (a) HAM with ResNet-50 as backbone. (b) HAM with DenseNet-169 as backbone. (c) HAM (with ResNet-50) utilizes IRB as post-processing in Section III-D.

One-anchor 512 represents only one anchor, e.g., $W = H = \{512\}$ for both the width anchor and height anchor in the detector. Similarly, **One-anchor-64** means $W = H = \{64\}$. Our **HAM** is characterized by $W = H = \{16, 32, 64, 128, 256, 512\}$ as described in Section III-A. **Average** means that we utilize the average weights for the hidden anchors instead of the predicted weights in the HAM, but the anchors W and H are the same as those of the HAM. The results show that the proposed HAM is effective; even if we utilize just the average of the hidden anchors' predictions, we can obtain a 0.41% improvement in the F-measure. The performance of **One-anchor-512** is almost the same as that of **Basemodel** since 512 is the base value for regression in **Basemodel**. **One-anchor-64** outperforms **One-anchor-512** by 0.24 percentage points in terms of the F-measure since most texts in the ICDAR2015 dataset are small in size, and the average width and height are closer to 64 than 512. This finding also shows that appropriate anchors can improve performance. When we utilize the whole HAM, we can achieve a 1.01% improvement in the F-measure.

4) Ablation Study for IRB: As shown in Fig. 9, we can easily determine that the capability of the network to predict very long text is limited when we utilize ResNet-50 as the backbone. When we utilize DenseNet-169 as the backbone, our method can detect long text better than when we utilize ResNet-50 as the backbone since DenseNet-169 is deeper than ResNet-50. In Fig. 9(d), we utilize the IRB described in Section III-D. We can determine that with IRB, our HAM can also regress very long text well, but it would make the bounding box slightly larger than it should be. We compare our HAM with/without IRB in all the datasets, as shown in Tables IV, V, VI, and VII. The MSRA-TD500 dataset is annotated at the line level, and on this dataset (Table VI), our HAM+IRB method can achieve an F-measure of 83.48%, outperforming HAM by 3.27%.

Fig. 10. Detection examples of our model on ICDAR2015 Incidental Scene Text benchmark.

Fig. 11. Detection examples of our model on ICDAR 2017 MLT.

5) Ablation Study for the Batch Size: The batch size in the training procedure can affect the performance, as is commonly known. As shown in Table III, we can see that the performance of our HAM increases as the training batch size increases. Because of the limitation of GPU memory (GTX 1080Ti 11 GB), we set the batch size to 6 for ResNet-50; for comparison, LOMO [20] is trained with a batch size of 32. It takes 2 GPUs to train a HAM-ResNet-50 model (batch size of 12), which takes longer to train than a HAM-ResNet-50 model (batch size of 6). Therefore, all the experiments in this paper are trained with a batch size of 6 when ResNet-50 is utilized as the backbone and a batch size of 3 when ResNet-169 is utilized as the backbone except when noted.

D. Compararing With State-of-the-Art Methods

We compare our work with state-of-the-art methods on several public benchmark datasets: ICDAR2015, ICDAR2017 MLT, MSRA-TD500, and ICDAR 2013. We utilize the proposed IRB if the longest side of a predicted box is longer than 256 pixels in ICDAR2013, ICDAR2015 and ICDAR2017 MLT and utilize IRB for all the predicted boxes on MSRA-TD500 since it is a line-level dataset.

1) ICDAR2015: The samples in the training set include 229 images from the ICDAR2013 dataset, $1,000$ training images from the ICDAR2015 dataset and $7,200$ training images from the ICDAR2017 MLT dataset. The training steps are described in Section IV-B. When testing, we resize the images to a resolution of $1,920 \times 1,024$. As shown in Table IV, we can conclude that our method outperforms the state-of-the-art method [20] by 0.59% in terms of the F-measure when we use ResNet-50 [46] and one-scale testing. FOTS [42] with the recognition branch can achieve an F-measure of 87.99%, and FOTS with the detection branch can achieve an F-measure of only 85.31%. Our method with ResNet-50 achieve an F-measure of 87.87% and outperforms FOTS (with the detection branch only) by 2.56% in terms of the F-measure. FOTS with the recognition branch can improve the F-measure by 2.68% compared with FOTS with only the detection branch, which shows that recognition can improve the detection performance. In addition, our method with DenseNet-169 can achieve an F-measure of 89.21%. The results are shown in Fig. 10.

2) ICDAR2017 MLT: The results are shown in Fig. 11. We use $7,200$ images from the MLT training set for training. The first step is to train the model with similar steps to those

Fig. 12. Detection examples of our model on MSRA-TD500 and ICDAR 2013. Images is the first row are from MSRA-TD500, and the second row are from ICDAR 2013. Bounding boxes in MSRA-TD500 are line-level, meanwhile boxes ib ICDAR 2013 are word-level.

TABLE IV

RESULTS ON ICDAR2015, WHERE * MEANS MULTI-SCALE TEST AND ^{recg} MEANS PREDICTION IS COMBINED WITH RECOGNITION BRANCH

Methods	Recall	Precision	F-measure
SegLink [29]	76.80	73.10	75.00
MCN [60]	80.00	72.00	76.00
ITN [61]	74.10	85.70	79.50
RRPN* [16]	77.00	84.00	80.00
EAST* [14]	78.30	83.30	80.70
He et al. [15]	80.00	85.00	82.00
RRD [62]	79.00	85.60	82.20
TextSnake [40]	84.90	80.40	82.60
Textboxes++*[17]	78.50	87.80	82.90
DDR* [13]	80.00	88.00	83.80
PixelLink [23]	82.00	85.50	83.70
FTSN [36]	80.00	88.60	84.10
Lyu et al. [39]	79.76	89.50	84.30
IncepText [18]	80.60	90.50	85.30
FOTS [42]	82.04	88.84	85.31
Tian et al. [63]	85.00	88.30	86.60
IncepText* [18]	84.30	89.40	86.80
LOMO [20]	83.50	91.30	87.20
Mask TextSpotter^{recg}[35]	81.00	**91.60**	86.00
FOTS^{recg} [42]	85.17	91.00	**87.99**
HAM-ResNet-50 (Batch 6)	85.84	89.82	87.79
HAM-ResNet-50+IRB (Batch 6)	85.80	89.90	87.80
HAM-ResNet-50 (Batch 12)	**86.03**	89.80	87.87
HAM-DenseNet-169 (Batch 3)	**87.77**	90.69	**89.21**

TABLE V

RESULTS ON ICDAR17 MLT. WHERE ^{recg} MEANS THE PREDICTION IS COMBINED WITH THE RECOGNITION BRANCH

Methods	Recall	Precision	F-measure
He et al. [15]	57.94	76.69	66.01
Border(ResNet-50) [38]	60.60	73.90	66.60
Lyu et al. [39]	55.60	**83.80**	66.80
FOTS [42]	57.45	79.48	66.69
FOTS^{recg} [42]	57.51	80.95	67.25
LOMO [20]	60.60	78.80	68.50
HAM-ResNet-50	59.83	77.26	67.44
HAM-ResNet-50+IRB	60.28	80.34	68.88
Border(DenseNet-121) [38]	62.10	77.70	69.00
HAM-DenseNet-121	62.14	81.03	70.34
HAM-DenseNet-169	**62.38**	82.57	**71.07**

used on the ICDAR2015 dataset. One-quarter of the images are added by an extra 90°, another quarter are added by an extra $-90°$, and the rest are not added. The second step is to use random multiscale, cropped images, i.e., 640×640, $1,024 \times 1,024$ and $1,280 \times 1,280$, for training. The mini-batch size is set to (3, 1, 1) for the DenseNet-169 backbone and (6, 2, 1)) for the ResNet-50 backbone for the three scales of cropped images. In testing procedure, we resize the images to 2 times of the original resolution, but the longest side is no larger than 2, 400 pixels. With ResNet-50, our method can also achieve an F-measure of 68.88% and outperforms LOMO [20] by 0.3% in terms of the F-measure. For a fair comparison with Border (DenseNet-121) [38], we also trained our HAM with DenseNet-121 as the backbone, and the result shows that our HAM (70.34%) can outperform Border (69.0%) by 1.34% in terms of the F-measure.

3) MSRA-TD500: The results are shown in the first row of Fig. 12. The model trained on the ICDAR2017 MLT dataset is used as the pretrained model for MSRA-TD500 due to the limited number of training images from the MSRA-TD500 dataset (300). Then, we fine-tune it on the training datasets of MSRA-TD500 and HUST-TR400 [64]. We resize the images to 0.5 times the original resolution, but the longest side is no larger than 960 pixels when testing. In the MSRA-TD500 dataset, there is much large text and very long text with a large area. Without the help of the IRB, our method achieves relatively unsatisfactory performance (HAM-ResNet-50 (80.21%) vs IncepText (83.0%)) [18].

TABLE VI
RESULTS ON MSRA-TD500

Methods	Recall	Precision	F-measure
He et al. [15]	70.00	85.00	76.00
EAST [14]	67.43	87.28	76.08
Border(ResNet-50) [38]	73.30	80.70	76.80
SegLink [29]	70.00	86.00	77.00
PixelLink [23]	73.20	83.00	77.80
TextSnake [40]	73.90	83.20	78.30
Border(DenseNet-121) [38]	77.40	83.00	80.10
ITN [61]	72.30	**90.30**	80.30
Lyu et al. [39]	76.20	87.60	81.50
FTSN [36]	77.10	87.60	82.00
Tian et al. [63]	81.70	84.20	82.90
MCN [60]	79.00	88.00	83.00
IncepText [18]	79.00	87.50	83.00
HAM-ResNet-50	78.69	81.78	80.21
HAM-ResNet-50+IRB	**79.89**	87.40	**83.48**
HAM-DenseNet-169	82.99	87.65	85.26
HAM-DenseNet-169+IRB	**83.33**	89.32	**86.22**

TABLE VII
RESULTS ON ICDAR2013. WHERE ^{recg} MEANS THE PREDICTION IS COMBINED WITH THE RECOGNITION BRANCH

Methods	Recall	Precision	F-measure	FPS
RRPN [16]	72.00	90.00	80.00	-
Textboxes [31]	74.00	88.00	81.00	11.0
Textboxes++ [17]	74.00	88.00	81.00	-
RRD [62]	75.00	88.00	81.00	-
PixelLink [23]	83.60	86.40	84.50	-
SegLink [28]	83.00	87.70	85.30	20.6
Lyu et al. [39] [17]	79.40	93.30	85.80	10.4
DDR [13]	81.00	92.00	86.00	1.1
FOTS [42]	-	-	86.96	23.9
Border(ResNet-50) [38]	86.90	87.8	87.4	-
CTPN [28]	83.00	93.00	88.00	7.1
MCN [60]	87.00	88.00	88.00	34.0
FOTS^{recg} [42]	-	-	88.23	23.9
Border(DenseNet-121) [38]	87.10	91.50	89.20	-
He et al. [15]	**89.00**	**95.00**	**91.00**	-
Mask TextSpotter ^{recg} [35]	88.10	94.10	**91.00**	4.6
HAM-ResNet-50	81.28	89.62	85.25	27.1
HAM-ResNet-50+IRB	82.55	92.30	87.17	25.0
HAM-DenseNet-169	83.47	94.42	88.60	18.5
HAM-DenseNet-169+IRB	83.56	94.52	88.70	17.7

However, our method achieves an F-measure of 83.48% with IRB. This finding demonstrates the effectiveness of our proposed IRB. In Section V, we discuss some other performance of our HAM in the long text in detail.

4) ICDAR2013: The results are shown in the second row of Fig. 12. We utilize the trained model of the ICDAR2017 MLT dataset as a pre-trained model for the ICDAR2013 dataset. We resize the longest side of the images to 960 pixels when testing, and we also add IRB for ICDAR2013. On the ICDAR2013 dataset, IRB takes approximately 3.53 iterations/image and 0.86 ms/iteration. It takes approximately 3 ms/image. The ICDAR2013 dataset contains a large amount of text that stretches across the whole image. However, our HAM may miss detecting very large text, since the Instance-Balance mechanism [23] we utilized reduces the weight of large text during training. However, if we do not utilize Instance-Balance, the HAM may not detect small text. The Instance-Balance mechanism remains to be optimized in future work. In addition, the large text is scarce, and our data augmentation method does not deal with especially large text.

5) Speed: Our method achieves 10.5 FPS (8.0 FPS) at a resolution of $1,920 \times 1,024$ pixels with ResNet-50 (DenseNet-169) as the backbone. The running environments are a GTX 1080Ti GPU and an E5-2650 V4 @ 2.2 GHz CPU. The HAM appends only several convolution channels and a softmax layer, so the difference in runtime between the HAM and Basemodel can be negligible, as shown in Table I.

V. CONCLUSION

This paper presents a novel framework with a hidden anchor mechanism (HAM) and a decoupled anchor mechanism (DAM), especially for scene text detection. Our HAM can keep the simplicity property of direct regression-based methods while being reinforced by integrating anchor mechanisms for regression. Compared with traditional anchor mechanisms, the DAM can accommodate to the variances in text bounding boxes with fewer anchors. In addition, we propose a post-processing IRB for processing very long text. Our method achieves state-of-the-art performance on several public benchmarks. In future work, we will combine our detection framework with the recognition branch in an end-to-end manner to further improve performance.

REFERENCES

[1] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.
[2] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.
[3] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.
[4] C. Yang *et al.*, "Tracking based multi-orientation scene text detection: A unified framework with dynamic programming," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3235–3248, Jul. 2017.
[5] S. Tian, W. Pei, Z. Zuo, and X. Yin, "Scene text detection in video by learning locally and globally," in *Proc. IJCAI*, 2016, pp. 2647–2653.
[6] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.
[7] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from Web videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 542–554, Mar. 2018.
[8] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas, "Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation," in *Proc. CVPR*, 2018, pp. 2226–2234.
[9] X. Peng, X. Yu, K. Sohn, D. N. Metaxas, and M. Chandraker, "Reconstruction-based disentanglement for pose-invariant face recognition," in *Proc. ICCV*, 2017, pp. 1623–1632.

[10] X. Zhu, Z. Li, X. Li, S. Li, and F. Dai, "Attention-aware perceptual enhancement nets for low-resolution image classification," *Inf. Sci.*, vol. 515, pp. 233–247, Apr. 2020.

[11] C. Li, F. Wei, W. Dong, X. Wang, Q. Liu, and X. Zhang, "Dynamic structure embedded online multiple-output regression for streaming data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 323–336, Feb. 2019.

[12] C. Li, X. Wang, W. Dong, J. Yan, Q. Liu, and H. Zha, "Joint active learning with feature selection via CUR matrix decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1382–1396, Jun. 2019.

[13] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. ICCV*, 2017, pp. 745–753.

[14] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. CVPR*, 2017, pp. 2642–2651.

[15] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Multi-oriented and multi-lingual scene text detection with direct regression," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5406–5419, Nov. 2018.

[16] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[17] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[18] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "IncepText: A new inception-text module with deformable psroi pooling for multi-oriented scene text detection," in *Proc. IJCAI*, 2018, pp. 1071–1077.

[19] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, 2017, pp. 6517–6525.

[20] C. Zhang *et al.*, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. CVPR*, 2019, pp. 10552–10561.

[21] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "DenseBox: Unifying landmark localization with end to end object detection," *CoRR*, vol. abs/1509.04874, 2015. [Online]. Available: http://arxiv.org/abs/1509.04874

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[23] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting scene text via instance segmentation," in *Proc. AAAI*, 2018, pp. 6773–6780.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, 2016, pp. 779–788.

[25] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement, *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[27] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, 2015, pp. 1440–1448.

[28] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. ECCV*, 2016, pp. 56–72.

[29] B. Shi, X. Bai, and S. J. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. CVPR*, 2017, pp. 3482–3490.

[30] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[31] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. AAAI*, 2017, pp. 4161–4167.

[32] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. CVPR*, 2017, pp. 3454–3461.

[33] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. ICCV*, 2017, pp. 764–773.

[34] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proc. ICCV*, 2017, pp. 2980–2988.

[35] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. ECCV*, 2018, pp. 71–88.

[36] Y. Dai *et al.*, "Fused text segmentation networks for multi-oriented scene text detection," in *Proc. ICPR*, 2018, pp. 3604–3609.

[37] M. R. L. Gomez, A. Mafla, and D. Karatzas, "Single shot scene text retrieval," in *Proc. ECCV*, 2018, pp. 728–744.

[38] C. Xue, S. Lu, and F. Zhan, "Accurate scene text detection through border semantics awareness and bootstrapping," in *Proc. ECCV*, 2018, pp. 370–387.

[39] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. CVPR*, 2018, pp. 7553–7563.

[40] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. ECCV*, 2018, pp. 19–35.

[41] Y. Liu, L. Jin, S. Zhang, and S. Zhang, "Detecting curve text in the wild: New dataset and new solution," *CoRR*, vol. abs/1712.02170, 2017. [Online]. Available: http://arxiv.org/abs/1712.02170

[42] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. CVPR*, 2018, pp. 5676–5685.

[43] L. Xie, Y. Liu, L. Jin, and Z. Xie, "Derpn: Taking a further step toward more general object detection," in *Proc. AAAI*, 2019, pp. 9046–9053.

[44] P. O. Pinheiro, T. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. ECCV*, 2016, pp. 75–91.

[45] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, 2017, pp. 936–944.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[47] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 2261–2269.

[48] J.-B. Hou *et al.*, "Detecting text in scene and traffic guide panels with attention anchor mechanism," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 10, 2020, doi: 10.1109/TITS.2020.2996027. [Online]. Available: https://ieeexplore.ieee.org/document/9113429/

[49] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. CVPR*, 2016, pp. 761–769.

[50] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. S. Huang, "Unitbox: An advanced object detection network," in *Proc. ACM MM*, 2016, pp. 516–520.

[51] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. ICDAR*, vol. 1, 2013, pp. 1484–1493.

[52] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. ICDAR*, 2015, pp. 1156–1160.

[53] N. Nayef *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification—RRC-MLT," in *Proc. CDAR*, 2017, pp. 1454–1459.

[54] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. CVPR*, 2012, pp. 1083–1090.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds. May 2015.

[56] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947.

[57] L. Gomez, A. Nicolaou, and D. Karatzas, "Improving patch-based scene text script identification with ensembles of conjoined networks," *Pattern Recognit.*, vol. 67, pp. 85–96, Jul. 2017.

[58] Y. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, "Learning to detect human-object interactions," in *Proc. WACV*, 2018, pp. 381–389.

[59] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Alternating regression forests for object detection and pose estimation," in *Proc. ICCV*, 2013, pp. 417–424.

[60] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and W. L. Goh, "Learning Markov clustering networks for scene text detection," in *Proc. CVPR*, 2018, pp. 6936–6944.

[61] F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao, "Geometry-aware scene text detection with instance transformation network," in *Proc. CVPR*, 2018, pp. 1381–1389.

[62] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. CVPR*, 2018, pp. 5909–5918.

[63] Z. Tian *et al.*, "Learning shape-aware embedding for scene text detection," in *Proc. CVPR*, 2019, pp. 4234–4243.

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

Jie-Bo Hou received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include text detection, pattern recognition, and deep learning.

Xiaobin Zhu received the M.E. degree from Beijing Normal University in 2006 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2013. He is currently an Associate Professor with the School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include machine learning and image content analysis and classification.

Chang Liu received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include text detection, few-shot learning, and text recognition.

Kekai Sheng received the B.Eng. degree in telecommunication engineering from the University of Science and Technology Beijing in 2014, and the Ph.D. degree from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, in 2019. He is currently a Researcher Engineer with Youtu Laboratory, Tencent Inc. His research interests include image quality evaluation, domain adaptation, and AutoML.

Long-Huang Wu received the MA.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2018. He is currently working as a Researcher with Tencent Computer System Company, Limited. His main research interests include scene text detection, computer vision, and deep learning.

Hongfa Wang received the master's degree in operation science and control theory from the Chinese Academy of Sciences, Beijing, China, in 2008. He is currently an Expert Researcher with Tencent Computer System Company Limited. His research interests include computer vision, machine learning, and pattern recognition.

Xu-Cheng Yin (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from the University of Science and Technology Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2006. He was a Visiting Professor with the College of Information and Computer Sciences, University of Massachusetts at Amherst, Amherst, MA, USA, for three times from January 2013 to January 2014, from July 2014 to August 2014, and from July 2016 to September 2016. He is currently a Full Professor and the Director of the Pattern Recognition and Information Retrieval Laboratory, Department of Computer Science and Technology, University of Science and Technology Beijing.