RLST:A Reinforcement Learning Approach to Scene Text Detection Refinement

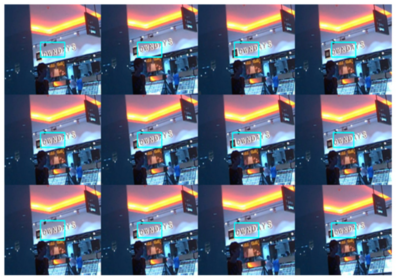
Xuan Peng, Zheng Huang*, Kai Chen, Jie Guo, Weidong Qiu
School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai, China. *Westone Cryptologic Research Center, Beijing, China

Introduction

In this paper, we present a context-attention-based approach consisting of two modules. The first module glances the whole scene of an image and provides suspicious text regions as rough detection results. The second module iteratively refines the rough detected text box using reinforcement learning. Multi-scale features extracted within context will assist the RL agent in selecting the next action to adjust the position, size and angle of an attention. The second module as the core of this paper, is built on a convolutional neural network (CNN), parallelly processing the information from the rough context and attention and subsequently making decisions.

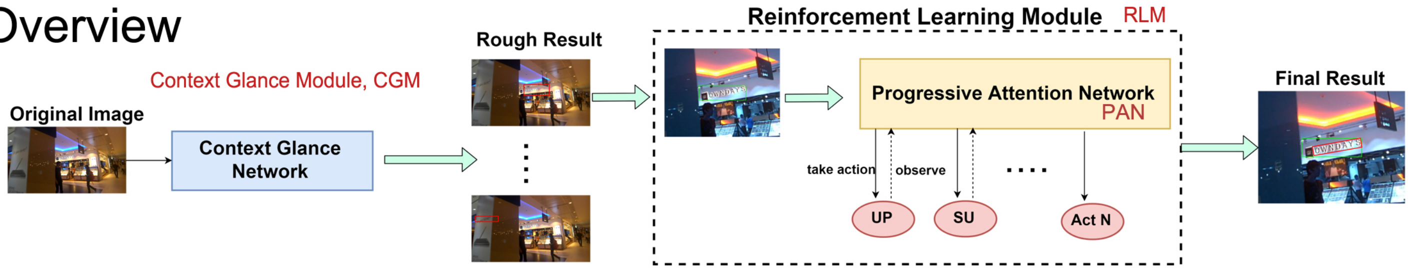
Take action!

Within RLM, PAN takes actions on image and thus slightly changes the context and attention view. Action strategy is changed according to reinforcement learning reward. Multiple scalesof image called auxiliary is used to caption image info on multiple scales while training. In our model,



the state of current RL environment can be represented by context along with the coordinates of attention and auxiliaries. All coordinates will go through the RoI Align layer together with the Context Feature Map to extract features on seven scales.

Overview



During inference, an original image first goes through CGM and there outputs enough rough detected boxes to RLM. The original image will be cropped into fixed-size contexts which shares the same centroid with the rough detected boxes. Attention will be generated as rectangle inside at the center of each context and the size of attention is designed to be as approximate to the rough detected boxes as possible. The core of RLM, a trained Progressive Attention Network (PAN), is capable of selecting the best (or good enough) action to act on the context and attention. By iterating limited steps, the attention at center of a context would finally converge to the border of target text. Following sections will introduce internal structures of CGM and RLM in detail.

Algorithm

Algorithm 1: RLST DQN training

Initialize: Fixed-size Contexts of target image based on distorted ground truth;
Initialize: Attention and Auxiliaries in the center of context;
Initialize: Progressive Attention Network f with random weight parameters W
Initialize: Action set A

```
for epoch = 0, E do
    for i = 0, C where C=number of available contexts do
        while t < T do
            Observe current state: context s_c, attention coordinates s_a and auxiliary coordinates s_aux;
            for ∀a ∈ A do
                Execute action a in emulator and observe reward r_i;
                Revert state;
            end
            Set reward vector R⃗ ← (r_1, r_2, ...r_n);
            Set approximated reward vector R⃗' ← f(s_c, s_a, s_aux, W);
            Perform a gradient descent step on Loss(R⃗', R⃗);
            Select random action a_i ∈ A and execute action a_i in emulator;
            Incr t;
        end
    end
end
```

Determining next action by evaluating current state and the reward caused by last action should be a classification problem. However, it's clear that reward of other actions also implies something important so it would be ideal if PAN can track all action rewards. During the training process, the actual reward can be calculated by ground truth, so reinforcement learning strategy within one step can be transformed into a regression problem. Moreover, RLST is a model used for refinement, i.e., our adjusting area is quite close to the text. Thus, we may assume that at least one action will result in a non-zero reward during each iteration.

The Shape of PAN looks like a lollipop

Performance

Action	Precision	Recall	F1 Score
up	0.56	0.65	0.60
down	0.61	0.62	0.61
left	0.48	0.55	0.51
right	0.5	0.57	0.53
rr	0.26	0.26	0.26
rl	0.24	0.26	0.25
awi	0.68	0.67	0.67
awd	0.51	0.4	0.45
ahi	0.79	0.74	0.76
ahd	0.77	0.84	0.80
su	0.72	0.72	0.72
sd	0.42	0.3	0.35

This table depicts the ability of our model to correctly predict each of the 12 actions.