C^2TD: a fast and robust text detector via centre line regression and centre border probability

Jesse wang

AiYunxiao Research

wangjunjie@iyunxiao.com

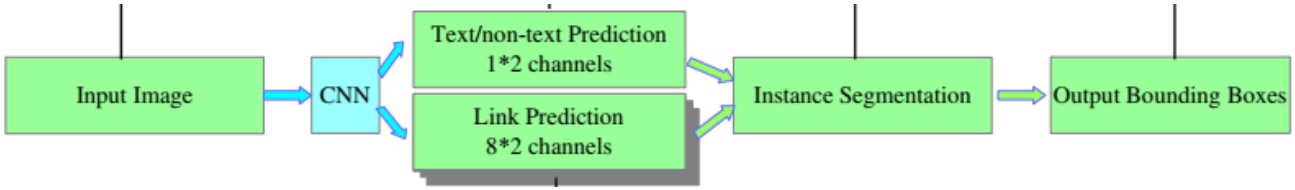
August 8, 2019

Overview

Computation complexity

Heavy Computation hampers real word use

- pixel-link: densely predicting link directions, which leads to a complex post-processing.



- PSENet: predicting socre maps mutiple times.

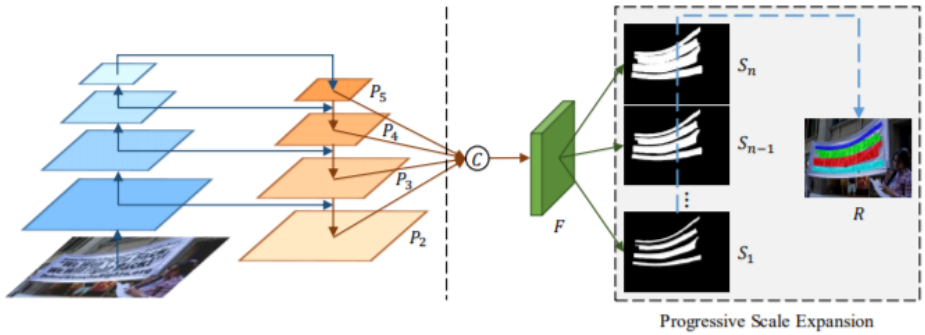


Figure 2: Illustration of our overall pipeline. The left part is implemented from FPN [16]. The right part denotes the feature fusion and the progressive scale expansion algorithm.

Link directions are used to separate different instances. The scale expansion in PSENet has similar effects. Do we have to do it this way?

predicting text line throughout a image

Some methods (TextBox++, East, etc. al) are not suitable for predicting extremely long texts in nature.

- East: A fast anchor-free method.

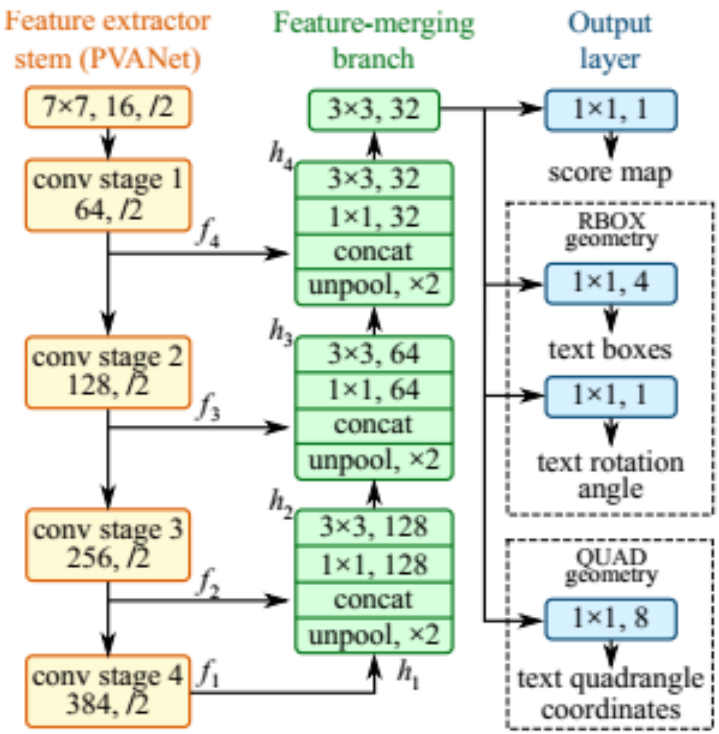
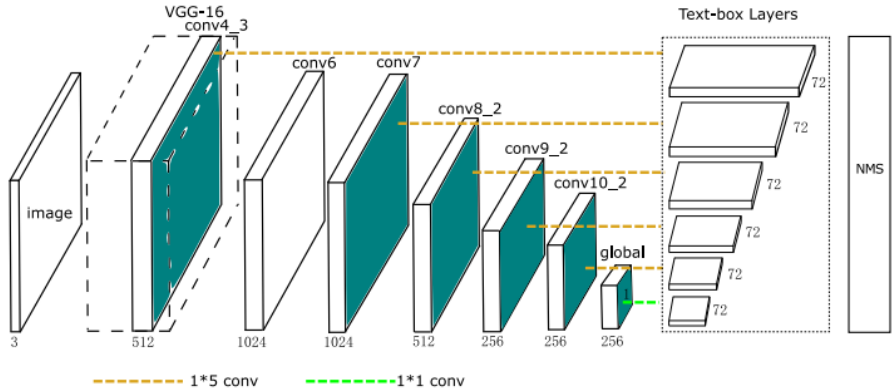


Figure 3. Structure of our text detection FCN.

- Textboxes: modify the shape of conv kernels & anchor boxes



This detection-based methods run fast, but may only suitable for predicting language that can be separate into words easily.

Robust

Only predicting key points(head & tail of a bounding box) is not robust, especially when data contain too much noise.

- Advanced East: predicting two key points. when the key points are not correctly predicted, things get complicated.

What is the problems? give your ideas.

内部像素?	边界像素?	头(or尾)?	左上(右上)X	左上(右上)Y	左下(右下)X	左下(右下)Y

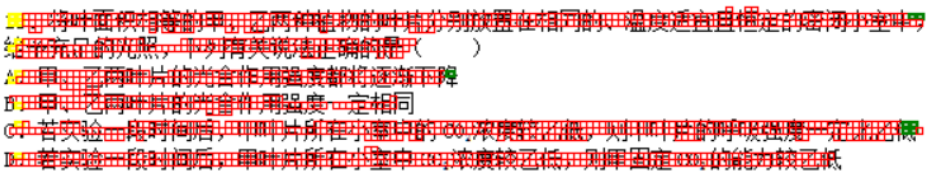
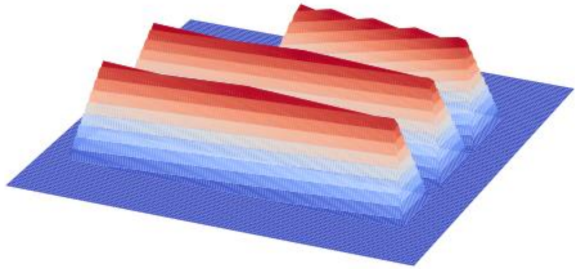


Figure: Activation pixels of advanced East

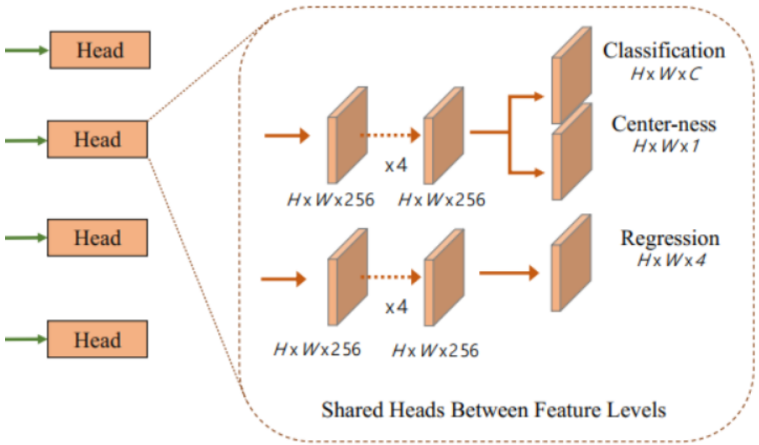
Inspiration: TextMountain & FCOS

- TextMountain: claim that *the mountaintop can separate text instances which cannot be easily achieved using semantic segmentation map*



Center-border Probability

- FCOS: Fully Convolutional One-Stage Object Detection.



$$centerness = \sqrt{\frac{min(l,r)}{max(l,r)} * \frac{min(t,b)}{max(t,b)}} \quad (1)$$

when $l = r$ and $t = b$, the centerness reachs extremum.

FCOS was accepted by CVPR 2019

C^2TD: a fast and robust text detector via centre line regression and centre border probability

Key ideas:

- taking advantage of the idea of TextMountain & FCOS. Predicting the center of a object is enough to separate texts.
- instance segmentation is not accurate and fast as regression methods , especial when we only use 1/4 feature map.

Our Approach

- The FPN network outputs center-border probabilities and the top/bottom offsets at the same time.
- The regression targets are encoded as:

$$t^* = t - y * stride \tag{2}$$

$$b^* = b - y * stride \tag{3}$$

  where t and b are top/bottom coordinates, y is vertical location at the feature map
- opencv is applied to cluster active pixels.

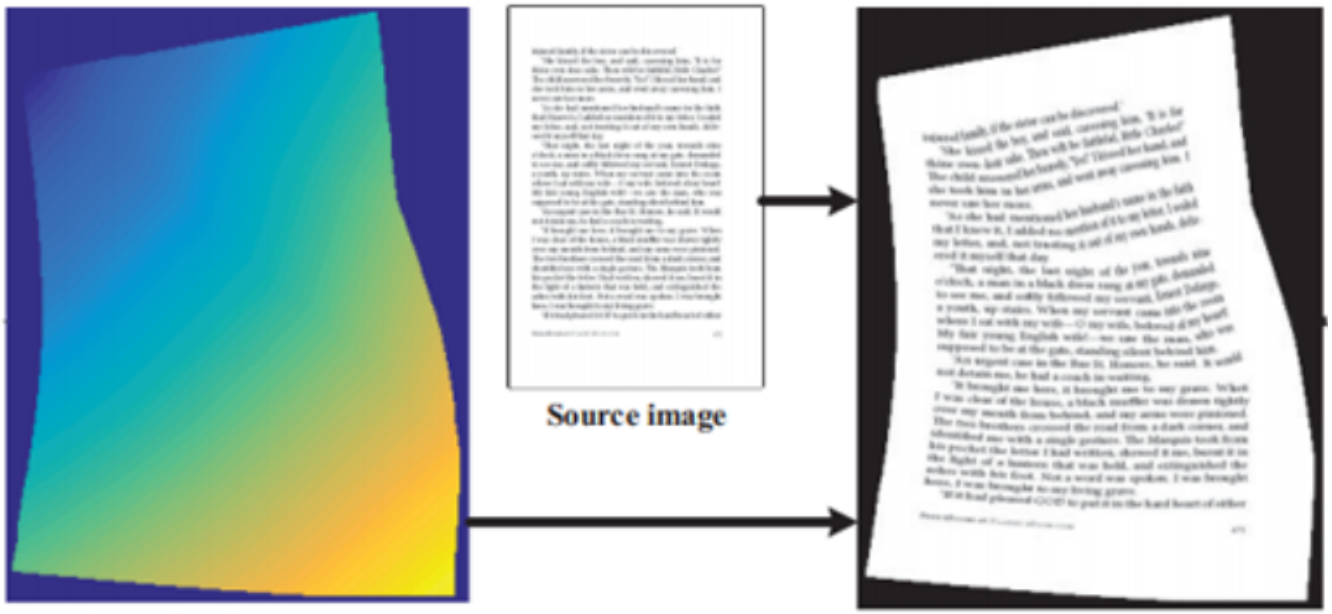
C^2TD: a fast and robust text detector via centre line regression and centre border probability

Advantages:

- Fast: up to 10 fps (end to end tests, including read, write images and skew correction), 38 fps (only C^2TD)
- Owing to the *link fuction* of centre lines, the methods can handle long text
- easy to handle small text lines

C^2TD: a fast and robust text detector via centre line regression and centre border probability

Dewarp Text lines: Understanding Transform Grid



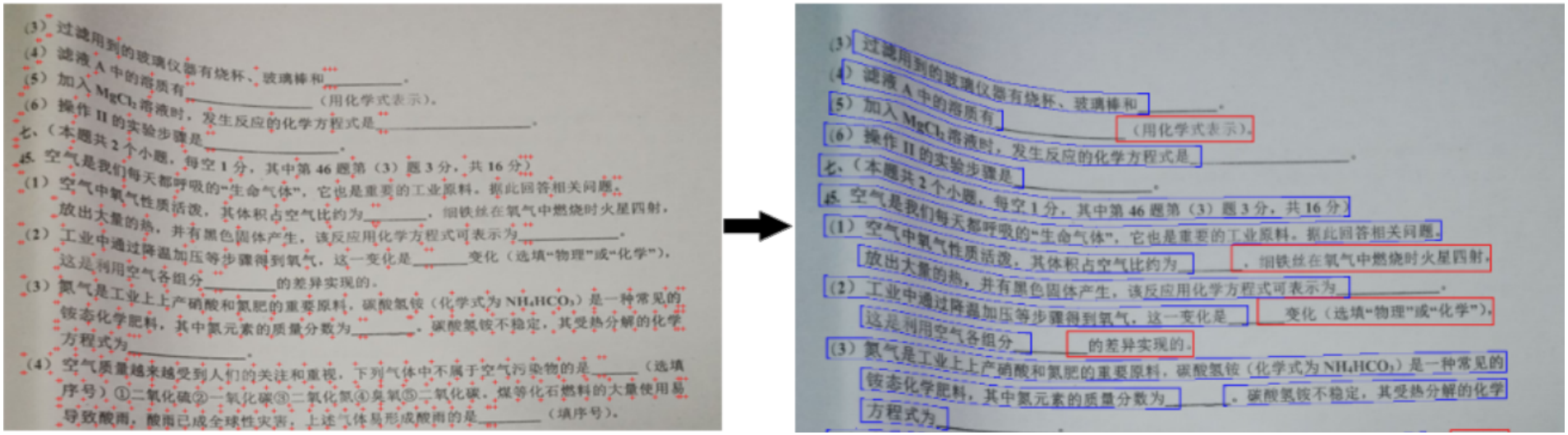
$$F^{-1} = 2 * referGrid - F \tag{4}$$

F^{-1} recorded the locations at warped images.
We use F^{-1} to match the text lines between source images and warped images.

Detecting Text of Arbitrary Shapes

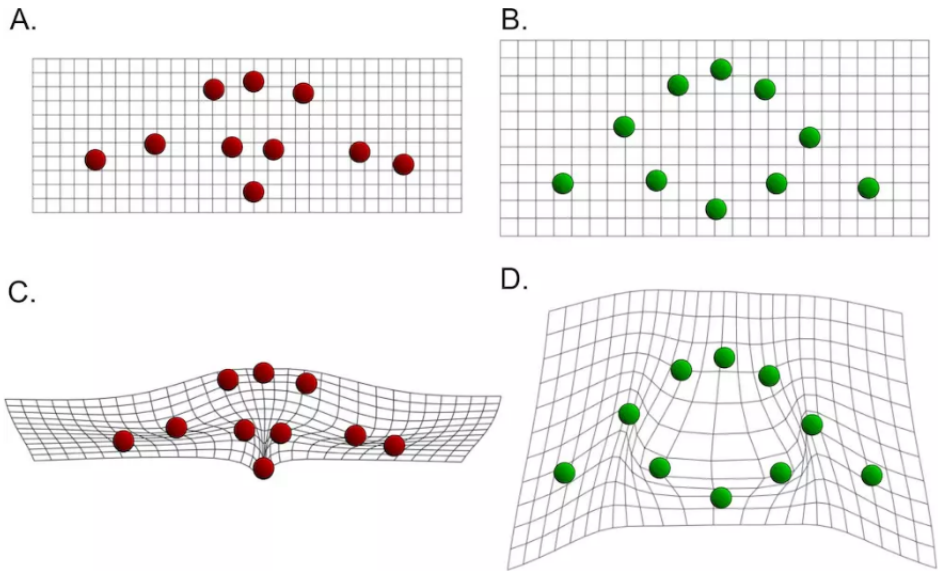
Similar to LOMO

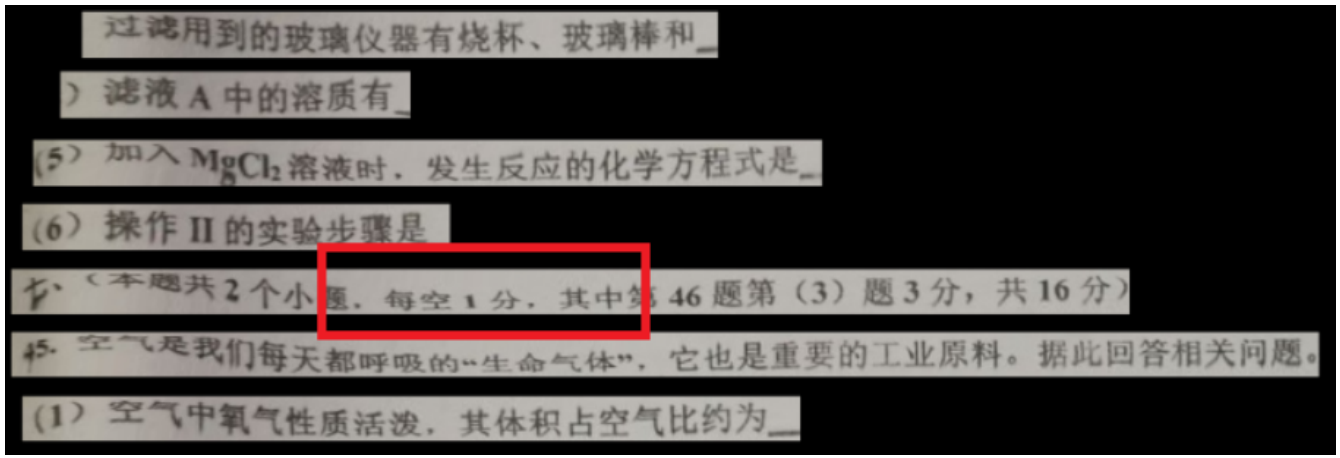
- Dynamic sampling: The sampling rate is related to the height of a text line.
- For a short textline, we treat it as a horizontal line.



Dewarp text lines based on key points

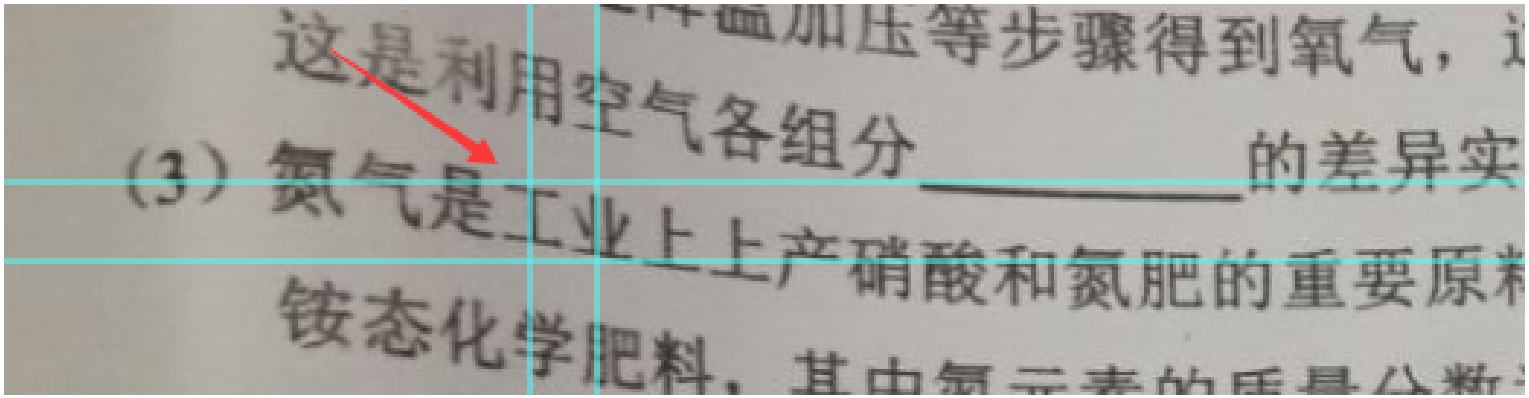
- Thin Plate Spline (TPS): Is TPS transformation too sensitive?



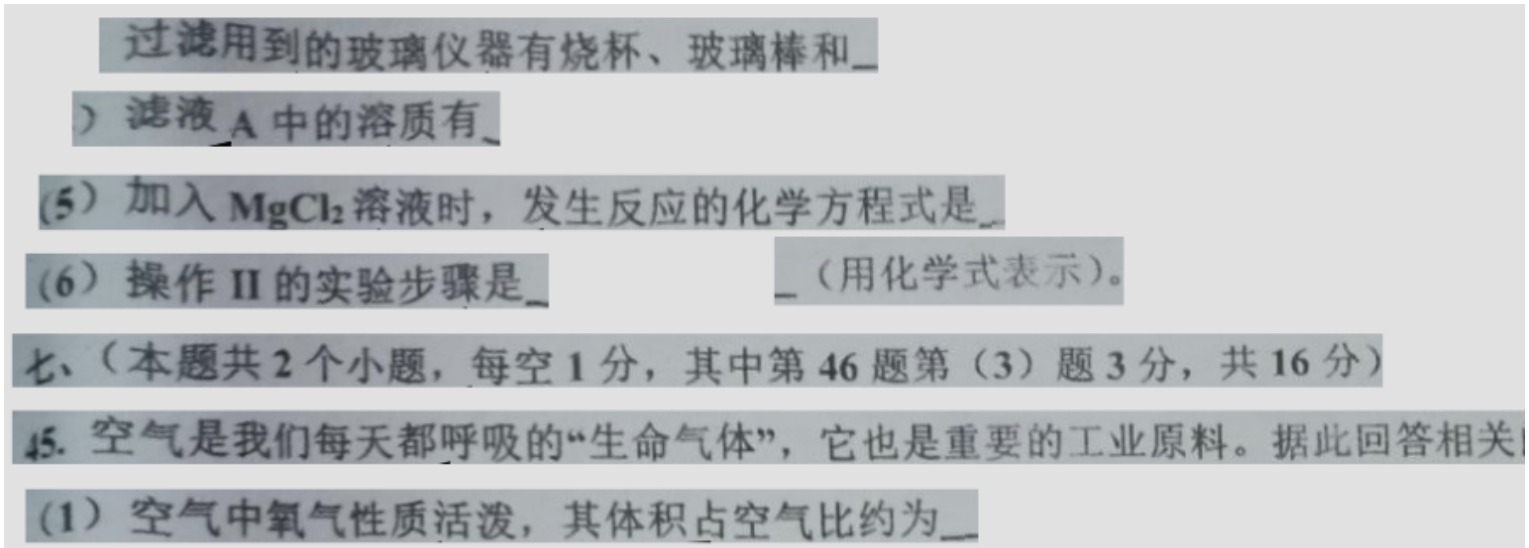


Our solution: Piecewise Affine Transformation

- Parallel lines remain parallel after an affine transformation
- Does paraller lines remain parallel in real wraped images?

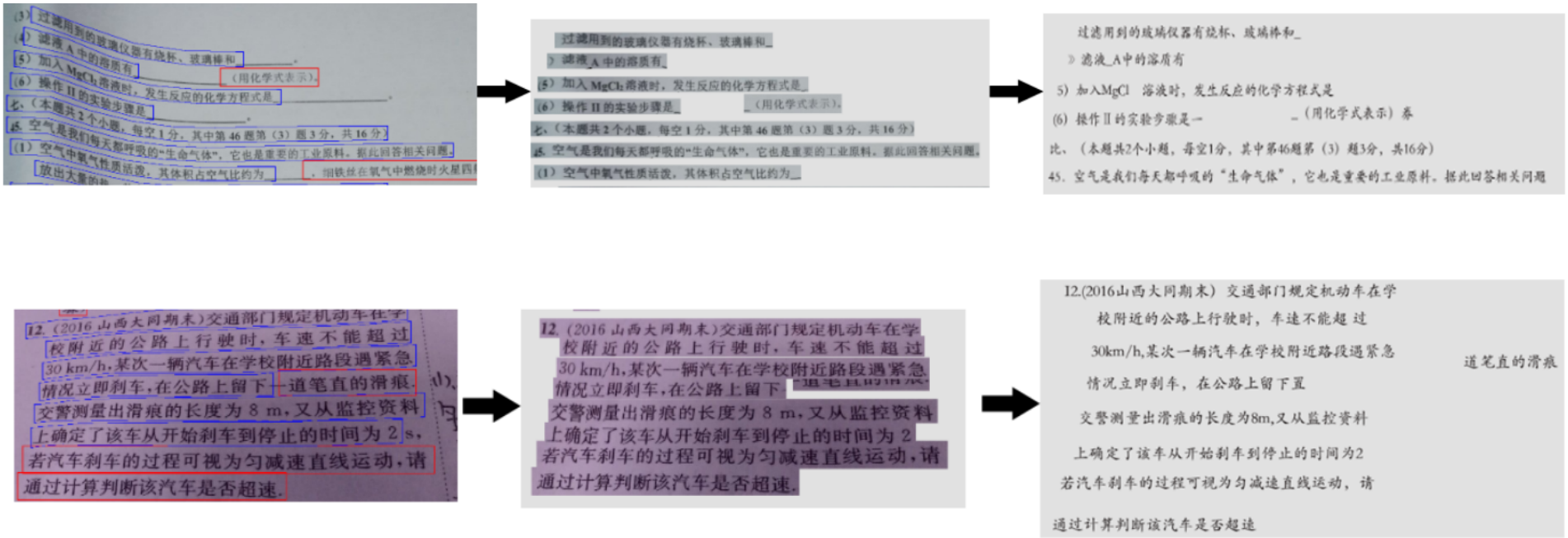


- Piecewise Affine transformation shows a better performance based on our observation.



End to End Recognition

CNN based recognition system show a robust performance towards slightly distorted lines.



How to apply hard mining and weighting?