

H1 Scene text detection with Local Context-Aware Upsampling

Sven Nelson

nelso531

Reimplementing RSCA network (RSCAnet) based on the paper:

Li, Jiachen, Yuan Lin, Rongrong Liu, Chiu Man Ho, and Humphrey Shi. "RSCA: Real-Time Segmentation-Based Context-Aware Scene Text Detection." In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2349-58. Nashville, TN, USA: IEEE, 2021. <https://doi.org/10.1109/CVPRW53098.2021.00267>.

H2 Original code

No code implementation was provided by the author of the paper above.

All code in `training.py` is original code written by me (Sven Nelson).

H2 The implementation

- Implemented in PyTorch using PyTorch Lightning
- Trained on NVIDIA GeForce RTX 2080 SUPER with 8GB GPU memory
- Train dataset split 80:20 for training and validation
- Separate Test dataset of 300 unseen images
- Outputs plot of training and validation loss curves

Parameters

- Batch size: 5 (unable to increase > 5 due to GPU limits)
- SGD with decaying learning rate
 - $lr = \left(1 - \frac{epoch}{max_epoch}\right)^{power}$
 - Weight decay: 0.0001
 - Momentum: 0.9
 - Power: 0.9
- Initial learning rate:
 - Paper used 0.007, but I optimized based on network

H2 Dataset

Trained using TotalText (Ch'ng and Chan 2017) dataset of 1255 train and 300 test image images including curved, angled, and diverse scene text examples

TotalText dataset obtained from official repository:

- <https://github.com/cs-chan/Total-Text-Dataset>

Which links to the dataset via this URL:

- <https://drive.google.com/file/d/1bC68CzsSVTusZVvOkk7imSZSbgD1MqK2/view?usp=sharing>

Subfolder names must be renamed to match the directory structure below for running experiments

H2 Directory structure

The code assumes this directory structure and that the file is run from within the `RSCAnet` folder.

```
.
├── Scripts
│   ├── RSCAnet
│   │   ├── training.py
│   │   ├── README.md
│   │   ├── README.html
│   │   └── README.pdf
│   ├── RSCAnet_checkpoints
│   └── lightning_logs
│       └── version_0 (folder(s) will appear after training)
└── data
    ├── totaltext
    │   ├── Images
    │   │   ├── Train
    │   │   └── Test
    │   └── Text_Region_Mask
    │       ├── Train
    │       └── Test
```

H2 Code execution

1. Navigate to `Scripts/RSCAnet`
2. Make sure all libraries are installed and you are running from a computer with CUDA support for PyTorch.
3. Run any of the following:

```
# Quick test using 1 batch only to check everything is working
> python training.py --quick_test
# Run full training and testing with LCAU blocks
> python training.py
# Run with network using transpose convolutions in place of LCAU
> python training.py --no_lcau
```

4. While training, TensorBoard can be loaded using VScode or from the terminal and pointed to the `lightning_logs` folder to view live loss curves and loss curves from past runs.
5. A folder will be generated within the `lightning_logs` folder with a name `version_0` (the zero will increment as you train more times). This folder will contain logs, the saved model, and a plot of training vs validation loss named `train-val_loss_plot.jpg`

H2 Reproducing experiments

1. Lines 436 - 474 of `training.py` (reproduced below for reference) contain parameters for training. To adjust any of these parameters, change the value and resave the file. This can be used to repeat the initial optimization of parameters.
 - Adjust learning rate, epochs, or batch size
 - Run training for $RSCA_{LCAU}$ by running `python training.py`

```

#### Set parameters for training and validation ####
#bs = 16 # Same as used in the paper
bs = 5 # Had to reduce due to out of memory errors with bs = 6 or
greater
workers = 4
epochs = 50
#learning_rate = 0.007 # Initial learning rate from paper
learning_rate = 1e-5 # Lower learning rate (got better results than
initial from paper)
# Set path to data directory (where folders of images have been
downloaded to)
data = "../..data/totaltext/"
# Set random seed for consistency in experimentation
randomseed = 42
#####

```

2. To reproduce training comparison for 30 epochs:

- Change epochs to 30 on line 467 and save `training.py`
- Run training for $RSCA_{LCAU}$ by running `python training.py`
- Run training for $RSCA_{TC}$ by running `python training.py --no_lcau`

3. To reproduce training comparison for 50 epochs:

- Change epochs to 50 on line 467 and save `training.py`
- Run training for $RSCA_{LCAU}$ by running `python training.py`
- Run training for $RSCA_{TC}$ by running `python training.py --no_lcau`