PixelLink Preliminary Summary

Shishir Jakati
University of Massachusetts Amherst
sjakati@umass.edu

Abstract

Many scene text detector techniques have recently taken to deep convectional networks for two major tasks: text localization and bounding box regression. PixelLink's novel approach presents a scene text detector technique does not require the a bounding box regressor routine, but rather utilizes connected component segmentation of text regions. This ultimately results in a computationally sound and succinct detector network.

1. Network Structure

1.1. Instance Segmentation

Segmentation tasks are divided between two major goals. Semantic segmentation can be considered the simpler task, as it is the act of assigning pixel-wise labels to every pixel in an image. Objects segmented by a semantic segmentation model are not differentiated, whereas instance segmentation has the goal of assigning every instance of a class a pixel-wise label. The major contrast between the two segmentation tasks is how pixel-wise labels are assigned.[4] Within the PixelLink[1] architecture, this instance segmentation model is used both for text extraction and for bounding box creation.

The first step in the PixelLink pipeline is to generate two separate feature maps from the input image. These two are generated from the two network headers: one for text/non-text prediction, the other for link prediction. The text prediction network is self-explanatory. The feature map is the activated output of locations the network believes there is a text instance.

1.2. Pixel Linking

It must first be established that each pixel has 8 neighbors. These are: left, left-down, left-up, right, right-down, up, down. Each of these neighbors are useful in generating connected components between pixels. For every pixel that is predicted as positive, a link between it and it's neighbors is created as being part of a connected component. One of the two network headers is specifically responsible for predicting these links.

This linking process is useful for refining the original text/non-text prediction that is performed by the other network header. While text/non-text prediction is useful for generating larger instance segmentation, the pixel linking routine is useful in separating between different words.

1.3. Training

To account for the relative size of different text instances, the PixelLink training routine defines a central weight for each text instance. Given N text instances, all text instances are assigned the central total weight B_i. Within each text instance, every pixel is assigned the weight $w_i = B_i/S_i$, where S_i is the area of some text instance. B_i is trivially defined to be the total number of pixels within text instances divided by the number of separate text instances.

The training loss is thus defined as

$$L = \lambda L_{pixel} + L_{link}$$

The loss function is separated between the two tasks, text/non-text prediction and link prediction. The loss function for text/non-text prediction is defined as

$$L_{pixel} = \frac{1}{(1+r)S} W L_{pixel\_CE}$$

Here, r is defined as the negative-positive ratio, as used in Online Hard Example Mining (OHEM). In this specific case, is used to select negative text predicted pixels with the top-K loss. It was found best to set r to 3. S is defined to be the area of all text instance areas. W is the weight matrix of the pixels being considered, the weights are assigned as prescribed above. Finally, L_{pixel_CE} is defined to be the Cross-Entropy loss matrix on text/non-text prediction.

Within the link loss, losses for positive and negative links are calculated separately and on positive pixels only. This loss is defined as

$$L_{link} = \frac{L_{link\_pos}}{rsum(W_{pos\_link})} + \frac{L_{link\_neg}}{rsum(W_{neg\_link})}$$

Here,

$$L_{link\_pos} = W_{pos\_link} L_{link\_CE}$$

Model	**Recall**	**Precision**	**F-Score**	**FPS**
PixelLink+VGG16 2s	82.0	85.5	83.7	3.0
PixelLink+VGG16 4s	81.7	82.9	82.3	7.3
EAST+PVANET2x MS	78.3	83.3	81.0	N/A
EAST+PVANET2x	73.5	83.6	78.2	13.2
EAST+VGG16	72.8	80.5	76.4	6.5
SegLink+VGG16	76.8	73.1	75.0	N/A
CTPN+VGG16	51.6	74.2	60.9	7.1

Table 1. Performance Results Reproduced from [1]

and

$$L_{link\_neg} = W_{neg\_link} L_{link\_CE}$$

Where L_{link_CE} is the previously defined Cross-Entropy loss on the link prediction, and W_{pos_link} and W_{neg_link} are defined to be the weight of the pixel conditioned on whether or not the pixel is predicted to be a link or not.

PixelLink is optimized using SGD with momentum 0.9, and weight decay $5 * 10^{-4}$. Additionally, when training the learning rate is set to 10^{-3} for the first 100 iterations, and then increased to 10^{-2} for the remaining iterations.

2. Results

Although PixelLink was evaluated on various datasets such as: ICDAR 2015 Challenge 4[2], ICDAR 2013[3], and MSRA - TD500[], it is most relevant to discuss ICDAR 2015 Challenge 4.

The ICDAR 2015 Challenge consists of 1000 training images and 500 testing images. The results presented in the PixelLink publication are reproduced here. It is evident that PixelLink outperforms the next best presented, EAST + PVANET2x, method by 2.7 with respect to F-Score.

3. Relevance to Research

Extraction of low resolution text instances may be accomplished using instance segmentation.[5] Previously, instance segmentation powered a majority of the connected component methods submitted to ICDAR 2015. PixelLink's use of CNNs to perform pixel-wise instance segmentation introduces an additional level of sophistication, that has empirically produced more accurate results. Within the low-resolution/small-text problem, these segmentation methods can be employed to better extract text instances which were not recognizable using other traditional detection techniques.

Listed advantages of PixelLink include producing more accurate results while utilizing less data[]. Supposedly, each prediction neuron in a network header is responsible for fewer pixels than in a traditional quadrangle regression network. Ultimately, the receptive field is much less stringent, as it must only focus on its pixel and the direct neighbors.

References

[1] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation, 2018. 1, 2

[2] 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 *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, ICDAR '15, pages 1156–1160, Washington, DC, USA, 2015. IEEE Computer Society. 2

[3] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras. Icdar 2013 robust reading competition. In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, ICDAR '13, pages 1484–1493, Washington, DC, USA, 2013. IEEE Computer Society. 2

[4] G. Stockman and L. G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. 1

[5] Y. Wei, Z. Zhang, W. Shen, D. Zeng, M. Fang, and S. Zhou. Text detection in scene images based on exhaustive segmentation. *Signal Processing: Image Communication*, 50:1 – 8, 2017. 2