IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 July 2022, accepted 19 August 2022, date of publication 1 September 2022, date of current version 13 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203691

RESEARCH ARTICLE

GMIF: A Gated Multiscale Input Feature Fusion Scheme for Scene Text Detection

TOFIK ALI[1], MOHAMMAD FARIDUL HAQUE SIDDIQUI[2], SANA SHAHAB[3], AND PARTHA PRATIM ROY[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India
[2]Department of Computer Science, West Texas A&M University, Canyon, TX 79016, USA
[3]Department of Business Administration, College of Business Administration, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding author: Tofik Ali (tali@cs.iitr.ac.in)

This work was supported by the Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia, under Project PNURSP2022R259.

ABSTRACT The feature fusion of the multi-scale features plays a significant role in localizing text instances of different sizes in the scene text detection (STD) paradigm. The existing approaches are not sufficient to tackle the issues of multi-scale text; consequently, their performance also varies with the text size. Here, we propose a gated multi-scale input feature fusion (GMIF) approach to overcome this issue in STD. The GMIF generates the local features from down-scaled input images and propagates these features from low resolution to the higher resolution global features through a gated recurrent unit-like mechanism. The consistent performance of the GMIF is validated with different text instance sizes of the test-set of the Total-text dataset. The GMIF obtained the performance in range (Precision 88.554-89.106, Recall 85.452-85.790, and f-measures 87.072 - 87.417) with marginal deviation, whereas the current state-of-the-art method, DBNet++, acquired in range (Precision 73.005-82.666, Recall 80.912-87.274, and f-measures 76.755 - 84.183) with significant deviation. Besides this, GMIF also achieved the best performance (f-measures) over ICDAR 2015 (as 88.0), Total-Text (as 87.4), and the second-best over the MSRA-TD500 (as 85.2) dataset. We have conducted an ablation study to show the impact of different components of the GMIF on the STD tasks, which shows the effectiveness of the overall GMIF approach.

INDEX TERMS Scene text detection, multi-scale text, multi-scale feature, feature-fusion, deep neural networks.

I. INTRODUCTION

Many different pieces of information are present in an image, but the text contains a wide range of valuable information. Text appears in our day-to-day life as part of the road signs, shops, buildings, vehicle license plates, product packaging, and information media (online and offline). The advent of deep learning provides a wide range of possibilities for text analysis in practical applications like image/video understanding, visual search, instant translation, automatic driving, blind assistance, scene understanding, and geolocation [1], [2], [3], [4], [5]. The detection and localization of all text instances in the natural images are the prerequisites for text understanding, also known as STD.

The current state-of-the-art text detection methods can be categorized into:

1) The bounding box regression methods
2) The segmentation-based methods

The bounding box regression methods generate the bounding box's representation, including the target text instance from the input image. This representation includes horizontal rectangles, oriented rectangles, quadrilateral boxes, and polygons from the contour of the text instances. One of the drawbacks of the bounding box regression method is its inability to constitute curved text instances. Even for a polygon, the number of points is changeable for all types of text instances, and predicting these points through regression-based methods is challenging.

The segmentation-based methods, on the contrary, represent the text instance by text region masking. The advantage

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

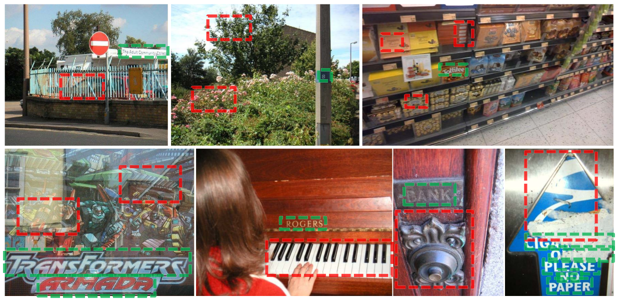


FIGURE 1. Background Ambiguity: Some examples of ambiguity present in real world images. The examples are selected from different dataset under consideration. The green color box represents the text regions, and the red color box represent some background region containing text similarity.

of these methods is that they can handle all types of text shapes and orientations but require post-processing to get the localization information of the text instances.

A. CHALLENGES IN STD

Significant progress has been effectuated in STD by the techniques based on deep learning [6], [7], [8], [9], [10], [11], [12], [13]; nevertheless, the SDT continues to pose a challenge. The major obstacles for STD in natural images range from arbitrary orientations, scale, aspect ratio, and shapes to foreground, background, and texture interference. These impediments are discussed below.

1) RECEPTIVE FIELD OF THE NEURAL NETWORK

A deep neural network (DNN) provides promising results for STD, but it has a fixed receptive field that can provide text detection for the corresponding text size range. A multi-scale testing (MST) approach can extend this range and improve the recall of the text detection system as it can process a text with multiple scales. However, the MST approach decreases the system's precision by producing multiple results for the same text instance.

2) TextSize

The size of the receptive field of the DNN regulates the recognition of text of varying sizes. If the effective receptive field of the DNN is large, the existing text localization methods are hindered from recognizing small text. Rescaling the input image can possibly resolve this issue. This method of flushing the input image to fit the text instances into the network effective receptive field is called multi-scale testing. References [6], [14] attained a better performance by increasing the size of the input image, indicating that the effective receptive field of the trained network [15] was quite large. The up-scaling of the input image to make text instances larger is not the desired solution as the existing resizing methods lack textual information, and generated image needs more computations. Besides this, the up-scaling also requires a solution from another research area named image super-resolution [16]. The down-scaling of the input image, on the contrary, is relatively easy and produces quite good results [8], [17].

3) OVERLAPPING TEXT REGIONS

Despite the small text size issues, the segmentation-based text detection methods also suffer from overlapping text instance masks. This overlapping complicates the text region extraction task; therefore, these methods require a few post-processing steps to separate these masks. The bounding box regression-based methods generate multiple boxes for the same text instances and need non-max suppression [18] to get the best representation.

4) REAL-TIME PERFORMANCE

A conventional approach to increase the receptive field of the output neurons of a DNN is to apply a pooling operation followed by a sequence of convolution operations. This approach facilitates covering a larger text region [19], [20], [21], [22], [23]. Since most of the computation done by a DNN is dense at the initial layers, the number of convolutional kernels used at these layers is kept relatively low. However, these layers provide the core feature for text understanding. So increasing the capacity of these layers might help in decision making, but real-time performance is compromised. Besides this, the vanishing gradient [24] and dying ReLu issue make learning these kernels obstinate.

B. MOTIVATION

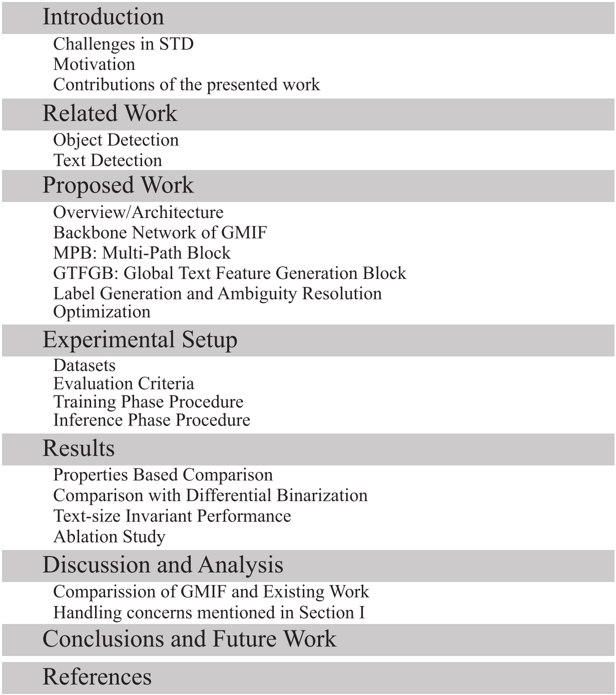
The techniques discussed in the previous sections are capable of detecting text that is contingent on the size of the text. This limitation is exposed further when the real world presents an untold variation of text, and it becomes very challenging to detect these variations. These limitations carved the way for the presented work. The authors have identified a couple of significant flaws in the existing methods, which are as follows:

One of the major concerns about the existing methods is over dependence on the text size. This is conspicuous that as the performance of the existing state-of-the-art methods line, DB [6] and DB++ [25] fluctuate with the variation in the size of the text in the test images. (refer to table 5).

The second concern perpetuates the first one. The existing methods try to curtail the influence of text size by learning for all text sizes. This turns out to be highly infeasible, and hence the reliance on such systems decreases for detecting text of larger sizes. To overcome this, a multi-scale testing approach is employed by these methods. The multi-scale testing produces multiple text instances at different scaled inputs and hence, it is able to detect text of varying sizes. However, this comes with the cost of a decrease in the precision of the overall system. Another overhead, although nominal, associated with multi-scale testing is the requirement of post-processing.

C. CONTRIBUTIONS OF THE PRESENTED WORK

The following are the significant contributions of the presented work:



Introduction
- Challenges in STD
- Motivation
- Contributions of the presented work

Related Work
- Object Detection
- Text Detection

Proposed Work
- Overview/Architecture
- Backbone Network of GMIF
- MPB: Multi-Path Block
- GTFGB: Global Text Feature Generation Block
- Label Generation and Ambiguity Resolution
- Optimization

Experimental Setup
- Datasets
- Evaluation Criteria
- Training Phase Procedure
- Inference Phase Procedure

Results
- Properties Based Comparison
- Comparison with Differential Binarization
- Text-size Invariant Performance
- Ablation Study

Discussion and Analysis
- Comparission of GMIF and Existing Work
- Handling concerns mentioned in Section I

Conclusions and Future Work

References

FIGURE 2. Paper structure.

- A framework that generates consistent performance across text instances of varying sizes.
- A shallow backbone network architecture capable of detecting all text instances of smaller sizes.
- A block within the main network known as GTFGB is responsible for feature propagation, and it adapts text features to various scales.
- The combination of the backbone network with GTFGB produces a single text segmentation map comprising every instance of text in the input image. This lessens the burden of post-processing that is required to select the best mask for a text instance.

The rest of the paper is organized as shown in Figure 2.

II. RELATED WORK

Text detection is the process of identifying the existence of text in the input image. It is an integral part of the text analysis and aims to localize every occurrence of the text. The text detection methods fall under a sub-category of non-exclusive object detection, one of the fundamental problems in the computer vision research domain. This section provides an overview of the text detection methods related to the presented research. Firstly, in subsection II-A, we present the methods of text detection falling under the realm of object detection. Then we discuss some specific text detection methods in subsection II-B

A. OBJECT DETECTION

A range of text detection methods [7], [8], [26], [27] have been proposed, which are inspired by the object detection methods. Remarkable object detection works have been presented with a sequence of evolved advancements. R-CNN [28] used object classification to detect an object in regions generated by the selective search [29]. The classification is performed on the features generated by a convolutional neural network (CNN) on the cropped image region according to a selective search. This method produces promising results, but it is slow as it applies the CNN on every cropped image region separately. This limitation is softened in the Fast R-CNN [30] by altering the sequence of crops and CNN. The Fast R-CNN first generates the CNN feature map from the complete image, then extracts features of the proposals by RoIpooling [30]. This allows the CNN feature map to be shared in multiple region proposals as their overlapping area. Consequently, it provides a significant speed-up to R-CNN. The selective search is an external method to the CNN, which impedes the system's end-to-end learning.

Faster R-CNN [31] presented a region proposal network alongside the CNN of Fast R-CNN, which gave the system end-to-end trainable capabilities. Although the Faster R-CNN has a speed improvement than the R-CNN and Fast R-CNN, cannot produce real-time results (≥25 frames per second). The root cause for this is RoIpooling, as it generates a fixed-size feature map for all valid boxes which require the alignment of features. This alignment is removed in YOLO [32] which achieves real-time performance. The performance of the YOLO is further enhanced by the SSD [33]. SSD uses default boxes to predict the bounding region box at each location. It utilizes the default boxes of different aspect ratios and scales at different stages of the backbone network. Thus, SSD achieves translation, aspect ratio, and scale invariance and enhances the original YOLO's performance with its real-time computation. Recently, a faster R-CNN-based technique known as OLCN [34] has been developed for detecting small objects in remote sensing images. However, it is still difficult to recognize small objects in remote sensing images due to the fact that very few pixels are available for the targeted small object and the majority of the visible space within the network's receptive field is occupied by other objects or the background information.

B. TEXT DETECTION

The available methods are broadly categorized into two types, as segmentation based methods [6], [9], [17], [35], [36], [37], and bounding-regions(box regression) based methods [7], [8], [26], [38], [39].

1) SEGMENTATION BASED METHODS

A segmentation-based method classifies every cell (a region in input, which size depends on the stride used) as text and non-text or provides some character-ness score. The Character-ness, text segmentation mask, as a mean of text saliency measure, is introduced in [37] and further utilized in many others [6], [9], [17], [35], [36]. These methods are capable of detecting text of any shape, including different orientations and curved text. This capability is also utilized in [36] for multi-oriented text segmentation with a CNN-based feature. Further, besides the Text-nonText classes, some more classes as the text border elements are utilized in [35] for better separation of the text instances. The PSENet [40] proposed progressive scale expansion by segmenting the text instances with different scale kernels. PixelLink [41] utilizes a VGG16 based convolution neural network to generate the text-nontext segmentation mask as well as an 8-neighbor link-connections prediction map for each pixel location. Thus, using link connections between the pixels of various text instances, PixelLink [41] was able to separate texts that were close to one another.

2) BOUNDING-REGIONS BASED METHODS

With the massive success of RCNN [28] based methods for object detection tasks, a range of its derivative-based methods are also suggested for text localization. Here, the text coverage region is learned as a regression task with anchor boxes of different scales and aspect ratios. The TextBoxes [42] is one of the pioneers in exploring the possibilities of SSD [33] (a derivative of RCNN) by modifying the convolution kernel shape and the aspect ratio of anchor boxes to cover the longer word/text. This method has limited capacity to identify oriented text instances as it used the horizontal rectangle box to represent text regions. This limitation was softened by DMP-Net [38] and TextBoxes++ [8] with quadrangle/quadrilaterals box representation of text region, which improves the localization and detection of text instances with arbitrary orientation. The work presented by SSTD [27] incorporated an attention mechanism to identify promising text regions at a coarse level and further refine it with a hierarchical inception module. The works proposed by EAST [7] and DeepReg [43] explore the possibilities of direct regression of text region by quadrilateral boxes. Their proposed methods are anchor-box-free and predict the text instances at each pixel concerning its position. RRD [39] disengaged the learning task as classification and regression. It utilized the rotation-dependent feature to enhance the performance of the multi-oriented and long text instances. The rotation-invariant features are used for classification, whereas the rotation-sensitive features facilitate the regression. A dimension-decomposition region proposal network is proposed in DeRPN [44] to handle the scale variation in text instances. The box-regression-based methods require simple post-processing, generally non-maximum suppression (NMS). However, these methods lack the representation of accurate bounding boxes for curved-shaped text instances.

III. PROPOSED WORK

This section provides the details of the proposed GMIF model. Subsection III-A gives a broad overview of the overall system, and the subsequent subsections(III-B to III-D) explain its different components. Subsection III-E is about the process of label generation and the ambiguity resolution scheme. Finally, subsection III-F deals with the loss function used to train the model.

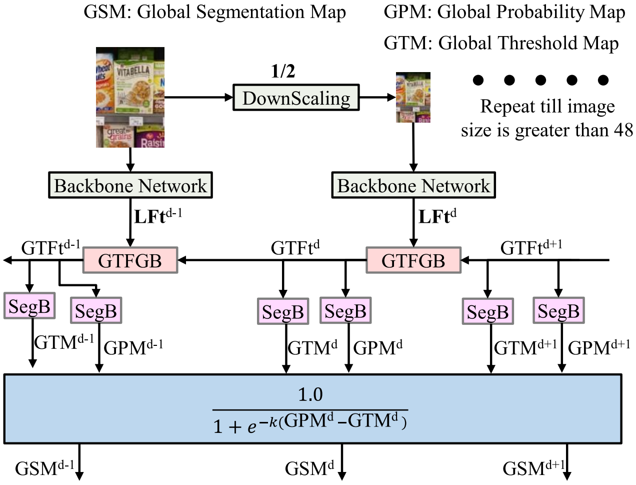


FIGURE 3. Overview of the proposed GMIF model: The different downscaled versions (till the smallest size reached below 48 pixels) of the input image are created and pass through the backbone network, which creates the local feature LFt^{d}, here d stands for the d^{th} downscaled input images. The global text feature (GTFt^{d}) for a scaled input image is generated by global text feature generation block (GTFGB) from its LFt^{d} and GTFt^{d+1} of the low scaled input image. The GTFt^{d} is also utilized to generate the segmentation map at its corresponding scale. The detail of different components of the GMIF model is given in Figure 4, and Figure 5.

A. OVERVIEW/ARCHITECTURE

The system architecture of the GMIF model is illustrated in Figure 3, and the detail of its different components is shown in Figure 4. The GMIF model takes the original image and its downscaled version as an input and generates the text-segmentation maps. The backbone network of the GMIF produces the local features (LFt) for a scaled version of the input image. The global text feature generation block (GTFGB) generates the global text feature (GTFt) for any scaled input image. The GTFGB utilizes the current LFt and GTFt from the lower-resolution input image to develop the GTFt of the current scale. The GTFGB at the lowest resolution input image uses a zeros valued map as GTFt of lower-resolution input. The GTFt at any scale is classified by the segmentation-Block and generates GPM (Global Probability Map) and GTM (Global Threshold Map). The GSM (Global Segmentation Map) is calculated from the GPM and GTM with the equation 1. The equation 1 generates the segmentation map with the concept of differentiable binarization (refer DB [6] and DB++ [25]). The layers and blocks used in different scaled inputs have shared parameters (the backbone, GTFGB, and Segmentation-Block are all the same across the input scale).

$$GSM_{i,j} = \frac{1.0}{1 + \exp -k(GPM_{i,j} - GTM_{i,j})} \tag{1}$$

where the value of k is set as 50 same as the DB [6].

B. BACKBONE NETWORK OF GMIF

The backbone network of the GMIF model is responsible for acquiring the text-nontext feature from the input image. The backbone is depicted in Figure 4 with its components. The proposed backbone does not use any pooling operation for feature map size reduction. Instead, it uses a convolution layer of large kernels (256) with a stride of 4 × 4. This operation provides detailed information for small regions and does not require significant computation. The backbone network should capture the input image features for a range of text sizes; therefore, we utilize square and non-square kernels in convolution as MPB: text inception block. MPBs capacitate the backbone network to cover a large area with fewer parameters and computation. MPBs also help to detect text instances with curvature. Here, we also aim to maintain the real-time performance of the GMIF, so the number of kernels at different layers is also restricted. Besides, the backbone's architecture is partially dense connected; before feature map reduction, all previous stage features are concatenated and followed by a convolution operation with a stride of 2 × 2, which provides more robust and detailed features for subsequent stage feature extraction. The segmentation of the text's boundary regions is challenging; therefore, we fuse high-level and low-level features to create the local scale-level features, termed as LFt.

C. MPB: MULTI-PATH BLOCK

The receptive field of the Backbone network after the first convolution layer is increased by only multi-path block (MPB). The MPB has kernels of different shapes (1 × 3, 3 × 1, and 3 × 3), which is the concept borrowed from GoogLeNet [21]. The text segmentation task is equivalent to the classification of the center of an image patch. The text inside this patch can appear at any location and with any orientation. Thus the kernels with different shapes help to extract the underline information efficiently. The performance of the text detection also validates that this behavior decreases if only a square (3 × 3) kernel shape is used in MPB (refer to ablation study V-D section). Here we target only a small range of text sizes, so we have included only elementary kernel shapes such as 1 × 3, 3 × 1, and 3 × 3. The design details of MPB are given in the top-right corner of Figure 4.

D. GTFGB: GLOBAL TEXT FEATURE GENERATION BLOCK

GTFGB gets LFt of the current scale and GTFt from the lower-resolution (higher scale) input image and produces the updated and upscaled GTFt for the current scale. The GTFt of the lowest resolution input image is generated from its LFt, and a zero-valued map as GTFt from the lower resolution. The design details of GTFGB are given in the bottom-right corner of Figure 4. Here the operation of GTFGB resembles the GRU [45] followed by a transpose convolution layer. Here the GTFt is equivalent to the hidden state (h_t) and LFt as the current input (x_t) of GRU. The primary use of the GTFGB is to propagate the valuable information of the previous scale's GTFt and update it according to the current LFt. Thus, GTFGB relieves the backbone network from covering the whole input image and acquires the more prominent neighborhood information through feature fusion of LFt and GTFt. The smaller text information is weekend at the down-scaled input image, so the GTFt at any scale acquires the small text information from the LFt at that scale. Therefore GTFt at any scale has a lower bound on text size, which it can localize efficiently. LFt encode the information from the image patch under its receptive field only, whereas the GTFt has the information from the whole input image. The approach for updating the GTFGB is bottom-up, where the features from the low-resolution input image are incorporated into the GTFt first.

E. LABEL GENERATION AND AMBIGUITY RESOLUTION

1) LABEL GENERATION

The label generation depends on two properties of a pixel location in an input image. We will refer to these properties as pixel location properties (PLPs) for onward discussion. The PLPs are as follows

- The distance of a pixel from the text boundary (we refer to it as DistB)
- The corresponding text size at a pixel location (we refer to it as RT). It is represented by the radius of the circle that covers the text height.

The DistB and RT of a pixel outside of the text mask are considered ones. The PLPs of a pixel inside a text mask is determined by the algorithm 1. The PLPs are calculated at the original input image resolution(refer to Figure 6). The pixel class for a pixel location is decided by algorithm 2, please refer to Ambiguity Resolution section III-E2).

2) AMBIGUITY RESOLUTION

We refer d^{th} downscaled image as IMG^d, DistB as DistB^d, and RT as RT^d. The class of a pixel location is decided by algorithm 2. The GMIF model learns the global probability map (GPM^d), the global segmentation map (GSM^d) and the global threshold map (GTM^d) at different scales of the input image. The GPM, GSM, and GTM are responsible for only a small corresponding range of text-size regions segmentation. If a text region occupies this range, then only it is considered TEXT. Besides this, the GTM is a two-valued $\{0.1, 0.9\}$ map whereas the DB [6] and DBNet++ [25] used a continuous-valued map in the range (0,1).

F. OPTIMISATION

We adopted the same loss function as the DB++, which is the weighted sum of the loss for the global probability map L_p, the loss for the global segmentation map L_s, and the loss for the global threshold map L_t.

$$L = L_p + \alpha \times L_s + \beta \times L_t \tag{2}$$

The weights α and β are selected as 1.0 and 10.0 according to their numeric values in losses L_s and L_t.

We have utilized the binary cross-entropy (BCE) loss with hard negative mining for both losses of L_p, L_s.

$$L_p = L_s = \sum_{i \in Sl} y_i log x_i + (1 - y) log(1 - x_i) \tag{3}$$

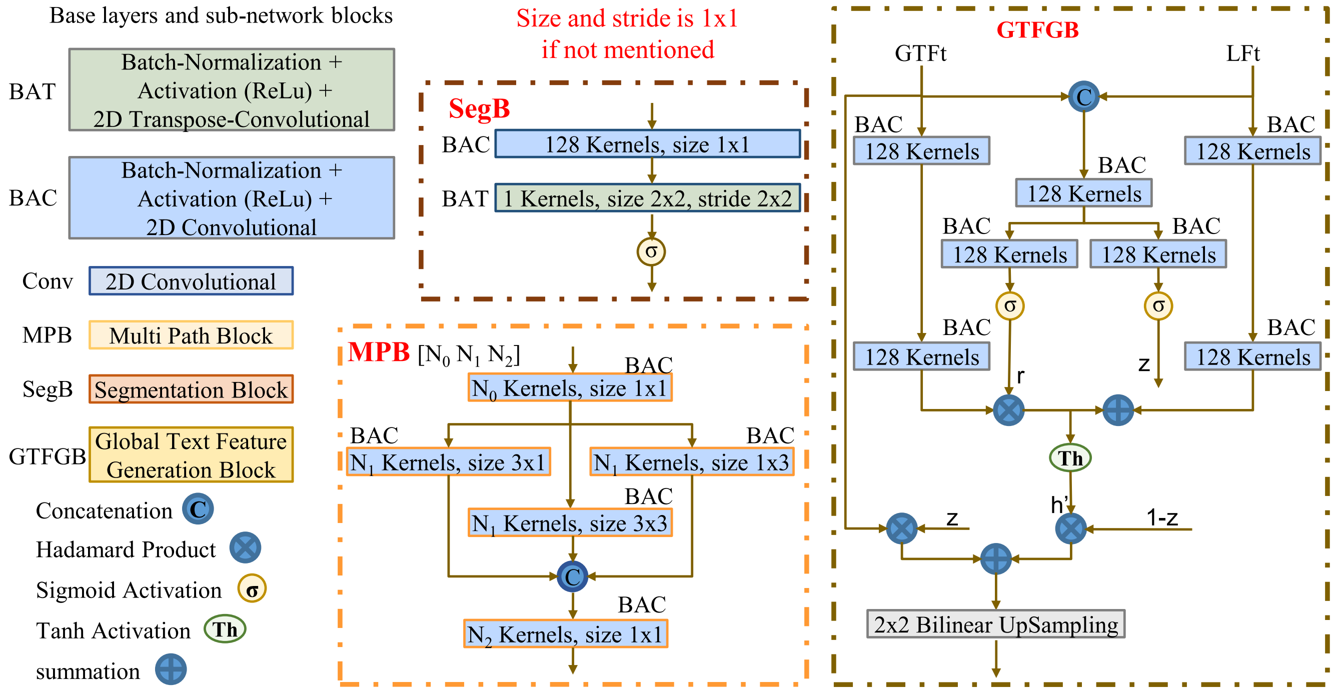


FIGURE 4. Different Components of GMIF: The leftmost section provides abbreviation details of the different layers and sub-network blocks used in GMIF. The right most section depicts the architecture of the global text feature generation block (GTFGB). The mid section provides the detail of the different block used in GMIF as MPB: mult-path block (mid-bottom), SegB: segmentation block (mid-top).

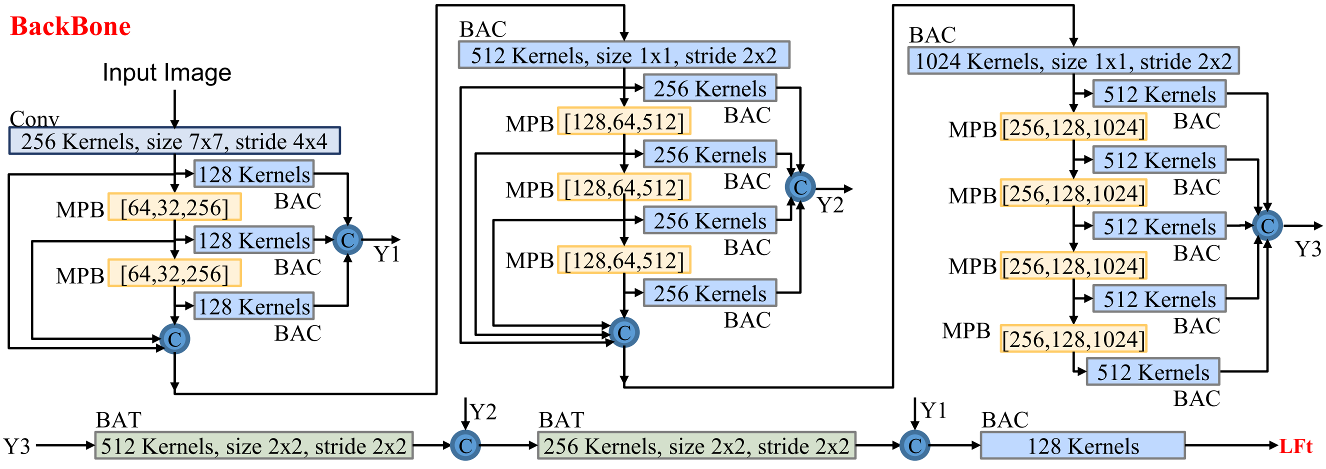


FIGURE 5. Backbone architecture of GMIF: the architecture detail of the backbone network is presented here. The details of the different layers and blocked used in backbone are depicted in Figure 4.

where Sl is the sampled set having positive and negative samples's ratio as 1:3. The Sl contain all positive samples and then select top-k (based on the absolute error) in negative prediction. Here we are using the TEXT pixels location as positive and NONTEXT as negative.

The global threshold map loss L_t is computed as the sum of L1 distances between the prediction and GTM label.

$$L_t = \sum_{i \in R_d} | \, y_i^* - x_i^* \, | \tag{4}$$

where R_d is a set of pixels locations inside the GTM with numeric values (pixel location don't have DONTCARE class); y^* is the numeric value {0.1, 0.9} for the threshold map.

IV. EXPERIMENTAL SETUP

The learning of the GMIF model with different datasets and settings is performed by a shared NVIDIA DGX system having 8 V100 GPUs each with 32 GB of memory. The learning is done in multiple sessions with 1/2/4 GPUs depending on the availability of GPUs on the system. Besides this, all the

Algorithm 1 Pixel Location Properties (PLPs) Calculation

```
Input: Text Instances Mask
Output: DistB, RT at each pixel locations.
    {EuclidDist is the euclidean distance between two pixel locations}
    Initialisation:
TIs ← All Text Insances in Text Mask
for all $TI \in TIs$ do
    $BCs \leftarrow$ Boundary pixels of current TI
    for all $pixel \in TI$ do
        $DistB_{pixel} \leftarrow \min_{BC_i \in BCs} EuclidDist(pixel, BC_i)$
    end for
    $MAXT \leftarrow \max_{pixel \in TI} DistB_{pixel}$
    $MAPs \leftarrow$ Medial axis locations of current TI
    for all $pixel \in TI$ do
        for all $pixel_i \in MAPs$ do
            if $DistB_{pixel_i} + 1 \geq EuclidDist(pixel, pixel_i)$ then
                $RT^1_{pixel,pixel_i} \leftarrow DistB_{pixel_i}$
            else
                $RT^1_{pixel,pixel_i} \leftarrow 1$
            end if
        end for
        $RT_{pixel} \leftarrow \max_{pixel_i \in MAPs} RT^1_{pixel,pixel_i}$
        {The corner refinement: The corner of text instances may not have the correct text size in RT. Therefore, we need to fix them as the boundary of TI.}
        if $DistB_{pixel} \geq 2 \wedge RT_{pixel} \leq \frac{MAXT}{3}$ then
            $RT_{pixel} \leftarrow \frac{MAXT}{3}$
        end if
    end for
end for
return $DistB, RT$
```

Algorithm 2 Pixel Location Class Categorization

```
Input: $IMG, DistB, RT$
Output: $GPM, GSM, GTM$
    Initialisation:
$S \leftarrow$ Smaller Side of IMG
$d \leftarrow 0$
{We need to downscale the input image so the maximum text-size becomes less than 48 pixels}
while $S > 48$ do
    $IMG^d \leftarrow$ Pooling by Stride $2^d \times 2^d$
    $DistB^d \leftarrow \frac{DistB+2^d-1}{2^d}$
    $RT^d \leftarrow \frac{RT+2^d-1}{2^d}$
    for all $Pixel_{Location} \in IMG^d$ do
        if $RT^d_{Pixel} < 3$ then
            $GPM^d_{Pixel} \leftarrow NONTEXT$
            $GTM^d_{Pixel} \leftarrow DONTCARE$
        else
            if $RT^d_{Pixel} > 6$ then
                if $DistB^d_{Pixel} > \frac{RT^d_{Pixel}}{3}$ then
                    $GPM^d_{Pixel} \leftarrow TEXT$
                    if $DistB^d_{Pixel} < \frac{2*RT^d_{Pixel}}{3}$ then
                        $GTM^d_{Pixel} \leftarrow 0.9$
                    else
                        $GTM^d_{Pixel} \leftarrow DONTCARE$
                    end if
                else
                    $GPM^d_{Pixel} \leftarrow NONTEXT$
                    $GTM^d_{Pixel} \leftarrow 0.1$
                end if
            else
                $GPM^d_{Pixel} \leftarrow DONTCARE$
                $GTM^d_{Pixel} \leftarrow DONTCARE$
            end if
        end if
    end for
    $S \leftarrow \frac{S+1}{2}$
    $d \leftarrow d + 1$
end while
$GPM \leftarrow \{GPM^d, \forall d\}$
$GTM \leftarrow \{GTM^d, \forall d\}$
$GSM \leftarrow GPM$
return $GPM, GSM, GTM$
```

inference computation is done by a separate system having a GTX 1080Ti GPU, which is the same as used by DB [6] for a fair comparison.

A. DATASETS

The performance of the GMIF is evaluated on three publicly available benchmarking datasets: ICDAR 2015 (first introduced in the ICDAR 2015 Robust Reading Competition) [46], MSRA-TD500 [47], and the Total-Text [48]. Besides these datasets, we have also incorporated two more instances of the datasets. Firstly, we have created a text size-specific dataset from the test set of the total-text dataset [48] for evaluation of different methods under text size constraints. Secondly, we have incorporated 400 training images from HUST-TR400 [49] as suggested and used by DB [6]. The detail of the main three datasets are summarized in the Table 1.

1) RESIZED TOTAL-TEXT

We have created a test set that validates the proposed work's superior performance. Here we choose the test set of the Total-Text dataset as it has text instances of different shapes and orientations. First, we identify the average RT_{avg} of every text instance of all test images, then we resized these images such that the RT_{avg} of the targeted text instance in the image becomes according to the desired RT_{avg}. After resizing, if the text instance is not according to the desired size, it is marked as a don't care instance. Some sample images from this dataset is shown in figure 8. The consistent performance of the proposed GMIF model over this dataset is validated in table 5.

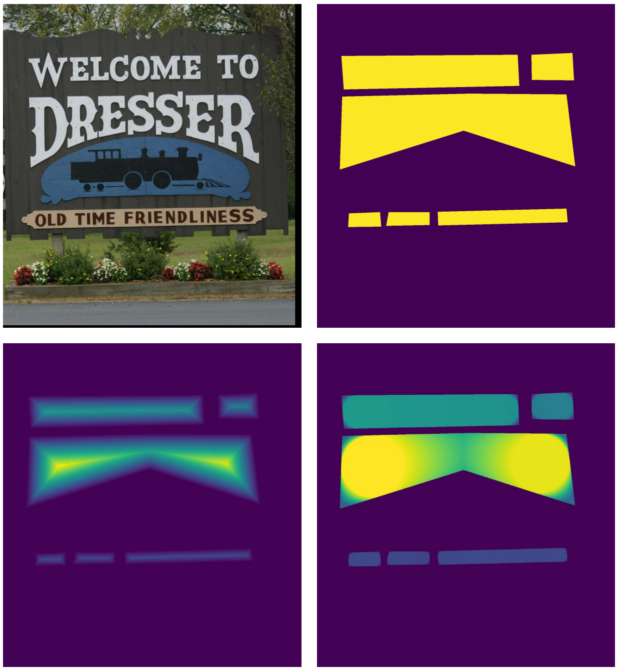


FIGURE 6. Label Generation: The top-left is the original image and its downscaled versions, and on the top-right, their corresponding text instances mask. The bottom-left is the Border distance map, and the bottom-right is the text height map for the input image and its downscaled versions.

TABLE 1. Summary of datasets used for GMIF evaluation.

Dataset	# Images	IR	A	AO	CI
ICDAR 2015 [46]	Total:1500 Train:1000 Test:500	720 X 1280	Word-level (oriented rectangles as bounding box)	Yes	No
MSRA-TD500 [47]	Total:500 Train:300 Test:200	1296 X 864 & 1920 X 1280	Text-line level (enclosing text polygons)	Yes	No
Total-Text dataset [48]	Total:1555 Train:1255 Test:300	Variable	Word-level (enclosing text polygons)	Yes	Yes

Legend:IR:Image Resolution, **A**:Annotation, **AO**:Arbitrary Orientation, **CI**:Curved Instances

B. EVALUATION CRITERIA

The text detection evaluation relies on the *precision* (How many detected regions are correct) (P) and *recall* (How many regions were retrieved). Generally, a text detection method uses some threshold to decide a text region. The precision and recall vary with this threshold. Decreasing the threshold can improve the recall, but it results in the fall in precision. Another measure, $f - measure$ (harmonic mean of P and R), is adopted to counter the tradeoff between *precision* and *recall* and soften the threshold selection effect. The $f - measure$ is obtained by equation5, where TP stands for true-positive, which is the number of correctly identified regions, FP stands for false-positive, which is the number of incorrectly identified regions, and FN stands for false-negative, which is the number of regions that are not identified. A predicted region is considered correctly identified if its IOU for the actual region is greater than a predefined threshold (generally 0.5).

TABLE 2. Text Detection performance comparison on the ICDAR2015 dataset. The blue and red colors show the best and second-best performance in the table.

Method	Performance			
	Precision	Recall	F-measure	FPS
CTPN [52]	74.0	52.0	61.0	-
DeepReg [43]	82.0	80.0	81.0	-
SegLink [53]	73.1	76.8	75.0	-
EAST* [7]	83.6	73.5	78.2	13.2
SSTD [27]	80.0	73.0	77.0	-
Pixellink [41]	85.5	82.0	83.7	3.0
WordSup [54]	79.3	77.0	78.2	-
TextCorner* [55]	89.5	79.7	84.3	1
TextBoxes++* [8]	87.8	78.5	82.9	2.3
PSENet-1s-Ext [40]	84.0	78.0	80.9	3.9
RRD* [39]	88.0	80	83.8	-
MSR [56]	86.6	78.4	82.3	4.3
LOMO* [57]	87.8	87.6	87.7	-
TextSnake [58]	84.9	80.4	82.6	1.1
SPCNet [59]	88.7	85.8	87.2	-
CRAFT [14]	89.8	84.3	86.9	8.6
SAE(L1760) [60]	88.3	85.0	86.6	-
DB [6]				
ResNet18(H736)	86.8	78.4	82.3	48
ResNet50(H736)	88.2	82.7	85.4	26
ResNet50(H1152)	91.8	83.2	87.3	12
DBNet++ [25]				
ResNet18(H736)	90.1	77.2	83.1	44
ResNet50(H1152)	90.9	83.9	87.3	10
Proposed GMIF (H736)	92.1	84.4	88.0	21

$$\begin{aligned} precision \quad P &= \frac{TP}{TP+FP} \\ recall \quad R &= \frac{TP}{TP+FN} \\ f - measure \quad F &= \frac{2 \times P \times R}{P+R} \end{aligned} \tag{5}$$

C. TRAINING PHASE PROCEDURE

The synthetic data is generated with text size in the range of 8 *to* 256 pixel text height (4 to 128 RT). Due to the large text size, large background images are required. The text is fused with the background images by the SynthText [50]. All real and synthetic training images are augmented in three steps, 1) 3D rotation (assuming it is in XY plane), 2) Projection (in XY plane), and 3) Scaling. Besides, this training of GMIF is done with a batch size of one. The GMIF is trained in two phases 1) Mixed dataset and 2) Targeting dataset. The first phase uses the synthetic data and a training set of different datasets. This phase is trained over 600K iterations. The second phase utilized the training set of the target dataset and trained with 200K iterations. Adam [51] is adopted to optimize the GMIF

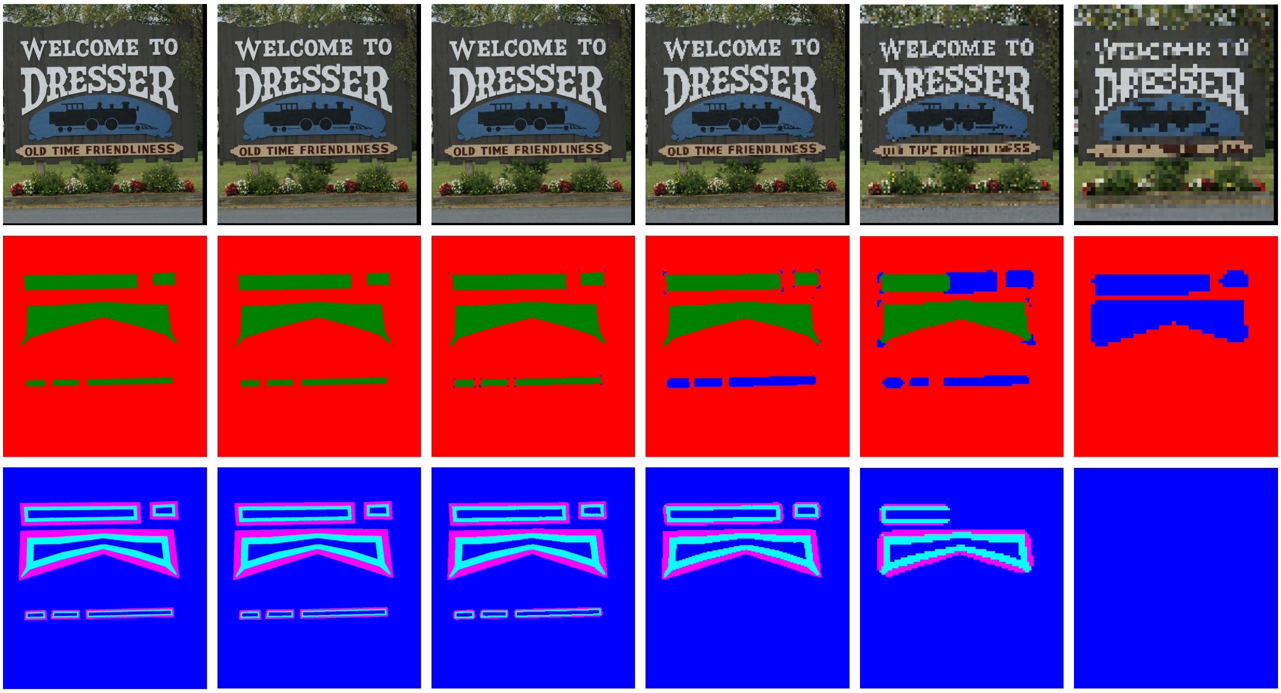


FIGURE 7. Label Generation: First-row is input image at different resolution, second-row is their corresponding global segmentation map (GSM^d), third-row is their global threshold map (GTM^d). The color encoding of different regions is as red is NONTEXT, green is TEXT, blue is DONTCARE, magenta is Threshold = 0.1, and cyan is Threshold = 0.9.

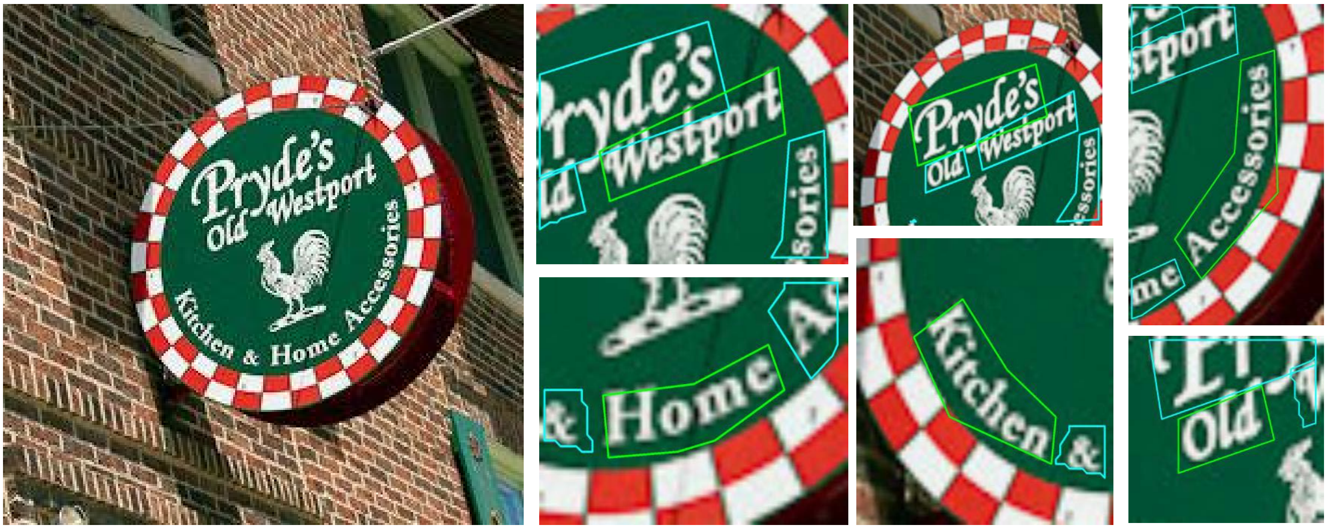


FIGURE 8. Sample images from the resized Total-Text dataset. Here green bounding region is TARGET TEXT instance and the cyan bounding regions are DONTCARE instances.

model. The hyper-parameters for Adam are $\alpha = 0.001,\ \beta_1 = 0.9,$ *and* $\beta_2 = 0.999$.

D. INFERENCE PHASE PROCEDURE

The original test image and its down-scaled versions are processed through the backbone network, and the resultant LFt feature-map is stored in a list. Then, this LTFt feature-map is passed to GTFGB for the global text feature GTFt generation. The last (the original image) GTFt is further processed with segmentation block and yields the final global probability map GPM. We use threshold 0.6 to binarize this GPM into the final segmentation map. This segmentation map extracts different connected components as the final text instances.

V. RESULTS

In this section, we are presenting the different experimental results obtained. This includes the text-properties-based comparison results, the ablation study results, and the study's results regarding the effect of the text size on the existing state-of-the-art methods. Here H# means that the height of

TABLE 3. Text detection performance comparison of the proposed GMIF model with existing works on MSRA-TD500 dataset. The blue and red colors show the best and second-best performance in the table.

Method	Performance			
	Precision	Recall	F-measure	FPS
DeepReg [43]	77.0	70.0	74.0	-
CRAFT [14]	88.2	78.2	82.9	8.6
SAE [60]	84.2	81.7	82.9	-
SegLink [53]	86	70	77	8.9
MSR [56]	87.4	76.7	81.7	-
ATRR [61]	85.2	82.1	83.6	-
TextSnake [58]	83.2	73.9	78.3	1.1
TextCorner [55]	87.6	76.2	81.5	5.7
PixleLink [41]	83	73.2	77.8	-
DB [6]				
ResNet18(H512)	85.7	73.2	79.0	82
ResNet18(H736)	90.4	76.3	82.8	62
ResNet50(H736)	91.5	79.2	84.9	32
DBNet++ [25]				
ResNet18(H512)	89.7	76.5	82.6	80
ResNet18(H736)	87.9	82.5	85.1	55
ResNet50(H736)	91.5	83.3	87.2	29
Proposed GMIF H736	90.8	80.2	85.2	26

the test images is resized as # pixels keeping the aspect ratio constant. For instance, H736 means that the test image is resized by making the height of the input image 736 pixels long.

A. PROPERTIES BASED COMPARISON

1) MULTI-ORIENTED TEXT DETECTION

ICDAR 2015 dataset is used to evaluate the effectiveness of the GMIF model in detecting multi-oriented text instances. The results obtained by GMIF with other state-of-the-art methods are presented in Table 2. Again, the GMIF obtained the highest f-measure as 88.0% (for H736 test images) and surpassed the DB [6] and DBNet++ [25] (with backbone ResNet50). The GMIF also maintains the computation efficiency with 21 FPS, which is twice the second-best performer (DBNet++ 87.3%). Some examples of results obtained by GMIF are depicted in Figure 9.

2) MULTI-LANGUAGE TEXT DETECTION

The MSRA-TD500 dataset is used for this purpose as it has text instances from the Chinese and English languages. The results obtained by GMIF with other state-of-the-art methods are presented in Table 3. The GMIF (85.2% f-measure and 80.2% recall) outperform the DB [6] (84.9% f-measure, 79.2 recall), and lagging DBNet++ [25] (87.2% f-measure, 83.3% recall) only.

3) ARBITRARY SHAPE TEXT DETECTION

The Total-Text dataset consists of text with arbitrary shapes, including horizontal, multi-oriented, and curved text, in most images. Therefore, we are incorporating this dataset to evaluate the effectiveness of GMIF in detecting arbitrarily shaped text instances. The results obtained by GMIF with other state-of-the-art methods are presented in Table 4. The GMIF (H800: 87.4%f-measure, 89.1% precision, and 85.8% recall) outperform the DB [6] and DBNet++ [25] in term of all performance measure listed. Some examples of obtained results by GMIF are shown in Figure 9.

TABLE 4. Text Detection performance comparison on the Total-Text dataset. The blue and red colors show the best and second-best performance in the table.

Method	Performance			
	Precision	Recall	F-measure	FPS
PSENet-1s-Ext [40]	84.0	78.0	80.9	3.9
SPCNet [59]	83.0	82.8	82.9	-
CRAFT [14]	87.6	79.9	83.6	-
MSR [56]	83.8	74.8	79.0	-
TextSnake [58]	82.7	74.5	78.4	-
LOMO* [57]	87.6	79.3	83.3	-
ATRR [61]	80.9	76.2	78.5	-
DB [6]				
ResNet18(H800)	88.3	77.9	82.8	50
ResNet50(H800)	87.1	82.5	84.7	32
DBNet++ [25]				
ResNet18(H800)	87.4	79.6	83.3	48
ResNet50(H800)	88.9	83.2	86.0	28
Proposed GMIF H800	89.1	85.8	87.4	26

B. COMPARISON WITH DIFFERENTIAL BINARIZATION

1) GMIF VS DB/DBNet++

The DB [6] and DBNet++ [25] uses a learnable threshold map to separate text instances. Their approach effectively separates text instances, but small text instances are not captured with the same efficiency. However, DB/DBNet++ utilizes the feature fusion of the output from a different level of their backbone network (ResNet). The feature fusion approach of DB is feature pyramidal addition, whereas the DBNet++ used an adaptive scale fusion approach. The leading cause of losing small text is the low resolution of the feature map after the first convolution (followed by a max pool operation). This feature map has four times lower resolution than the input image, which puts much stress on the first convolution (size 7 × 7) to maintain the information.

The GMIF outperforms the text detection performance of DB on all benchmarking datasets under consideration. GMIF does not use pooling operations that lose spatial information. The spatial information is helpful in segmentation. Besides this, GMIF also utilizes a large number of convolution kernels (256) at the first convolution, which capacitates it to acquire more information regarding small text instances. GMIF uses partially densely connected blocks to acquire more robust and detailed features. The total number of layers used in the backbone of GMIF is comparatively small than the backbone used in DB.

C. TEXT-SIZE INVARIANT PERFORMANCE

The Low precision of DB and DBNet++ shows that they are generating more predictions than the valid and don't-care text instances. The larger number of predictions arises due to the false-positive predictions and splitting a valid text

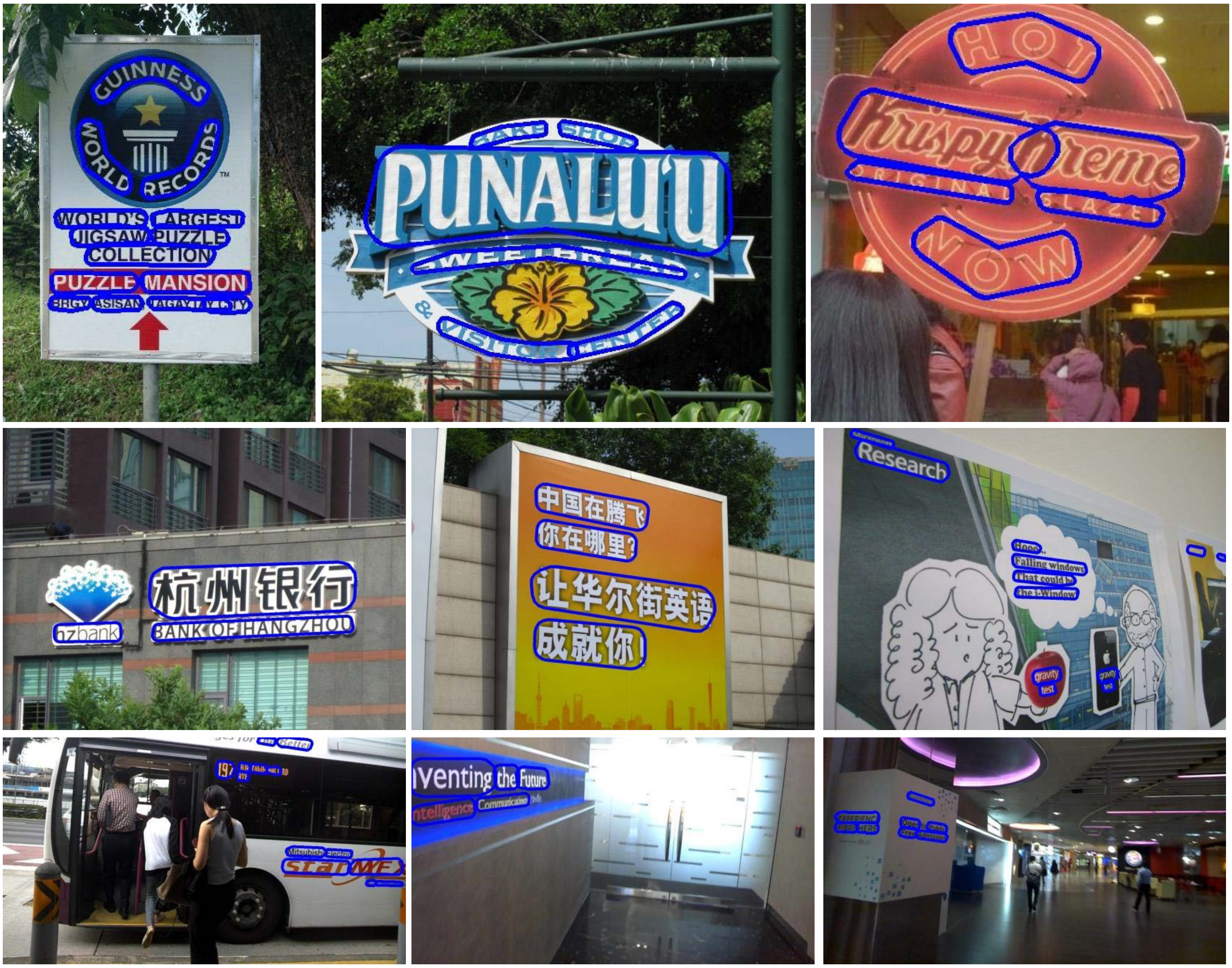


FIGURE 9. Qualitative results form test samples of datasets under considerations. The first row shows the results from the total-text dataset, second row shows the results of MSRA-TD500, and the last row shows the qualitative results for the ICDAR 2015 datset.

instance into more predictions. The Recall of the DB is high at $RT_{avg} = 40$ and decreases as going further. This also shows that the effective receptive field of their trained model is around 160 × 160 pixels (For a segmentation task, the network needs to cover the text instance from the boundary of the text, RT is the radius). The Recall of the DBNet++ is higher than the DB for a range of RT_{avg} which validates the effectiveness of the ASF approach for multi-scale feature fusion over the FPN of DB.

D. ABLATION STUDY

We conducted an ablation study on the Total-Text dataset to show the efficacy of the different components of the GMIF model: the backbone, the MPB, and the MPB with only square kernels, the GFGB. The detailed experimental results are shown in Table 6.

For this ablation study, the proposed GMIF model is compared with DB [6] and DBNet++ [25]. The DBNet++ system has been considered as the baseline for this study. The DBNet++ system can be considered a two-component system. 1) The backbone network of DBNet++ (we are undertaking the resnet-50 as DBNet++ backbone). 2) The adaptive scale fusion approach.

We performed a few modifications to this baseline model to convert it to the proposed GMIF Model. These modifications are as follows:

- *Resnet50 + GTFGB*: Here, we are using the Resnet50 as the backbone of the proposed GMIF model.
- $3 \times 3MPB + GTFGB$: Here, we are using the proposed backbone network with only square kernels in the MPB.
- *Backbone with MPB + GTFGB*: Here, we are using the proposed backbone network with MPB.

VI. DISCUSSION AND ANALYSIS

The proposed GMIF model is capable of handling the issues that were discussed in section I-A. Furthermore, the model has been evaluated and compared with the state-of-the-art in STD. The tables 2, 3 and 4 compares our model with existing work for the performance based on precision, recall, F-measure and speed.

TABLE 5. Text Detection performance comparison on the RT_{avg} specific total text dataset. The blue and red color shows the best and second-best performance in the table.

Method	Performance RT_{avg}=20 pixels			Performance RT_{avg}=30 pixels			Performance RT_{avg}=40 pixels		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
DB [6] ResNet50	81.467	85.695	83.528	80.471	86.282	83.275	78.418	85.424	81.771
DBNet++ [25] ResNet50	82.666	84.792	83.716	81.698	86.823	84.183	80.819	87.274	83.923
Proposed GMIF	89.106	85.790	87.417	88.956	85.645	87.269	88.855	85.548	87.171
Method	Performance RT_{avg}=60 pixels			Performance RT_{avg}=80 pixels			Performance RT_{avg}=100 pixels		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
DB [6] ResNet50	76.866	83.664	80.121	74.494	81.318	77.756	70.980	76.490	73.632
DBNet++ [25] ResNet50	79.833	86.462	83.016	76.678	84.567	80.429	73.005	80.912	76.755
Proposed GMIF	88.554	85.548	87.171	88.755	85.452	87.072	89.106	85.790	87.417

TABLE 6. Ablation Study with Total-Text Dataset. blue and red colors show the best and second-best performance in the table.

Method	Performance			
	Precision	Recall	F-measure	FPS
baseline				
ResNet50 DB	87.1	82.5	84.7	32
ResNet50 DBNet++	88.9	83.2	86.0	28
GMIF				
Resnet50 + GTFGB	89.7	84.4	87.0	17
3x3MPB + GTFGB	86.3	83.9	85.1	34
Proposed GMIF	89.1	85.8	87.4	26

A. HANDLING CONCERNS MENTIONED IN SECTION I

1) RECEPTIVE FIELD OF THE NEURAL NETWORK

The DB [6] and DBNet++ [25] reported their best performance with the resnet50 backbone. The resnet50 has four convolution blocks with the stacking of 3,4,6 and 3 convolution layers. In contrast, the proposed backbone network uses only three blocks with 2,3 and 4 layers. The proposed backbone concatenates the side features at each convolution and uses them as the final feature instead of using only the last features from a convolution block. This facilitates the proposed GMIF model to acquire detailed information from its receptive field.

2) TextSize

The proposed model extracts the required features from an appropriate down-scaled input image, and then the final masking is done at a higher resolution.

3) OVERLAPPING TEXT REGIONS

The proposed model primarily classifies any text region as a scaled version of it, which is completely covered (only text height) by the network(refer to section III for detail). The model also uses feature fusion from different scaled input images. Consequently, the overlapping of text instances is minimized by acquiring their segmentation mask at their respective scale. GMIF maintains these low-resolution segmentation masks at a higher resolution.

4) REAL-TIME PERFORMANCE

The proposed model utilizes a comparatively shallow backbone network with a hybrid architecture of InceptionNet [62] and VGG [20]. The GMIF performs comparatively faster while maintaining the text detection performance.

B. COMPARISSION OF GMIF AND EXISTING WORK

1) BACKBONE NETWORK AND FEATURE FUSION

The current state-of-the-art methods [6], [7], [8] use a convolution neural network as a backbone network to generate essential features. This backbone covers the entire input image by its receptive field [15]. Therefore these methods need to downscale the input image if a text with a size bigger than the network's receptive field appears. Besides this, a large receptive field tends to lose focus on the smaller text; these methods upscale the input image to overcome this issue [6]. The GMIF is also a convolution neural network, but it uses only three pooling operations (the pooling is done using stride) at its backbone network. GMIF has a small receptive field, uses the downscale input image for more extensive coverage, and does not require upscaling to detect a text. The overall architecture of GMIF is depicted in Figure 3. The RefineNet [63] model also use the downscale images and then fuses their local feature to get robust features for the semantic segmentation task. Their approach differs from the GMIF on the backbone network, the feature fusion, and the target task. The RefineNet fused all the features and created a single prediction map, whereas the GMIF fuses the feature according to the target text size. Besides this GMIF target a different segmentation map at different downscale image(refer to III).

2) TEXT AREA SHRINKING

The text instance region is shrunk to reduce the overlapping between nearby text instances. The existing methods [6] use

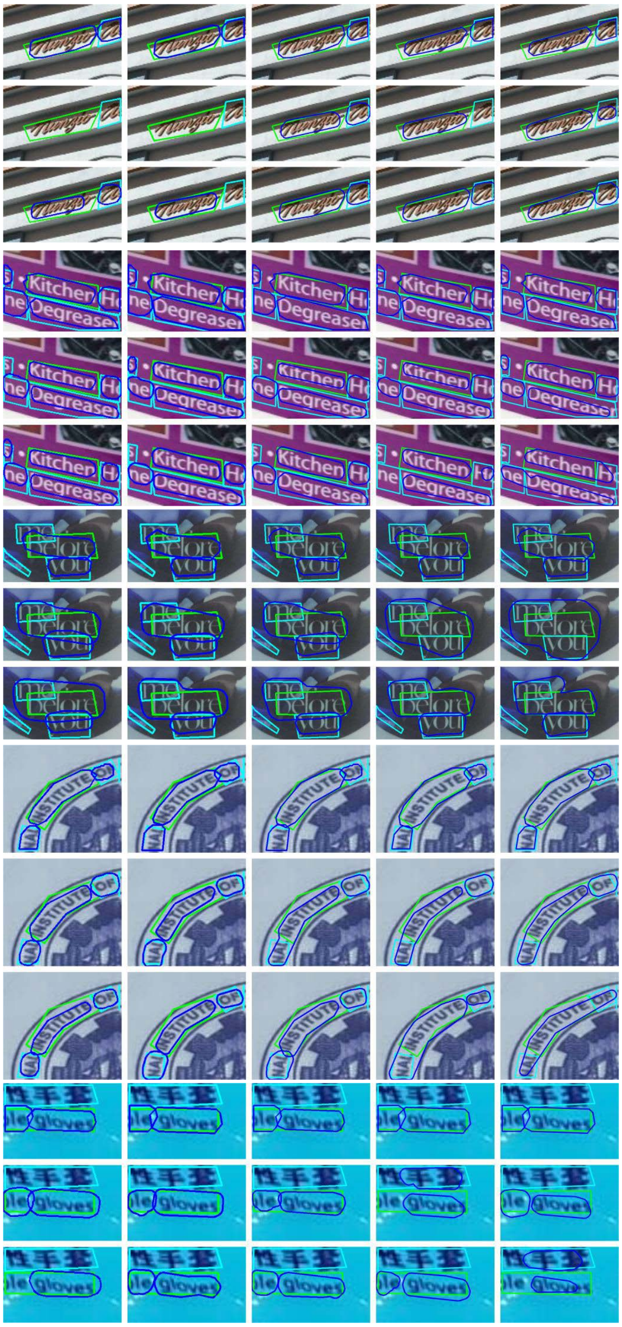


FIGURE 10. **Example images showing the results obtained by proposed GMIF, DB, and DBNet++ over the resized total-text dataset. For every test sample the first row shows the results obtained through GMIF, second row shows the DB results, and the third row shows the results from DBNet++. Here green is TARGET TEXT, cyan are DONTCARE, and the blue bounding regions are PREDICTED text instances.**

the Vatti clipping algorithm [64] for shrinking the text region area. The shrinking is done according to the text region size to separate the different words better. This shrinking is a constant for a given text region. The label generation for GMIF is different from these existing methods. The GMIF shrinks the text regions according to the text height at that location. This shrinking is not fixed for entire text regions but is adaptive to the pixel location. A visual sample of this shrinking is shown in Figure 7. The detail of the label generation is provided in section III-E. Here, the character mask is already downscaled four times due to the pooling used in the backbone network; therefore, one pixel of the character mask is four pixels in the backbone network's input data.

3) A PIXEL/CELL CATEGORIZATION AND ITS WEIGHTS

The Differential Binarization (DB) [6] method uses the supervised learning for the text probability map and the text-threshold map to enhance the learning of boundary pixel segmentation. The method in [35] proposed text border elements for better separation of the text instances. The UNet [65] suggests a loss weighting scheme that assigns a higher weight to the boundary pixels for the object segmentation. The GMIF model categorizes different pixels/cell regions into three categories 1) TEXT, 2) NONTEXT, and 3) DONTCARE. Besides this, a cell location has a threshold map score depending upon the text height and distance from the text border(refer to sectionIII-E for details).

4) EXISTING METHODS UTILIZING TEXT SIZE

The SRPN+TextDetector [66] (will be referred to as SRPN+TD) also utilizes the text size as the proposed GMIF for the performance improvement. The SRPN+TD is a two-phase method. It first estimates the text proposal and the size of their texts. The second phase generates the bounding box for text instances on a scaled and cropped input version. This approach follows the coarse to fine methodology, but the performance of the first phase is the bottleneck. It cannot also detect the curved text instances effectively. The GMIF is a single-phase model and target all text shapes and orientation.

VII. CONCLUSION AND FUTURE WORK

This paper has offered GMIF, a text detection method for text instances of arbitrary shapes, orientations, and scales. The model's architecture consists of a backbone network that learns the text features for a small range of text sizes, improving its text detection capacity at some down-scaled version of the input image. Concurrently, the architecture's GTFGB propagates the background information learned from the low-resolution input images to the higher resolution segmentation. This helps the GMIF to suppress the background and extract the text correctly. The experimental analysis of the results obtained by GMIF shows its effectiveness for detecting text instances of any scale. GMIF achieves the state-of-the-art performance in detecting text with the arbitrarily shaped images of the Total-text dataset and ICDAR 2015 dataset. Besides this, GMIF also achieves second-best results in F-measure performance over MSRA-TD500 (multi-language script) dataset. The GMIF also sustains real-time performance, and its computation speed can be adjusted by down-scaling the input images accordingly. The proposed GMIF model provides a consistent STD performance over an extensive range of text instance sizes. The GTFGB propagates the text features through different scales of the text instances. Similarly, the GTFGB should also propagate the features regarding the text-character classes, but its behavior needs to be explored in the text recognition task.

REFERENCES

[1] C. Yi and Y. Tian, ''Scene text recognition in mobile applications by character descriptor and structure configuration,'' *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2972–2982, Jul. 2014.

[2] B. Xiong and K. Grauman, ''Text detection in stores using a repetition prior,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[3] X. Rong, C. Yi, and Y. Tian, ''Recognizing text-based traffic guide panels with cascaded localization network,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 109–121.

[4] Y. Zhu, M. Liao, W. Liu, and M. Yang, ''Cascaded segmentation-detection networks for text-based traffic sign detection,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 209–219, Jan. 2018.

[5] C. Kang, G. Kim, and S. Yoo, ''Detection and recognition of text embedded in online images via neural context models,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 4103–4110.

[6] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, ''Real-time scene text detection with differentiable binarization,'' in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11474–11481.

[7] 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. 5551–5560.

[8] 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.

[9] X. Rong, C. Yi, and Y. Tian, ''Unambiguous scene text segmentation with referring expression comprehension,'' *IEEE Trans. Image Process.*, vol. 29, pp. 591–601, 2020.

[10] Z. Wang, J. Jiang, Y. Wu, M. Ye, X. Bai, and S. Satoh, ''Learning sparse and identity-preserved hidden attributes for person re-identification,'' *IEEE Trans. Image Process.*, vol. 29, pp. 2013–2025, 2020.

[11] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, ''Person reidentification via discrepancy matrix and matrix metric,'' *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 3006–3020, Oct. 2018.

[12] T. Qiao, J. Zhang, D. Xu, and D. Tao, ''MirrorGAN: Learning text-to-image generation by redescription,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1505–1514.

[13] M. Gao, Y. Du, Y. Yang, and J. Zhang, ''Adaptive anchor box mechanism to improve the accuracy in the object detection system,'' *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27383–27402, Oct. 2019.

[14] 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. 9365–9374.

[15] W. Luo, Y. Li, R. Urtasun, and R. Zemel, ''Understanding the effective receptive field in deep convolutional neural networks,'' in *Proc. Conf. Neural Inf. Process. Syst* Red Hook, NY, USA: Curran Associates, 2016, pp. 4905–4913.

[16] S. C. Park, M. K. Park, and M. G. Kang, ''Super-resolution image reconstruction: A technical overview,'' *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.

[17] P. Keserwani, T. Ali, and P. P. Roy, ''TRPN: A text region proposal network in the wild under the constraint of low memory GPU,'' in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 286–291.

[18] E. Rosten and T. Drummond, ''Machine learning for high-speed corner detection,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2006, pp. 430–443.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[20] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. Int. Conf. Learn. Represent.*, 2015.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ''Going deeper with convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[22] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[24] S. Hochreiter, ''The vanishing gradient problem during learning recurrent neural nets and problem solutions,'' *Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[25] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, ''Real-time scene text detection with differentiable binarization and adaptive scale fusion,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Mar. 3, 2022, doi: 10.1109/TPAMI.2022.3155612.

[26] 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. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 67–83.

[27] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, ''Single shot text detector with regional attention,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3047–3055.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ''Rich feature hierarchies for accurate object detection and semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[29] J. Uijlings, K. van de Sande, and T. Gevers, ''Selective search for object recognition,'' *Int. J. Comput. Vis.*, vol. 104, pp. 154–171, Oct. 2013.

[30] R. Girshick, ''Fast R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[31] 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.

[32] 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.

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot multibox detector,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[34] Y. Yuan and Y. Zhang, ''OLCN: An optimized low coupling network for small objects detection,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[35] C. Xue, S. Lu, and F. Zhan, ''Accurate scene text detection through border semantics awareness and bootstrapping,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 355–372.

[36] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, ''Multi-oriented text detection with fully convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4159–4167.

[37] Y. Li, W. Jia, C. Shen, and A. van den Hengel, ''Characterness: An indicator of text in the wild,'' *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1666–1677, Apr. 2014.

[38] 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. 1962–1969.

[39] 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.

[40] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, ''Shape robust text detection with progressive scale expansion network,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9336–9345.

[41] D. Deng, H. Liu, X. Li, and D. Cai, ''PixelLink: Detecting scene text via instance segmentation,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 6773–6780.

[42] 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 Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 1–7.

[43] 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.

[44] L. Xie, Y. Liu, L. Jin, and Z. Xie, ''DeRPN: Taking a further step toward more general object detection,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9046–9053.

[45] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, ''Empirical evaluation of gated recurrent neural networks on sequence modeling,'' 2014, *arXiv:1412.3555*.

[46] 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, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[47] 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.

[48] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 935–942.

[49] 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, Nov. 2014.

[50] 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.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[52] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 56–72.

[53] 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. 2550–2558.

[54] 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. 4940–4949.

[55] 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.

[56] C. Xue, S. Lu, and W. Zhang, "MSR: Multi-scale shape regression for scene text detection," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 989–995.

[57] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10552–10561.

[58] 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. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 20–36.

[59] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proc. Conf. Artif. Intell.*, vol. 33, Aug. 2019, pp. 9038–9045.

[60] 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. 4234–4243.

[61] 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. 6449–6458.

[62] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 1–7.

[63] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[64] B. R. Vatti, "A generic solution to polygon clipping," *Commun. ACM*, vol. 35, no. 7, pp. 56–63, 1992.

[65] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[66] W. He, X.-Y. Zhang, F. Yin, Z. Luo, J.-M. Ogier, and C.-L. Liu, "Real-time multi-scale scene text detection with scale-based region proposal network," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107026.

TOFIK ALI received the B.Tech. and M.Tech. degrees in computer science from the Aligarh Muslim University, Aligarh, India, in 2010 and 2012, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India. His current research interests include computer vision and machine learning.

MOHAMMAD FARIDUL HAQUE SIDDIQUI was born in Aligarh, Uttar Pradesh, India, in 1987. He received the B.Tech. degree in computer engineering and the M.Tech. degree in software engineering from the Aligarh Muslim University, Aligarh, in 2005 and 2012, respectively, and the Ph.D. degree in computer science and engineering from The University of Toledo, Toledo, OH, USA, in 2019.

From 2014 to 2019, he was a Research Assistant at the Paul A. Hotmer Cybersecurity and Teaming Research (CSTAR) Laboratory, The University of Toledo. From 2020 to 2021, he worked at the University of North Carolina Greensboro, as a Lecturer of computer science. Since 2021, he has been with the College of Engineering, West Texas A&M University, as an Assistant Professor of computer science. He is the author of several IEEE journal and conference publications. He has also served as a reviewer for several high-impact journals and conferences. His research interests include multimodal human–computer interaction, computer vision with deep learning, affective computing, and augmented reality.

SANA SHAHAB is currently an Assistant Professor with the College of Business Administration, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. She has authored or coauthored more than 20 research papers in international journals and conferences. Her current research interests include interdisciplinary applications of statistics, computer and management science to serve the broad areas of problem-solving, and decision-making in the organization.

PARTHA PRATIM ROY (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Universitat Autònoma de Barcelona, Spain. He was at the Synchromedia Laboratory, Canada, in 2013, and the RFAI Laboratory, France, from 2011 to 2012, as a Postdoctoral Research Fellow. He worked at the TATA Consultancy Services, from 2003 to 2005, and the Advanced Technology Group, Samsung Research Institute Noida, India, from 2013 to 2014. He is currently an Associate Professor with the Department of Computer Science and Engineering, IIT Roorkee, India. He has published more than 225 papers in international journals and conferences. His research interests include pattern recognition, human–computer interaction, bio-signal analysis, and multilingual text recognition. He is an Associate Editor of *IET Image Processing*, *IET Biometrics*, *IEICE Transactions on Information and Systems*, and *Springer Nature Computer Science*. He is a Regional Editor of the *Journal of Multimedia Information System*, and the Guest Editor of the *International Journal of Distributed Sensor Networks*.