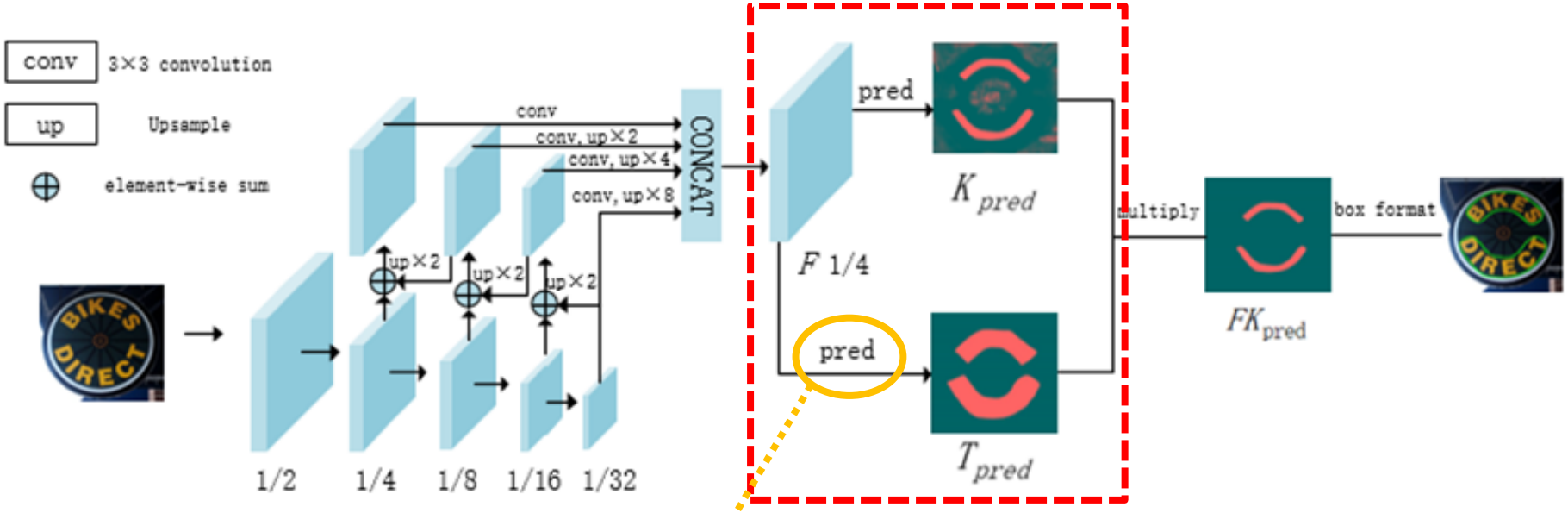
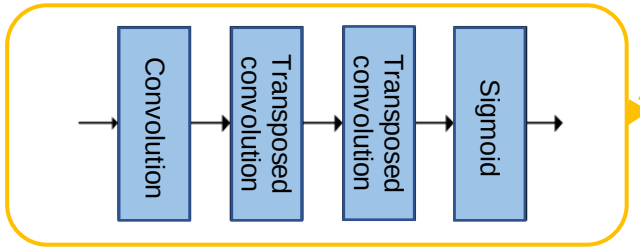
An Accurate Threshold Insensitive Kernel Detector for Arbitrary Shaped Text

Speaker: Xijun Qian

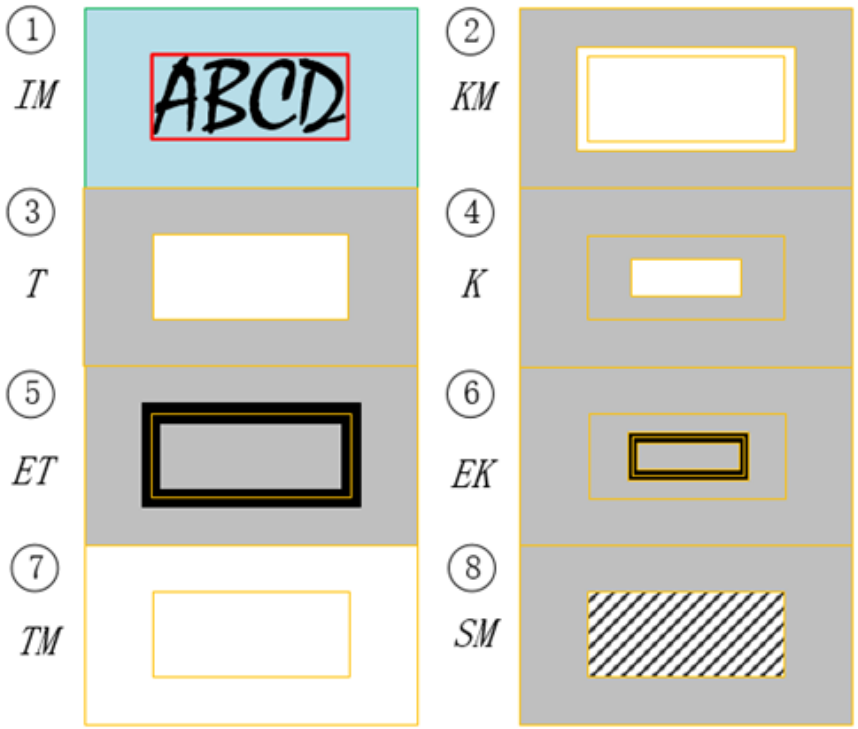
Detection Process





We divide the detection into two parts, respectively detecting the text core and the text core area

Label Generation



Generate 7 label images for each training image

- Decay Learning weight(DLW)

$$ep = \frac{current\_epoch}{MAX\_epoch}$$

$$TDLW = \lambda^{SM\times(1-ep)} \otimes \lambda^{ET\times ep} \otimes TM$$

$$KDLW = \lambda^{SM\times(1-ep)} \otimes \lambda^{EK\times ep} \otimes KM$$

The white area is filled with 1, the gray area is filled with 0, the black area is filled with -1, and the shaded area is filled with the weight related to the text width

Loss Function

- Loss function
  - Total loss: $L = L_T + \lambda L_K$
    - *Loss of text area*: $L_T = BCE_T + Dice_T$
      - The cross entropy loss and Dice loss are：

$$BCE_T = \frac{1}{|S_l|}\sum_{i \in S_l}[y_i \log x_i + (1-y_i)\log(1-x_i)] \otimes TDLW$$

$$\text{Dice}_T = 1 - \frac{2|T_{pred} \cap T \cap TDLW|}{|T_{pred} \cap TDLW| + |T \cap TDLW|}$$

Use OHEM to deal with the imbalance of positive and negative samples

    - *Loss of text kernel*: $L_K = BCE_K + Dice_K$
      - The cross entropy loss and Dice loss are：

$$BCE_K = \frac{1}{|km|}\sum_{i \in km}[y_i \log x_i + (1-y_i)\log(1-x_i)] \otimes KDLW$$

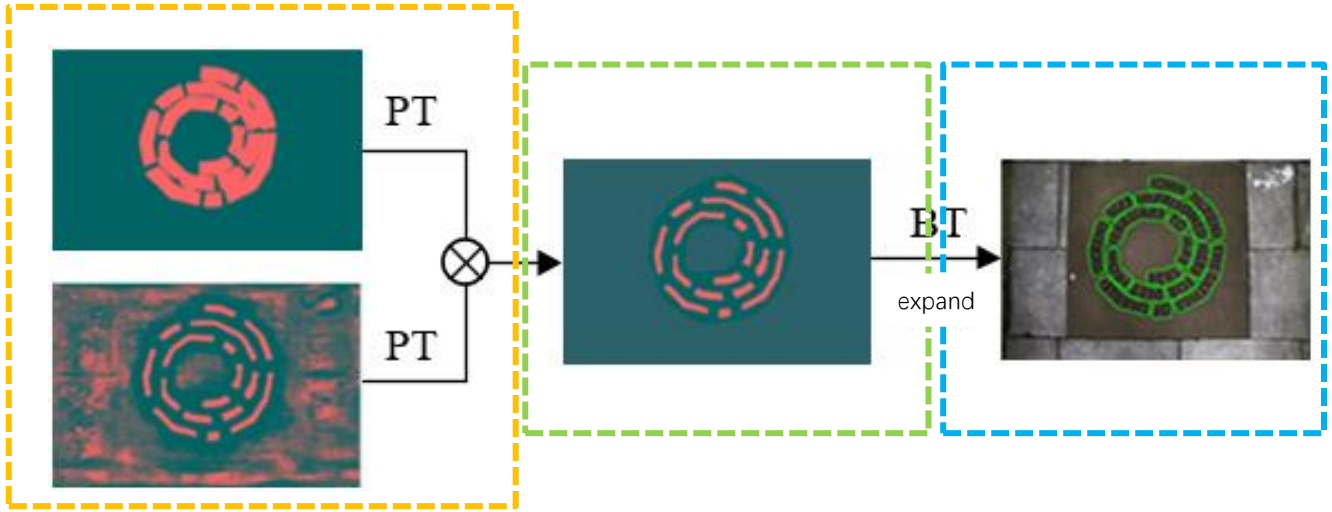
$$Dice_K = 1 - \frac{2|K_{pred} \cap K \cap KDLW|}{|K_{pred} \cap KDLW| + |K \cap KDLW|}$$

Only count the pixels in the text area

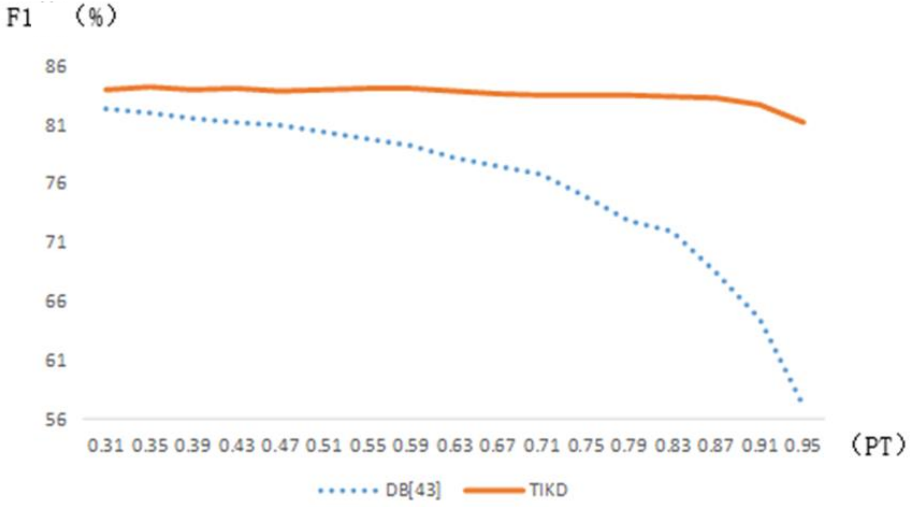
Post-processing

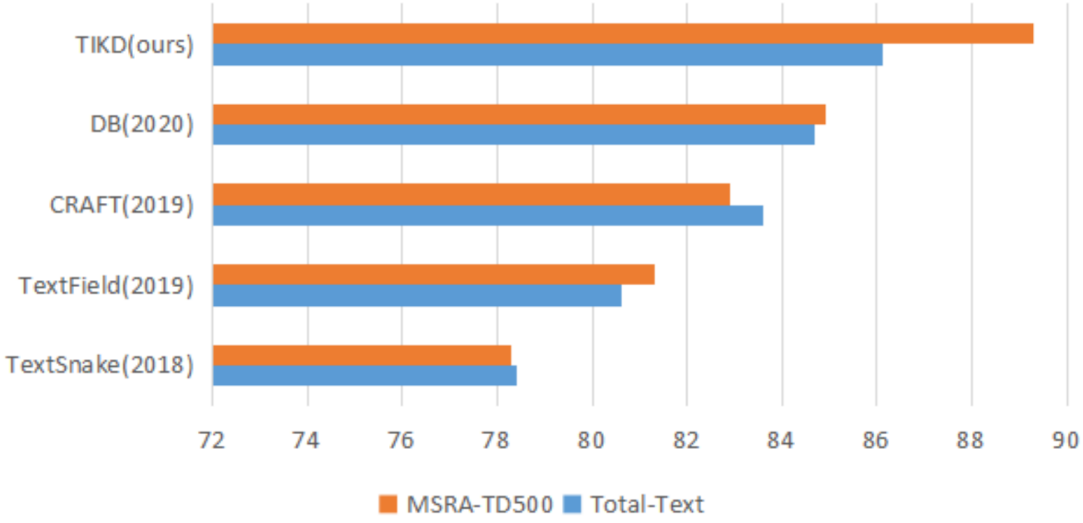
- Obtain the prediction of the text core area
- Calculate the text core score and filter out areas with low scores
- Expand the text core according to a fixed ratio to obtain the result

Key hyperparameters：
Pixel threshold （PT）
Box threshold（BT）



Some examples and results





Method	P	R	F	FPS
TextSnake [32]	82.7	74.5	78.4	1.1
TextField [2]	81.2	79.9	80.6	-
PSENet [1]	84.0	78.0	80.9	3.9
LOMO [34]	88.6	75.7	81.6	-
SPCNet [16]	83.0	82.8	82.9	-
CRAFT [14]	87.6	79.9	83.6	-
DB (800) [3]	87.1	82.5	84.7	**32**
TIKD (736)	**88.7**	**83.7**	**86.1**	16.3