

---

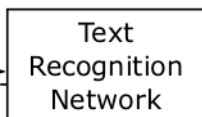
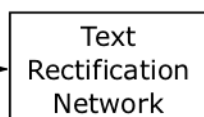
## ASTER: Attentional Scene Text Recognizer with Flexible Rectification

ASTER is an accurate scene text recognizer with flexible rectification mechanism. The research paper can be found [here](#).

Input Image



Rectified Image



→ "Storage"

The implementation of ASTER reuses code from Tensorflow Object Detection API.

### Update

**[07/13/2019] A PyTorch port has been made by @ayumiymk.**

### Correction (10/22/2018)

We have identified a bug we accidentally made in the code that causes only part of SVT images being tested and results in higher results. The bug has been fixed in commit a7e8613. Below are the corrected numbers on SVT. The results are still state-of-the-art, so the conclusions are not affected.

- SVT (50) ASTER: 97.4%; ASTER-A: 96.3%; ASTER-B: 96.1%;
- SVT (None): ASTER: 89.5%; ASTER-A: 80.2%; ASTER-B: 81.6%

### Prerequisites

ASTER was developed and tested with **TensorFlow r1.4**. Higher versions may not work.

ASTER requires Protocol Buffers (version  $\geq 2.6$ ). Besides, in Ubuntu 16.04:

```
1 sudo apt install cmake libcupti-dev
2 pip3 install --user protobuf tqdm numpy editdistance
```

### Installation

1. Go to `c_ops/` and run `build.sh` to build the custom operators
2. Execute `protoc aster/protos/*.proto --python_out=.` to build the protobuf files
3. Add `/path/to/aster` to `PYTHONPATH`, or set this variable for every run

---

## Demo

A demo program is located at `aster/demo.py`, accompanied with pretrained model files available on our release page. Download `model-demo.zip` and extract it under `aster/experiments/demo/` before running the demo.

To run the demo, simply execute:

```
1 python3 aster/demo.py
```

This will output the recognition result of the demo image and the rectified image.

## Training and on-the-fly evaluation

Data preparation scripts for several popular scene text datasets are located under `aster/tools`. See their source code for usage.

To run the example training, execute

```
1 python3 aster/train.py \  
2   --exp_dir experiments/demo \  
3   --num_clones 2
```

Change the configuration in `experiments/aster/trainval.prototxt` to configure your own training process.

During the training, you can run a separate program to repeatedly evaluates the produced checkpoints.

```
1 python3 aster/eval.py \  
2   --exp_dir experiments/demo
```

Evaluation configuration is also in `trainval.prototxt`.

## Citation

If you find this project helpful for your research, please cite the following papers:

```
1 @article{bshi2018aster,  
2   author  = {Baoguang Shi and  
3             Mingkun Yang and  
4             Xinggang Wang and  
5             Pengyuan Lyu and  
6             Cong Yao and  
7             Xiang Bai},
```

---

```
8   title   = {ASTER: An Attentional Scene Text Recognizer with Flexible
9             Rectification},
10  journal = {IEEE Transactions on Pattern Analysis and Machine
11             Intelligence},
12  volume  = {},
13  number  = {},
14  pages   = {1-1},
15  year    = {2018},
16 }
17
18 @inproceedings{ShiWLYB16,
19   author    = {Baoguang Shi and
20               Xinggang Wang and
21               Pengyuan Lyu and
22               Cong Yao and
23               Xiang Bai},
24   title     = {Robust Scene Text Recognition with Automatic
25               Rectification},
26   booktitle = {2016 {IEEE} Conference on Computer Vision and Pattern
27               Recognition,
28               {CVPR} 2016, Las Vegas, NV, USA, June 27-30, 2016},
29   pages     = {4168--4176},
30   year      = {2016}
31 }
```

IMPORTANT NOTICE: Although this software is licensed under MIT, our intention is to make it free for academic research purposes. If you are going to use it in a product, we suggest you contact us regarding possible patent issues.