Representation in Scene Text Detection and Recognition

Prof. Xiang Bai

Huazhong University of Science and Technology

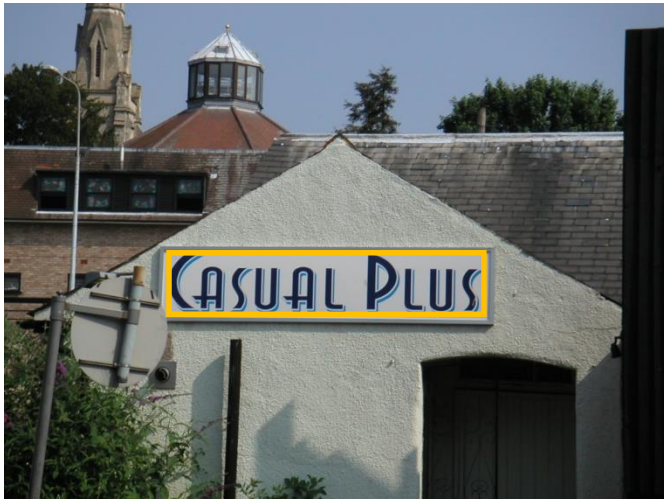
Contents

- Problem definition
- Significance and challenges
- Previous works
- Our algorithms
- Conclusion

Contents

- Problem definition
- Significance and challenges
- Previous works
- Our algorithms
- Conclusion

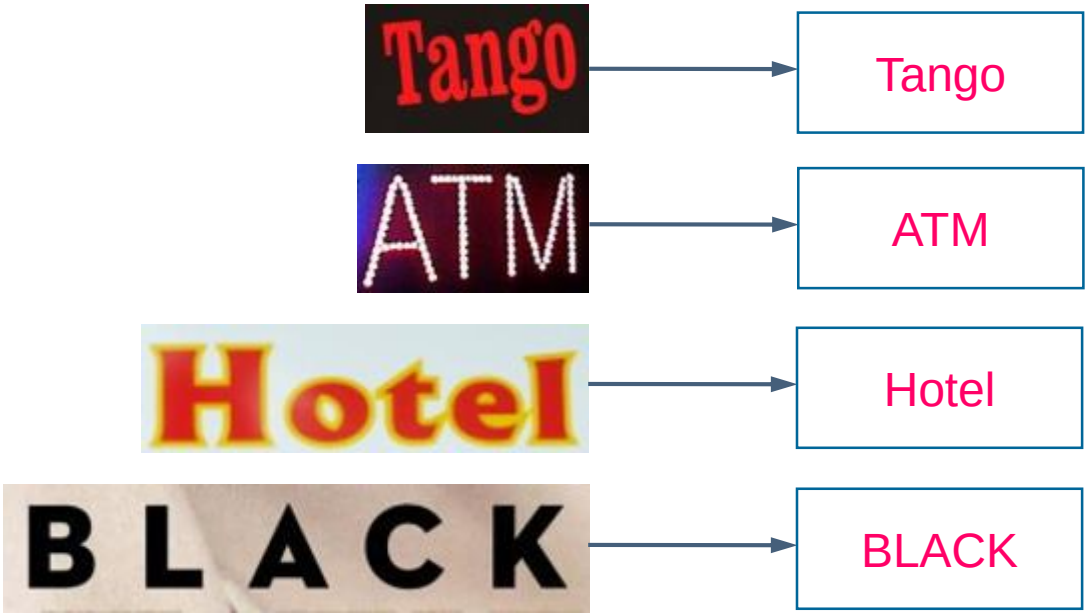
Problem definition



Scene text detection:

the process of predicting the presence of text and localizing each instance (if any), usually at word or line level, in natural scenes

Problem definition



Scene text recognition:

the process of converting text regions into computer readable and editable symbols

Contents

- Problem definition
- Significance and challenges
- Previous works
- Our algorithms
- Conclusion

Significance

- text in natural scenes carries rich and precise high level semantics
- text information can be useful to a variety of applications:
  scene understanding, product search, HCI, virtual reality…

challenges

Diversity of scene text:

different colors, scales, orientations, fonts, languages…

challenges

Complexity of background:

elements like signs, fences, bricks, and grasses are virtually undistinguishable from true text

challenges

Various interference factors:

noise, blur, non-uniform illumination, low resolution, partial occlusion…

challenges

These challenges make
scene text detection and recognition
extremely difficult problems

Contents

- Problem definition
- Significance and challenges
- Previous works
- Our algorithms
- Conclusion

Previous works

Three categories:

1. text detection

only localize text regions, no need to recognize the content

2. text recognition

only recognize the content, assume text regions are given

3. end-to-end text recognition

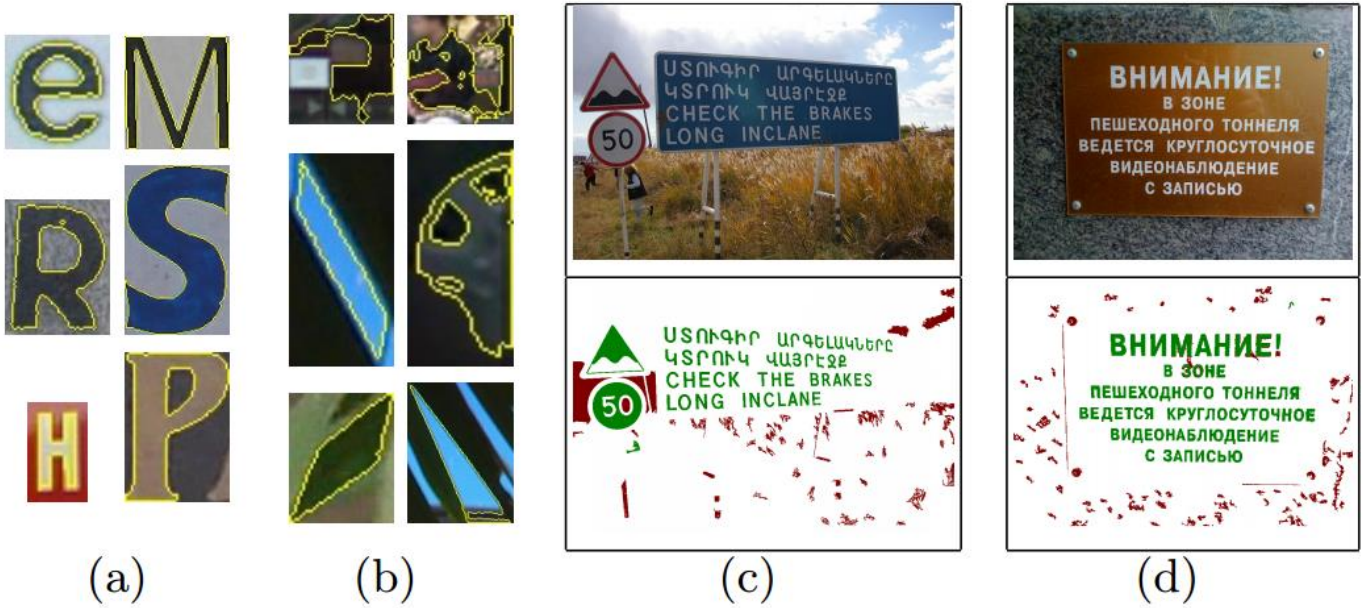
perform both text detection and recognition

Previous works

In the following slides, we will review a number of previous algorithms. We pay more attention to the deep learning based methods

Text Detection

MSER

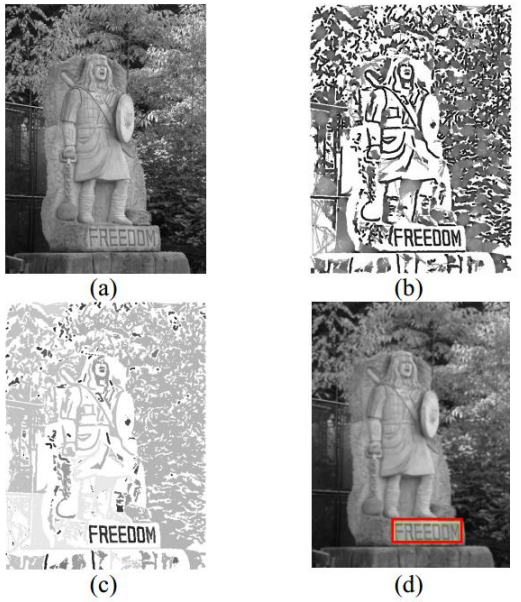


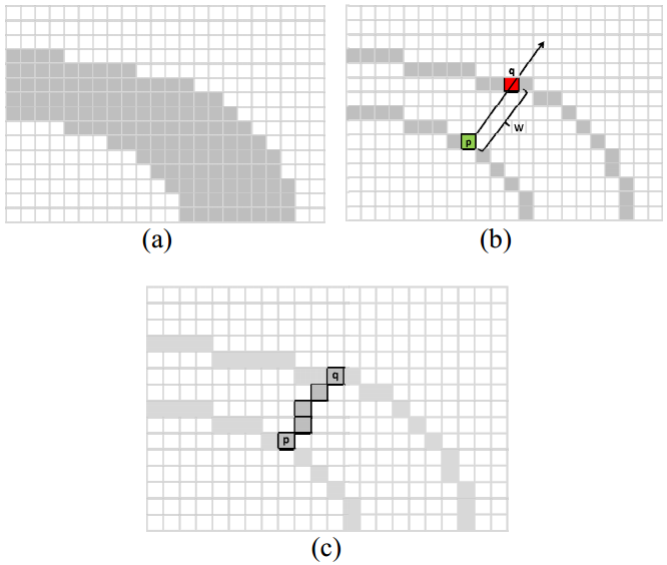
[Neumann and Matas, ACCV 2010]

- extract character candidates using Maximally Stable Extremal Regions, assuming similar color within each character
- robust, fast to compute, independent of scale and orientation

Text Detection

SWT





[Epshtein et al., CVPR 2010]

- extract character candidates with Stroke Width Transform, assuming consistent stroke width within each character
- robust, fast to compute, independent of scale and orientation

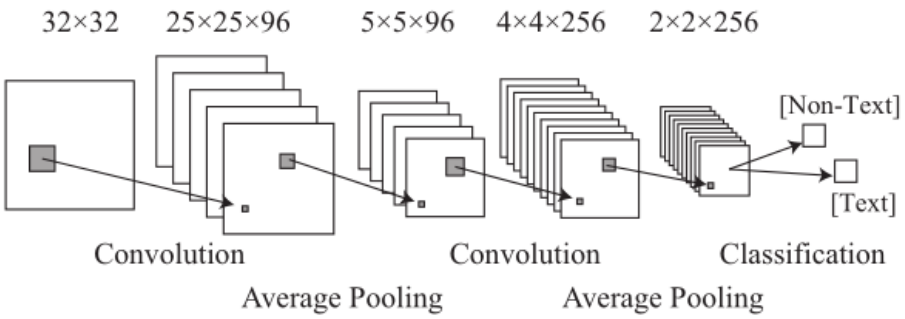
Text Detection

MSER and SWT are representative methods in scene text detection, which constitute the basis of a lot of subsequent works

[Chen et al., ICIP 2011],
[Yao et al., CVPR 2012],
[Neumann and Matas, CVPR 2012],
[Novikova et al., ECCV 2012],
[Huang et al., ICCV 2013],
[Yinet al., SIGIR 2013],
[Koo et al., TIP 2013],
[Yin et al., TPAMI 2014],
[Yao et al., TIP 2014],
[Huang et al., ECCV 2014],
.....

Text Detection

CNN for character detection

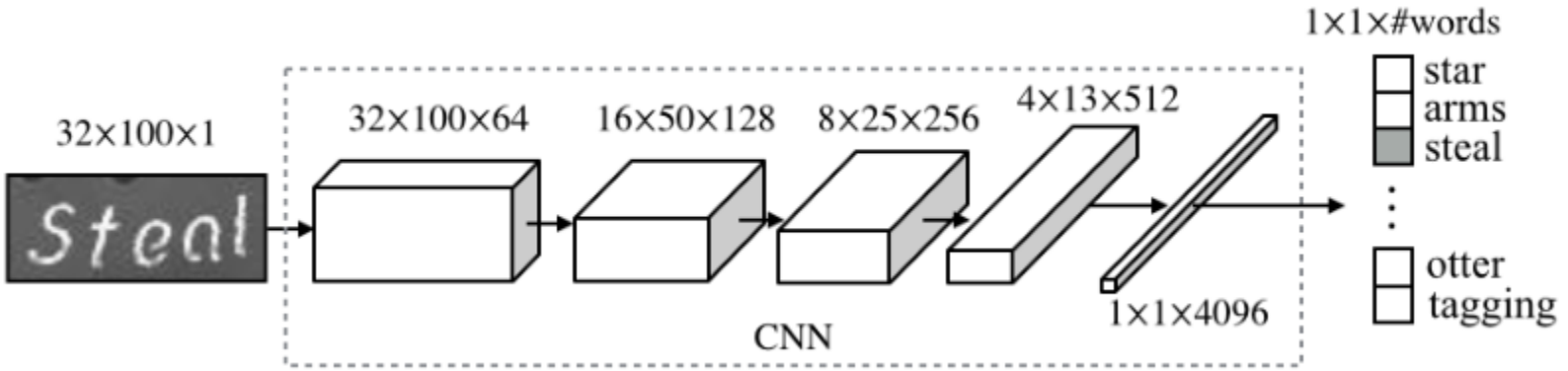


[Tao Wang et al., ICPR 2012]

- Use CNN as the character detector (binary classifier)
- Highly-accurate and robust deep model

Text Recognition

90k-way word classification CNN



[Max Jaderberg et al., NIPSW 2014]

- Recognize English words by direct CNN classification
- 90k-way CNN trained with an incremental training strategy
- Synthesize 9 million word images for training the model

End-to-End Text Recognition

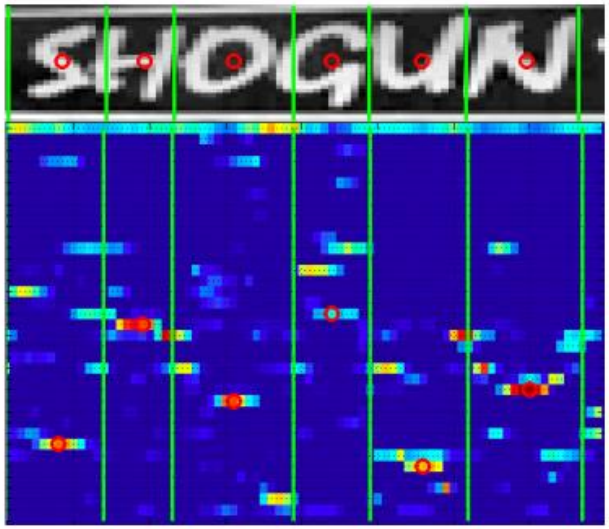
PhotoOCR

[Bissacco et al., ICCV 2013]

- localize text regions by integrating multiple existing detection methods
- recognize characters with a DNN running on HOG features, instead of raw pixels
- use 2.2 million manually labelled examples for training

End-to-End Text Recognition

Deep Features

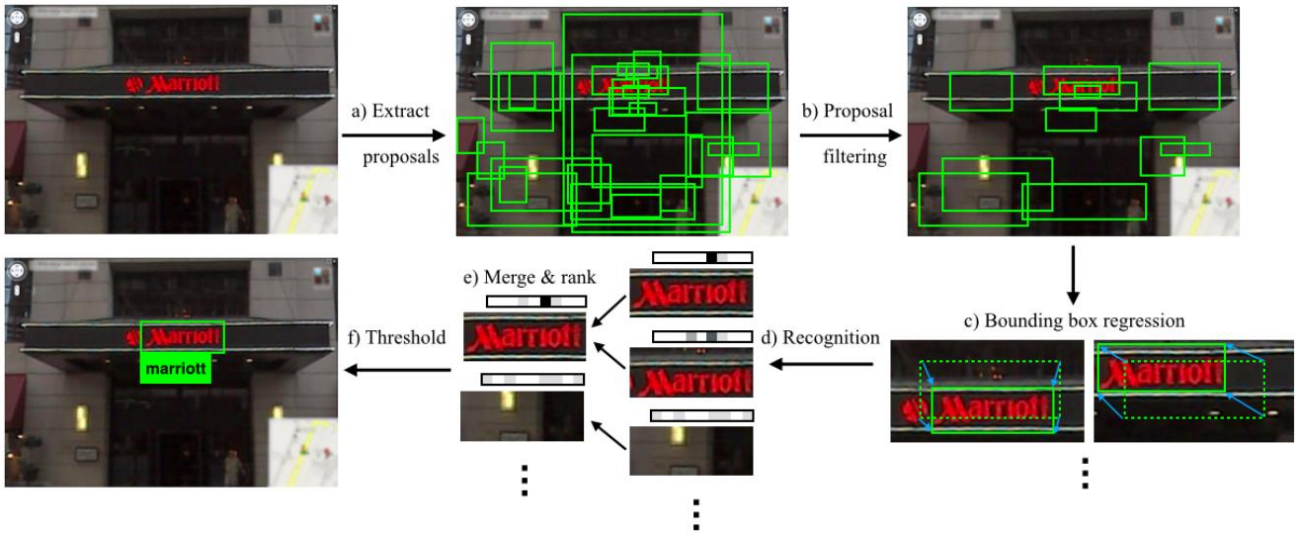


[Jaderberg et al., ECCV 2014]

- propose a novel CNN architecture, enabling efficient feature sharing for text detection and character classification
- generate word and character level annotations via automatic data mining of Flickr

End-to-End Text Recognition

Scene text reading in the R-CNN way



[Max Jaderberg et al., IJCV 2015]

- Proposals: combine the results of EdgeBox and ACF
- Filtering: HOG + random forest
- Bounding box regression: CNN regression
- Recognition: 90k-way CNN classification

End-to-End Text Recognition

Deep learning + Big data
seem to dominate this field

For more detais:
[1] Y. Zhu, C. Yao, and X. Bai, Scene Text Detection and Recognition: Recent Advances and Future Trends, Frontier of Computer Science, to appear.

Contents

- Problem definition
- Significance and challenges
- Previous works
- Our algorithms
- Conclusion

Our algorithms

We will introduce some of our works
that propose novel representations
for better text detection and recognition

Symmetry-Based Text Line Detection in Natural Scenes

- Text lines always bear distinctive symmetry and self-similarity properties. By considering these properties, we could find text region without seeking for individual characters.

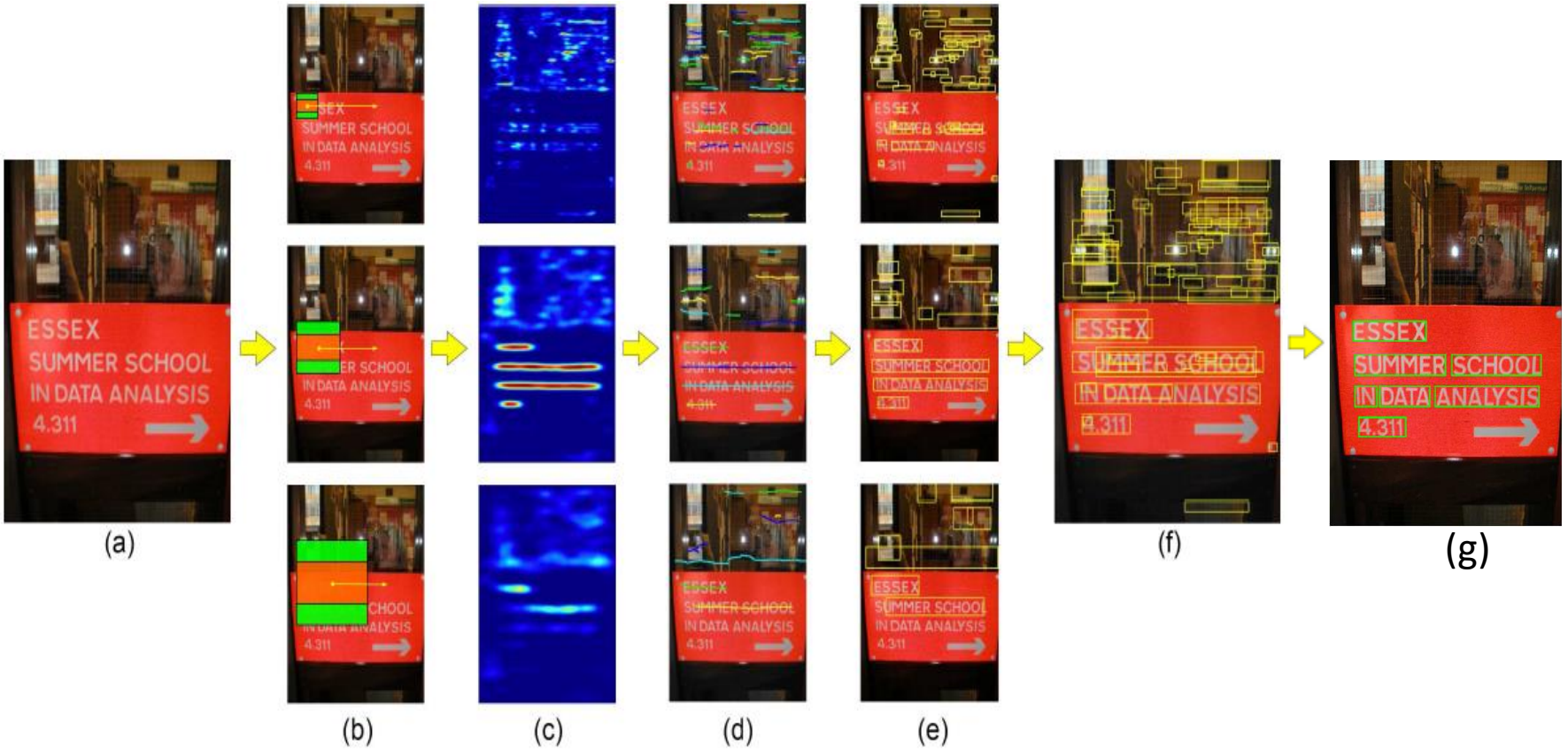
[1] Zheng Zhang, Wei Shen, Cong Yao, Xiang Bai. Symmetry-based Text Line Detection in Natural Scenes, IEEE CVPR, 2015.

Overview of the proposed methodology

1.Feature extraction at multiple scales.
2.Symmetry probability estimation.
3.Axes sought in the symmetry probability maps.
4.Bounding box estimation and proposals generation.
5.False positive removal and word partition

Symmetry-Based Text Line Detection in Natural Scenes

Overview of the proposed methodology

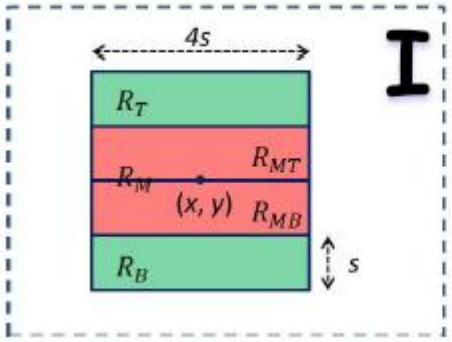


Feature Extraction and Symmetry probability estimation

1.Symmetry feature
2.Appearance Feature (LBP)
3.Probability estimation by Random Forest at Multiple scales

Symmetry-Based Text Line Detection in Natural Scenes

Symmetry feature



- Self-Similarity

$$S^c_{x,y} = \chi^2(h^c_{x,y}(R_{MT}), h^c_{x,y}(R_{MB}))$$

- Disimilarity

$$Ct^c_{x,y} = \chi^2(h^c_{x,y}(R_T), h^c_{x,y}(R_{MT}))$$

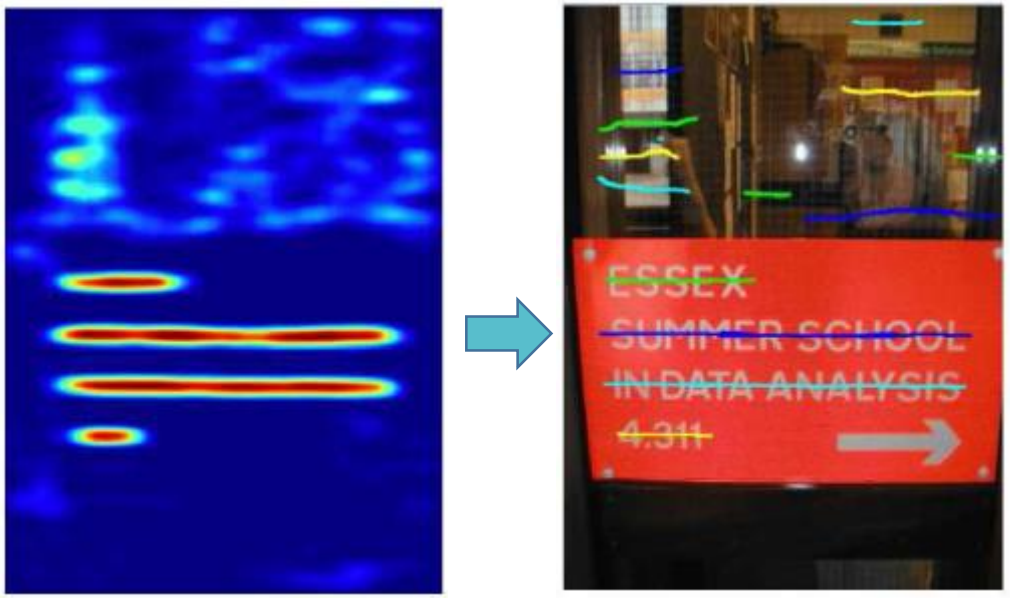
$$Cb^c_{x,y} = \chi^2(h^c_{x,y}(R_B), h^c_{x,y}(R_{MB}))$$

- Calculation at LAB, Gradient and Textons channels

Symmetry-Based Text Line Detection in Natural Scenes

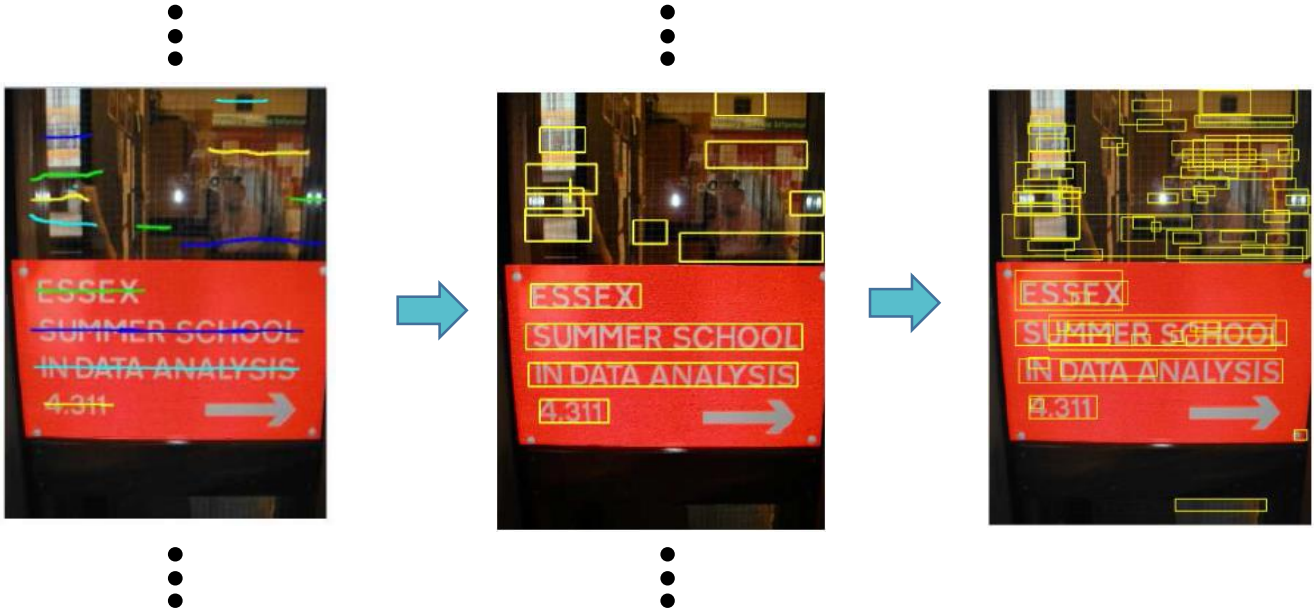
Axes sought in the symmetry probability maps

1. Non-Maximum Suppression
2. Axes linking
*Angular Difference Constraint
*Distance Constraint
3. Above two steps are applied at each scale respectively



Symmetry-Based Text Line Detection in Natural Scenes

Bounding box estimation and proposals generation



Symmetry-Based Text Line Detection in Natural Scenes

False positive removal and word partition

1.Character level CNN classifier(Text Spotting, ECCV2014, Zisserman)
  * Word partition
  * Preliminary false positive removal
2.Textline level CNN classifier for further filter

Symmetry-Based Text Line Detection in Natural Scenes

Experimental result

ICDAR 2011

Algorithm	Precision	Recall	F-measure
Proposed	0.84	**0.76**	**0.80**
Huang *et al.* [9]	**0.88**	0.71	0.78
Yin *et al.* [40]	0.863	0.683	0.762
Koo *et al.* [13]	0.814	0.687	0.745
Yao *et al.* [35]	0.822	0.657	0.730
Huang *et al.* [8]	0.82	0.75	0.73
Neumann *et al.* [24]	0.793	0.664	0.723
Shi *et al.* [29]	0.833	0.631	0.718
Kim *et al.* [28]	0.830	0.625	0.713
Neumann *et al.* [23]	0.731	0.647	0.687
Yi *et al.* [38]	0.672	0.581	0.623
Yang *et al.* [28]	0.670	0.577	0.620
Neumann *et al.* [28]	0.689	0.525	0.596
Shao *et al.* [28]	0.635	0.535	0.581

ICDAR 2013

Algorithm	Precision	Recall	F-measure
Proposed	**0.88**	**0.74**	**0.80**
iwrr2014 [41]	0.86	0.70	0.77
USTB TexStar [40]	**0.88**	0.66	0.76
Text Spotter [23]	**0.88**	0.65	0.74
CASIA_NLPR [1]	0.79	0.68	0.73
Text_Detector_CASIA [29]	0.85	0.63	0.72
I2R_NUS_FAR [1]	0.75	0.69	0.72
I2R_NUS [1]	0.73	0.66	0.69
TH-TextLoc [1]	70	0.65	0.67

Symmetry-Based Text Line Detection in Natural Scenes

Contributions of different types of feature

Feature	Precision	Recall	F-measure
symmetry	0.80	0.65	0.72
appearance	0.79	0.57	0.66
symmetry+appearance	0.84	0.76	0.80

Character detection rates of different methods on the ICDAR 2013 dataset

Algorithm	Detection Rate	Proposal Number
Proposed	**0.977**	**1310**
MSER (Gray+LUV)	0.964	8415

Symmetry-Based Text Line Detection in Natural Scenes

Examples

Symmetry-Based Text Line Detection in Natural Scenes

Limitations

1.Distinguish ability of features is not good enough(especially appearance feature).
2.Axes sought is not robust enough in street view dataset.
3.High time consumption

Future works

1. To explore better feature representation
2. To explore better axes sought method.
3. To expand our works to multi orientations text detection.

CRNN: End-to-End Trainable Network for Scene Text Recognition

Overview

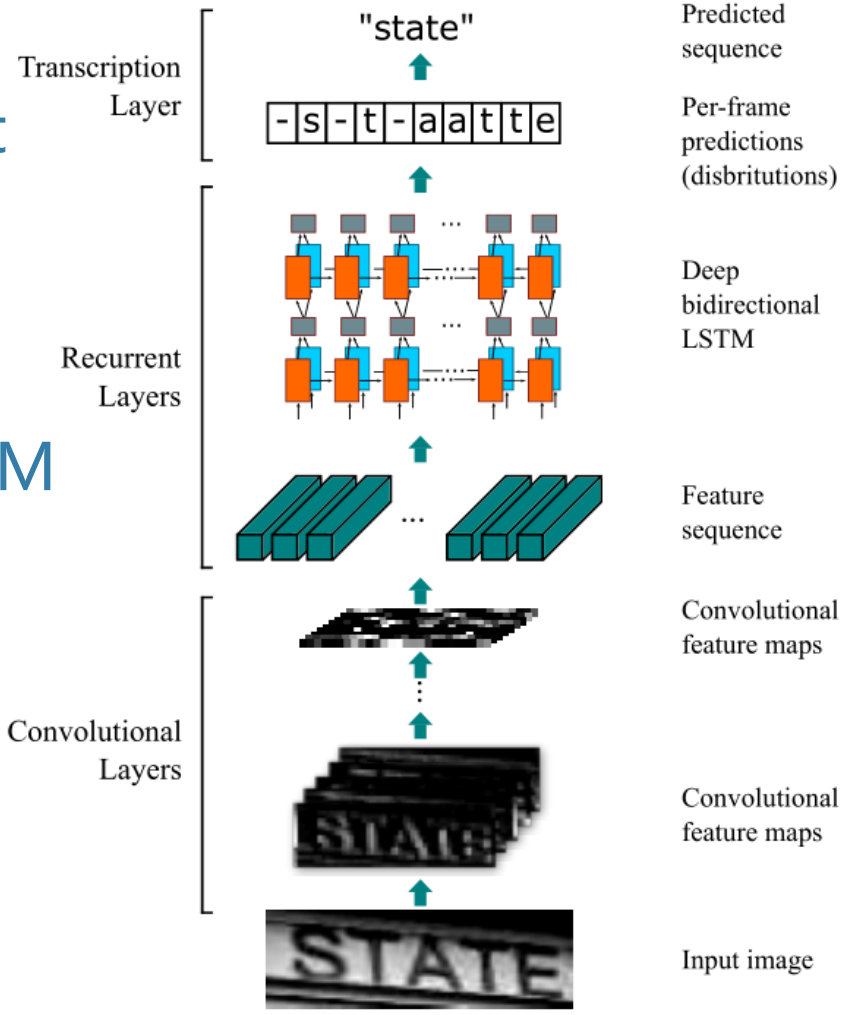
- Text recognition as image-based sequence recognition problem
- Recognition within one network
- Directly trained from images and text strings (end-to-end)

[1] Baoguang Shi, Xiang Bai, Cong Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. arxiv.org/abs/1507.05717, 2015

CRNN: End-to-End Trainable Network for Scene Text Recognition

Network Structure

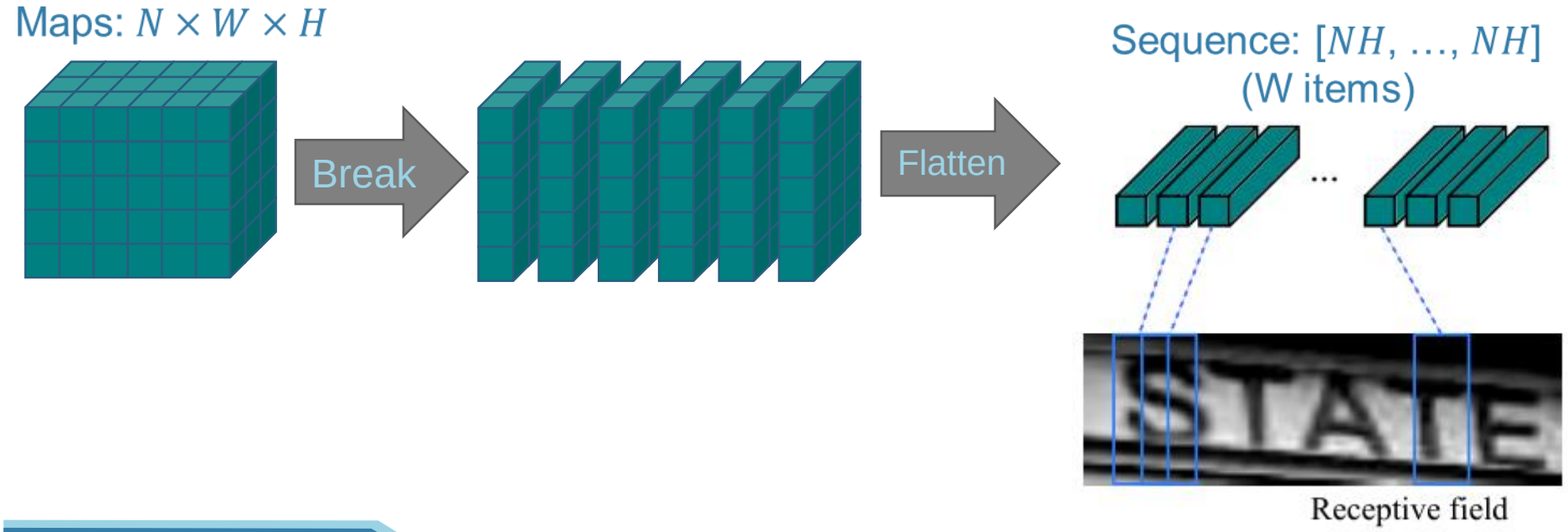
- Convolutional layers extract feature maps
- Convert feature maps into feature sequence
- Sequence labeling with LSTM
- Convert labeling into text



CRNN: End-to-End Trainable Network for Scene Text Recognition

Convolutional Layers

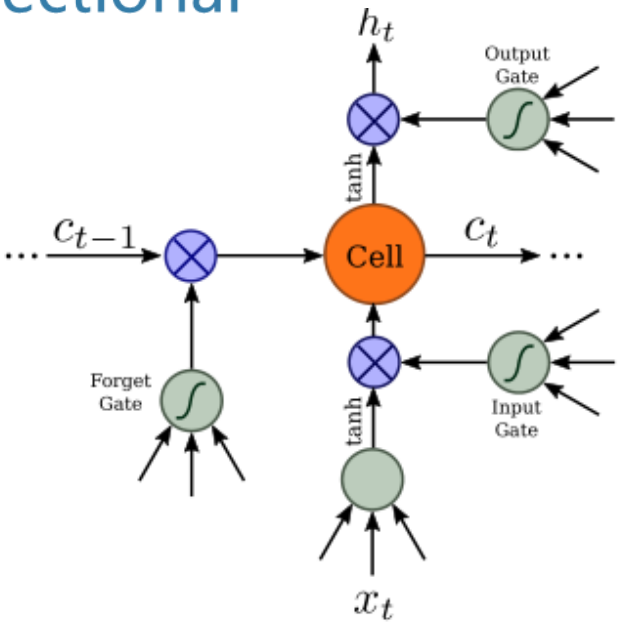
- Extracts feature maps
- Feature maps are converted into a sequence, resulting in a sequential representation

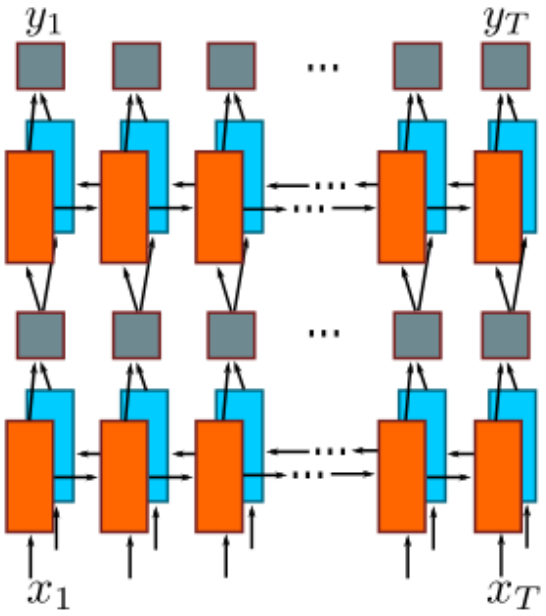


CRNN: End-to-End Trainable Network for Scene Text Recognition

Recurrent Layers (Deep Bidirectional LSTM)

- Predict a distribution over {a-z, 0-9} ∪ {blank} for each frame in the sequence
- Utilize context of nearby frames
- Deep and bidirectional





CRNN: End-to-End Trainable Network for Scene Text Recognition

Transcription Layer (CTC layer)

- Output the conditional probability of a text string. Calculated by the forward-backward algorithm.
- Lexicon-free transcription: naïve CTC decoding ("--hh-e-l-ll-oo--" → "hello")
- Lexicon-based transcription: choose the word with the highest probability in the dictionary
- In case of large lexicon: use lexicon-free transcription to find l'. Choose the best among its neighbors (edit-distance metric), found by a BK-tree.

$$\mathbf{l}^* = \arg \max_{\mathbf{l} \in \mathcal{N}_\delta(\mathbf{l}')} p(\mathbf{l}|\mathbf{y})$$

[1] Alex Graves, Santiago Fernández, Faustino J. Gomez, Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, ICML, 2006.

CRNN: End-to-End Trainable Network for Scene Text Recognition

Network Training

- Back-Propagation Through Time (BPTT) in recurrent layers
- Back-Propagation (BP) in convolutional layers
- Random initialization, ADADELTA optimization, batch normalization layers
- Trained on 8 millions purely synthesized samples (released by Jaderberg et al.)
- ~2 days on single Tesla K40 GPU

CRNN: End-to-End Trainable Network for Scene Text Recognition

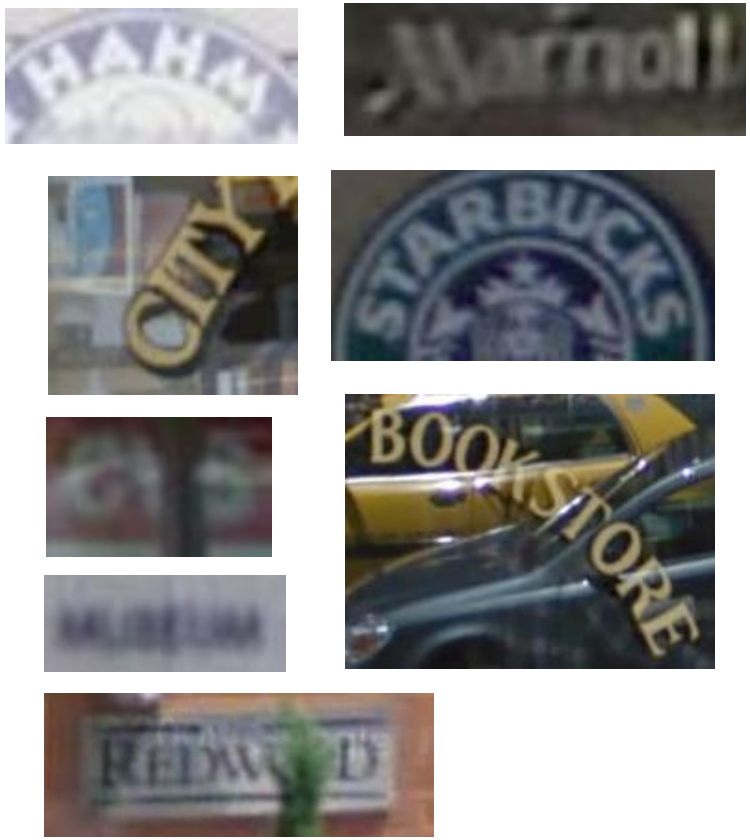
Recognition Performance

	IIIT5k			SVT		IC03				IC13
	50	1k	None	50	None	50	Full	50k	None	None
ABBYY [34]	24.3	-	-	35.0	-	56.0	55.0	-	-	-
Wang *et al.* [34]	-	-	-	57.0	-	76.0	62.0	-	-	-
Mishra *et al.* [28]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-
Wang *et al.* [35]	-	-	-	70.0	-	90.0	84.0	-	-	-
Goel *et al.* [13]	-	-	-	77.3	-	89.7	-	-	-	-
Bissacco *et al.* [8]	-	-	-	90.4	78.0	-	-	-	-	87.6
Alsharif and Pineau [6]	-	-	-	74.3	-	93.1	88.6	85.1	-	-
Almazán *et al.* [5]	91.2	82.1	-	89.2	-	-	-	-	-	-
Yao *et al.* [36]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-
Rodrguez-Serrano *et al.* [30]	76.1	57.4	-	70.0	-	-	-	-	-	-
Jaderberg *et al.* [23]	-	-	-	86.1	-	96.2	91.5	-	-	-
Su and Lu [33]	-	-	-	83.0	-	92.0	82.0	-	-	-
Gordo [14]	93.3	86.6	-	91.8	-	-	-	-	-	-
Jaderberg *et al.* [22]	97.1	92.7	-	95.4	80.7*	**98.7**	**98.6**	93.3	**93.1***	**90.8***
Jaderberg *et al.* [21]	95.5	89.6	-	93.2	71.7	97.8	97.0	93.4	89.6	81.8
CRNN	**97.6**	**94.4**	**78.2**	**96.4**	**80.8**	**98.7**	97.6	**95.5**	89.4	86.7

CRNN: End-to-End Trainable Network for Scene Text Recognition

Correctly Recognized

Failure Cases



CRNN: End-to-End Trainable Network for Scene Text Recognition

Compare with other methods

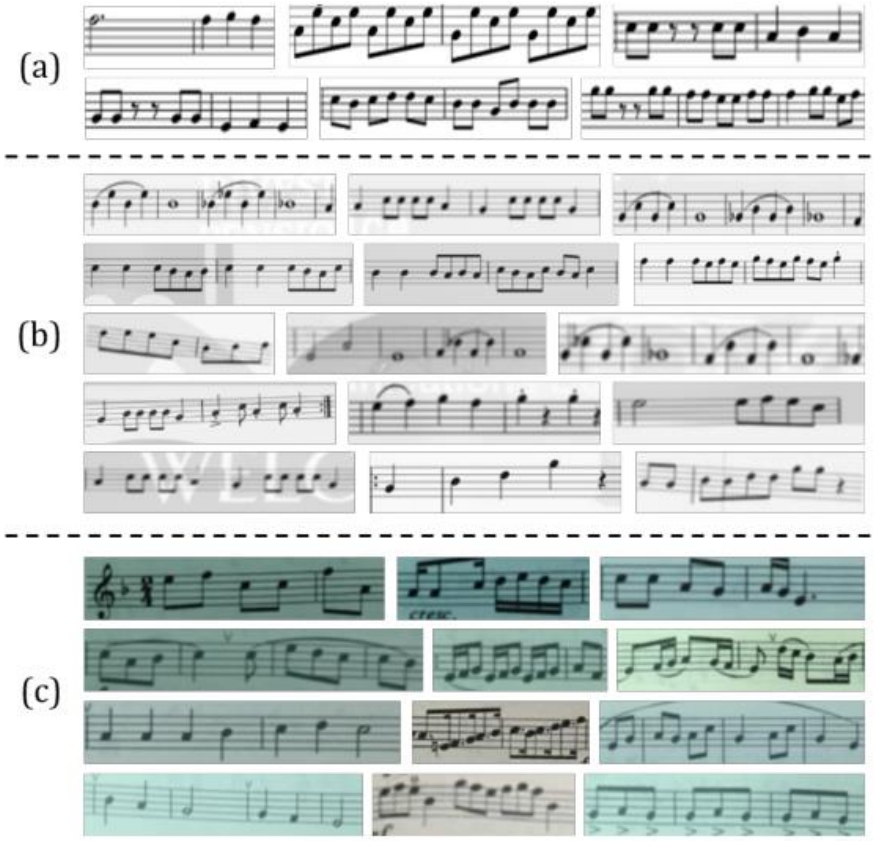
- End-to-end trainable
- No need to detect/segment individual characters
- Unconstraint to any lexicon
- Small model size (8.3M parameters, 33M RAM)

	E2E Train	Conv Ftrs	CharGT-Free	Unconstrained	Model Size
Wang *et al.* [34]	✗	✗	✗	✔	-
Mishra *et al.* [28]	✗	✗	✗	✗	-
Wang *et al.* [35]	✗	✔	✗	✔	-
Goel *et al.* [13]	✗	✗	✔	✗	-
Bissacco *et al.* [8]	✗	✗	✗	✔	-
Alsharif and Pineau [6]	✗	✔	✗	✔	-
Almazán *et al.* [5]	✗	✗	✔	✗	-
Yao *et al.* [36]	✗	✗	✗	✔	-
Rodrguez-Serrano *et al.* [30]	✗	✗	✔	✗	-
Jaderberg *et al.* [23]	✗	✔	✗	✔	-
Su and Lu [33]	✗	✗	✔	✔	-
Gordo [14]	✗	✗	✗	✗	-
Jaderberg *et al.* [22]	✔	✔	✔	✗	490M
Jaderberg *et al.* [21]	✔	✔	✔	✔	304M
CRNN	✔	✔	✔	✔	**8.3M**

CRNN: End-to-End Trainable Network for Scene Text Recognition

Another Application: Musical Score Recognition

- Apply CRNN to recognize musical notes (pitches only) in musical score photos



	Clean	Synthesized	Real-World
Capella Scan [3]	51.9%/1.75	20.0%/2.31	43.5%/3.05
PhotoScore [4]	55.0%/2.34	28.0%/1.85	20.4%/3.00
CRNN	**74.6%/0.37**	**81.5%/0.30**	**84.0%/0.30**

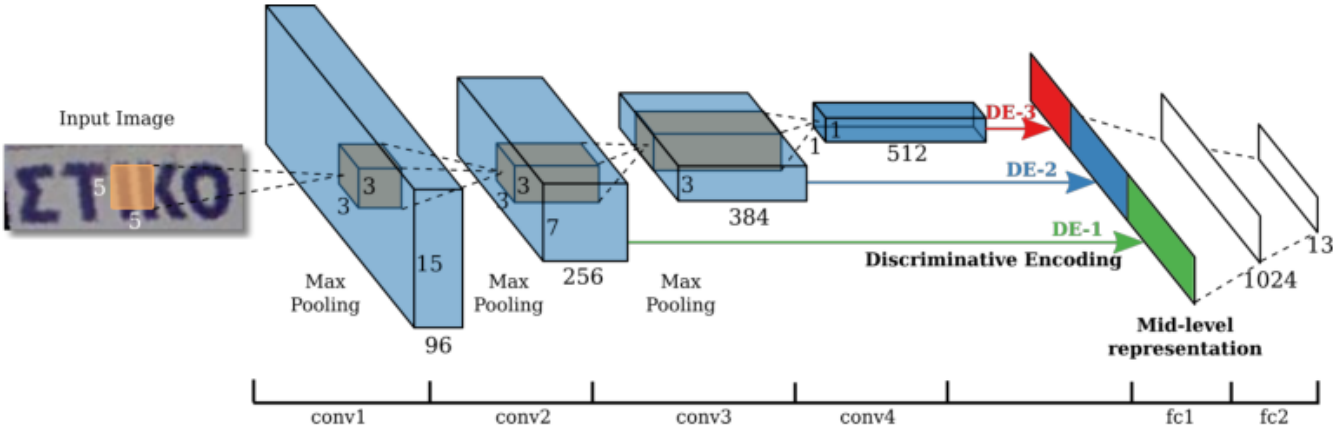
Script Identification with Discriminative CNN

[1] Baoguang Shi, Xiang Bai, Cong Yao, "Script Identification in the Wild via Discriminative Convolutional Neural Network", Pattern Recognition, 2015 (accepted)

- A large-scale dataset for script identification in the wild (SIW-13, available for download)
  - –16,291 cropped text images
  - –13 classes
  - –9,791 for training
  - –6,500 for testing

Script Identification with Discriminative CNN

- Method overview
  - Scripts are easier to distinguish by the *discriminative patches* (e.g. special characters)
  - We model the convolutions, *discriminative encoding*, and horizontal pooling into one network



Script Identification with Discriminative CNN

- Steps

  1. Discover discriminative patches from the training data, resulting in a *discriminative codebook* (a set of linear classifiers)

  2. Initialize the *discriminative encoding layer* using the codebook weights

  3. Jointly fine-tune the whole network

Script Identification with Discriminative CNN

- Discover discriminative patches
  - Extract dense local descriptors from the convolutional feature maps, extracted by a trained CNN
  - Perform discriminative clustering [1] on the descriptors

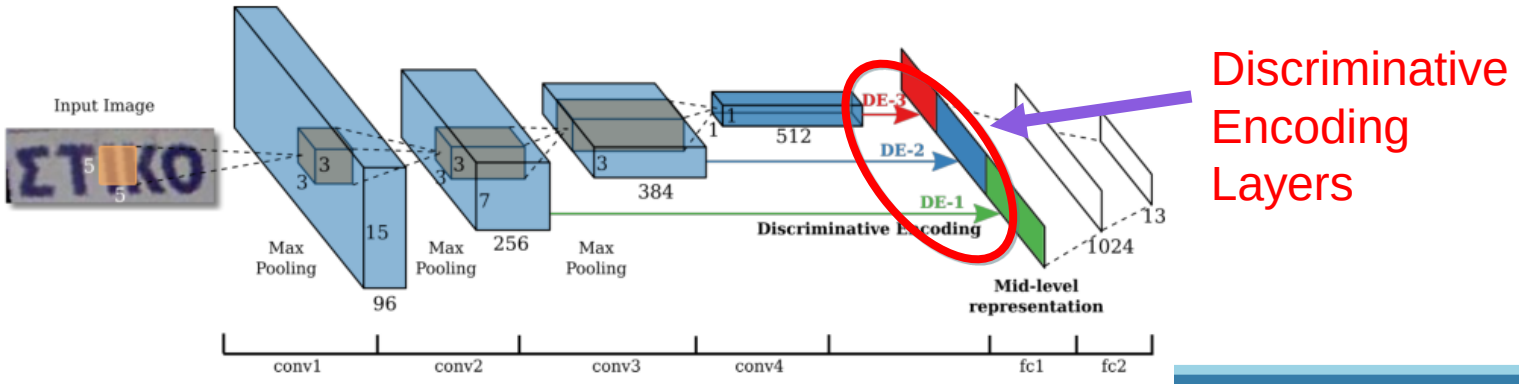
[1] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In Proc. of ECCV, 2012.

Script Identification with Discriminative CNN

- Visualizations of part of the codebook (a row for a cluster)

Script Identification with Discriminative CNN

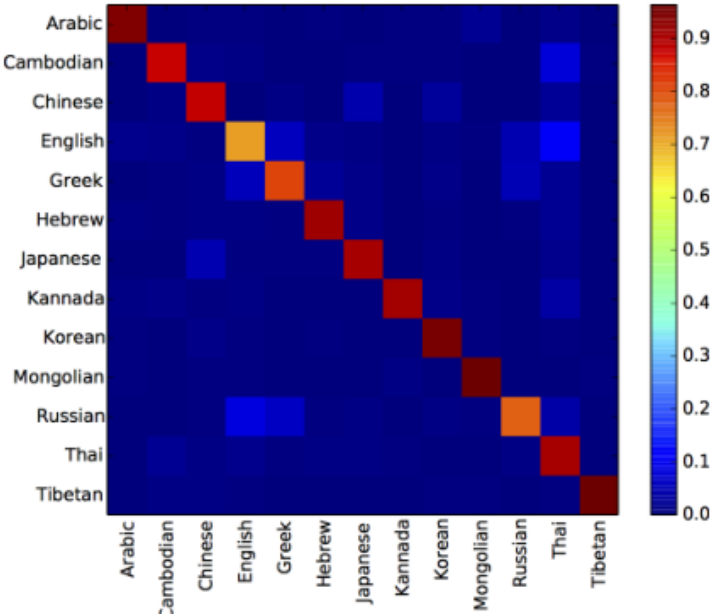
- The weights of the codebook are used for initializing the *discriminative encoding layer*
  - Discriminative encoding layer:
    - $y_i = \mathrm{hpool}_j(\max(0, Wx_{i,j} + b))$
    - $x_{i,j}$ is the feature vector at location (i, j) of the feature map
    - "hpool" is the horizontal pooling operation
    - W and b are initialized from the codebook
- Finally, we jointly fine-tune the network



Script Identification with Discriminative CNN

- The model is evaluated on the SIW-13 dataset (extended from SIW-10)
  - "Alphabetic" and "Logographic" are two subsets

	Alphabetic				Logographic				Full			
	LBP	SLN	CNN	Ours	LBP	SLN	CNN	Ours	LBP	SLN	CNN	Ours
Ara	0.80	0.91	0.94	**0.96**	-	-	-	-	0.64	0.87	0.90	**0.94**
Cam	-	-	-	-	-	-	-	-	0.46	0.76	0.83	**0.88**
Chi	-	-	-	-	0.82	0.90	0.86	**0.91**	0.66	0.87	0.85	**0.88**
Eng	0.63	0.77	0.72	**0.83**	-	-	-	-	0.31	0.64	0.58	**0.71**
Gre	0.70	0.79	0.74	**0.86**	-	-	-	-	0.57	0.75	0.70	**0.81**
Heb	-	-	-	-	-	-	-	-	0.61	**0.91**	0.89	0.91
Jap	-	-	-	-	0.85	0.93	0.88	**0.93**	0.58	0.88	0.75	**0.90**
Kan	-	-	-	-	-	-	-	-	0.56	0.88	0.82	**0.91**
Kor	-	-	-	-	0.87	0.94	0.93	**0.96**	0.69	0.93	0.90	**0.95**
Mon	0.88	0.97	0.94	**0.98**	-	-	-	-	0.77	0.95	0.96	**0.96**
Rus	0.62	0.78	0.71	**0.82**	-	-	-	-	0.44	0.70	0.66	**0.79**
Tha	-	-	-	-	-	-	-	-	0.61	0.91	0.79	**0.94**
Tib	-	-	-	-	0.96	0.97	**0.99**	0.98	0.88	**0.97**	0.97	0.97
Avg.	0.73	0.84	0.81	**0.89**	0.88	0.93	0.92	**0.94**	0.60	0.85	0.82	**0.89**



Compare with our previous ICDAR version (horizontal pooling network)

	Alphabetic	Logographic	Full
MSPN	0.870 ± 0.005	0.930 ± 0.019	0.866 ± 0.014
DiscCNN	**0.892 ± 0.008**	**0.942 ± 0.007**	**0.887 ± 0.007**

Automatic discrimination of text and non-text natural images

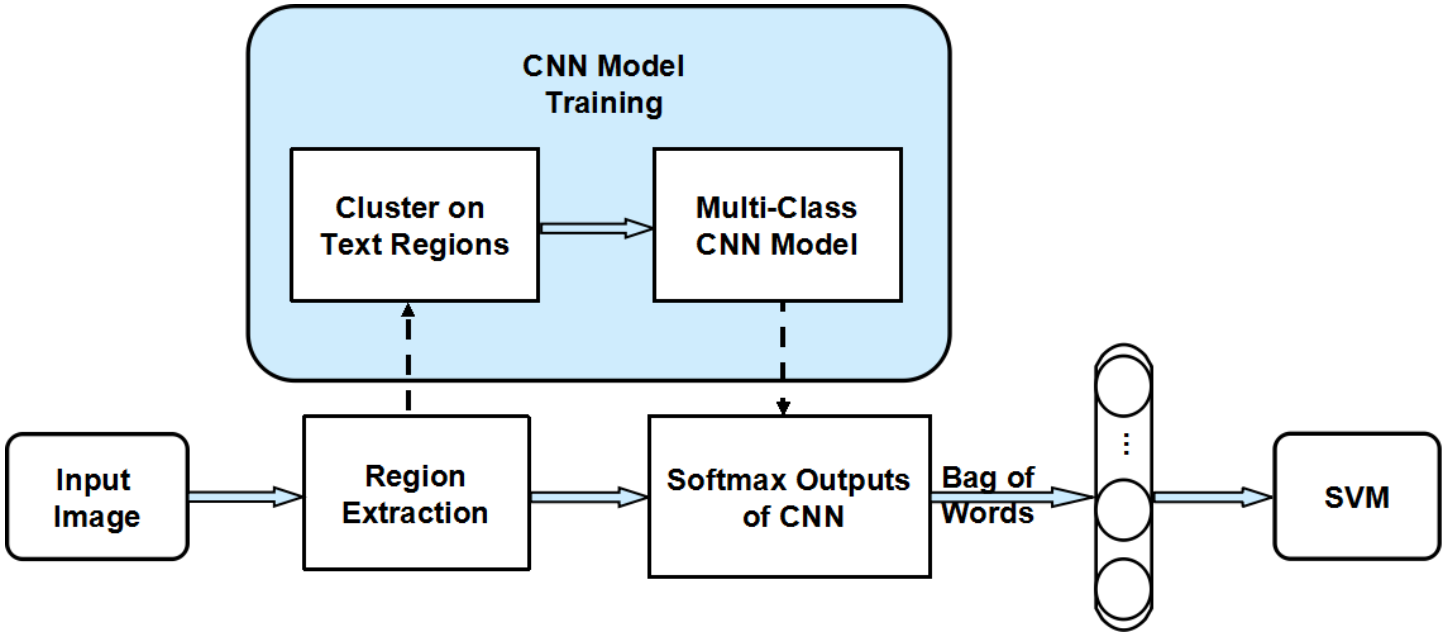
- propose an effective algorithm for text image discrimination
- establish a bechmark of text and non-text images

Text Image Discrimination

[1] Chengquan Zhang, Cong Yao, Baoguang Shi and Xiang Bai. Automatic Discrimination of Text and Non-Text Natural Images. ICDAR 2015.

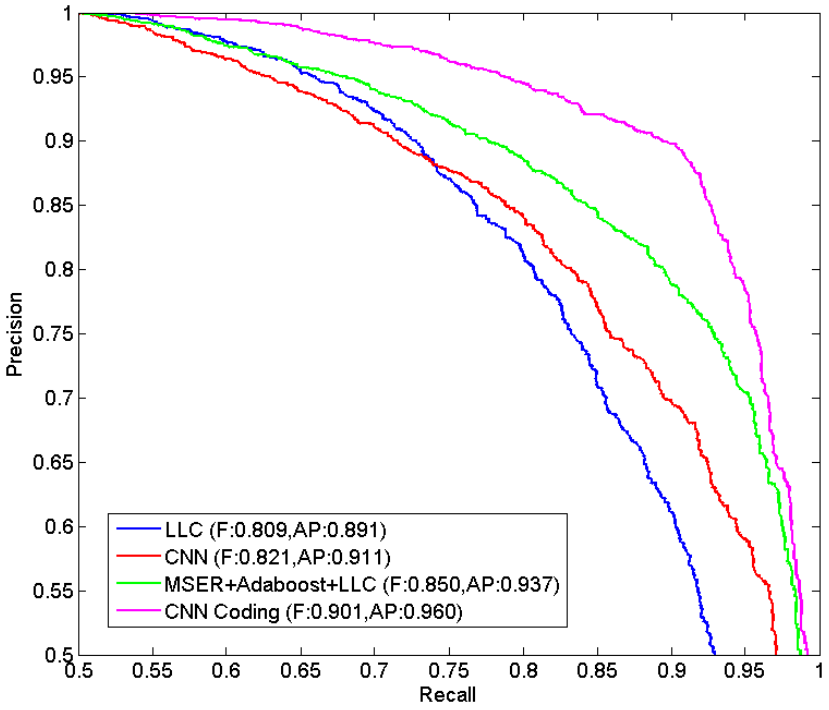
Algorithm

- 3 matural techniques: MSER,CNN and BoW
- major training steps: Region Extraction and Clustering, Multi-class CNN training, CNN Coding and SVM Training



Pipeline for text image discrimination

Performance & Time Cost



P-R curves of different methods on our dataset

Stage	Time
MSER Extraction	0.18~0.23s
CNN Coding	0.25~0.26s
SVM Classification	0.24ms
Total	0.43~0.49s

time cost for each step with single CPU & GPU

Contents

- Problem definition
- Significance and challenges
- Previous works
- Our algorithms
- Conclusion

Conclusion

The common key to the success of the above surveyed text detection and recognition methods is representation, just as in many other vision problems

Conclusion

Conventional methods rely on human designed representations (MSER, SWT, HOG),
while CNN based algorithms directly learn representations from data

Conclusion

Learning representation from data
is a future trend

Conclusion

But there is still a long way to go,
since challenges remain:
multi-scale,
multi-orientation,
multi-language,
…

Thank You!

Email: xbai@hust.edu.cn