IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received April 25, 2020, accepted May 23, 2020, date of publication June 16, 2020, date of current version June 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002808

Cluttered TextSpotter: An End-to-End Trainable Light-Weight Scene Text Spotter for Cluttered Environment

RANDHEER BAGI, (Graduate Student Member, IEEE), TANIMA DUTTA, (Member, IEEE), AND HARI PRABHAT GUPTA, (Member, IEEE)
Department of Computer Science and Engineering, IIT Varanasi (Banaras Hindu University), Varanasi 221005, India

Corresponding author: Randheer Bagi (randheerbagi.rs.cse17@iitbhu.ac.in)

This work was supported in part by the Science and Engineering Research Board (SERB) under Grant ECR/2017/002419 and Grant ECR/2016/000406, and in part by the Early Career Research Award Scheme.

ABSTRACT Scene text spotting aims at simultaneously localizing and recognizing text instances, symbols, and logos in natural scene images. Scene text detection and recognition approaches have received immense attention in computer vision research community. The presence of partial occlusion or truncation artifact due to the cluttered background of scene images creates an obstacle in perceiving the text instances, which makes the process of spotting very complex. In this paper, we propose a light-weight scene text spotter that can address the issue of cluttered environment of scene images. It is an end-to-end trainable deep neural network that uses local part information, global structural features, and context cue information of oriented region proposals for spotting text instances. It helps to localize in scene images with background clutters, where partially occluded text parts, truncation artifacts, and perspective distortions are present. We mitigate the problem of misclassification caused by inter-class interference by exploring inter-class separability and intra-class compactness. We also incorporate multi-language character segmentation and word-level recognition in a light-weight recognition module. We have used six publicly available benchmark datasets in different smart devices to illustrate the efficacy of the network.

INDEX TERMS Deep learning, noisy images, scene text detection, text recognition, text spotting.

I. INTRODUCTION

The recent advancement in technology has played an important role in the rapid development of multimedia and computer vision using resource-constrained devices, like smartphones, as a research field. There has been a notable improvement in terms of processing power, internal memory, and power consumption of smartphones. These factors automate, facilitate, and improve the processing capability of the devices. High denser digital cameras used in smartphones capture natural scene images of the world around us in real-time. Natural scenes contain both text and various scene objects. Texts embedded in scene images contain a large amount of useful information.

Texts in scene images contain many important clues, such as notices, doorplates, and captions, that helps in scene understanding [1], [2]. Recognition of these texts has attracted significant attention of the research community for understanding the images. Unlike the characters in printed documents, scene texts are more difficult to recognize due to the large variations in backgrounds, textures, fonts, and illumination conditions. Scene images have attracted importance largely due to its various practical applications, such as robot navigation, autonomous vehicles, human-computer interaction, and multilingual translation [1], [3]–[7]. Multi-language text spotting is an essential but challenging task. *Scene text spotting* is the process of simultaneously detect and recognize all text instance occurrences, logos, and symbols in a natural scene image. The detected text instance is confined within a bounding box [1], [4], [8]. This is a challenging task because spotting results can be significantly affected by a wide variation in size, orientation, aspect ratio, color, script, and font of text instances in the scene images. Scene text spotting in a cluttered environment is a furthermore complex

The associate editor coordinating the review of this manuscript and approving it for publication was Feng Shao.

task due to the presence of perspective distortion, noise, truncation, occlusion, and inter-class interference, as shown in FIGURE 1.

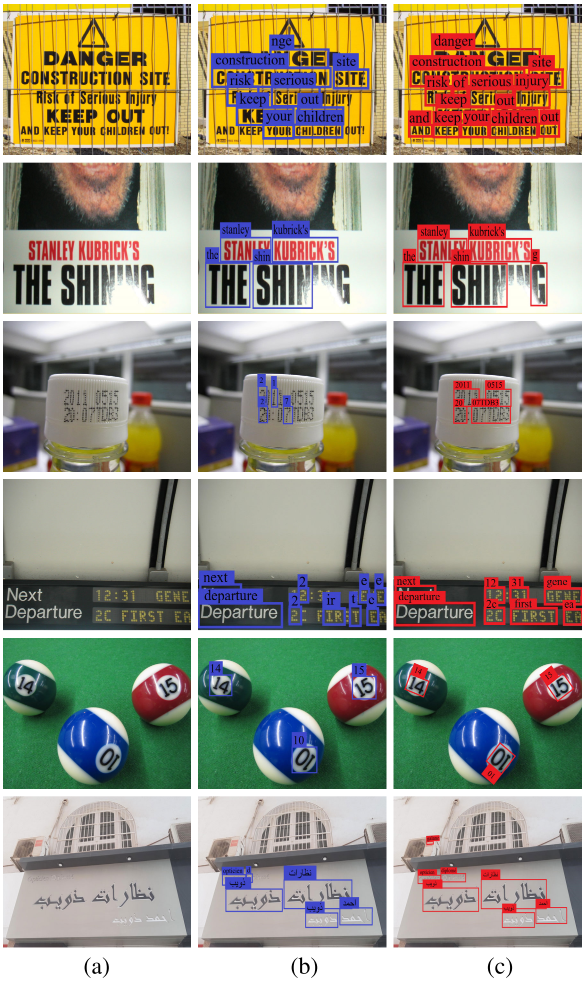


FIGURE 1. Exemplification of scene images illustrating the necessity of Cluttered TextSpotter. Columns (b) and (c) are the recognized text instances in the scene images of column (a) using baseline [1] and our network. Cluttered TextSpotter can spot multi-language oriented text instances in the scene images with cluttered environment.

Although significant progress has been reported in the literature of text spotting, the task remains challenging in the real-time applications due to the cluttered background environment, which results in occluded text parts, truncation artifacts, and low contrast. However, we humans when we have to identify a text region in a scene image, no matter how complicated the background clutters are, the localization of text instance is subserved by both a local process sensitive to local (regional) features and a global process of retrieving structural context.

Contextual information is critical for feature representation. In a classification task, focusing on informative regions in images is beneficial to generate discriminative features. The presence of cluttered background noise however severely restricts the feature matching process in real application scenarios. Therefore, it is advantageous to identify and focus on the informative regions and their structural context. In generic object detection using deep neural networks (in short deep networks), global structures and context information of an object are used to learn and discriminate salient features [9]. To the best of our knowledge, local and global structural context information of text regions, especially in the presence of occluded text parts and truncation artifacts, has still not been fully utilized for text spotting in scene images. Furthermore, in most literature [1], [4], [8], [10]–[14], the authors mainly consider scenes images with good illuminating conditions and also text instances have good resolution and contrast with respect to other objects in the scene images. Also, partial occlusion or truncation artifacts are not considered.

The aforementioned analysis motivates us to focus on global structural context information and local features of text instances. In this paper, we propose a light-weight deep network for efficient scene text spotting, especially when partial occlusion or truncation is present due to the cluttered environment. The contributions of the proposed Cluttered TextSpotter (in short CTS) are briefly summarized as follows:

- Firstly, we propose a robust text spotter that can efficiently be used in resource-constraint devices, like smartphones and tablets, to localize and recognize text instances and symbols in natural scene images, even in the presence of cluttered background.
- Secondly, we include multi-scale contextual information and encoder-decoder model in the backbone network to preserve the finer spatial information that prevents from degradation problem. We use bilinear sampling to map the features of a light-weight oriented region proposal network into a canonical dimension to normalize rotation and scale persisting aspect ratio and positioning of individual characters.
- Thirdly, we utilize a context encoding and refinement module for precisely regress the oriented bounding boxes of text instances and symbols. It helps to localize accurately in scene images even with cluttered background, where partially occluded text parts, truncation artifacts, perspective distortions, and inter-class interference are present. It is subserved by both a local process sensitive to regional features and a global process of retrieving structural context.
- Next, we address the misclassification problem caused by inter-class interference using Gaussian softmax. In the recognition module, we incorporate C-LSTM [15] in Bi-LSTM cells for word-level recognition and multi-language character segmentation.
- Finally, we conduct an exhaustive set of experiments on publicly available benchmark datasets to show the effectiveness of our network. We use metrics, like precision, recall, f-measure, and amount of training parameters for the evaluation. Our network performs better on publicly available benchmark datasets in terms of recall.

The rest of the paper is organized as follows. The next section investigate the state-of-the-art literature for scene text detection, recognition, and spotting using deep networks.

Section III reports the proposed Cluttered TextSpotter. Section IV exhibits the experimental results and finally we conclude the paper in Section V.

II. LITERATURE SURVEY

In this section, we briefly present a recent survey on scene text detection, recognition, and spotting using deep networks.

A. SCENE TEXT DETECTION

EAST [16] utilizes a fully convolutional network to performs word-level and text-line-level predictions. The distance of an oriented bounding box from a point to its every side is predicted. Deep Direct Regression [17] utilizes a fully convolutional network to directly project the coordinate offsets from a related bounding box. SegLink [18] decomposes a text instance into segments connected by links, which are detected densely by a fully convolutional neural network at multiple scales. An oriented box that covers a part of a word is a segment and such segments are combined by links. DMPNet [19] recalls text instances with a higher overlapping area by sliding quadrilateral windows in intermediate convolutional layers. It incorporates a shared Monte-Carlo process and a sequential protocol that helps in relative regression of text instances with quadrangles. RefineText [20] processes features at multiple levels to produce dense text regions with higher semantic value. TextEdge [21] uses the text-region edge map for classification, edge prediction, and boundary regression, which are relevant to text instances. The authors in [22] obtain inclined proposals that have the information of orientation angle of text instances. They use oriented region proposal network and oriented region-of-interest pooling layer to map arbitrary-oriented region proposals to a feature tensor for text classification. Tian *et al.* project pixels onto an embedding space, where they consider pixels of same text instances appear closer to each other [23]. The authors in [24] incorporate normalization of scale and orientation of text instances to map to a desired canonical geometry range. Liu *et al.* make use of a tightness-aware intersect-over-union metric that quantifies completeness of ground truth, tightness of matching degree, and compactness of word and text-line detection [25]. The authors in [26] describe visual and geometrical correlations to exploit texts having higher pairwise similarity using cycle consistency constraint and permutation matrix. In [27], a teacher-student learning based method and proposal-free multi-level feature mimicking approach are introduced to improve the accuracy by mimicking multi-level convolutional feature maps. The authors in [28] utilize instance segmentation network that combines prototype masks, per-instance mask coefficients, and self-distillation for precise text detection. To address the issue of complex background in scene images, GISCA [29] uses adaptive soft attention to capture the context of salient areas in U-Net architecture and make the gradient back-propagation process stable. In [30], a multilingual multi-oriented text detector is proposed exploiting instance segmentation and context information through channel and spatial attention. LATD [31] makes use of learnable anchors to refine scales and locations for regressing the offsets. The authors in [32] localize corner points of bounding boxes in test time and segmenting text regions in relative positions and generate candidate boxes using sampling and grouping corner points during the interference stage. Liao *et al.* utilize rotation-sensitive features for regression and rotation-invariant features for classification [33]. WordSup [34] is a weakly supervised approach exploiting word annotations to train a character detector and generate loose bounding boxes. CRAFT [2] estimates affinity between characters in the detection of arbitrarily-oriented, curved, or deformed text instances in scene images by providing character level annotations. The authors in [3] utilize two convolutional neural network in cascaded manner to perform word-level spotting of scene text without any post processing. In [5], authors focus on learning of strong features by using global and local information for detection of occluded and long text at word-level. The authors of [13] develop a semantic rich information feature map using feature pyramid text with novel loss function for end-to-end trainable system to detect small text instances. In [35], text components are identified by exploring most significant bit information of a bit plane slice. Also the authors fixed the window for character components of arbitrary oriented words which is based on angular relationship between sub-bands and a fused band for oriented scripts detection and recognition. The authors in [36], proposed a ring radius transform (RRT) technique to perform detection of oriented and multi-script scene text. Histogram Oriented Moments (HOM) was introduced in [37] for text detection in video. It is invariant to rotation, scaling, font, and font size variations. The authors in [38] proposed a multi-scale text detection technique that uses feature pyramid network for small text detection. In [39], a character graph grouping algorithm is utilized based on local context information to distinguish background noise from scene texts. Tang and Wu include a combination of superpixel-based stroke feature transform, hand-crafted features, and deep learning based region classification for scene text detection [40]. In [41], two convolution neural networks are used for detection and classification for coarse segmentation of scene texts.

TextField [42] detects irregular scene texts by learning a direction field that points each text pixel away from the nearest boundary of the text instance. A morphological operation is then used as post-processing for final detection. TextContourNet [43] extract instance-level text contour to increase the accuracy of curve text detection. In [44], to enhance the feature representation ability, a pyramid attention network is used text detection tasks. Mask-Most Net [45] use instance-level mask approximation method through a combination of high-level semantic and low-level features. It applies the auxiliary regression task on center and corner points followed by a contextual information to increase the accuracy of detection. In [46], arbitrary shape text is detected by extracting text proposals, which are refined using a recurrent neural network (RNN) and an adaptive number of boundary points. The authors in [47] use text frontier learning and a tightness prior that refine pixel-wise mask prediction and assign polygonal boundary to each text region for arbitrary shaped text detection. In [48], feature enhancement and a region proposal network are explored to utilize the prior knowledge about the shape of the text instance and representations of enhanced features to generate the bounding boxes. A pyramid region-of-interest pooling attention is further introduced that extracts the features of fixed-size text segmentation. A bounding box refinement network is also used to extract a curve text. OPMP [49] applies many arbitrary-shape fitting mechanisms to enrich the backbone layers followed by a re-classification of text instance using multi-grain classification. Yao *et al.* extract individual characters of a text region and their relationship as a part of a semantic segmentation problem using a single fully convolutional network for multi-oriented and curved text detection [50].

B. SCENE TEXT RECOGNITION

A comparative study of encoder-decoder approaches with attention module on large-scale text recognition tasks in natural scene images is performed in [51]. Shi *et al.* [52], [53] involve a spatial transformer network for rectifying and recognizing a text image using an attention-based sequence network. In [53], [54], the authors apply an attention mechanism that learns adaptively and selects the suitable features for recognizing text instances. Focusing Attention [55] learns character-wise annotations by supervising a learnable attention module. AON [56] applies an arbitrary orientation network to extract features of text instances in four different directions and the placement semantics of characters. Bai *et al.* [57] address expensive annotation problem by introducing an edit probability loss considering the occurrences of missing or unnecessary characters. Char-Net [58] utilizes an auxiliary dense detection task of characters to address the issue of having irregular text. ESIR [59] has involved line-fitting transformation recursively to eradicate text-line curvature and perspective distortion by estimating the pose of text-lines. In [60], text recognition is performed by extracting discriminative features and increasing the alignment between the target character region and attention region. The authors in [61] utilize text shape descriptors, such as center line, scale, and orientation to deal with highly curved or distorted text. NRTR [62] dispenses with recurrences and convolutions with a stacked self-attention module, where an encoder extracts features and a decoder perform the recognization of texts based on the output of the encoder. In [63], a realization of asynchronous training and inference behavior is performed to classify images irrespective of the presence of text instances, which leads to multimodal recognition tasks.

C. SCENE TEXT SPOTTING

Jaderberg *et al.* enables feature sharing for detecting text instances. They further utilize character case-sensitive and insensitive classification and bigram classification using multi-mode learning. Downsampling is avoided for a per-pixel sliding window [8]. The authors in [64], localize and recognize text instances using region proposal networks and deep networks, respectively. Deep TextSpotter [10] obtains text proposals from a region proposal network, which are normalized by bilinear sampling to obtain a variable-width feature map. Each region is then mapped with a sequence of characters. The authors in [65] apply region-of-interest (RoI) pooling to obtain feature maps only once and shared by both detection and recognition, where RNN encoder encodes feature maps of different lengths into the fixed-size. Curriculum learning is utilized to learn the character-level language and appearance models. He *et al.* uses a text-alignment layer to compute arbitrarily orientated text features [66]. An explicit supervision based on spatial information of characters and recurrent neural network for word recognition is also included. FOTS [1] introduces feature sharing with rotated text proposals to develop an end-to-end trainable system for detection and recognition of scene text instances. In a single network farword pass, TextBoxes [67] localize and recognize horizontal text and TextBoxes++ [68] deal with arbitrarily-oriented text instances. In [69], detection and recognition processes are serially connected yielding a straightforward relation between the text detection task and the followup text recognition task. A rectification of mask is used to adapt to incidental recognition of text instances with arbitrary orientation and shape. TextPlace [70] performs topological metric localization considering spatial-temporal dependence between text instances.

MLTS [4] is a multi-language text spotter that maps text proposals to a fixed height keeping the aspect ratio same, which is followed by detection and recognition. Each proposal uses a connectionist temporal classification to decode multi-language texts. E2E-MLT [71] is one of the popular multilingual optical character recognition for scene text detection and recognition. It uses a single shared fully convolutional network. OctShuffleMLT [72] is a hardware-efficient network for multi-language detection and recognition that also uses fully-convolutional neural network with a less number of parameters and layers.

Mask TextSpotter [11] uses semantic segmentation to detect text of arbitrary shapes and spatial attention for handling text instances of irregular shapes by simultaneously considering local and global textual information. TextDragon [73] describes the shape of text with a sequence of quadrangles to handle the text of arbitrary shapes and RoISlide that connect a deep network and connectionist temporal classification based text recognizer. The labeling of locations of characters is not needed. WACNET [12] applies a shared convolutional backbone between word-level segmentation and char-level detection and recognition. ASTS [74] customizes the mask R-CNN [75] to exploit the holistic-level semantics and pixel-level semantics for text spotting, simultaneously. It further delivers sequence-level semantics for text recognition using an attention-based sequence-to-sequence network.

III. PROPOSED CLUTTERED TEXTSPOTTER (CTS)

In this section, we propose a deep network, known as *Cluttered TextSpotter (in short CTS)*, for text spotting in scene images that has higher efficiency and consume less time and memory. It has the following modules:

- First, we introduce MobileNetV2 in our backbone network to make it light-weight in nature. To encode multi-scale contextual information, we integrate light-weight atrous spatial pyramid pooling with our network. We further include encoder-decoder model to recover the finer spatial information. This helps to exploit low-level features along with high-level and also reduces the degradation problem.
- Second, we design an oriented region proposal network, which is light-head is nature, to obtain oriented region proposals. Bilinear sampling is used for mapping the rotated region proposals into the canonical dimension. It also normalizes the rotation and scaling keeping aspect ratio and position of individual characters intact.
- Third, we develop a context encoding and refinement module for precisely regressing the oriented bounding boxes of text instances even in the presence of partially occluded text parts, truncation artifacts, perspective distortion, and inter-class interference. It is being served by both a global process of retrieving structural context and a local process sensitive to regional features.
- Finally, we use Gaussian softmax to mitigate the misclassification problem due to inter-class interference. We also incorporate a light-weight recognition module using C-LSTM within Bi-LSTM for multi-language character-level segmentation and word-level recognition.

A. LIGHT-WEIGHT BACKBONE NETWORK

The goal of this section is to obtain a light-weight backbone network that can encode multi-scale contextual information preserving finer spatial information. We therefore use the basic building block of MobileNetV2 [76] as a bottleneck depth separable convolution with residuals. Our backbone network contains initially a convolution layer with a kernel of size 3 × 3 and 32 filters, then 37 residual bottleneck layers, and finally one convolution layer with 1 × 1 kernel and 1280 filters, as shown in FIGURE 2. Let the size of the input image in the backbone network is $I \in \mathbb{R}^{2H \times 2W \times 3}$. Each residual bottleneck layer has kernel of size 3 × 3 along with RELU6, dropout, and batch normalization.

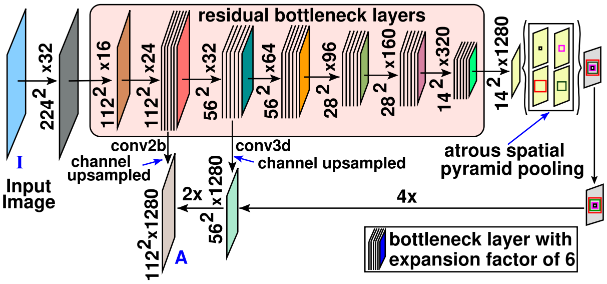


FIGURE 2. Architecture of the backbone network.

We adopt output stride as 16 to get a dense feature extraction by removing the striding in the third layer and apply a layer of atrous spatial pyramid pooling (ASPP) [77] after the last layer. With the increase in stride, the computation complexity will decrease, but dense feature matching will not be extracted. Scene texts are usually small in nature which requires dense features for precise detection. Therefore, there is always a trade-off between running time and high recall. We incorporate atrous separable convolution to able to encode multi-scale contextual information. Thus, an atrous convolution is divided into a depthwise convolution followed by a point-wise convolution. It drastically reduces computation complexity probing convolutional features at multiple scales using atrous convolution with four different rates {1,2,3,4}. Since we are not performing feature pooling in a deeper stage of the network, but feature computation in the initial stage of our network, therefore our sampling rates are much smaller in comparison with [77].

With the introduction of encoder-decoder architecture, the finer spatial information is recovered that helps to prevent from degradation problem. The integration of MobileNetV2 with ASPP helps to obtain the encoder of the backbone network. We further include the decoder to preserve rich spatial information in the backbone network. In the encoder, we bilinearly upsampled the output features of ASPP by a factor of 4. We apply 1 × 1 convolution on the low-level features of *residual conv3b* to increase the number of channels and concatenated with the corresponding upsampled features. We then apply a 3 × 3 convolution to refine the features, which is followed by another bilinear upsampling by a factor of 2. This reduces the number of computations in comparison with one bilinearly upsampling 8× directly. The output feature map of the backbone network is $A \in \mathbb{R}^{H \times W \times 1280}$, which is fed as input to the proposed R-CNN subnet.

B. ORIENTED REGION PROPOSAL NETWORK

In this section, we develop a region-based oriented proposal subnet, which is light-weight in nature, as depicted in FIGURE 3. We incorporate RoI warping to generate fixed-size feature maps. We produce feature maps with a smaller number of channels (thin feature maps) accompanied by traditional RoI warping. During training and inference, we find that RoI warping on thin feature maps will save memory and computation, keeping the accuracy same. Considering position sensitive RoI (PSRoI) pooling of rotated region proposals on thin feature maps, we decrease the R-CNN overhead to improve our performance. We reduce the channels from 1280 to 128 for obtaining thin feature maps.

To obtain rotated region proposals, we utilize rotated anchors [22]. Intersection-over-union (**IoU**) between a ground truth and an anchor is the overlap over skew rectangles. We consider a positive anchor that has an **IoU** > 0.7 with respect to the ground truth or the largest **IoU** overlap

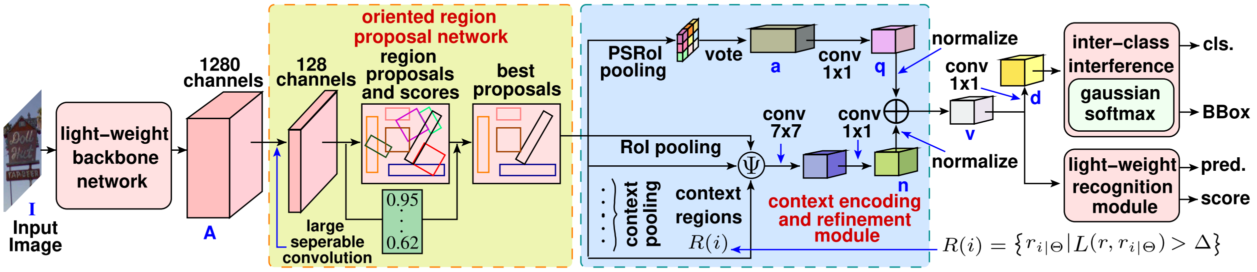


FIGURE 3. Architecture of the proposed Cluttered TextSpotter, where cls., BBox, pred., and score represent classification, bounding box, prediction, and recognition score, respectively.

with an intersection angle $\mho < \pi/12$ respect to the ground truth. Similarly, we choose a negative anchor that has an **IoU** < 0.3 or **IoU** > 0.7 with $\mho > \pi/12$ with a ground truth. All the regions which are not selected either by a positive anchor or a negative anchor are not utilized in training. We empirically found that it is reasonable to use **IoU** value as a trade-off between speed and accuracy.

We propose a region proposal $\mathbf{r}$ with a bounding box of 5 tuples $\mathbf{u} = (x, y, h, w, \theta)$ and a confidence score $\mathbf{s}$. The center coordinate of a bounding box is denoted by (x, y). The height h and width w represents the small and long sides of the bounding box, respectively. The angle between the $x-$axis (positive direction) and in a direction parallel to the long side of the rotated bounding box is represented by θ. The rotated anchors use scale, aspect ratio, and orientation parameters. The value of $\theta = \{-\pi/6, 0, \pi/6, \pi/3, \pi/2, 2\pi/3\}$ helps to control the runtime complexity and orientation coverage. The aspect ratio has the values {1:2, 1:5, 1:8} to cover a broad range of text-lines. We keep scales of text instances as 8, 16, and 32. Thus, we obtain 270 anchors [22] for the proposed region-based oriented subnet.

Each detected region has a different shape (aspect ratio and rotation). We therefore map the features of a region into a canonical dimension. We map a region $\widehat{\mathbf{r}} \in \mathbb{R}^{h' \times w' \times \varsigma}$ into a fixed height tensor of $\mathbf{r} \in \mathbb{R}^{h \times w \times \varsigma}$, we incorporate bilinear sampling of [10] and it is define as:

$$\mathbf{r}_{x,y} = \sum_{x'=1}^{h} \sum_{y'=1}^{w} \widehat{\mathbf{r}}_{x',y'} \Gamma(x' - \Upsilon_{x'}(x)) \Gamma(y' - \Upsilon_{y'}(y)), \tag{1}$$

where the kernel of bilinear sampling is $\Gamma(\vartheta) = \mathbf{max}(0, 1 - |\vartheta|)$. Υ is a point-wise coordinate transformation that maps the coordinates $\{x', y'\}$ of $\widehat{\mathbf{r}}$, fixed-size tensor, to the coordinates $\{x, y\}$ of the detected region $\mathbf{r}$, where we keep the number of channels unchanged. The transformation allows the shifting and scaling along the $x-$ and $y-$axes and the rotational parameters are extracted directly from the region parameters. The standard RoI warping lacks in comparison to feature representation as it not only allows the network to normalize scale and rotation, but also persist the aspect ratio and location of each character at the same time. This makes feature representation crucial for persisting accuracy in text spotting task.

C. CONTEXT ENCODING AND REFINEMENT MODULE

This section aims to address the challenges due to the truncation artifacts, partial occlusions, perspective distortion, and background clutter. We therefore incorporate a local process sensitive to regional features and a global process of retrieving structural context followed by context refinement to address the problem of unreliable results due to ill-localized regions, as illustrated in FIGURE 3.

Local part information captures the specific fine-grained parts of text instances and hence is important for classification task. We adopt PSRoI pooling of rotated region proposals to exploit local part information of text instances. PSRoI pooling typically ignores global structure information. If local part information is only used, a long word with partially occluded text parts will not be recovered as a single word. As a result, the long word will appear to be a combination of many small words. Global structure information is thus necessary for detection of texts in scene images, especially in a cluttered environment. In the presence of global structure information, the network will have structural information of a large region. If global structure information is only provided, then the network will have the complete structural information of a large region, but no specific fine-grained part related information will be present, which may lead to misclassification. We therefore utilize both global and local features for detection.

We also include context information to enhance the representation of global features. Context information also plays an essential role in classification task. For example, a boat is surrounded by water. The information about the surroundings (or background) for a particular object is the context for that object. Similarly, scene texts mostly occur in regions having a uniform texture and color, such as vehicle number plate, billboards, and door plates.

In the first branch, we used one fully-connected layer with 128 channels and no dropout for PSRoI pooling of each rotated proposal. It extracts text-specific parts followed by a voting task to obtain the resized tensor $\mathbf{q} \in \mathbb{R}^{32 \times 64 \times 128}$ using average pooling. It serves as the ensemble of multiple part models. In the second branch, we introduce global context encoding using RoI pooling of rotated region proposals. We incorporate a global context encoding to tackle the partial detection issue taking into account that the background regions provide informative clues and auxiliary discriminators. It describes a text region with an accurate status. Due to the wide diversity in size of text instances, we introduce an RoI pooling layer on rotated region proposals to extract a feature vector with fixed-length as the global descriptor. We also obtain a set of context regions R for a given region proposal $\mathbf{r}$ with variations based on $\Theta = \{h, w, \theta\}$ of the original proposal $\mathbf{r}$, such that,

$$R(i) = \{\mathbf{r}_{i|\Theta} | L(\mathbf{r}, \mathbf{r}_{i|\Theta}) > \Delta\}, \tag{2}$$

where $\mathbf{r}_{i|\Theta}$ is $i - th$ context region of the original proposal $\mathbf{r}$ by increasing height h or width w by $\delta-$times larger than the size of $\mathbf{r}$ or varying the orientation θ by $\pounds\pi$ ensuring that $\theta + \pounds\pi$ is within the interval $[-\frac{\pi}{4}, \frac{3\pi}{4}]$. In our implementation, we consider the threshold Δ as 0.5. We further use the **IoU** score of two regions as a measure of correlation level between them, such that,

$$L(\mathbf{r}, \mathbf{r}_{i|\Theta}) = \mathbf{IoU}(\mathbf{u}, \mathbf{u}_{i|\Theta}). \tag{3}$$

RoI pooled features from the original region and a set of context regions are then aggregated together. We adopt an adaptive weighting mechanism for aggregation. The visual features $\mathbf{v}$ extracted from the region $\mathbf{r}$ is bounded by $\mathbf{u}$ and its confidence score $\mathbf{s}$. We use $\mathbf{v}_{i|\Theta}$ and $\mathbf{w}_{i|\Theta}$ to denote the contextual information and the corresponding weight carried by $\mathbf{r}_{i|\Theta}$ for the original proposal $\mathbf{r}$. We implement the aggregation operation Ψ as follows:

$$\Psi(\mathbf{r}, \mathbf{v}_{i|\Theta}, \mathbf{w}_{i|\Theta}) = \frac{\sum_i (\mathbf{w}_{i|\Theta} \times \mathbf{v}_{i|\Theta})}{\sum_i \mathbf{w}_{i|\Theta}}, \tag{4}$$

where the value of $i = 0$ represents the original proposal $\mathbf{r}$. Two convolutional layers are incorporated having a kernel of size 7×7 and 1×1 to further exploit the global representation of oriented RoI and obtain a resized tensor, denoted by $\mathbf{n} \in \mathbb{R}^{16 \times 32 \times 128}$. The accuracy of precision is improved by refining the process of regressing the bounding boxes.

We apply 1×1 convolution for normalization separately on local part information enriched tensor $\mathbf{q}$ and global context enhanced feature tensor $\mathbf{n}$ followed by element-wise sum to concatenate and obtain the output tensor $\mathbf{v}$ for each rotated region proposal. The tensor $\mathbf{v}$ is fed into 1×1 convolution layer and a classifier to produce the output vector $\mathbf{d}$, a two-dimensional vector that indicates the probability whether a region is a text instance or not.

D. INTER-CLASS INTERFERENCE PROBLEM

The inter-class separability and intra-class compactness have a significant role in quantifying the efficiency of a network. Therefore, we address the inter-class interference by exploring inter-class separability and intra-class compactness of features learned by deep networks, as shown in FIGURE 3. The closeness of features within the same class is denoted by intra-class compactness and how discriminative the features of different classes are from each other is represented by inter-class separability. We assume that the distribution of text-specific features with reference to the background is subject to Gaussian distribution and Gaussian softmax [78] is thus used for classification on $\mathbf{d}$ as follows:

$$\mathfrak{P}(\mathbf{d}_i) = \frac{\mathbf{exp}(\lambda \times \Omega(\mathbf{d}_i, \mu_i, \sigma_i) + \mathbf{d}_i)}{\sum_{j=1} \mathbf{exp}(\lambda \times \Omega(\mathbf{d}_j, \mu_j, \sigma_j) + \mathbf{d}_j)}, \tag{5}$$

where μ and σ represent the mean and standard deviation of Gaussian distribution. Ω is the cumulative density function (CDF) and $\mathbf{d}_i$ is the $i - th$ element of $\mathbf{d}$. The use of softmax function depends on the value of λ. When $\lambda = 0$, traditional softmax function is obtained. Gaussian softmax helps in approximating distributions of text-specific features with large variations on the training samples. On the other hand, the softmax function learns only from the current observing sample. In addition, the use of distribution parameters μ and σ are used to directly quantify inter-class separability and intra-class compactness. It shows that the average accuracy of detection will be improved with the improvement in inter-class separability and intra-class compactness, even in the presence of cluttered background.

E. LIGHT-WEIGHT RECOGNITION MODULE

In this section, we predict the label sequence of each text instance globally. A cluttered environment leads to false label prediction of each character in the word. Inspired by [11], we include a position embedding mechanism to convert each word into a real-value vector and to specify a relative position of each character in a word as a pair of entity. We use position embedding on the input feature tensor $\mathbf{v}$ to obtain the embedded feature tensor F. Assume the feature tensor $\mathbf{v}$ has ν characters. The tensor $\mathbf{v}$ is divided by a fixed sliding window with the value $\mathfrak{m}$ and the step size of each slide is $\mathfrak{s}$ to obtain a hierarchical sequence $X = \{x_1, x_2, x_3, \cdots, x_\nu\}$. Each word has two annotated entities e_1 and e_2. The aim of relation classification is to identify the semantic relation between entities e_1 and e_2 in a given word. During training, we have used 36 classes are for alphanumeric characters in English, χ standard symbols and characters from other five languages, like Arabic, Bengali, Chinese, Japanese, and Latin, and 1 class for end-of-sequence symbol (EOS). We estimate χ based on the characters in RRC-MLT 2017 dataset.

The position embedding specifies the position of each character in a word as a pair of entity. We thus replace the target position of a character with its corresponding entity pair. We further compute the relative distance $\partial^{\phi}_{i,j}$ between $\phi - th$ character x^{ϕ}_i and target entity e^{ϕ}_j as follows:

$$\partial^{\phi}_{i,j} = x^{\phi}_i - e^{\phi}_j, \quad \forall i \in \{1, 2, \cdots, \nu\}, \text{ and } \forall j \in \{1, 2\}, \tag{6}$$

where x^{ϕ}_i and e^{ϕ}_j represent the position of current character x_i and target entity e_j in a word. For instance, in a given word "HOSPITAL", the relative position from the character "P" to entity "O" (e_1) is +2 ($\partial_{4,1}$) and entity "T" (e_2) is -2 ($\partial_{4,2}$). Each relative distance between current character and target entity is mapped to position vector. In the output feature tensor F contains the characters with relative position vectors. We aggregate the embedding feature tensor F with

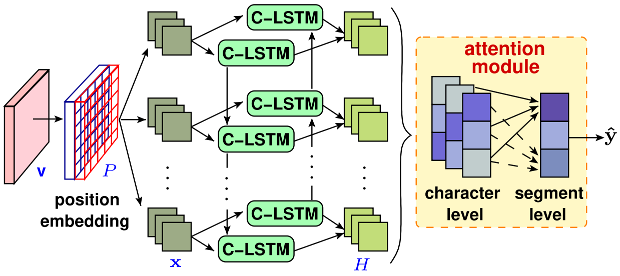


FIGURE 4. Architecture of the recognition module.

input feature tensor $\mathbf{v}$ to obtain a feature tensor P having channel $\zeta = 128$.

Next, we predict a label sequence of character class by incorporate C-LSTM [15], a variant of LSTM, in Bi-LSTM network (BiC-LSTM). Each cell in BiC-LSTM network is a C-LSTM cell. It is capable in capturing both local and global semantic information of a word. We therefore utilize it for label prediction, especially in cluttered environment. The feature tensor P is fed into the Bi-LSTM network with both a forward and a backward sequence to learn most discriminate spatial information. Let C-LSTM accepts the input sequence $\mathbf{x} = (\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T)$ from P, where each of $\mathbf{x}_t$ is a vector corresponding to step $t \in \{1, 2, \cdots, T\}$. Here, we formulate the equations of C-LSTM as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathfrak{W}_{ix} * \mathbf{x}_t + \mathfrak{W}_{ih} * \mathbf{h}_{t-1} + \mathfrak{W}_{ic} \circ \mathbf{c}_{t-1} + \mathbf{b}_{\boldsymbol{i}}),\\
\mathbf{f}_t &= \sigma(\mathfrak{W}_{fx} * \mathbf{x}_t + \mathfrak{W}_{fh} * \mathbf{h}_{t-1} + \mathfrak{W}_{fc} \circ \mathbf{c}_{t-1} + \mathbf{b}_f),\\
\mathbf{m}_t &= \mathbf{tanh}(\mathfrak{W}_{mx} * \mathbf{x}_t + \mathfrak{W}_{mh} * \mathbf{h}_{t-1} + \mathbf{b}_m),\\
\mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{m}_t,\\
\mathbf{o}_t &= \sigma(\mathfrak{W}_{ox} * \mathbf{x}_t + \mathfrak{W}_{oh} * \mathbf{h}_{t-1} + \mathfrak{W}_{oc} \circ \mathbf{c}_{t-1} + \mathbf{b}_o),\\
\mathbf{h}_t &= \mathbf{o}_t \circ \mathbf{tanh}(\mathbf{c}_t),\\
\mathbf{y}_t &= \mathfrak{W}_{yh} * \mathbf{h}_t,
\end{aligned}$$

where $\circ$ and $*$ denote the Hadamard product and convolution operation, whereas $\mathbf{b}_t$, $\mathfrak{W}_t$, $\sigma(\cdot)$, and $\mathbf{tanh}(\cdot)$ represent bias vectors, weight matrices, logistic element-wise sigmoid function, and hyperbolic tangent function, respectively. Also, the symbols $\mathbf{c}$, $\mathbf{f}$, $\mathbf{m}$, $\mathbf{i}$, $\mathbf{o}$, $\mathbf{h}$, and $\mathbf{y}$ present the cell state, forget gate, input modulation gate, input gate, output gate, hidden state, and cell output, respectively. On passing through BiC-LSTM, the feature tensors are transformed into several segments and each segment is associated with multiple characters. As we are dealing with detection of scene text in cluttered environment, the identification of semantic relation between the characters in a word is also important. The semantic relation of a word is determined by certain key characters. We therefore apply attention mechanism to automatically provide weights to the important characters in a word.

BiC-LSTM outputs $H = \{\mathbf{h}_1, \cdots, \mathbf{h}_t, \cdots, \mathbf{h}_T\}$, a matrix of hidden states, on which we apply character based attention mechanism to determine the most influential characters in $t - th$ segment $\mathbf{h}_t$. The character-level attention is computed by weighted sum of characters in $\mathbf{h}_t$ as follows:

$$\begin{aligned}
\mathbf{g}_t &= \mathbf{tanh}(\mathbf{h}_t \cdot \alpha_t^T),\\
\alpha_t &= \mathbf{softmax}((\mathfrak{W}_t^{\alpha})^T \cdot \mathbf{tanh}(\mathbf{h}_t)),
\end{aligned} \tag{7}$$

where $\mathbf{g}_t$ is the character-level attention vector, $\mathfrak{W}_t^{\alpha}$ is the model parameter, and α_t is the weight vector of $\mathbf{h}_t$. Furthermore, we apply segment-level attention mechanism on the output matrix $G = (\mathbf{g}_1, \cdots, \mathbf{g}_t, \cdots, \mathbf{g}_T)$ to weigh the importance of each segment in a word with the semantic relation η, we use bilinear function as follows:

$$\xi_t = (\mathbf{g}_t)^T \cdot \mathfrak{D} \cdot \eta_{emb}, \tag{8}$$

where $\mathfrak{D}$ is the weighted diagonal matrix learned during the training process and η_{emb} is the embedding of relation η. The normalized importance weight Φ_t is calculated by:

$$\Phi_t = \frac{\mathbf{exp}(\xi_t)}{\sum_{t'=1}^{T} \mathbf{exp}(\xi_t')}. \tag{9}$$

The final attention based vector $\mathbf{z}$ is obtained by a linear weighted combination of all segments as follows:

$$\mathbf{z} = \sum_{t=1}^{T} \Phi_t \mathbf{g}_t. \tag{10}$$

Finally, we utilize a softmax function and a linear transformation to calculate the conditional probability ($\mathcal{P}$) as follows:

$$\begin{aligned}
\hat{\mathbf{y}} &= \underset{\dot{\mathbf{y}}}{\mathbf{argmax}}\, \mathcal{P}(\dot{\mathbf{y}}),\\
\mathcal{P}(\dot{\mathbf{y}}|O) &= \mathbf{softmax}(\mathfrak{W}_o \times \mathbf{z} + \mathbf{b}_o),
\end{aligned} \tag{11}$$

where O denotes all parameters of our model. $\mathfrak{W}_o$ and $\mathbf{b}_o$ are learnable weight and bias.

IV. EXPERIMENTAL RESULTS

The effectiveness of our Cluttered TextSpotter (CTS) is validated by conducting abundant experiments on publicly available benchmark datasets for scene text detection, word spotting, and end-to-end recognition. We pretrain our model using synthtext [79] dataset for three epochs and initialize the weights from ImageNet [80] for detection process. On the other hand, weights are randomly initialized from $\mathcal{N}(0, 1)$ distribution for the recognition process. Data augmentation is also implemented to improve the robustness of our network. We split the training/testing in a three-fold manner for evaluation and use standard metrics to measure the performance in terms of accuracy and training parameters.

- **ImageNet** [80] is a large-scale image database. It is quite accurate and diverse in nature. It contains 80,000 noun synsets of WordNet and have 500-1000 clean and full-resolution images to illustrate each synset. ImageNet has tens of millions of annotated images that are built upon by the semantic hierarchy of WordNet [81].
- **synthtext** [79] is a popular synthetic dataset for scene text detection and recognition. It has a huge number of multi-oriented text instances that are annotated with character-level and word-level rotated bounding boxes.

It is composed of 800k images having 10 synthetic words per image, which are placed on real scene background. An image is annotated with a ground truth word and not at a character-level.

A. BENCHMARK DATASETS

We conduct an exhaustive set of experiments on six publicly available benchmark datasets.

ICDAR 2013 dataset [82] is a dataset that focuses on detection and recognition of horizontal text instances in natural scene images. There are 255 images in training set with 716 annotated words and 233 images in the test set. Apart from the bounding box, transcriptions are also assigned for each character-level and word-level text instance.

SVT Street View Text dataset [83] is composed of images from Google Street View that consist of frontal texts of street names, pavement markings, and shop names. It has 647 images of cropped words with lower resolutions and perspective distortion. Also, the dataset has only word-level annotations (no character bounding boxes).

ICDAR 2015 Incidental Text dataset [84] contains 1000 training and 500 testing images, collected using Google glasses and of somewhat low resolutions. It has multi-oriented text instances in each image, having word-level annotations for bounding boxes.

COCO-Text dataset [85] is one of the largest and challenging dataset that composed of 43686 training images and 20000 testing or validation images. It has text instances in arbitrary orientations. Some images in this dataset have blur edges.

MSRA-TD500 dataset [86] contains 500 indoor and outdoor scenes images that are captured using a pocket camera. The indoor office and mall images contain caution plates, signs, and door plates, while the outdoor street images are mostly guided billboards and boards in English and Chinese languages with complex backgrounds.

RRC-MLT 2017 dataset [87] consists of multi-language and multi-oriented text instances in scene images. It is widely applicable for the task like multi-lingual text detection, crop word script identification, and joint text detection and script identification. It contains 18000 images, which comprised of text of six scripts belonging to nine languages. The training set has a total of 9000 images for nine languages.

B. IMPLEMENTATION DETAILS

We use pretrained model on ImageNet dataset [80]. The complete training process contains two stages. First, we utilize synthtext dataset [79] to train the network and then fine-tuned on the benchmark datasets on which it is to be tested. Our network is implemented with NVIDIA Titan X graphic card and Intel E5-2670v3 CPU running at 2.30 GHz.

•**Training.** We jointly train detection and recognition module of our proposed network for three epochs on a merged dataset of RRC-MLT 2017, ICDAR 2015, ICDAR 2013, SVT, COCO-Text, and MSRA-TD500 with a fixed sample ratio 2:2:1:1:1:1. We randomly cropped up to 30% of height and width each image. We set the mini-batch to 8 for initial experiments. In case of RPN and Fast R-CNN, the batch sizes are maintained at 256 and 512.

The end-to-end training use curriculum learning [88] technique to train the model from gradual to complex data efficiently. We train multiple tasks jointly in a single model by propagating the generalization capability from synthesized images to real-world data; (1) we randomly pick 600k images from 800k synthetic images. The training of the recognition branch is performed by freezing the detection branch, where 120k iterations are utilized in the training process with a learning rate of 10^{-3}; (2) the next 80k iterations is utilized for detection only. The learning rate is set to 10^{-4}. In the next 20k iterations, we get sampling tensors from detection task. The network is trained end-to-end in this stage; and (3) in the next 70k iterations, nearly 70k real-world images from the COCO-Text, RRC-MLT 2017, ICDAR 2013, MSRA-TD500, SVT, and ICDAR 2015 datasets are chosen. We also conduct data augmentation [18], [89] to enhance the generalization ability. Furthermore, we incorporate character-level supervision by making a batch of size 4. The images are taken from the synthetic dataset. We are maintaining the learning rate as 10^{-4}.

Stochastic gradient descent with adam optimizer, a weight decay of 10^{-3}, and a momentum of 0.9 are used. The input is fed in mini-batches of ψ images, where $\psi = 8$. Due to the presence of less number of real samples, we use data augmentation and multi-scale training in the fine-tuning stage. Moreover, we perform a random rotation of input images in a range of $[-30°, 30°]$ angles. Due to the presence of cluttered background noise, some background textures appear similar to the text instances in noisy scene images, which makes it hard for a network to distinguish between text and non-text instances. This compels the training process to be unbalanced and leads to slow convergence. We therefore use hard a negative mining strategy [18] to suppress training unbalance. We perform two-stage training on a dataset. In first stage, the negative ratio between negatives and positives is set to 3:1, whereas it changed to 6:1 in second stage. We select a multi-scale training mechanism [90] to make our network robust. Furthermore, we also use several data augmentation techniques [89], [91]. Also, we provide an input image

TABLE 1. Effect of different variations of MobileNetV2, ShuffleNetV2 and IGCV2 as backbone networks on ICDAR 2015 dataset.

Backbone	**F-measure**	**Flops**	**Params**	**MAdds**
		(G)	(M)	(B)
MobileNetV2 [76]	92.4	0.7	2.4	2.9
MobileNetV2+ASPP [76]	93	1.1	5.2	5.9
MobileNetV2+ASPP +Encoder-Decoder (Ours)	**93.7**	1.2	5.9	6.1
SSDLite [76]	92.8	**0.8**	**3.4**	**1.4**
ShuffleNetV2 [92]	91.7	2.8	10.9	8.7
IGCV2 [93]	91.3	4.6	23.1	10.4

TABLE 2. Effect of different branches of context encoding and refinement module.

Models	ICDAR 2013			SVT		
	Precision	Recall	F-measure	Precision	Recall	F-measure
CTS-LP	96.9	94.4	95.5	84.8	76.1	78.6
CTS-GS	96.2	94.9	95.4	84.2	76.4	78.9
CTS-GC	97.2	95.1	95.9	85.3	76.8	79.1
Ours	**97.6**	**95.4**	**96.1**	**85.8**	**77.1**	**79.6**

TABLE 3. Effect of different softmax functions on COCO-Text dataset.

Function	Precision	Recall	F-measure
Softmax	68.1	59.2	63.2
G-softmax ($\mu = 0, \sigma = 1$)	72.3	58.3	63.7
G-softmax ($\mu = 0, \sigma = 5$)	71.3	58.6	62.9
G-softmax ($\mu = -0.1, \sigma = 1$)	74.2	**59.8**	**67.4**
G-softmax ($\mu = 0.1, \sigma = 1$)	**75.1**	57.9	66.8

TABLE 4. Specifications of the smartphones with Adreno-640 GPU that are used for experimentation.

	Smartphone	Operating System	Internal Memory	RAM
D1	Samsung Galaxy S10+	Android 9	1 TB	12 GB
D2	Asus ROG Phone II	Android 9	1 TB	12 GB
D3	Xiaomi Mi 9 Pro 5G	Android 10	512 GB	12 GB
D4	Oneplus 7 Pro	Android 9	256 GB	12 GB
D5	Google Pixel XL4	Android 10	128 GB	6 GB
D6	LG G8X ThinQ	Android 9	1 TB	6 GB
D7	Sony Xperia 5 Plus	Android 10	1 TB	6 GB

with a larger size to achieve better detection of multi-scale text at the third stage of training. For multi-scale training, we randomly resize the shorter sides of the input images to (600, 800, 1000, 1200, 1400) scales randomly.

Interference: In the inference stage, we use a single model to evaluate all datasets, but the scales of the input images depend on the datasets. We obtain a predefined number, *i.e.*, 300 text region proposals from RPN in our experiment through a forward pass and then get the outputs for detection and recognition tasks. We provide *strong*, *weak*, and *generic* dictionaries for testing reference [66]. In the strong lexicon, 100 words per-image are assigned including all words that appear in the image. The weak lexicon consists of all words that present in entire test set of the dataset, whereas the generic dataset has 90k words. We kept the words of length greater than three in dictionaries, where signs and numbers are excluded. We prefer to use two models for evaluation, *i.e.*, end-to-end and word-spotting. The end-to-end model recognizes all the words accurately even if a detected string is not present in the dictionary. While the word-spotting model only inspects about the presence of the word of the dictionary in the images. It is therefore less strict than end-to-end model for avoiding numbers, symbols, and words whose length is less than 3. We evaluate all results in one single scale without referring to any lexicon.

TABLE 5. Performance comparison on ICDAR 2013 dataset.

Methods	Precision	Recall	f-measure
[27]	94.4	90.1	92.2
[29]	92.8	91.4	92.1
[18]	92.4	83.8	87.9
[21]	92	84	88
[43]	93	85.2	88.9
[66]	91	88	90
[3]	90	83	86
[20]	91.2	85.5	88.3
[45]	90.3	87.1	88.7
[50]	88.8	80.2	84.3
[31]	93	86	90
[7]	94	91	93
[68]	84	91	88
[32]	92	84.4	88
[33]	92	86	89
[34]	93.3	87.5	90.3
[17]	92	81	86
[22]	95	88	91
[79]	92	83	75.5
[35]	84.5	87.7	86
[39]	90	75	81
[40]	91.1	86.1	88.5
[36]	67	87	75
[41]	91.9	87.1	89.5
[67]	89	83	86
[37]	57.3	63.4	60.2
[2]	97.4	93.1	95.2
[11]	94.8	89.5	92.1
Ours	**97.6**	**95.4**	**96.1**

C. ABLATION STUDY

In this section, we perform extensive analyses and ablation studies to evaluate the detection and recognition accuracy and computational efficiency of the proposed network. We conduct a comprehensive set of experiments to study different aspects of our network. The experiments were undertaken on split 1 of ICDAR 2013, ICDAR 2015, SVT, RRC-MLT 2017, COCO-Text, and MSRA-TD500 datasets.

- **Impact of backbone network.** We have carried out a large set of experiments to select a backbone network enriched in spatial information with an optimal number of parameters. We study the impact of MobileNetV2, MobileNetV2+ASPP, MobileNetV2+ASPP+Encoder-Decoder (Ours), SSDLite, ShuffleNetV2 [92], and IGCV2 [93] as backbone network in the evaluation of f-measure of our proposed network. TABLE 1 depicts that our backbone network outperforms other networks in terms

TABLE 6. Performance comparison on ICDAR 2015 dataset.

Methods	Precision	Recall	f-measure
[69]	83.7	96.1	89.5
[27]	89.6	84.9	87.2
[24]	90.4	86.7	88.5
[29]	89.1	85.5	87.3
[18]	88	76.8	82
[21]	88	84	86
[1]	91.5	87.9	89.8
[16]	83.2	78.3	80.7
[65]	91.4	80.5	85.6
[28]	87	86.2	86.6
[43]	86.1	79.9	82.9
[23]	88.3	85	86.6
[66]	87	86	87
[3]	61	40	48
[20]	83.9	80.7	82.3
[49]	89.1	85.5	87.3
[45]	88.2	85.7	86.9
[30]	87.9	81.6	84.6
[7]	**93**	90	91
[5]	83	81	82
[13]	82	83	82
[44]	90.8	81.5	85.9
[50]	72.2	58.6	64.7
[31]	89.9	80.4	84.9
[68]	87.8	78.5	82.9
[32]	89.5	79.7	84.3
[33]	88	80	83.8
[34]	79.3	77	78.2
[17]	82	80	81
[22]	84	77	80
[41]	91.9	87.1	89.5
[42]	84.3	83.9	84.1
[47]	86.6	87.6	87.1
[48]	86.2	82.7	84.4
[73]	92.4	83.7	87.8
[2]	89.8	84.3	86.9
[11]	86.6	87.3	87
[23]	88.3	85	86.6
Ours	91.8	**96.4**	**93.7**

TABLE 7. Performance comparison on MSRA-TD500 dataset.

Methods	Precision	Recall	F-measure
[29]	86.3	77.1	81.4
[16]	87.2	67.4	76
[20]	83.2	80.2	81.7
[49]	86	83.4	84.7
[45]	85.5	74.1	79.4
[50]	76.5	75.3	75.9
[32]	87.6	76.2	81.5
[17]	77	70	74
[22]	82	69	75
[35]	67.2	77.2	72.4
[36]	52	85	65
[37]	45	53.3	48.8
[42]	87.4	75.9	81.3
[47]	85.7	81.1	83.3
[23]	84.2	81.7	82.9
Ours	**87.8**	**86.1**	**86.5**

TABLE 8. Performance comparison on RRC-MLT 2017 dataset.

Methods	Precision	Recall	F-measure
[27]	77.6	67.4	72.1
[24]	79.6	70	74.5
[1]	81.8	62.3	70.7
[4]	71.4	65.8	68.5
[28]	78	67.4	72.3
[49]	**82.9**	70.5	**76.2**
[30]	73.9	66.9	70.2
[44]	80	69.8	74.3
[32]	74.3	70.6	72.4
[48]	79.5	66.8	72.6
[2]	80.6	68.2	73.9
Ours	80.9	**71.2**	75.9

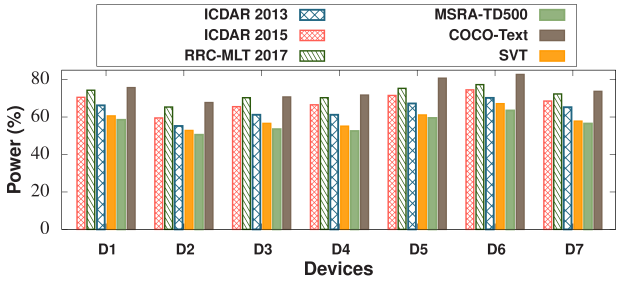
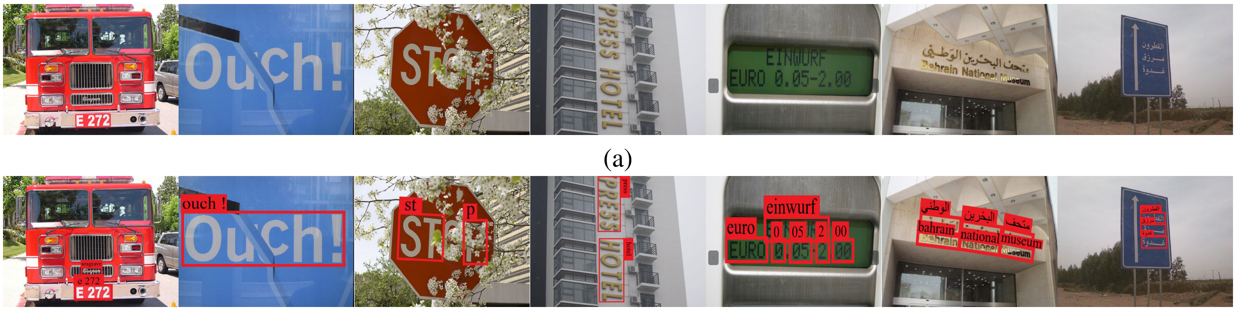


FIGURE 5. Effect of datasets on power consumption for different devices.

of training parameters and computational complexity of the network. Our backbone network has enriched with low-level spatial details and high-level context information. The network is not kept much deeper to restrict the number of training parameters.

• **Impact of Context Encoding and Refinement module.** In the Cluttered TextSpotter, the context encoding and refinement module consist of local, global, and context branches. We evaluate the impact that each branch on the overall performance of the CTS network, as shown in TABLE 2. For this evaluation, we make ablation studies, where we consider the CTS network as the baseline and create three more models. We drop the global and context branches and name the network as *CTS-LP* and perform experiments on it using split 1 of ICDAR 2015 and SVT datasets. Likewise, only the global structure information of the global context branch is used in this model is denoted by *CTS-GS*. This model has both global structure information and context information of the global context branch. It provides a means to evaluate the contribution of context branches to the performance of the overall network. Context information helps to extract text information efficiently even in cluttered environment. This branch is known as *CTS-GC*.

• **Impact of inter-class interface.** We perform experiments with softmax (vanilla) and variations of G-softmax functions for detection of text instances. The G-softmax function consistently outperforms the softmax function on COCO-Text datasets, as shown in TABLE 3. G-softmax function is utilized to quantify the compactness and separability of features. In our analysis, we observe that the improvement of



(a)

(b)

FIGURE 6. Text spotting results of the proposed network, where images are taken from different benchmark datasets.

TABLE 9. Performance comparison on COCO-Text dataset.

Methods	Precision	Recall	F-measure
[38]	60	33	42
[50]	43.2	27.1	33.3
[31]	74	51	61
[68]	60.8	56.7	58.7
[32]	61.9	32.4	42.5
[33]	64	57	61
[34]	45.2	30.9	36.8
[11]	66.8	58.3	62.3
Ours	**74.2**	**59.8**	**67.4**

TABLE 10. Performance comparison on SVT dataset.

Methods	Precision	Recall	F-measure
[16]	50.3	32.4	39.4
[79]	26.2	26.7	27.4
[35]	60.4	68.7	64.2
[39]	**87**	73	79
[40]	54.1	75.8	63.1
[36]	55	68	61
[41]	54.1	75.8	63.1
[67]	67.2	60.8	63.8
[37]	41.6	44.3	42.9
Ours	85.8	**77.1**	**79.6**

intra-class compactness and inter-class separability improves the average precision of the detection network.

- **Impact of different devices.** We implement the proposed text spotter on several smartphones, as shown in FIGURE 5. The technical specifications, such as processing speed and memory, are shown in TABLE 4. We use a Monsoon power monitor that can measure the power consumption of smart devices, alike literature [94]. It is evident from the result that our CTS network is competent with smart devices.

D. COMPARISON WITH STATE-OF-THE-ART RESULTS

In this section, we compare our network with the state-of-the-art approaches [1], [3], [10], [11], [16], [18], [20], [21], [23], [24], [27]–[29], [43], [45], [49], [65], [66], [69], [69], [71]–[73] on six different benchmark datasets. We consider recall, precision, and f-measure as the metrics for evaluation of accuracy of detection. Alike [1], [10], [69], we conduct experiments to compare our recognition results based on *strong*, *weak*, and *generic* lexicons. We also perform a comparative study for parameter count of our network with the existing approaches.

TABLE 11. Performance comparison on SVT dataset.

Methods	Word-Spotting	
	strong	generic
[67]	84	64
[64]	76	53
[79]	67.7	55.7
[65]	84.9	66.1
Ours	**85.8**	**67.6**

TABLE 12. Performance comparison on RRC-MLT 2017 dataset.

Methods	Word-Spotting	End-to-End
[71]	**65.1**	48
[72]	60.7	56.6
[72]+[71]	63.9	58.8
Ours	**65.1**	**59.3**

- **Detection results on different datasets.** Cluttered TextSpotter is compared with the recent literature for both detection and recognition on standard evaluation metrics, like precision, recall, and f-measure. It is clear from the results tabulated in TABLE 5 and TABLE 6 that detection accuracy of our network is improved by more than 1% in terms of f-measure on both ICDAR 2013 and ICDAR 2015 datasets, respectively, with respect to the baseline literature FOTS [1]. TABLE 7 and TABLE 9 show that our network outperforms existing approaches on MSRA-TD500 and COCO-Text datasets. CTS performs better in terms of recall on RRC-MLT 2017 and SVT datasets, as depicted in TABLE 8 and TABLE 10, respectively.
- **Recognition results on different datasets.** The proposed network achieves state-of-the-art end-to-end text recognition accuracy for all three lexicons, as shown in TABLE 13, for ICDAR 2013 and ICDAR 2015 datasets. CTS performs better existing literature for both strong and generic lexicons for word spotting. TABLE 11 and TABLE 12 illustrate that our network outperforms recent literature for multi-language and distorted text instances for RRC-MLT 2017 and SVT datasets, respectively.

TABLE 13. Performance comparison on ICDAR 2013, and ICDAR 2015 datasets for the recognition.

Methods	ICDAR 2013						ICDAR 2015					
	End-to-End			Word-Spotting			End-to-End			Word Spotting		
	strong	weak	generic	strong	weak	generic	strong	weak	generic	strong	weak	generic
[69]	91.4	90.3	84.8	86	83.4	66.9	96.1	95.1	89	89.6	87.1	70
[1]	91.9	90.1	84.7	83.5	79.1	65.3	95.9	93.9	87.7	87	82.3	67.9
[65]	-	-	-	91	89.8	84.5	-	-	-	94.1	92.4	88.2
[4]	87	84	70	73	67	59	-	-	-	-	-	-
[66]	91	89	86	82	77	63	93	92	87	85	80	65
[10]	89	86	77	54	51	47	92	89	81	58	53	51
[68]	93	**92**	85	73.3	65.8	51.9	96	95	87	76.4	69	54.3
[67]	91	89	84	-	-	-	94	92	87	-	-	-
[73]	-	-	-	82.5	78.3	65.1	-	-	-	86.2	81.6	68
[11]	93.3	91.3	88.2	92.7	**91.7**	87.7	83	77.7	73.5	82.4	78.1	73.6
[74]	92.8	91.5	85.9	84.8	79.8	66.5	95.9	94.7	88.5	87.7	82.9	68.9
Ours	**93.8**	**92**	**88.5**	**93.1**	**91.7**	**88.1**	**96.3**	**95.4**	**89.2**	**94.3**	**92.5**	**73.8**

TABLE 14. Test time speed in terms of on FLOPS, number of training parameters, and frames per second (FPS) on ICDAR 2015 dataset for detection (D), recognition (R), or spotting (S).

Methods	Flops (G)	Params (M)	fps	D/R/S
[1]	9.997	34.98	9.0	S
[16]	4.685	24.1	13.2	D
[71]	2.946	4.7	-	R
[72]	1.525	4.8	-	R
[72]+[71]	**0.829**	**1.6**	-	R
[2]	10.239	20.8	-	D
[69]	-	-	3.7	S
[10]	-	-	9	S
Ours	1.139	5.914	**24.8**	S

- **Qualitative results.** FIGURE 6 illustrates the qualitative results of text spotting (detection and recognition) of scene text instances. It is evident from outputs that the proposed Cluttered TextSpotter can accurately spot scene texts in the scene images even in the presence of a cluttered background environment.
- **Speed and Model Size.** We have reduced both the parameter count and computational cost by using a hardware-efficient backbone network and region proposal network. The test time speed of the proposed CTS network is reported in TABLE 14 and compared with state-of-the-art scene text detection methods. To evaluate the run time complexity, we have exhibit the flops, number of training parameters (params), and frames-per-second (fps) of our network.

V. CONCLUSION

In this paper, we have proposed an efficient network for scene text spotting that uses local, global, and context information of multi-scale feature maps of light-weight backbone network. The proposed text spotter works efficiently, even with the cluttered background environment in scene images in resource-constrained devices, like smartphones, for the detection of multi-lingual text instances, logos, and symbols with high accuracy and speed. We have addressed the problem of misclassification caused by inter-class interference using Gaussian softmax. We also evaluated our Cluttered TextSpotter on publicly available benchmark datasets, which outperformed previous methods in terms of efficiency and performance.

ACKNOWLEDGMENT

This work was supported by Science and Engineering Research Board (SERB) file numbers ECR/2017/002419 and ECR/2016/000406 and scheme Early Career Research Award.

REFERENCES

[1] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5676–5685.

[2] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9357–9366.

[3] S. Qin and R. Manduchi, "Cascaded segmentation-detection networks for word-level text spotting," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 1275–1282.

[4] Y. Zhou, S. Fang, H. Xie, Z.-J. Zha, and Y. Zhang, "MLTS: A multilanguage scene text spotter," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 163–168.

[5] S. Mohanty, T. Dutta, and H. P. Gupta, "Recurrent global convolutional network for scene text detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2750–2754.

[6] S. Mohanty, T. Dutta, and H. P. Gupta, "An efficient system for hazy scene text detection using a deep CNN and patch-NMS," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2588–2593.

[7] R. Bagi, S. Mohanty, T. Dutta, and H. P. Gupta, "Leveraging smart devices for scene text preserved image stylization: A deep gaming approach," *IEEE Multimedia*, vol. 27, no. 2, pp. 19–32, Apr./Jun. 2020.

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

[9] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, "CoupleNet: Coupling global structure with local parts for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4146–4154.

[10] M. Busta, L. Neumann, and J. Matas, "Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2231.

[11] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 26, 2019, doi: 10.1109/TPAMI.2019.2937086.

[12] Y. Gao, Z. Huang, Y. Dai, K. Chen, J. Guo, and W. Qiu, "Wacnet: Word segmentation guided characters aggregation net for scene text spotting with arbitrary shapes," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3382–3386.

[13] S. Mohanty, T. Dutta, and H. PrabhatGupta, "Robust scene text detection with deep feature pyramid network and CNN based NMS model," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3741–3746.

[14] S. Mohanty, T. Dutta, and H. P. Gupta, "Text preserving animation generation using smart device," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 1039–1044.

[15] S. Wang, Z. Li, C. Ding, B. Yuan, Q. Qiu, Y. Wang, and Y. Liang, "C-LSTM: Enabling efficient LSTM using structured compression techniques on FPGAs," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, Feb. 2018, pp. 11–20.

[16] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2642–2651.

[17] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 745–753.

[18] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3482–3490.

[19] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3454–3461.

[20] P. Xie, J. Xiao, Y. Cao, J. Zhu, and A. Khan, "RefineText: Refining multi-oriented scene text detection with a feature refinement module," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1756–1761.

[21] C. Du, C. Wang, Y. Wang, Z. Feng, and J. Zhang, "TextEdge: Multi-oriented scene text detection via region segmentation and edge classification," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 375–380.

[22] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[23] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4229–4238.

[24] J. Duan, Y. Xu, Z. Kuang, X. Yue, H. Sun, Y. Guan, and W. Zhang, "Geometry normalization networks for accurate scene text detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9136–9145.

[25] Y. Liu, L. Jin, Z. Xie, C. Luo, S. Zhang, and L. Xie, "Tightness-aware evaluation protocol for scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9604–9612.

[26] C. Wang, H. Fu, L. Yang, and X. Cao, "Text co-detection in multi-view scene," *IEEE Trans. Image Process.*, vol. 29, pp. 4627–4642, 2020.

[27] Z. Zhong, L. Sun, and Q. Huo, "A teacher-student learning based born-again training approach to improving scene text detection accuracy," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 281–286.

[28] P. Yang, G. Yang, X. Gong, P. Wu, X. Han, J. Wu, and C. Chen, "Instance segmentation network with self-distillation for scene text detection," *IEEE Access*, vol. 8, pp. 45825–45836, 2020.

[29] M. Cao, Y. Zou, D. Yang, and C. Liu, "GISCA: Gradient-inductive segmentation network with contextual attention for scene text detection," *IEEE Access*, vol. 7, pp. 62805–62816, 2019.

[30] Y. Xiao, M. Xue, T. Lu, Y. Wu, and S. Palaiahnakote, "A text-context-aware CNN network for multi-oriented and multi-language scene text detection," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 695–700.

[31] F. Sheng, Z. Chen, T. Mei, and B. Xu, "A single-shot oriented scene text detector with learnable anchors," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1516–1521.

[32] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.

[33] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5909–5918.

[34] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "WordSup: Exploiting word annotations for character based text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4950–4959.

[35] K. S. Raghunandan, P. Shivakumara, S. Roy, G. H. Kumar, U. Pal, and T. Lu, "Multi-script-oriented text detection and recognition in video/scene/born digital images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1145–1162, Apr. 2019.

[36] S. Dey, P. Shivakumara, K. S. Raghunandan, U. Pal, T. Lu, G. H. Kumar, and C. S. Chan, "Script independent approach for multi-oriented text detection in scene image," *Neurocomputing*, vol. 242, pp. 96–112, Jun. 2017.

[37] V. Khare, P. Shivakumara, and P. Raveendran, "A new histogram oriented moments descriptor for multi-oriented moving text detection in video," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7627–7640, Nov. 2015.

[38] P. Cheng, W. Wang, and Y. Cai, "Multi-scale scene text detection via resolution transform," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 988–993.

[39] D. He, X. Yang, W. Huang, Z. Zhou, D. Kifer, and C. L. Giles, "Aggregating local context for accurate scene text detection," in *Proc. ACCV*, 2017, pp. 280–296.

[40] Y. Tang and X. Wu, "Scene text detection using superpixel-based stroke feature transform and deep learning based region classification," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2276–2288, Sep. 2018.

[41] Y. Tang and X. Wu, "Scene text detection and segmentation based on cascaded convolution neural networks," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1509–1520, Mar. 2017.

[42] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.

[43] D. He, X. Yang, D. Kifer, and C. L. Giles, "TextContourNet: A flexible and effective framework for improving scene text detection architecture with a multi-task cascade," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 676–685.

[44] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, "Mask R-CNN with pyramid attention network for scene text detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 764–772.

[45] X. Guo, J. Li, B. Chen, and G. Lu, "Mask-most net: Mask approximation based multi-oriented scene text detection network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 206–211.

[46] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6442–6451.

[47] Y. Liu, L. Jin, and C. Fang, "Arbitrarily shaped scene text detection with a mask tightness text detector," *IEEE Trans. Image Process.*, vol. 29, pp. 2918–2930, 2020.

[48] P. Dai, H. Zhang, and X. Cao, "Deep multi-scale context aware feature aggregation for curved scene text detection," *IEEE Trans. Multimedia*, early access, Nov. 11, 2019, doi: 10.1109/TMM.2019.2952978.

[49] S. Zhang, Y. Liu, L. Jin, Z. Wei, and C. Shen, "OPMP: An omnidirectional pyramid mask proposal network for arbitrary-shape scene text detection," *IEEE Trans. Multimedia*, early access, Mar. 9, 2020, doi: 10.1109/TMM.2020.2978630.

[50] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," *CoRR*, vol. abs/1606.09002, pp. 1–10, Jun. 2016.

[51] F. Cong, W. Hu, Q. Huo, and L. Guo, "A comparative study of attention-based encoder-decoder approaches to natural scene text recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 916–921.

[52] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[53] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4168–4176.

[54] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2231–2239.

[55] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5086–5094.

[56] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5571–5579.

[57] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1508–1516.

[58] W. Liu, C. Chen, and K. Y. K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI*, 2018, pp. 5571–5579.

[59] F. Zhan and S. Lu, "ESIR: End-to-end scene text recognition via iterative image rectification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2054–2063.

[60] S. Wang, Y. Wang, X. Qin, Q. Zhao, and Z. Tang, "Scene text recognition via gated cascade attention," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1018–1023.

[61] M. Yang, Y. Guan, M. Liao, X. He, K. Bian, S. Bai, C. Yao, and X. Bai, "Symmetry-constrained rectification network for scene text recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9146–9155.

[62] F. Sheng, Z. Chen, and B. Xu, "NRTR: A no-recurrence sequence-to-sequence model for scene text recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 781–786.

[63] Z. Zhang, P. Chen, X. Shi, and L. Yang, "Text-guided neural network training for image recognition in natural scenes and medicine," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 25, 2019, doi: 10.1109/TPAMI.2019.2955476.

[64] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, Jan. 2016.

[65] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5248–5256.

[66] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5020–5029.

[67] 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.

[68] 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.

[69] J. Li, Z. Zhou, Z. Su, S. Huang, and L. Jin, "A new parallel detection-recognition approach for end-to-end scene text extraction," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1358–1365.

[70] Z. Hong, Y. Petillot, D. Lane, Y. Miao, and S. Wang, "TextPlace: Visual place recognition and topological localization through reading scene texts," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2861–2870.

[71] M. Bušta, Y. Patel, and J. Matas, "E2E-MLT—An unconstrained end-to-end method for multi-language scene text," in *Proc. ACCV Workshop*, 2019, pp. 127–143.

[72] A. Lundgren, D. Castro, E. Lima, and B. Bezerra, "OctShuffleMLT: A compact octave based neural network for end-to-end multilingual text detection and recognition," in *Proc. Int. Conf. Document Anal. Recognit. Workshops (ICDARW)*, Sep. 2019, pp. 37–42.

[73] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9075–9084.

[74] J. Liu, Z. Chen, B. Du, and D. Tao, "ASTS: A unified framework for arbitrary shape text spotting," *IEEE Trans. Image Process.*, vol. 29, pp. 5924–5936, Apr. 2020, doi: 10.1109/TIP.2020.2984082.

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

[76] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[77] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[78] Y. Luo, Y. Wong, M. Kankanhalli, and Q. Zhao, "G-softmax: Improving intraclass compactness and interclass separability of features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 685–699, Feb. 2020.

[79] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.

[80] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[81] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.

[82] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

[83] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[84] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "Robust reading competition," in *Proc. ICDAR*, 2015, pp. 1156–1160.

[85] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*. [Online]. Available: http://arxiv.org/abs/1601.07140

[86] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1083–1090.

[87] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, W. Khlif, M. M. Luqman, J.-C. Burie, C.-L. Liu, and J.-M. Ogier, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification–RRC-MLT," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1454–1459.

[88] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

[89] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[90] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[91] 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.

[92] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. ECCV*. Cham, Switzerland: Springer, 2018, pp. 122–138.

[93] G. Xie, J. Wang, T. Zhang, J. Lai, R. Hong, and G.-J. Qi, "Interleaved structured sparse convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8847–8856.

[94] T. Dutta and H. P. Gupta, "Leveraging smart devices for automatic mood-transferring in real-time oil painting," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1581–1588, Feb. 2017.

RANDHEER BAGI (Graduate Student Member, IEEE) received the M.Tech. degree in computer science from the Central University of South Bihar, India, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, IIT Varanasi (BHU), Varanasi, India. His research interests include deep learning, computer vision, and human–computer interaction.

TANIMA DUTTA (Member, IEEE) received the Ph.D. degree in computer science and engineering from IIT Guwahati, India, in 2014. She is currently an Assistant Professor with the Department of Computer Science and Engineering, IIT Varanasi (BHU), India. Her research interests include computer vision and deep networks.

HARI PRABHAT GUPTA (Member, IEEE) received the M.Tech. and Ph.D. degrees in computer science and engineering from IIT Guwahati, India, in 2010 and 2014, respectively. He is currently an Assistant Professor with the Department of Computer Science and Engineering, IIT Varanasi (BHU), India. His research interests include wireless networks and game theory.