DAFNet: Divide and Filter Network for Efficient Binarization and Script Recognition of Texts in Digital Images

Indra Narayan Dutta[1], Neelotpal Chakraborty[1,*], Ayatullah Faruk Mollah[2], Subhadip Basu[1] and Ram Sarkar[1]

[1]*Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India*
[2]*Department of Computer Science and Engineering, Aliah University, Kolkata-700160, India*

*Corresponding author: Neelotpal Chakraborty {chakraborty.neelotpal@gmail.com}

Abstract. Analysis of multi-script texts digital images is a popular research area involving information retrieval from the texts present in a digital image that may be either scene or born-digital. Several methods have been reported for accomplishing this task along with many end-to-end solutions. However, binarization of texts in digital image has a significant scope for improvement and relatively few methods are found in the literature. Besides, for script recognition from such text images, most techniques require classifier training, thereby becoming subject to challenges like dealing with inter-script similarities, occluded texts and trivial parameter selection of the classifier. To address the above challenges, a novel approach called "Divide And Filter" Network or DAFNet is proposed for binarization and script recognition of digital image texts. The aims of DAFNet are to "Divide" the intensity range of every color channel of an input image for generating binary outputs called Binned Images and then apply a fully convolutional 18-layer neural network called ResNet-18 to "Filter" out spurious Binned Images for identifying the final binarized text image. Binarization is done at both image and component levels. Component level binarized texts are further processed by the proposed framework for script recognition. The proposed system is evaluated on standard datasets like ICDAR 2003, 2011, 2019, SVT, Total-text and KAIST delivering comparable results with some state-of-the-art methods.

Keywords. Multi-script; Text in Digital Image; Binarization; Binning; Script Recognition; Deep Learning, Convolutional Neural Network; Divide and Filter Network

1. Introduction

Texts in Digital Image (TDI) refers to the textual information present in various types of digital images [1] like (1) Natural Scene Images (NSI) [1, 2] where texts are usually seen in road signs, billboards, hoardings, graffiti, shop signs, etc., and (2) Born Digital Images (BDI) [1] where texts are synthetically typed on some images using a digital device and can be shared as some meme, post or message across the Internet. Recently, there has been exponential rise in interest among researchers to retrieve and analyze TDI as these are significantly information-rich since it serves several applications related to visual assistance, image to text conversions, traffic and vehicle detection, text translations, sentiment analysis, identifying internet trends, gather public opinions, etc. [1-3]

Retrieval and analysis of such textual information from DIs mainly focus on text detection and recognition [3]. TDI detection requires background elimination [4] and spurious component filtering [5] which may include approaches like region proposal generation [5] or image binarization. Image binarization [6] is a faster approach than its former counterpart to make the TDI ready for Optical Character Recognition (OCR) engine [3] as the text components are exclusively segmented from the complex background. Given the widespread presence of multi-script TDI, an intermediate task i.e., script recognition [7] is now a necessity. In a culturally diverse world, the presence of multi-script TDI is quite common thus necessitating methods to be used for script recognition.

While this domain ensures huge scope for applications that attempt to satiate the growing demands of the general public, binarization and script recognition of TDI require addressing a number of challenges arising due to the factors such as presence of blur, noise, improper lighting conditions, uneven intensity distribution, arbitrary nature of text properties like size, font style and orientation, inter-script similarities, occluded texts. To deal with these, researchers look for the approaches that may be rule-based, machine learning or deep learning-based [1-3]. Several methods and approaches have been proposed for addressing this research need. The state-of-the-art has significantly improved in terms of robustness and efficiency especially with the application of various deep learning models since the last decade. However, despite this performance enhancement in TDI detection, relatively few efficient methods exist for TDI binarization. Most of these binarization methods are rule-based, thereby constraining the whole mechanism to parametric restrictions. As a result, certain objects other than texts may get binarized, thus necessitating the use of machine learning or deep learning models for text-specific binarization in DIs.

Another major challenge that is recently being addressed is text in multiple orientations [8]. Mostly, researchers work upon TDI assuming rectangular bounding boxes [9] for detected text. Curved, angled and multi-oriented text regions pose challenges to networks designed for just detecting rectangular bounding boxes which often need additional processing with extra convolution layers [10] that can find tight bounds within the detected bounding boxes.

To address these challenges, a novel deep learning based model is presented here which can efficiently perform binarization and script recognition of TDI. A new model called Divide and Filter Network (DAFNet) is designed which directly yields the tight regions as output from a single network without any additional parallel networks. Besides, the network used here is lightweight and can easily be used for multi-script and multi-oriented TDI retrieval.

The major contributions in this paper may be summarized as follows:

1. A relatively new deep learning based framework called DAFNet is designed where a combination of Bi-Level Overlapped Binning (BOB) [11] and ResNet-18 [10] is used for effective binarization as well as script recognition of TDI.
2. DAFNet proves to be a fast and efficient deep learning based architecture that uses the concept of Binned Images (BI) [11] and determining the optimal binarized text output along with script recognition using few convolution layers (only 18). This system achieves satisfactory precision and recall despite being trained on less number of samples.
3. The proposed framework proves to be versatile enough for binarizing texts at image level (anywhere across an image) as well as component level (word image) along with script

recognition of the binarized text. Overall, DAFNet provides a single shot mechanism for making multi-script TDI OCR-ready.

4. The proposed system is evaluated on standard datasets like, SVT [1-3], KAIST [1, 2], ICDAR 2003 [1-3], 2011 [1-3], 2019 [12] and Total-Text [8], performing satisfactorily along with achieving results comparable to that of some recent state-of-the-art methods.

2. Related Work

As earlier mentioned, text binarization is usually required for easier OCR processing. With more focus on multi-script texts in digital images, the region having text components need to be detected at word level and their scripts identified since most OCR engines are script specific. Then binarization of these text regions is preferred for effective recognition. As a result, most text binarization schemes focus only on images with text as the only foreground object. Furthermore, binarizing text components scattered across an image having multiple object foregrounds, is relatively challenging due to the complex intensity distribution, noise and blur. Hence, few contributions have been made for image level binarization of texts. Some of the text binarization methods are discussed below.

A classical yet popular method for image binarization was introduced by Otsu in his work in [6] where a global optimal threshold is determined using zeroth and first order cumulative moments of gray level histogram of an image. In the work [13], Kittler et al. applied gray level histogram information independent image statistics in view of the relation between luminance values and a specific value falling amidst the luminance values of the object and background.

A local threshold is determined in the work [14] by Niblack which is calculated using a rectangular window across the image using mean and variance of the gray values falling under this window. However, a drawback associated with noisy points would often appear in textless regions. This problem is addressed by the method proposed in [15] which calculates the intensity mean and variance for both the window as well as the whole image. A probabilistic Markov Random Field (MRF) model is described in [16] for binarization so as to normalize the uneven contrast of any input image. Laplacian of image intensity is implemented by Howe [17] and Milyaev et al. [18] to determine the energy function of a sliding window that passes across the input image for automatic parameter selection.

The above methods address most of the problems associated with document image binarization, but have limitations when dealing against complexities that come with different types of digital images. Problems like manual tuning of parameters, image complexities caused by uneven lighting, noise, blur, and similar foreground/background shades diminish the efficiency of these methods. An attempt is made by Kasar et al. in [19] to tackle such complexities by applying a Canny edge-based binarization for segmenting the text components. A bilateral regression-based method is developed by Feild & Learned-Miller in [20] where color clustering is initially applied to fit a regression model to produce multiple text proposals. The work [21] by Mishra et al. emphasizes on energy minimization by formulating an energy (cost) function using an iterative graph cut algorithm. This algorithm defines an inverse relation between the binarization quality and the energy value obtained from the color information in an image region falling under a sliding window. The method proposed in [22] applies game theory for text binarization by designing a two-player, non-zero-sum, non-cooperative game to extract local information of a pixel and classifying it as foreground or background using K-means algorithm. A probabilistic method is developed by Weinman et al. in [23] to crudely binarize a text region, identify baselines, and then simultaneously segment words and characters.

Adaptive binarization is performed by Ghoshal et al. in [24] where a gray-level variance map is utilized for text boundary detection in an image. Most of the above methods are based around English scene texts. Wang et. al [25] introduces a neural network based adaptive preprocessing algorithm for Chinese text which performs binarization, extraction of colors and segmentation. Qaroush et. al [26] introduces a font independent text segmentation algorithm for complex arabic scripts.

Recent binarization approaches explore the principle of component stability defined by Matas et al. in [27] where a concept called Maximally Stable Extremal Regions (MSER) is constructed that defines text components as stable foregrounds since the pixels representing them have near similar intensity values. However, its sensitivity against blur led Chen et al. to modify MSER technique by applying Canny edge-based mechanism in [28] along with Stroke Width Transform (SWT) [29] to filter out the spurious MSERs. SWT represents the idea of a text region having informative strokes of nearly uniform width throughout a Connected Component (CC). Similar combination of MSER and SWT has been employed by Li et al. in [30] to determine the characterness of the extracted candidate CCs. Ghosal et. al [31] combined Canny's edge information with Savola to obtain a binarization on which connected component region growing was applied to separate text from background. Deep learning based binarization is also becoming exceedingly popular for segmentation purposes. Deep learning-based methods often allows for better flexibility as demonstrated in Xu et al. [32] which introduces Prior Guided Text Segmentation Network (PGTSNet) for handing bi-lingual (English and Chinese) texts and complex text segmentation. Liao et. al [33] introduced a real time scene text detection based on differentiable binarization that did not rely on complex post processing algorithms. The above deep learning methods are supervised algorithms which require large datasets for training. Wang et. al [34] introduces a semi-supervised deep learning algorithm for pixel level segmentation.

Character grouping is an important issue for word or text line extraction from NSIs which is addressed by Yin et al. [35] by adaptively clustering character MSERs. Bai et al. [36] introduce strokelets which are a set of mid-level primitives for capturing hidden information of characters at different granularities. A multi-channel cum multi-resolution MSER approach is implemented by Tian et al. [37] after which spurious regions are morphologically filtered out and then text strings are generated using a graph model. Another stability-based technique is proposed by Dutta et al. [4] where gray levels are binned such that text components do not get broken and it proves to be more robust against blur. Adaptive stroke filters are developed by Paul et al. in the works [38] and [39] to retrieve the textual information in NSIs. In the work [40], Bhunia et al. select a color channel where a Hidden Markov Model (HMM) is applied for utilizing Pyramidal Histogram of Oriented Gradients (PHOG) features. These stability-based methods, however, are mostly prone to parameter tuning unsuitability for certain digital images where text regions are shadowed or inadequately lit because of night-time acquisition or poor light condition.

Script recognition of the localized texts is also another subdomain in this research field that has gained significant importance due to presence of texts written in multiple scripts. Usually it is performed with the help of a machine learning based classifier trained on standard features from multi-script text images or a deep learning model. The ICDAR competition [12] has greatly emphasized on this sub-problem in the recent times by featuring some best performing models. Most of these models are deep learning based. In an experimental analysis highlighted by Jajoo et al. in [7] some standard machine learning models like Naïve-Bayes, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) are trained on popular feature descriptors like

Histogram of Oriented Gradients (HOG) and Gray Level Co-occurrence Matrix (GLCM) to classify word-level images of scene texts into Bangla, Roman and Devanagari scripts where MLP achieves the highest accuracy. Islam et al. [41] used vertical scanning in combination with SVMs to achieve an accuracy of 99.16% on recognizing Bangla characters. A Convolutional Neural Network (CNN) based script recognition module serves as an integral part of the end-to-end scene text analysis in the work reported in [5] where text proposals are generated using a Generative Adversarial Network (GAN) based framework and fed to the script recognition mechanism. Khan et. al [42] utilized CNNs to perform component level script identification for Bengali, Latin and Devanagari. A lot of scene text segmentation methods rely on complex post-processing algorithms that often becomes time consuming. Although, deep learning-based systems are becoming very popular among researchers owing to their high performances, traditional machine learning models when trained with features like Daisy descriptor too give comparable performance as reported in [43] for script recognition from NSIs. Here, an end-to-end system is built where an adaptive K-means based intensity clustering is performed on input image to generate candidate proposals for identifying texts along with their corresponding script using a standard SVM classifier fed with Daisy features from the proposals. It is a common practice to train CNN based classification models with large dataset using advanced hardware resources. To cope with these requirements, an ensemble of CNN models is built in [44] where each CNN classifier is trained upon an image channel and their individual classification scores are combined using sum or product rule to give the final script classification result.

In this paper, the focus is on binarizing text components in digital images for effective text segmentation. Also, these binarized text components are combined at word-level to identify the scripts in which such texts are written. These objectives provide motivation to propose a filter network based on the hour-glass model [45] and is very similar to ResNet-18 architecture [46] with residual connections. The first half of the network decreases resolution by half after every block till the resolution is one-eighth of the original. It then uses transpose convolution to double the resolution till it reaches the original resolution. The final output of the network is a single channel image of just text segments. Most modern text proposal generation methods focus on bounding box generation followed by binarization or other advanced techniques to segment text regions. In this work, this proposal generation is altogether eliminated by directly introducing text specific binarization. This end-to-end system thus ensures that a single network directly segments the texts to be fed to an OCR system specific to a particular script. This is useful in OCR based applications that need lightweight networks for recognizing texts found in digital images because a single is enough to extract text regions in the form of CCs.

3. Proposed Method

CNNs are great at learning linear mappings from X → Y which is widely explored in this paper. Instead of directly feeding an input digital image to a CNN model, an image is processed to generate a collection of several binary images each of which considers a sub-range of the global pixel value range (0 – 255) for a color channel as a data pixel, determined by a specific mathematical model. Each binary image then consists of several CCs and some binary images also contain text or foreground CCs desired in the final binarized image.

DAFNet architecture comprises of two modules namely (1) BOB schema for generating multiple binary image solutions or BIs, and (2) a fully convolutional network called ResNet-18 for identifying the optimal solution among the BIs along with recognizing the script to which the text belongs. A flowchart of the proposed method is depicted in Fig. 1, and the working procedure of every module of DAFNet framework is discussed in the following subsections.

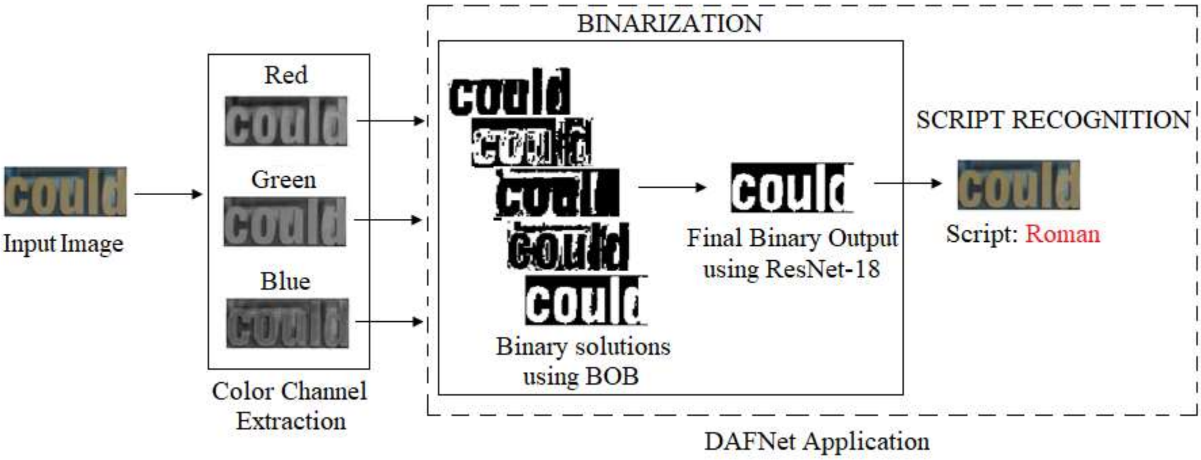

Fig. 1. Flowchart of the proposed methodology. DAFNet combines the working scheme of BOB and ResNet-18 to effectively binarize and identify script of a text in an input digital image.

3.1 Bi-Level Overlapped Binning

An input color image usually consists of 3 color channels: Red (R), Green (G) and Blue (B) which individually can be viewed as 2D grayscale map of values ranging from 0 to 255. Every such 3-channel input image can be represented as a collection of several binary images which are generated using BOB [11]. Each binary image maps to a particular range of pixel values within 0 to 255 through a mapping defined in BOB. This representation is called BI. The idea is that every BI captures some regions of the input channel for 3-channel image with one specific range of pixel values in the form of a CC. *Range* of pixel values for a CC is defined as the difference between the maximum and minimum pixel intensity values of the set of pixels in the original image that overlap with the pixels of the CC.

The goal of BOB is to capture homogenous text regions with a particular range of pixel values as a single CC by selecting a series of bin sizes S_i. For each size, several binary images are produced, with the goal of capturing all the regions (connected set of pixels) with a range of pixel values that is less than or equal to S_i and avoid capturing any background pixels in the process.

Algebraically, the goal can be represented as a mapping problem in the following manner:

Find the mapping, $I \rightarrow \{B^s_{0\ to\ a},\ B^s_{b\ to\ c},\ B^s_{d\ to\ e},\ B^s_{f\ to\ g} \dots \dots .\ B^s_{z\ to\ 255}\}$ such that,

$\forall\ R_k \in I$, there exists $CC_j\ \in\ B^s_i$ such that structural features of R_k are preserved in CC_j and if $I(r,c) \in R_k$ then $B^s_i\ (r,c) = 1$

where I is a single channel input image, R_k denotes the candidate regions in I and $(a, b, c, \dots, z)$ are integers representing a particular intensity value. $B^s_{x\ to\ y}$ are binary images of same dimensions as I. CC_j is the j^{th} CC in B^s_i. The pair (r, c) denotes the coordinates in the 2D space where r and c are number of rows and columns respectively of the 2D map of the image.

However, just dividing the global pixel value range into equal sized bins and mapping the pixel values to the bin does not ensure that a region R with a range of pixel values less than S_i will be retrieved as a CC in a binary image. If a homogenous region occurs in the range between $\frac{S_i}{2}$ and $\frac{3\times S_i}{2}$, despite the fact that the region has pixel value range equal to S_i, it is not retrieved as a single CC but instead gets distributed over two separate binary images i.e., binary images for ranges $\{0 \text{ to } S_i - 1\}$ and $\{S_i \text{ to } 2 \times S_i - 1\}$ with hardly any preservation of original structural features in most cases. The issue is resolved in the BOB algorithm where instead of stepwise division, it divides the global pixel intensity range into several overlapping bins.

First, it considers two levels of stepwise non-overlapped binning given size S. The bin images at each level are represented as follows:

Level 1: $I \rightarrow \{ B^S_{0\ to\ S-1},\ B^S_{S\ to\ 2\times S-1},\ B^S_{2\times S\ to\ 3\times S-1} \dots B^S_{M-S\ to\ M} \}$

Level 2: $I \rightarrow \{ B^S_{\frac{S}{2}\ to\ \frac{3}{2}\times S-1},\ B^S_{\frac{3}{2}\times S\ to\ \frac{5}{2}\times S-1}, \dots,\ B^S_{M-\frac{3\times S}{2}\ to\ M-\frac{S}{2}} \}$

where, $M = \lceil \frac{255}{S} \rceil \times S$

Here, Overlapped Bin (OB) images are generated from the two levels by combining the BIs B^S_{a-b} using OR operation alternatively from the two levels as follows:

$$OB^S_{0\ to\ \frac{3}{2}\times S-1} = B^S_{0\ to\ S-1}\ OR\ B^S_{\frac{S}{2}\ to\ \frac{3}{2}\times S-1}$$

$$OB^S_{\frac{S}{2}\ to\ 2\times S-1} = B^S_{\frac{S}{2}\ to\ \frac{3}{2}\times S-1}\ OR\ B^S_{S\ to\ 2\times S-1}$$

$$OB^S_{S\ to\ \frac{5}{2}\times S-1} = B^S_{S\ to\ 2\times S-1}\ OR\ B^S_{\frac{3}{2}\times S\ to\ \frac{5}{2}\times S-1} \text{ and so on...}$$

Fig. 2 shows the implementation of BOB on a single input image of size 56. Algorithm 1 provides the pseudo-code to generate OB images given a single image channel I and size S.

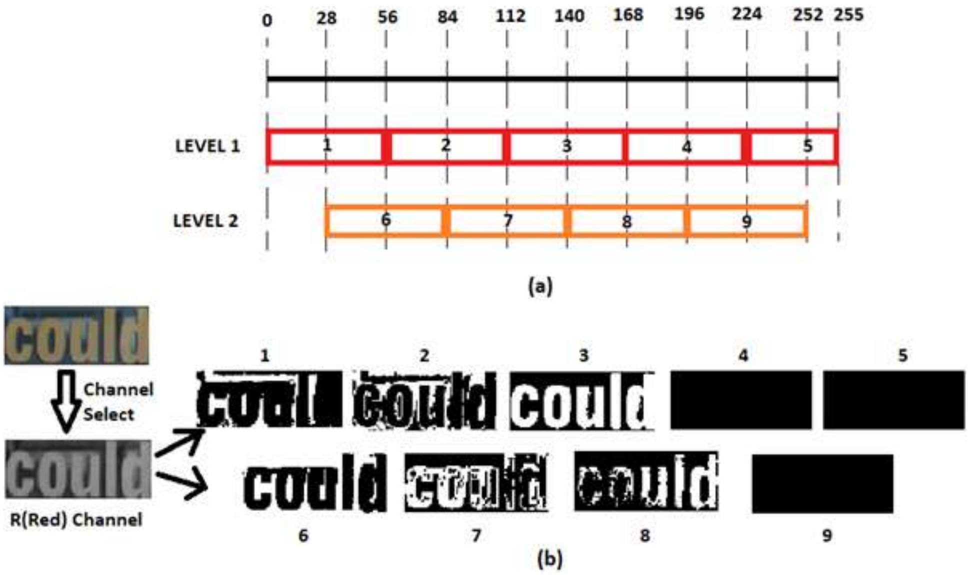

Fig. 2. An illustration of BOB schema applied on an input word image with bin size =56. (a) The intensity range is divided into multiple bins at two levels with each Level 2 bin (bin number 6 to 9) lying between the mid values of two consecutive Level 1 bins (bin number 1 to 5), (b) The candidate CCs generated from a particular channel (R in this case) are represented in binary images.

Algorithm 1 is utilized for generation of BIs from a single image channel and a single size. Selecting a single size can be problematic because the large diversity of images and that text region's pixel intensity values can have a variety of ranges. Too small sizes or too large sizes can distort the binary images and not provide accurate segmentation. This problem is depicted in Fig. 3 where too small bin sizes lead to only partial text components occurring and too large bin sizes lead to foreground pixels and background pixels merging into a single CC.

***Algorithm 1**. Overlapped Binned Image generation*

```
function Generate_OB_Images(I,S)
begin
     M = ceil(255/S)*S
     binned_images = zeros(I.height, I.width, 2*(M/S) +1)
     k = S//2
     index = 0
     for main_lower in (0 to 255) with step_size = k
     loop:
          main_upper = main_lower+S-1+k
          gt_lower = ( I >= main_lower )
          lt_higher = ( I <= main_upper )
          bin_image = ( gt_lower AND lt_higher )
          binned_images[index] = bin_image
          index = index +1
     end loop
     return Binned_Images
end function
```

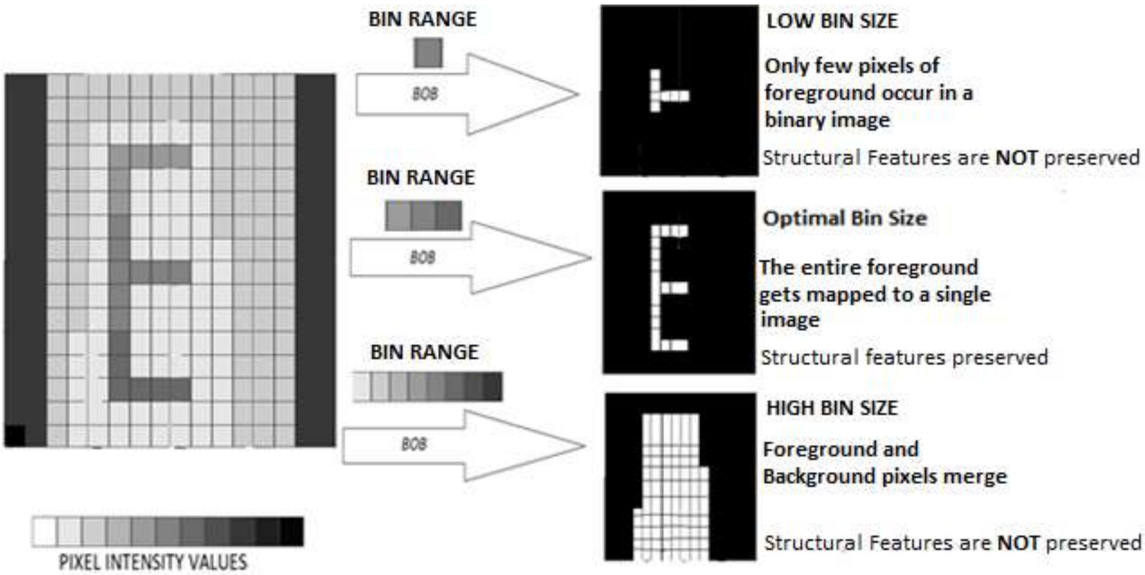
BIN RANGE
BOB
LOW BIN SIZE
Only few pixels of foreground occur in a binary image
Structural Features are NOT preserved
BIN RANGE
BOB
Optimal Bin Size
The entire foreground gets mapped to a single image
Structural features preserved
BIN RANGE
BOB
HIGH BIN SIZE
Foreground and Background pixels merge
Structural Features are NOT preserved
PIXEL INTENSITY VALUES

Fig. 3. A representative image showing the effect of low, optimal and high bin sizes on the CC generated at a binary image. For convenience, it is assumed that only 10-pixel intensity values exist and bin ranges that are 1, 3 and 8-pixel intensity values across are considered. The letter ‘E’ emerges as a perfect single CC in a binary image on applying bin range of 3. However, a lower bin range only leads to a few pixels of being mapped to a single CC while a high bin size captures too much of the background.

Therefore, instead of a single bin size, a large array of bin sizes is selected which ensures at least one of the OB images will contain the properly segmented text regions. Each of these OB images of size S maps to a range of pixel values of net range 1.5 × S with 75% overlap with the next OB image. The overlap and higher size ensure that any region R if it has a range of pixel values that is less than or equal to S will always be retrieved as a single CC or part of slightly larger CC with structural features preserved in one of the OB binary images provided its range of pixel values is not too less than S.

To generate the OB images, 3 parameters are chosen namely Lower Limit (LL), Upper Limit (UL) and Step Size (SS) that represent the bin size ranges. Initially, a low size value is considered and the step size is kept small to avoid problems that arise with too large and too small bin sizes as depicted in Fig. 3.

For initial bin size, LL is taken as the first size. At first, R channel is chosen and BOB is applied to generate OB images as follows:

Set 1 = Generate_OB_Images (I [:, :, 0], LL)

The bin size is increased to (LL+SS) and generate the next set of binary images for the second (G) image channel as:

Set 2 = Generate_OB_Images (I [:, :, 1], LL + SS)

Similarly, the bin size is increased by a further SS and OB images are generated from the third (B) channel as:

Set 3 = Generate_OB_Images (I [:, :, 2], LL + 2 × SS)

These OB images are generated for every channel alternatively until the size is greater than equal to UL which serves as the stopping criterion. Algorithm 2 describes the process for generating the complete set of OB images from all channels.

***Algorithm 2**. Generating complete set of OB images from all channels*

```
IMAGE I: Input Image of 3 Channels
function DAF_INPUT(I, Lower_Limit, Upper_Limit, Step_Size)
begin
    DAF_INPUT = None
    red_channel = I[:, :, 0]
    green_channel = I[:, :, 1]
    blue_channel = I[:, :, 2]
    channel_track = 0
    for size in (Lower_Limit, Upper_Limit) with step size = Step_Size
    loop:
        if channel_track % 3 == 0 then
            current_channel = red_channel
        else if channel_track % 3 == 1 then
            current_channel = green_channel
        else
            current_channel = blue_channel
        end if
        channel_track = channel_track + 1
        OB_Images = Generate_OB_Images(current_channel, size)
        if DAF_INPUT == None then
            DAF_INPUT = OB_Images
        else
            DAF_INPUT = Concatenate ([DAF_INPUT, OB_Images], axis=2)
        end if
    end loop
    return DAF_INPUT
end function
```

3.2 ResNet Architecture

Using BOB mechanism, a number of solutions are generated in the form of binary images. To determine the final solution among them, a CNN based classification mechanism is developed where simply changing the resolution of the input and output to the network gives an optimal outcome, as shown in Fig. 4. The architecture of DAFNet is based on the strategy of BOB based “Division” in intensity range of a channel while performing a CNN based “Filtering” to determine whether a particular CC should be part of the final binarized output or not. Unlike other text binarization schemes, this mechanism ensures optimal binarization for both cropped-word image as well as whole image. The CNN used here is comprised of two parts – Hour-Glass Residual Network (HGRN) [46] layer followed by a Channel Decrement (CD) layer as depicted in Fig. 5. HGRN consists of residual blocks like ResNet.

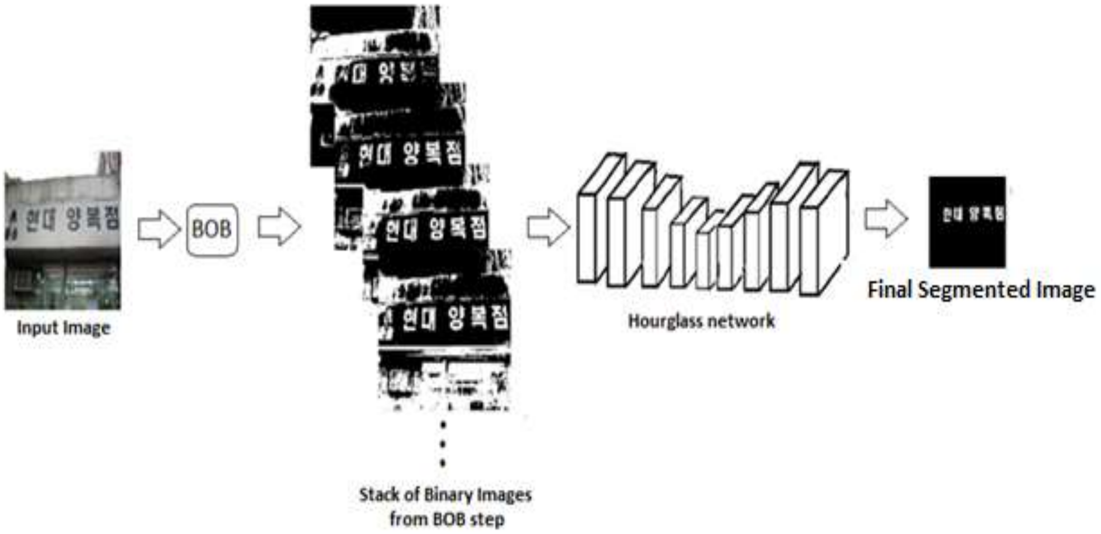

Fig. 4. Use of DAFNet to binarize text components in a sample digital image. This framework used for binarizing cropped word images, is also used for binarizing texts present anywhere in an image.

The number of BIs that is generated from BOB can be quite high. Hence, at first the number of channels is brought down to 64 using layers conv1 and conv2 with filter size 7. Each of these two-convolution layers is followed by a batch normalization layer [43]. Experimentally, it is observed that the most text components occur in around 60 of the total number of BIs hence the logic behind decreasing the number of channels to 64 using convolution. The next layers contain residual blocks with residual connections and batch normalization. Each alternate residual block decreases spatial resolution by half and doubles the number of channels as shown in Fig. 5.

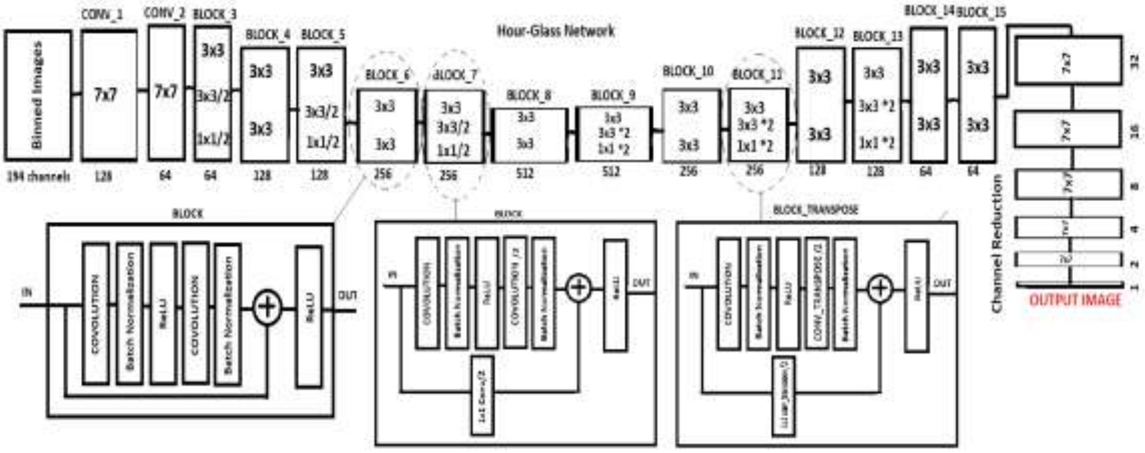

Fig. 5. Architecture of the network used in region detection. Blocks 3 to 8 are ResNet type blocks with residual connections. Blocks 9, 11 and 13 use convolution-transpose layers to double input resolution. Hour-Glass Network involves 5-channel reduction and convolutions each of filter 7×7 that reduces the number of channels by half.

Blocks 3 to 8 are the upper half of the hour-glass having 7 residual blocks that decrease the resolution (given input height is M and input width is N) from $\mathrm{M} \times \mathrm{N}$ to $\frac{M}{2} \times \frac{N}{2}, \frac{M}{4} \times \frac{N}{4}$ and finally $\frac{M}{8} \times \frac{N}{8}$. The number of channels is doubled from 64 to 128, 256 and 512 at each step in which the resolution is halved. The lower half of the hour-glass model from blocks 9 to 15 is the mirror of the first upper half and the resolution is doubled at each alternate residual block using convolution-transpose [47]. The number of channels is halved at each block from 512 to 256 then to 128 and finally to 64. The resolution is doubled from $\frac{M}{8} \times \frac{N}{8}$ to $\frac{M}{4} \times \frac{N}{4}$, then to $\frac{M}{2} \times \frac{N}{2}$ and finally to the original resolution of M × N.

At the end of the hour-glass model an M × N × 64 convolution matrix is obtained. The ground-truth image is an M × N binary image with the text regions labelled as 1. The number of channels is brought down from 64 to 1 using 6 convolution layers of filter size 7. Each convolution layer halves the number of channels from 64 to 32, 16, 8, 4, 2 and finally to 1.

Rectified Linear Unit (ReLU) [48] activation is used at each convolution layer and residual blocks except the final layer where tanh activation is applied. Tanh activation has output in range of [-1,1]. Here, 1 is added and divided by 2 to each output unit to bring it in the range of [0, 1].

The network architecture for script recognition shown in Fig. 6 is a ResNet-18 deep CNN. Here too, BOB is performed first to generate BIs. These BIs are fed to a ResNet for classification according to the scripts in which texts are written. The final number of output classes is n which denotes the number of scripts. The resolution of the input is 64 × 64 × B where B is the number of BIs produced.

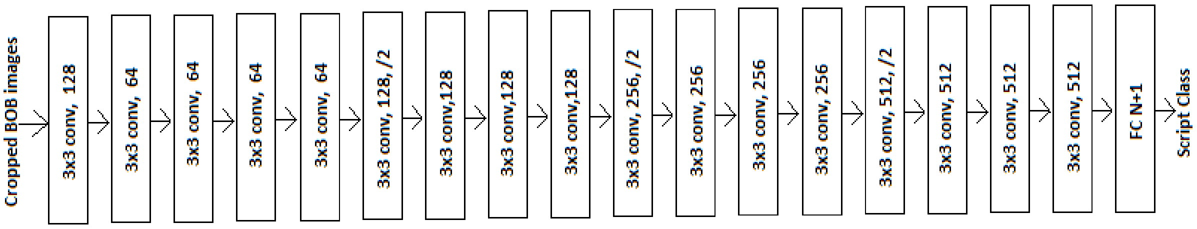

Fig. 6. ResNet-18 architecture used for script recognition.

3.3 Network Input, Output and Loss

As mentioned earlier, binarization can be done at both image level and cropped word level. For full-size images, on which BOB is applied to generate BIs that are resized to 256×256 pixels. If number of BIs after the BOB step is B, then a 256 × 256 × B input is passed to the HGRN that gives a 256×256 output image.

For binarizing cropped words as given in Fig. 7, BOB is performed and the BIs generated are resized to 32×64 pixels (width is greater than height in most input images, hence the width is chosen to be twice that of height while resizing). If number of BIs after the BOB step is B, then input to the hourglass network is 32 × 64 × B. The output is a 32×64 image.

For script recognition, the binarization outcome of the cropped words obtained from the above process, is utilized. If number of BIs after the BOB step is B, then input to the HGRN is 32 × 64 × B. The output is a N vector, where N is the number of possible scripts. To calculate loss, Intersection over Union (IoU) is calculated between the output of DAFNet and the Ground Truth (GT). The localization loss given a ground-truth G and output) is given in Eq. 1.

$$Loss_{localization} = 1 - \frac{\sum_{i=0}^{255}\sum_{j=0}^{255}(G_{ij}==1?\,O_{ij}:0)}{\sum_{i=0}^{255}\sum_{j=0}^{255}(G_{ij}==0?\,O_{ij}:1)} \qquad (1)$$

where O is the output from DAFNet and G is the GT image.

When output O and G are a perfect match (i.e., the value is 1 wherever G is 1 otherwise 0) then $\{G_{ij} == 1?\,O_{ij}\colon 0\} = 1$ and $\{G_{ij} == 0?\,O_{ij}\colon 1\} = 1$, thus $Loss = 1 - \frac{1}{1} = 0$. Since G_{ij} is either 0 or 1, the above loss will minimize when the output is low wherever G is 0 because all non-zero values at these points get added to the denominator. Correspondingly, the above loss will minimize when the output is high whenever G is 1 because those values get summed up in the numerator.

For script recognition, SoftMax Cross Entropy loss is calculated where there are n output nodes for n script classes. This SoftMax function takes as input an n vector Z and outputs an $n+1$ dimensional vector Y of real values between 0 and 1. This function is a normalized exponential and is defined in Eq. 2. The loss for script recognition is given in Eq. 3.

$$Y_c = \frac{e^{Z_c}}{\sum_{d=1}^{C} e^{Z_d}} \; for\ c = 1,2 \ldots\ldots n \qquad (2)$$

$$Loss_{recognition} = -log\ (Y_i) \qquad (3)$$

where i is the actual class of the cropped-word image.

4. Experimental Results and Discussion

The proposed system is experimentally evaluated on several standard datasets and compared with state-of-the-art for binarization as well as script recognition methods. Pixel level evaluation is used where a text pixel (=1) is True Positive if and only if the corresponding pixel in the GT is also a text pixel (=1), otherwise with the ground-truth pixel at that coordinate is 0 and flagged as False Positive. Similarly, if the pixel is 0 at the point where ground-truth pixel is 1, it is flagged as False Negative. Precision and Recall as given in Eqs. 4, 5 and 6 respectively.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad (5)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (6)$$

4.1 Dataset Description

For training the CNN, pixel level annotations are used to label the word CCs obtained from the bins. Pixel level annotations are provided by [49] for the following benchmark datasets:

1. **SVT:** The Street View Text (SVT) dataset consists of 647 images depicting outdoor scenes with mostly Roman texts in signboards, hoardings, license plates, banners and so

on. Annotations of these texts are done at both word and character level along with OCR GTs.

2. **ICDAR (2003, 2011, 2019):** This is a competition where several researchers across the world compete by proposing methods to address the challenges specifically in the TDI analysis domain for robust reading. ICDAR 2003 dataset is a collection of 1110 scene images having words in various font sizes and styles. Similarly, ICDAR 2011 provides a collection of 716 scene images NSIs as well as a collection of 918 BDIs that are sourced from web and email. Words in these images are mostly written in Roman and have variations in terms of color, size, font style, orientation, etc. The annotation of the words is done at pixel level of each character foreground. ICDAR 2019 comprises of NSIs with embedded text, such as street signs, street advertising boards, shop names, passing vehicles and users photos in microblogs. The images were captured using different mobile phone cameras or were collected from freely available images from the Internet. The dataset consists of 20,000 images containing text of 10 different scripts (2,000 images per script). Most images contain text of more than one script, but each script is represented in at least 2,000 images. The 10 scripts are: Arabic, Bangla, Chinese, Devanagari, Roman, French, German, Italian, Japanese and Hangul. The text in the scene images of the dataset is annotated at word level. A GT-word (What is GT – you have not said earlier!) is defined as a consecutive set of characters without spaces, i.e., words are separated by spaces, except in Chinese and Japanese, where the text is labelled at text-line level. Each GT-word is labelled by a 4-corner bounding box, and is associated with a script class.

3. **KAIST:** This is a multi-lingual scene text dataset comprised of 3000 images captured in both indoor and outdoor environments. The images shot under lighting conditions have words written in Roman and Hangul scripts. Pixel level annotations are done for representing the word components.

4. **Total-Text:** This dataset is a collection of images with multi-oriented words written in Roman and some Asian scripts like Chinese. The dataset is divided into train as well as test sets having 1255 and 300 images respectively. The words in these images vary in terms of size, font style, and shape etc. The annotations for the scene texts are available in word region level (both polygon and rectangle) as well as pixel-level.

4.2 Experimental Setup

For each dataset 20% of data are kept for testing and the rest are used for training. The training is done in batches of 5 images for binarizing full-size images, 15 for binarizing cropped-word images and for script recognition. Training is carried out on a NVIDA 1050Ti GPU with 4GB of graphics memory. It took around 10 hours to converge for a dataset of 1000 images. Adam optimization [50] is used to optimize the loss function. The parameters are set as: beta1 = 0.9 and gamma1 = 0.99. The base learning rate is set as 0.0002 which is reduced according to $lrbase \times (1 - \frac{iter}{max\ \_iter})^{power}$ with power = 0.9 by following the method in [51].

4.3 Binarization Performance

The performance of the proposed system is evaluated on datasets mentioned in Section 4.1. For full image binarization, evaluation is done on KAIST and Total-Text which have pixel-level GTs. For cropped-word binarization, evaluation is done on datasets like SVT, ICDAR 2003, ICDAR 2011 and Total-Text.

4.3.1 Image-level binarization

The resolution of the input and output is 256×256 for training, i.e., input image and ground-truth are resized to 256×256 before training the network. KAIST contains images in 2 different scripts organized into 3 types – images containing only Roman, images containing only Hangul, and images containing both Hangul and Roman. Each contains two different quality of images – captured using mobile phone camera and digital camera. There are a total of 2465 images of all the 6 categories where 80% of these images (1972) are randomly selected for training and the rest (492) for testing and evaluation.

Total-Text contains 1255 images for training and another 300 for testing, with many curved texts. In each image, text is annotated at word level, each word is labelled by a bounding polygon. No pre-training is required unlike that in [52]. Using polynomial annotations, binary images are first generated where polynomial regions are set as "true". These are used as ground-truth for comparison.

The evaluation results of the proposed DAFNet system are illustrated in Table 1 and compared with some standard methods for both KAIST as well as Total-Text datasets. A visual comparison between the GT and the system outcome of sample images from the test set of KAIST and Total-Text datasets, is shown in Fig. 7 highlighting a very high degree of segmentation capability of the DAFNet mechanism.

Table 1. Evaluation results on KAIST and Total-Text datasets for image-level binarization of TDI.

Dataset	Method	Precision	Recall	F-Score
KAIST	Otsu [6]	0.75	0.78	0.76
	MSER [25]	0.76	**0.86**	0.81
	SWT [27]	0.83	0.84	0.83
	Ozgen et al. [53]	0.57	0.54	0.55
	DAFNet (Proposed)	**0.88**	**0.86**	**0.87**
Total-Text	Liao et al. [54]	0.77	0.74	0.75
	MSR [55]	0.81	0.73	0.77
	Mask-TextSpotter [56]	0.83	0.75	0.79
	TextField [57]	0.83	**0.82**	0.83
	DAFNet (Proposed)	**0.86**	**0.82**	**0.84**

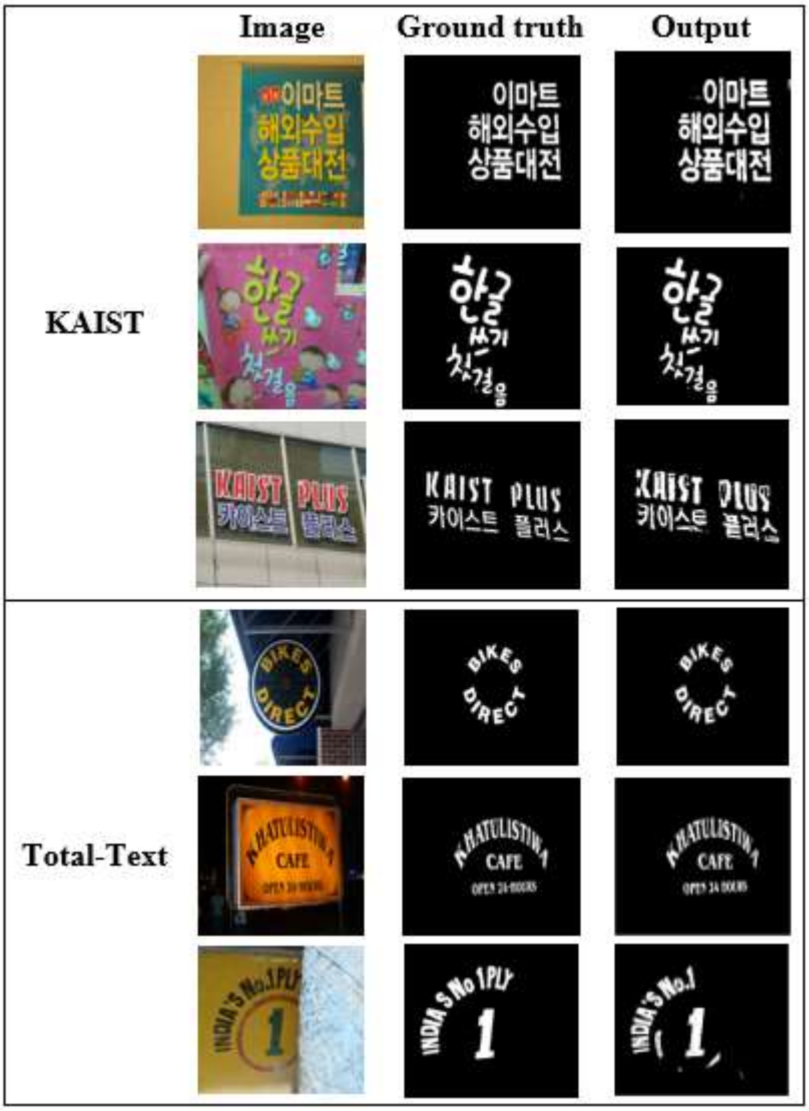

Fig. 7. Sample results on KAIST and Total-Text datasets for image-level binarization of TDI.

4.3.2 Component-level binarization

DAFNet can also be used for binarizing cropped-word images. Cropped-word images and their corresponding ground-truths are resized to 32×64, width being greater than height since most words are horizontally than vertically. SVT, ICDAR 2003, ICDAR 2011 Scene Text, ICDAR 2011 BDI as well as cropped words from the Total-Text dataset are used to evaluate accuracy of binarization on cropped words. Total-Text contains rectangular bounding box coordinates which are used to crop out both from the original images as well as the pixel level ground-truths. Fig. 8 provides sample evaluation results and Table 2 depicts the performance comparison between the proposed method and some state-of-the-art techniques for cropped binarization.

Table 2. Performance comparison of the proposed system with some the state-of-the-art methods on different standard datasets for cropped-word images.

Dataset	Method	Precision	Recall	F-Measure
SVT	Otsu [6]	0.64	**0.83**	0.72
	Kittler et al. [13]	0.55	0.81	0.66
	Niblack [14]	0.58	0.81	0.68
	Sauvola & Pietikäinen [15]	0.52	0.78	0.62

	Wolf & Doermann [16]	0.52	0.76	0.62
	Howe [17]	0.62	0.77	0.69
	Milyaev et al. [18]	0.52	0.66	0.58
	Kasar et al. [19]	0.70	0.71	0.70
	Feild & Learned-Miller [20]	0.64	0.79	0.71
	Mishra et al. [21]	0.64	0.82	0.72
	DAFNet (Proposed)	**0.89**	0.76	**0.82**
ICDAR 2003	Otsu [6]	0.86	0.90	0.88
	Kittler et al. [13]	0.75	0.89	0.81
	Niblack [14]	0.65	0.83	0.73
	Sauvola & Pietikäinen [15]	0.68	0.87	0.76
	Wolf & Doermann [16]	0.81	**0.91**	0.86
	Howe [17]	0.76	0.84	0.79
	Milyaev et al. [18]	0.71	0.69	0.69
	Kasar et al. [19]	0.72	0.64	0.68
	Feild & Learned-Miller [20]	0.84	0.85	0.84
	Mishra et al. [21]	0.82	**0.91**	0.86
	DAFNet (Proposed)	**0.98**	0.90	**0.94**
ICDAR 2011 (NSI)	Otsu [6]	0.87	**0.91**	**0.89**
	Kittler et al. [13]	0.79	0.89	0.84
	Niblack [14]	0.83	0.90	0.86
	Sauvola & Pietikäinen [15]	0.75	0.86	0.80
	Wolf & Doermann [16]	0.73	0.81	0.77
	Howe [17]	0.76	0.87	0.81
	Milyaev et al. [18]	0.72	0.73	0.72
	Kasar et al. [19]	0.65	0.47	0.55
	Feild & Learned-Miller [20]	0.89	0.87	0.88
	Mishra et al. [21]	0.86	**0.91**	0.88
	DAFNet (Proposed)	**0.92**	0.83	0.87
ICDAR 2011 (BDI)	Otsu [6]	0.77	0.92	0.84
	Kittler et al. [13]	0.57	0.88	0.69
	Niblack [14]	0.59	**0.94**	0.72
	Sauvola & Pietikäinen [15]	0.54	**0.94**	0.69
	Howe [17]	0.43	0.93	0.59
	Milyaev et al. [18]	0.48	0.68	0.56
	Kasar et al. [19]	0.55	0.65	0.59
	Feild & Learned-Miller [20]	0.75	0.86	0.80
	Mishra et al. [21]	0.70	0.90	0.79
	Ghoshal et. al [31]	0.71	**0.94**	**0.81**
	Bhattacharya et. al [58]	0.48	0.91	0.53
	Kumar et. al [59]	0.47	0.86	0.47
	DAFNet (Proposed)	**0.89**	0.86	**0.87**
Total-Text	Bonechi et al. [60]	0.73	0.57	0.64
	Liao et al. [61]	0.87	**0.83**	**0.85**

	Dai et al. [62]	0.85	0.78	0.81
	Xie et al. [63]	0.83	**0.83**	0.83
	Xu et. al. [64]	-	-	**0.85**
	Wang et. al. [65]	0.83	0.82	0.81
	DAFNet (Proposed)	**0.89**	0.81	**0.85**

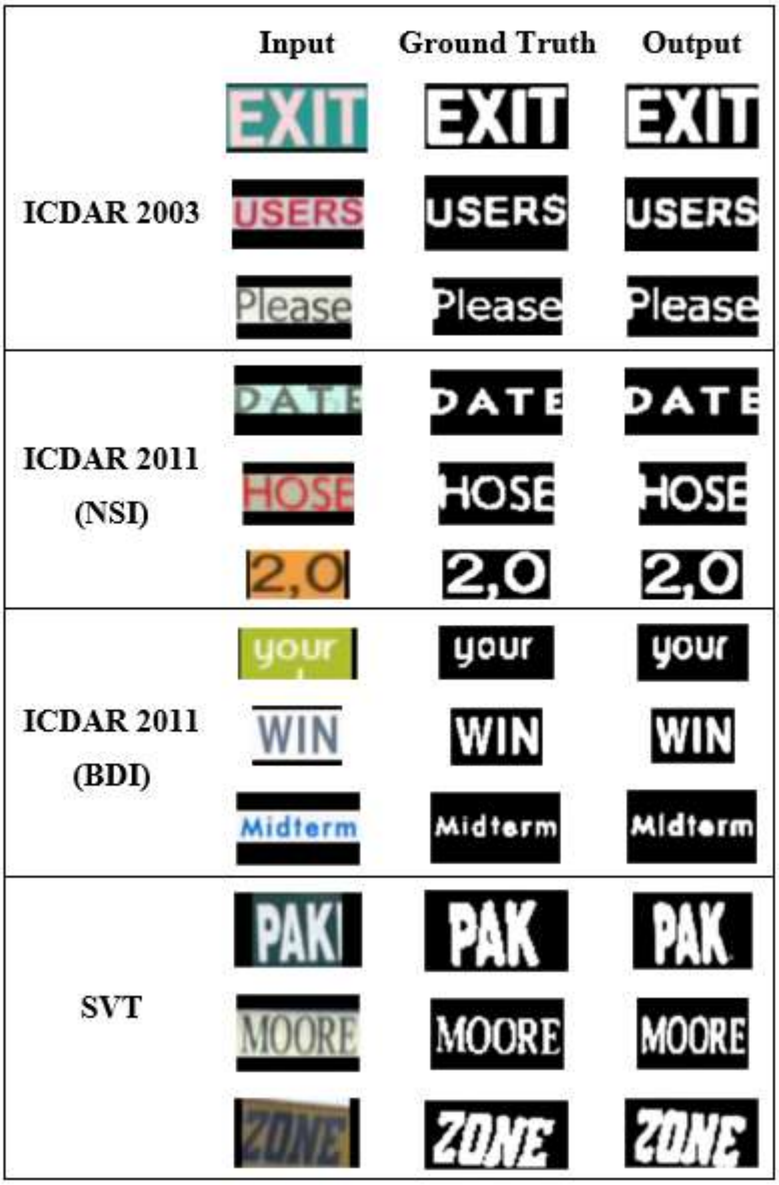

Fig. 8. Sample results for component-level binarization of TDI on different datasets.

4.4 Script Recognition Performance

Script recognition performance is reported for multi-script datasets like KAIST and ICDAR 2019. For script recognition, initially BOB is performed on the cropped-word images to get the candidate BIs. These BIs are then passed through the HGRN module for deciding the optimal BI that can be further analyzed by the same network for identifying the corresponding script of the word image. The evaluation for script recognition is done on ICDAR 2019 and KAIST datasets. For evaluation on KAIST, the bounding box coordinates at word-level are used to estimate the script identification accuracy. In ICDAR 2019, each GT-word is labelled by a 4-corner bounding box and is associated with a script class.

For each set of images for a script, 80% of the images are taken for training and the rest for testing the model. For each cropped word, the GT is a one-hot vector of dimension 10 for ICDAR 2019 and 2 for KAIST, where the corresponding values are 1 depending on which one of the scripts the cropped word belongs to. The evaluation results are given in Table 3 for both KAIST and ICDAR 2019 datasets and compared with the outcome of some standard methods.

Table 3. Script identification results applied on the extracted text regions for different datasets.

Dataset	Method	Accuracy (%)
ICDAR 2019	GSPA [12]	91.02
	Conv_attention [12]	88.41
	NXB_OCR [12]	84.88
	Res MUL SPP [12]	71.31
	ELE-MLT [12]	82.86
	DAFNet (Proposed)	**92.05**
KAIST	Saha et al. [5]	95.90
	Chakraborty et al. [35]	97.40
	Chakraborty et al. [36]	96.29
	DAFNet (Proposed)	**97.41**

4.5 Discussion

From the performance obtained, it is observed that the binarization technique gives better results for most of the datasets considered in this work. The same architecture can be used for binarizing TDI at image-level as well as word-level. As described earlier, DAFNet is based on the principle of "Divide and Filter" where "Divide" operation happens during BOB implementation, while "Filter" operation is performed by the fully convolutional HGRN that determines if a particular CC should be part of the final binarized output. The BOB step also serves an additional purpose – it prevents overfitting of networks by increasing number of input parameters for the network to fit to by almost 100 times and increasing input data complexity. In neural networks [66], overfitting can occur when data is few and input parameters are low – allowing the neural net to learn weights that fit to specific sequences to input parameters. When the number of input parameters is large compared to the number of learnable weights – the network does not easily overfit to this large number of input parameters. Small changes in the image introduced in the data augmentation step like random scaling and cropping of images, changes several input parameters at the same time making it difficult for the network to overfit to a specific sequence of input parameters. After the BOB step, each image gets converted to around 100-200 binary images each representing a particular bin or range of pixel values being captured. Large number of input parameters implies overfitting to a dataset requires overfitting to a very large dimensional space that is sparsely populated, which makes it difficult for the networks we use to overfit completely. Thus, even small sized dataset can be sufficient for training the network when BOB representation is used to feed into the network.

5 Conclusion

In this work, a novel deep learning-based text specific binarization method is presented where efficient segmentation of text from digital images is performed. A combination of BOB mechanism and deep CNN model called DAFNet provides a robust solution to the binarization problem. Most of the text segmentation is performed during the binning process itself. It also provides tight bounds on the regions detected in arbitrary oriented text without the need for additional networks or convolutions, i.e., it directly outputs the region masks. A deep network for efficient script recognition is also built that can be used to localize text in any script as well as to identify the script before feeding it to a script specific OCR system. Experiments are conducted on different standard datasets and comparisons are drawn between the performance of the proposed DAFNet based framework with that of the state-of-the-art systems for binarization as well as script identification of TIDs. From the performance reported, it is observed that while the recall achieved for various datasets lies proximally around that of the state-of-the-art (same, nearly same or higher), precision has significantly improved giving a boost to the F-measure scores. DAFNet achieves highest F-measure scores as 0.87 and 0.84 when applied on KAIST and Total-Text datasets respectively for image level binarization. For binarization of cropped word images, F-measure scores of 0.82, 0.94, 0.88, 0.87 and 0.85 are achieved for SVT, ICDAR 2003, 2011 (NSI), 2011 (BDI) and Total-Text respectively. Also, DAFNet proves its robustness and efficiency in script recognition of TDIs by achieving the highest accuracies of 92.05% and 97.41% for ICDAR 2019 and KAIST datasets respectively. One limitation of the system, however, is closely aligned words often end up being combined into a single CC. Additionally, with more focus on achieving tight bound text binarization that enhances the precision, the recall falls for some images as seen in case of ICDAR 2011 (NSI) dataset. Future aim includes dealing with these issues by constructing some heuristics that may help in optimal decision making and using a single-shot detector that can identify text boxes and scripts in a single feature map, thereby avoiding adjacent word regions from overlapping.

The code of this work will be made publicly available upon acceptance of the paper.

Acknowledgement: This work is partially supported by the CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India, PURSE-II and UPE-II, project. This work is partially funded by DBT grant (BT/PR16356/BID/7/596/2016) and DST grant (EMR/2016/007213).

References

1. Joan, S. F., & Valli, S. (2019). A survey on text information extraction from born-digital and scene text images. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, *89*(1), 77-101.
2. Zhang, H., Zhao, K., Song, Y. Z., & Guo, J. (2013). Text extraction from natural scene image: A survey. *Neurocomputing*, *122*, 310-323.
3. Lin, H., Yang, P., & Zhang, F. (2020). Review of scene text detection and recognition. *Archives of Computational Methods in Engineering*, *27*(2), 433-454.
4. Dutta, I. N., Chakraborty, N., Mollah, A. F., Basu, S., & Sarkar, R. (2019). Multi-lingual text localization from camera captured images based on foreground homogenity analysis. In *Recent Developments in Machine Learning and Data Analytics* (pp. 149-158). Springer, Singapore.

5. Saha, S., Chakraborty, N., Kundu, S., Paul, S., Mollah, A. F., Basu, S., & Sarkar, R. (2020). Multi-lingual scene text detection and language identification. *Pattern Recognition Letters*, *138*, 16-22.
6. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66.
7. Jajoo, M., Chakraborty, N., Mollah, A. F., Basu, S., & Sarkar, R. (2019). Script identification from camera-captured multi-script scene text components. In *Recent developments in machine learning and data analytics* (pp. 159-166). Springer, Singapore.
8. Ch'ng, C. K., & Chan, C. S. (2017, November). Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (Vol. 1, pp. 935-942). IEEE.
9. Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2016). Textboxes: A fast text detector with a single deep neural network. *arXiv preprint arXiv:1611.06779*.
10. Xing, L., Tian, Z., Huang, W., & Scott, M. R. (2019). Convolutional character networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 9126-9136).
11. Indra Narayan Dutta, Neelotpal Chakraborty, Ayatullah Faruk Mollah and Ram Sarkar. (In Press). BOB: A Bi-level Overlapped Binning procedure for Scene Word Binarization, Multimedia Tools and Applications, Springer.
12. Nayef, N., Patel, Y., Busta, M., Chowdhury, P. N., Karatzas, D., Khlif, W., ... & Ogier, J. M. (2019, September). ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition—RRC-MLT-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1582-1587). IEEE.
13. Kittler, J., Illingworth, J., & Föglein, J. (1985). Threshold selection based on a simple image statistic. *Computer vision, graphics, and image processing*, *30*(2), 125-147.
14. Niblack, W. (1985). An Introduction to Digital Image Processing, 215 Strandberg Publishing Company. *Copenhagen, Denmark*.
15. Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern recognition*, *33*(2), 225-236.
16. Wolf, C., & Doermann, D. (2002, August). Binarization of low quality text using a markov random field model. In *Object recognition supported by user interaction for service robots* (Vol. 3, pp. 160-163). IEEE.
17. Howe, N. R. (2013). Document binarization with automatic parameter tuning. *International journal on document analysis and recognition (ijdar)*, *16*(3), 247-258.
18. Milyaev, S., Barinova, O., Novikova, T., Kohli, P., & Lempitsky, V. (2015). Fast and accurate scene text understanding with image binarization and off-the-shelf OCR. *International Journal on Document Analysis and Recognition (IJDAR)*, *18*(2), 169-182.
19. Kasar, T., Kumar, J., & Ramakrishnan, A. G. (2007, September). Font and background color independent text binarization. In *Second international workshop on camera-based document analysis and recognition* (pp. 3-9).
20. Feild, J., & Learned-Miller, E. (2012). Scene text recognition with bilateral regression. *Department of Computer Science, University of Massachusetts Amherst, Tech. Rep. UM-CS-2012-021*.
21. Mishra, A., Alahari, K., & Jawahar, C. V. (2017). Unsupervised refinement of color and stroke features for text binarization. *International Journal on Document Analysis and Recognition (IJDAR)*, *20*(2), 105-121.
22. Bhowmik, S., Sarkar, R., Das, B., & Doermann, D. (2018). GiB: A ${G} Game Theory

Inspired Binarization Technique for Degraded Document Images. *IEEE Transactions on Image Processing*, *28*(3), 1443-1455.

23. Weinman, J. J., Butler, Z., Knoll, D., & Feild, J. (2013). Toward integrated scene text reading. *IEEE transactions on pattern analysis and machine intelligence*, *36*(2), 375-387.
24. Ghoshal, R., Roy, A., Banerjee, A., Dhara, B. C., & Parui, S. K. (2019). A novel method for binarization of scene text images and its application in text identification. *Pattern Analysis and Applications*, *22*(4), 1361-1375.
25. Wang, Fang. "Adaptive preprocessing of character recognition image based on neural network." Journal of Physics: Conference Series. Vol. 1982. No. 1. IOP Publishing, 2021.
26. Qaroush, Aziz, et al. "An efficient, font independent word and character segmentation algorithm for printed Arabic text." Journal of King Saud University-Computer and Information Sciences 34.1 (2022): 1330-1344.
27. Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, *22*(10), 761-767.
28. Chen, H., Tsai, S. S., Schroth, G., Chen, D. M., Grzeszczuk, R., & Girod, B. (2011, September). Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *2011 18th IEEE International Conference on Image Processing* (pp. 2609-2612). IEEE.
29. Epshtein, B., Ofek, E., & Wexler, Y. (2010, June). Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2963-2970). IEEE.
30. Li, Y., Jia, W., Shen, C., & van den Hengel, A. (2014). Characterness: An indicator of text in the wild. *IEEE transactions on image processing*, *23*(4), 1666-1677.
31. Ghoshal, Ranjit, and Ayan Banerjee. "Region Growing-Based Scheme for Extraction of Text from Scene Images." Proceedings of International Conference on Frontiers in Computing and Systems. Springer, Singapore, 2021”.
32. Xu, Xixi, et al. "BTS: A Bi-Lingual Benchmark for Text Segmentation in the Wild." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
33. Liao, Minghui, et al. "Real-time scene text detection with differentiable binarization and adaptive scale fusion." IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).
34. Wang, Chuan, et al. "Semi-supervised pixel-level scene text segmentation by mutually guided network." IEEE Transactions on Image Processing 30 (2021): 8212-8221.
35. Yin, X. C., Pei, W. Y., Zhang, J., & Hao, H. W. (2015). Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), 1930-1937.
36. Bai, X., Yao, C., & Liu, W. (2016). Strokelets: A learned multi-scale mid-level representation for scene text recognition. *IEEE Transactions on Image Processing*, *25*(6), 2789-2802.
37. Tian, C., Xia, Y., Zhang, X., & Gao, X. (2017). Natural scene text detection with MC–MR candidate extraction and coarse-to-fine filtering. *Neurocomputing*, *260*, 112-122.
38. Paul, S., Saha, S., Basu, S., & Nasipuri, M. (2015). Text localization in camera captured images using adaptive stroke filter. In *Information Systems Design and Intelligent Applications* (pp. 217-225). Springer, New Delhi.
39. Paul, S., Saha, S., Basu, S., Saha, P. K., & Nasipuri, M. (2019). Text localization in camera captured images using fuzzy distance transform based adaptive stroke filter. *Multimedia Tools and Applications*, *78*(13), 18017-18036.

40. Bhunia, A. K., Kumar, G., Roy, P. P., Balasubramanian, R., & Pal, U. (2018). Text recognition in scene image and video frame using Color Channel selection. *Multimedia tools and applications*, *77*(7), 8551-8578.
41. Islam, Rashedul, Md Rafiqul Islam, and Kamrul Hasan Talukder. "An efficient ROI detection algorithm for Bangla text extraction and recognition from natural scene images." Journal of King Saud University-Computer and Information Sciences (2022).
42. Khan, Tauseef, and Ayatullah Faruk Mollah. "AUTNT-A component level dataset for text non-text classification and benchmarking with novel script invariant feature descriptors and D-CNN." Multimedia Tools and Applications 78.22 (2019): 32159-32186.
43. Chakraborty, N., Chatterjee, A., Singh, P. K., Mollah, A. F., & Sarkar, R. (2020). Application of daisy descriptor for language identification in the wild. *Multimedia Tools and Applications*, 1-22.
44. Chakraborty, N., Kundu, S., Paul, S., Mollah, A. F., Basu, S., & Sarkar, R. (2020). Language identification from multi-lingual scene text images: a CNN based classifier ensemble approach. *Journal of Ambient Intelligence and Humanized Computing*, 1-12.
45. Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017). Image-based localization using hourglass networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 879-886).
46. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
47. Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.
48. Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
49. Kumar, D., Prasad, M. A., & Ramakrishnan, A. G. (2012, December). Benchmarking recognition results on camera captured word image data sets. In *Proceeding of the workshop on Document Analysis and Recognition* (pp. 100-107).
50. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
51. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834-848.
52. Qin, S., Bissacco, A., Raptis, M., Fujii, Y., & Xiao, Y. (2019). Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4704-4714).
53. Özgen, A. C., Fasounaki, M., & Ekenel, H. K. (2018, May). Text detection in natural and computer-generated images. In *2018 26th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.
54. Liu, Y., Jin, L., Zhang, S., Luo, C., & Zhang, S. (2019). Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, *90*, 337-345.
55. Xue, C., Lu, S., & Zhang, W. (2019). Msr: Multi-scale shape regression for scene text detection. *arXiv preprint arXiv:1901.02596*.
56. Liao, M., Lyu, P., He, M., Yao, C., Wu, W., & Bai, X. (2019). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE transactions on pattern analysis and machine intelligence*.
57. Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., & Bai, X. (2019). Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, *28*(11), 5566-5579.

58. Bhattacharya, U., Parui, S.K., Mondal, S.: Devanagari and bangla text extraction from natural scene images. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), pp. 171–175 (2009)
59. Kumar, D., Ramakrishnan, A.G.: Octymist:otsu-canny minimal spanning tree for born-digital images. In: Proceedings of the 10th IAPR International Workshop on Document Analysis Systems. DAS ’12, pp. 389–393 (2012)
60. Bonechi, S., Andreini, P., Bianchini, M., & Scarselli, F. (2019, September). COCO_TS Dataset: Pixel–Level Annotations Based on Weak Supervision for Scene Text Segmentation. In *International Conference on Artificial Neural Networks* (pp. 238-250). Springer, Cham.
61. Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2020). Real-Time Scene Text Detection with Differentiable Binarization. In *AAAI* (pp. 11474-11481).
62. Dai, Y., Huang, Z., Gao, Y., Xu, Y., Chen, K., Guo, J., & Qiu, W. (2018, August). Fused text segmentation networks for multi-oriented scene text detection. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 3604-3609). IEEE.
63. Xie, E., Zang, Y., Shao, S., Yu, G., Yao, C., & Li, G. (2019, July). Scene text detection with supervised pyramid context network. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 9038-9045).
64. Xu, Xingqian, et al. "Rethinking text segmentation: A novel dataset and a text-specific refinement approach." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
65. Wang, Chuan, et al. "Semi-supervised pixel-level scene text segmentation by mutually guided network." *IEEE Transactions on Image Processing* 30 (2021): 8212-8221.
66. Ahamed, P., Kundu, S., Khan, T., Bhateja, V., Sarkar, R., & Mollah, A. F. (2020). Handwritten Arabic numerals recognition using convolutional neural network. *JOURNAL OF AMBIENT INTELLIGENCE AND HUMANIZED COMPUTING*.