Scene Text Recognition

Share & Learn Seminar

Yi Li

06/09/2021

Table of Contents

- Problem Definition
- Papers
  - CRNN (2015): https://arxiv.org/abs/1507.05717
  - ASTER (2018): https://ieeexplore.ieee.org/document/8395027
  - MORAN(2019): https://arxiv.org/abs/1901.03003
  - TextScanner (2020): https://arxiv.org/abs/1912.12422 (not open sourced yet)
- References
- A Practical Ultra Lightweight OCR System
  - PP-OCR (2020): https://arxiv.org/abs/2009.09941

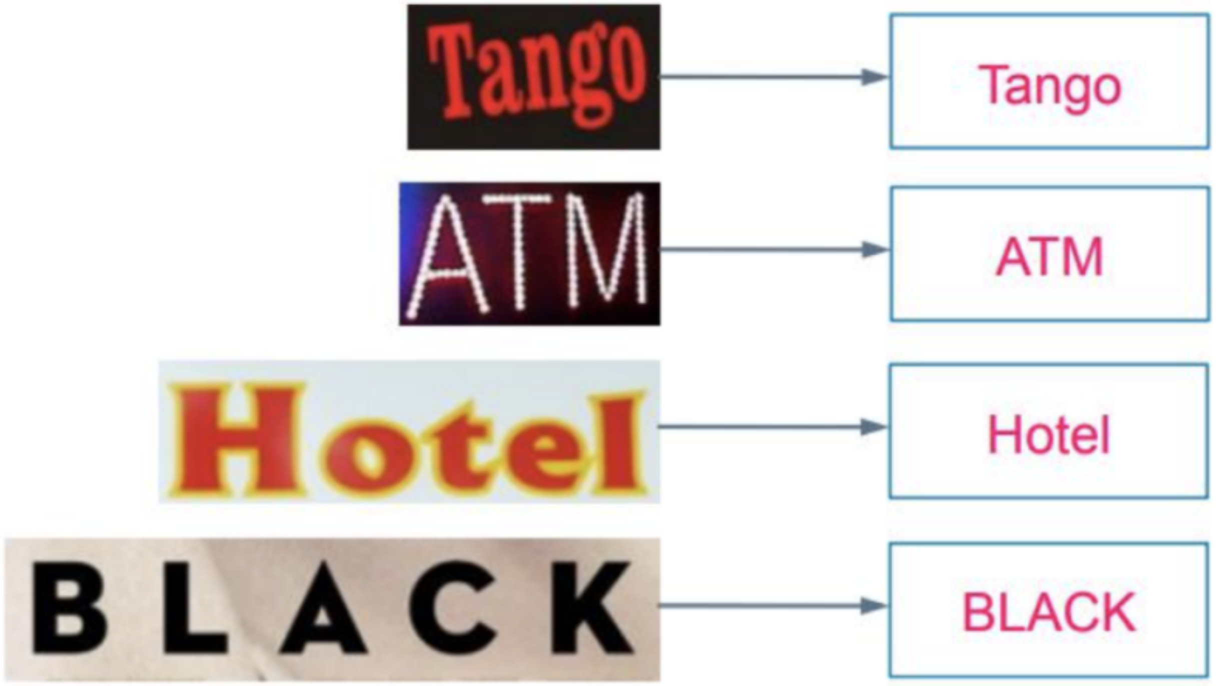
This slide comes from Megvii (Link).

Problem Definition

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

This slide comes from Megvii (Link).

Problem Definition

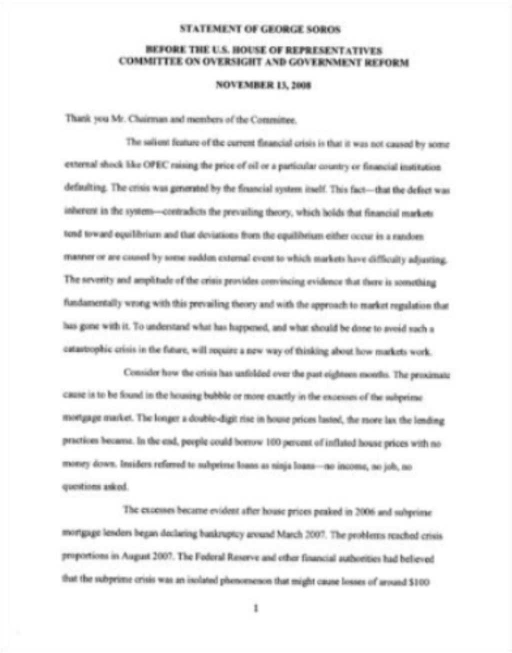


Scene text recognition is the process of converting text regions into computer readable and editable symbols

This slide comes from Megvii (Link).

Challenges

Traditional OCR *vs.* Scene Text Detection and Recognition



STATEMENT OF GEORGE SOROS

BEFORE THE U.S. HOUSE OF REPRESENTATIVES
COMMITTEE ON OVERSIGHT AND GOVERNMENT REFORM

NOVEMBER 13, 2008

Thank you Mr. Chairman and members of the Committee.

The salient feature of the current financial crisis is that it was not caused by some external shock like OPEC raising the price of oil or a particular country or financial institution defaulting. The crisis was generated by the financial system itself. This fact—that the defect was inherent in the system—contradicts the prevailing theory, which holds that financial markets tend toward equilibrium and that deviations from the equilibrium either occur in a random manner or are caused by some sudden external event to which markets have difficulty adjusting. The severity and amplitude of the crisis provides convincing evidence that there is something fundamentally wrong with this prevailing theory and with the approach to market regulation that has gone with it. To understand what has happened, and what should be done to avoid such a catastrophic crisis in the future, will require a new way of thinking about how markets work.

Consider how the crisis has unfolded over the past eighteen months. The proximate cause is to be found in the housing bubble or more exactly in the excesses of the subprime mortgage market. The longer a double-digit rise in house prices lasted, the more lax the lending practices became. In the end, people could borrow 100 percent of inflated house prices with no money down. Insiders referred to subprime loans as ninja loans—no income, no job, no questions asked.

The excesses became evident after house prices peaked in 2006 and subprime mortgage lenders began declaring bankruptcy around March 2007. The problems reached crisis proportions in August 2007. The Federal Reserve and other financial authorities had believed that the subprime crisis was an isolated phenomenon that might cause losses of around $100

1

Labeling/Annotation:

- Char-level
- Word-level
- Line-level

- clean background vs. cluttered background (uneven lighting, low resolution, heavy occlusions, etc.)
- regular font vs. various fonts
- plain layout vs. complex layouts
- monotone color vs. different colors

- Curved text
- Arbitrarily oriented text
- Perspective distortion
- Multi-language

CRNN (2015)

An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition

(https://arxiv.org/abs/1507.05717)

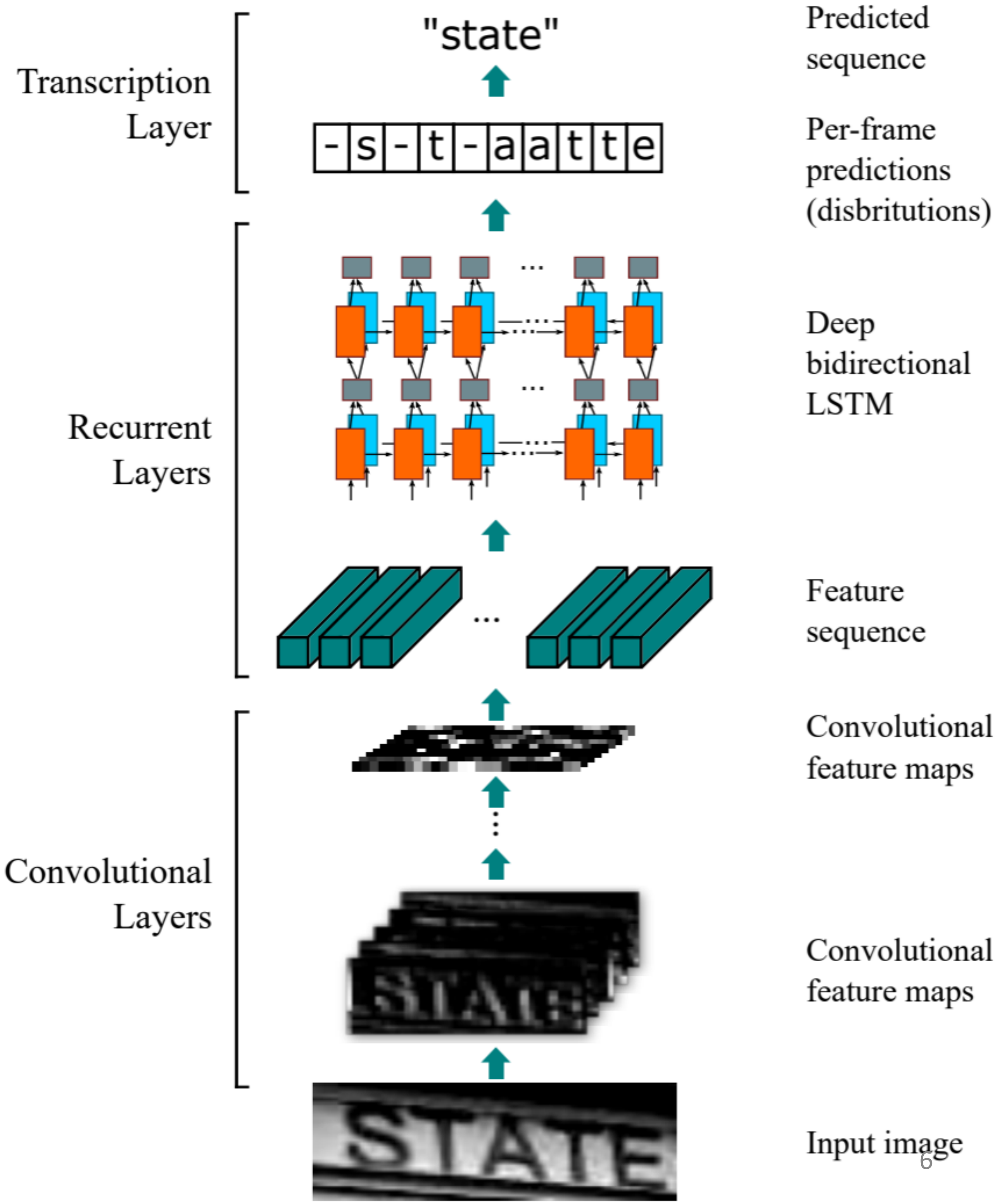
CRNN Network Architecture

“The architecture consists of three parts:

1) convolutional layers, which extract a feature sequence from the input image;

1) recurrent layers, which predict a label distribution for each frame;

1) transcription layer, which translates the per-frame predictions into the final label sequence.”



Convolution Layers: Feature Sequence Extraction

- Based on the VGG architectures
- 7 Conv + 4 MaxPool + 2 BatchNorm
- 1×2 pooling window: yield feature maps with larger width, hence, longer feature sequence.
- BatchNorm: speed up training process

Convolution	#maps:512, k:2 $\times$ 2, s:1, p:0
MaxPooling	Window:1 $\times$ 2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3 $\times$ 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 $\times$ 3, s:1, p:1
MaxPooling	Window:1 $\times$ 2, s:2
Convolution	#maps:256, k:3 $\times$ 3, s:1, p:1
Convolution	#maps:256, k:3 $\times$ 3, s:1, p:1
MaxPooling	Window:2 $\times$ 2, s:2
Convolution	#maps:128, k:3 $\times$ 3, s:1, p:1
MaxPooling	Window:2 $\times$ 2, s:2
Convolution	#maps:64, k:3 $\times$ 3, s:1, p:1
Input	W $\times$ 32 gray-scale image

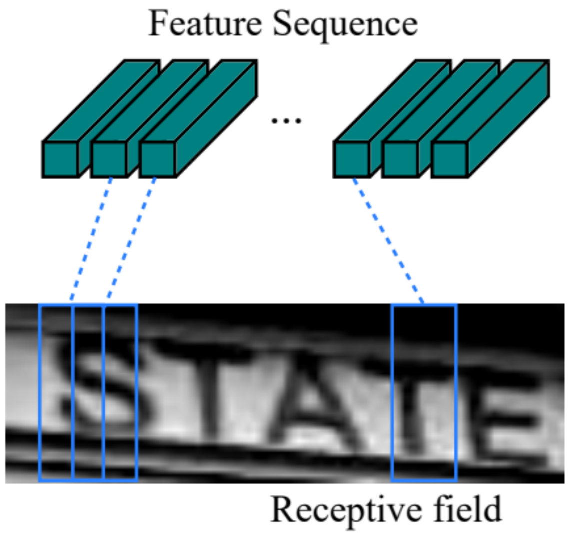
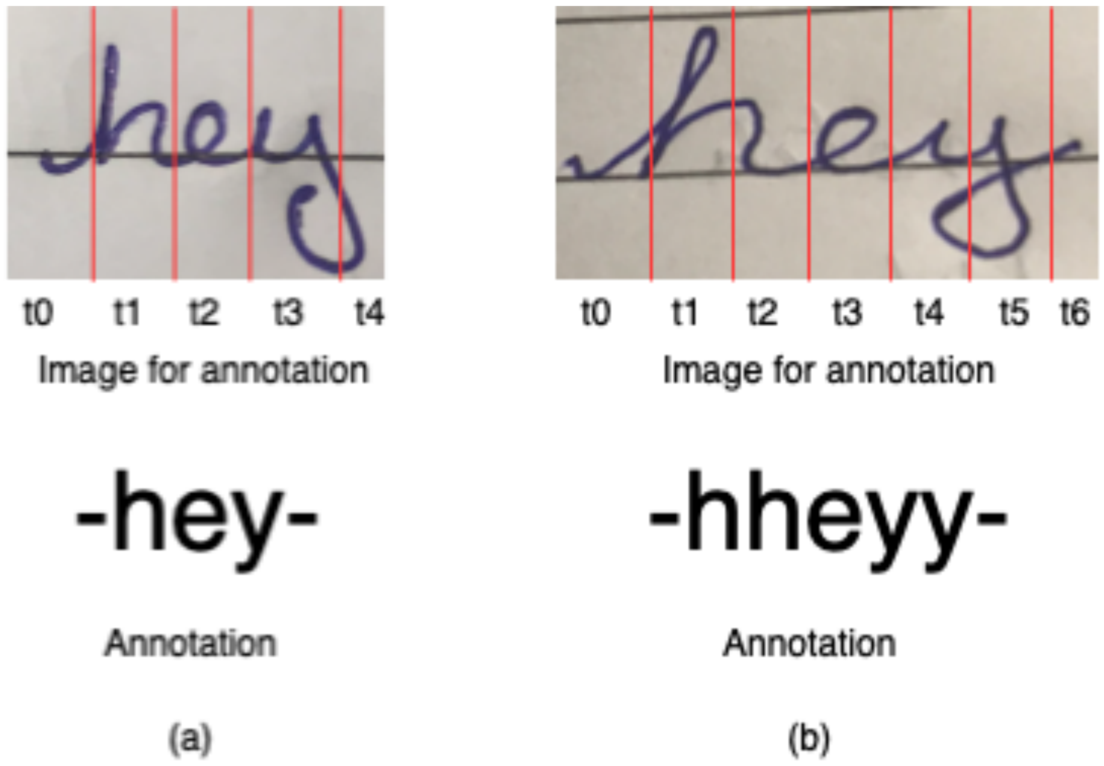


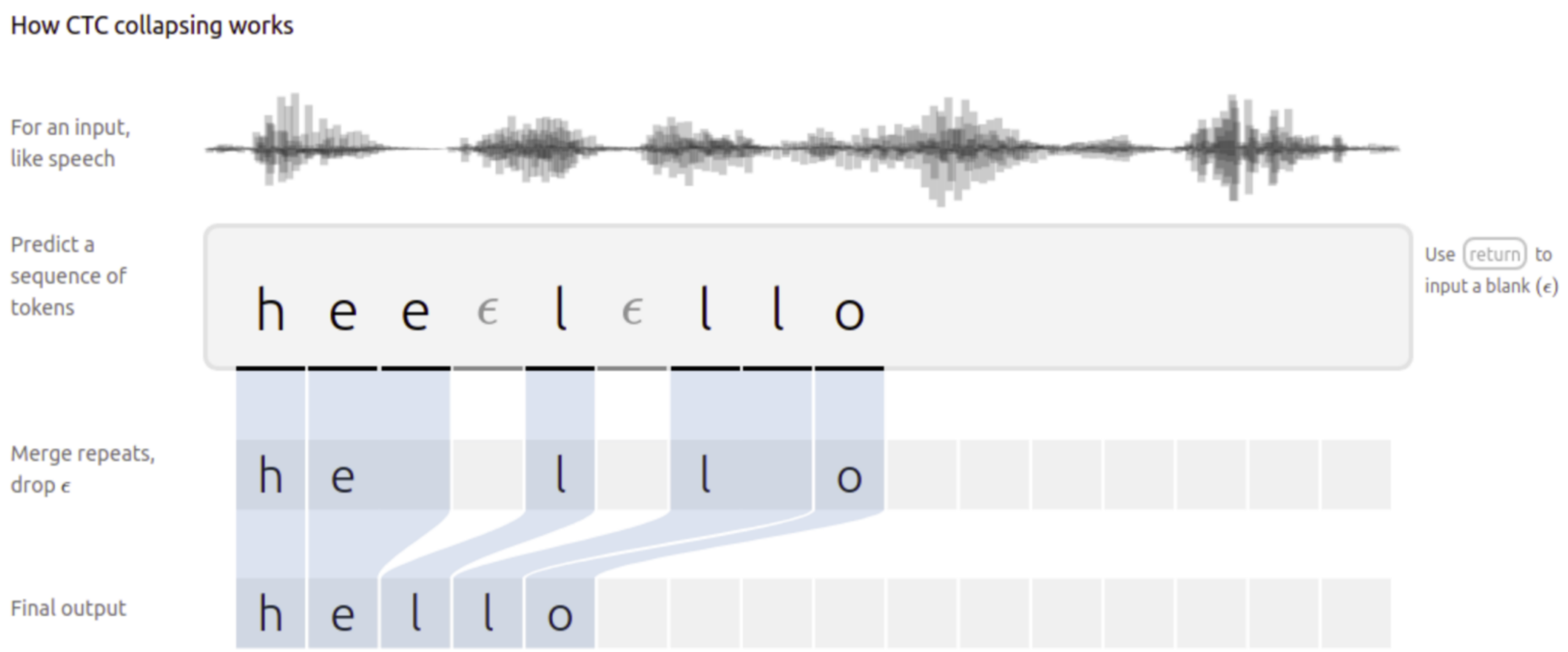
Figure 2. The receptive field. Each vector in the extracted feature sequence is associated with a receptive field on the input image, and can be considered as the feature vector of that field.

Transcription Layer: Connectionist Temporal Classification (CTC)



- Convert the per-frame predictions made by RNN into a label sequence
- Introduce “blank” labels (different from space labels)
- Solve alignment issue
- CTC loss uses forward-backward algorithm to quickly update gradients.

CTC (continued)



CTC (continued)

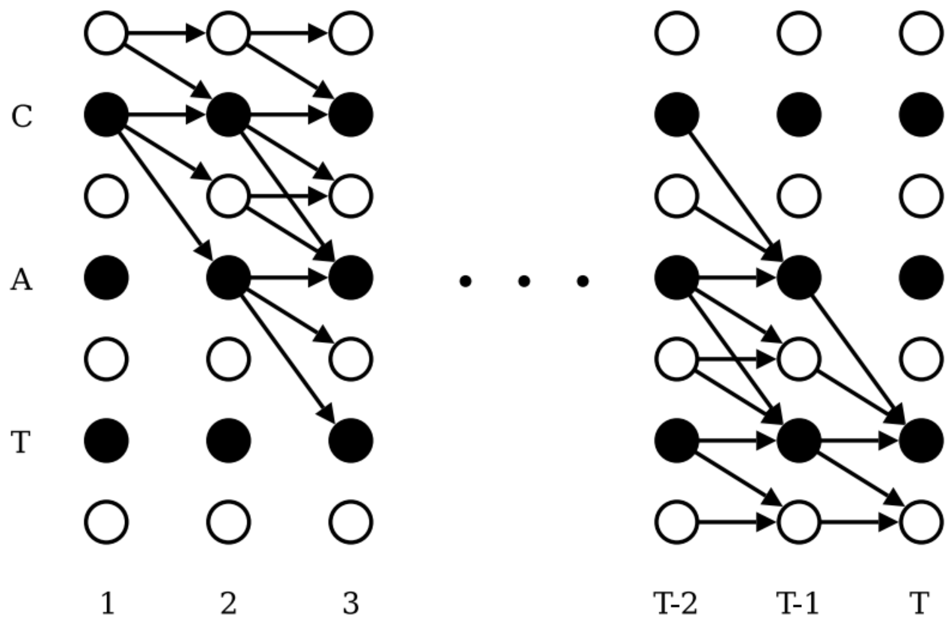


Figure 3. **illustration of the forward backward algorithm applied to the labelling 'CAT'**. Black circles represent labels, and white circles represent blanks. Arrows signify allowed transitions. Forward variables are updated in the direction of the arrows, and backward variables are updated against them.

The formulation of the conditional probability is briefly described as follows: The input is a sequence $\mathbf{y} = y_1, \ldots, y_T$ where T is the sequence length. Here, each $y_t \in \Re^{|\mathcal{L}'|}$ is a probability distribution over the set $\mathcal{L}' = \mathcal{L} \cup$, where $\mathcal{L}$ contains all labels in the task (*e.g.* all English characters), as well as a 'blank' label denoted by . A sequence-to-sequence mapping function $\mathcal{B}$ is defined on sequence $\boldsymbol{\pi} \in \mathcal{L}'^T$, where T is the length. $\mathcal{B}$ maps $\boldsymbol{\pi}$ onto $\mathbf{l}$ by firstly removing the repeated labels, then removing the 'blank's. For example, $\mathcal{B}$ maps "`--hh-e-l-ll-oo--`" ('$-$' represents 'blank') onto "`hello`". Then, the conditional probability is defined as the sum of probabilities of all $\boldsymbol{\pi}$ that are mapped by $\mathcal{B}$ onto $\mathbf{l}$:

$$p(\mathbf{l}|\mathbf{y}) = \sum_{\boldsymbol{\pi}:\mathcal{B}(\boldsymbol{\pi})=\mathbf{l}} p(\boldsymbol{\pi}|\mathbf{y}), \quad (1)$$

where the probability of $\boldsymbol{\pi}$ is defined as $p(\boldsymbol{\pi}|\mathbf{y}) = \prod_{t=1}^{T} y_{\pi_t}^t$, $y_{\pi_t}^t$ is the probability of having label π_t at time stamp t. Directly computing Eq. 1 would be computationally infeasible due to the exponentially large number

RARE (2016)

Robust Scene Text Recognition with **A**utomatic **Re**ctification
(https://arxiv.org/abs/1603.03915)

ASTER (2018)

ASTER: An **At**tentional **S**cene **Te**xt **R**ecognizer with Flexible Rectification
(https://ieeexplore.ieee.org/document/8395027)

RARE/ASTER

Replace CTC layer with an attention-based decoder.

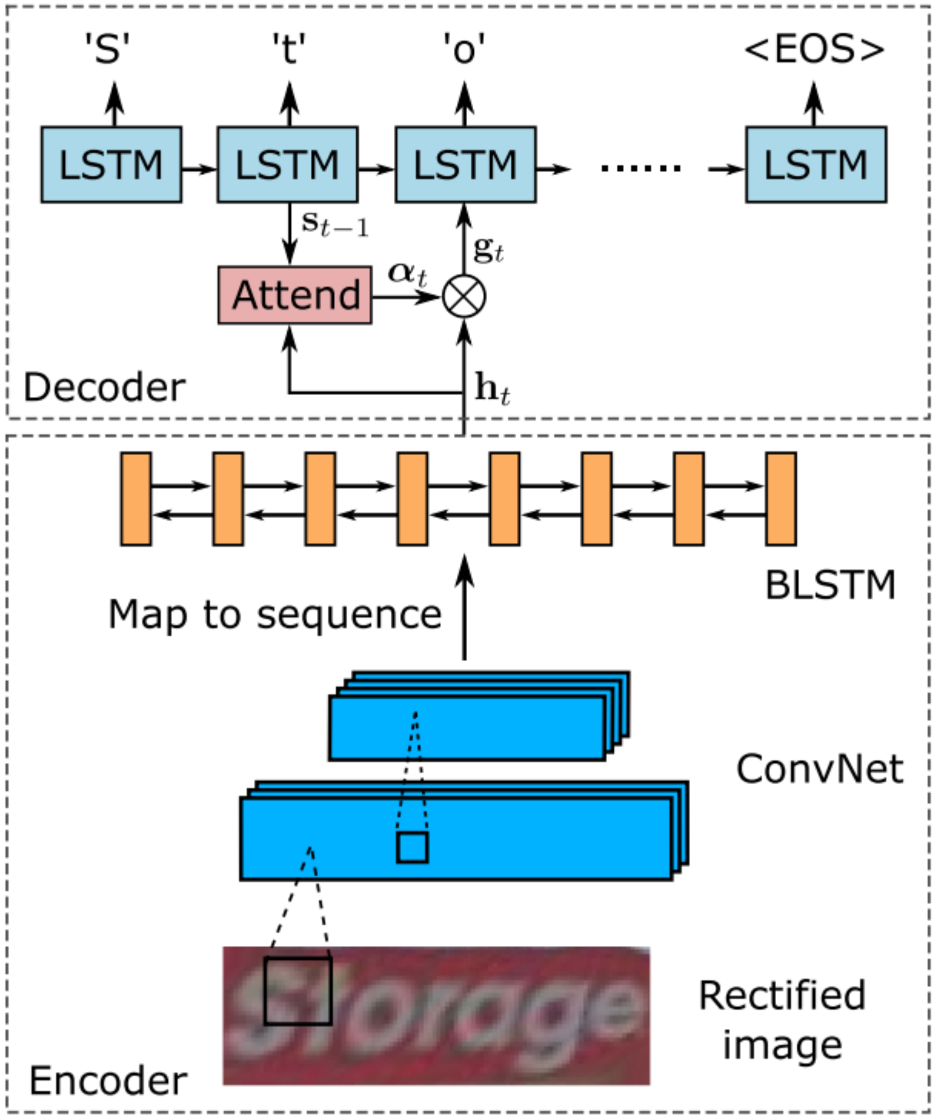


Fig. 7. Structure of the basic text recognition network.

Add a rectification module before recognition.

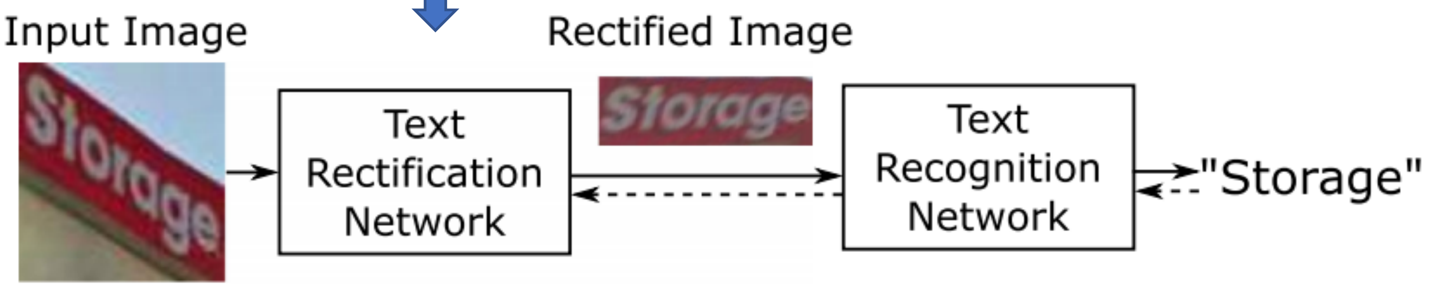


Fig. 2. Overview of the proposed model. Dashed lines show the flow of gradients.

Why do we need Spatial transformer networks?

Are Convolutional Neural Networks invariant to...

- Scale? **No**
- Rotation? **No**
- Translation? **Partially**

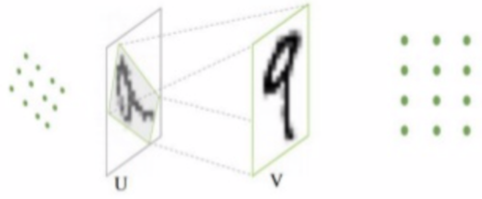
https://arxiv.org/abs/1506.02025

Spatial Transformer Networks (STN) is a differentiable module that can be inserted anywhere in ConvNet architecture to increase its geometric invariance. It effectively gives the network the ability to spatially transform feature maps at no extra data or supervision cost.

Examples

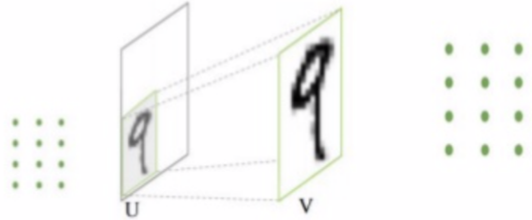
Affine transform

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



Attention model

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



(x_i^t, y_i^t) - coordinates in the target (output) feature map

(x_i^s, y_i^s) - coordinates in the source (input) feature map

This slide comes from SlideShare by Victor Campos (Link).

https://goo.gl/qdEhUu

ASTER Rectification Samples

For every two rows, the first row contains the input images (top), the predicted control points (visualized as green crosses), and the rectified images (bottom). The second row contains the recognition results.

MORAN(2019)

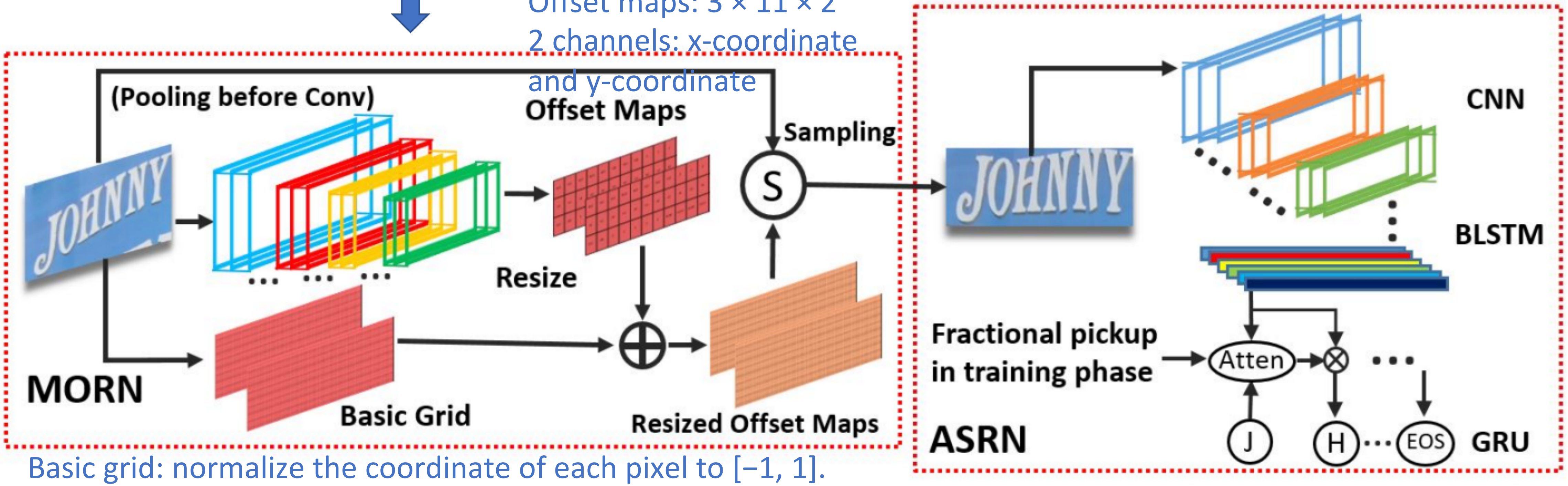
A **M**ulti-**O**bject **R**ectified **A**ttention **N**etwork for Scene Text Recognition
(https://arxiv.org/abs/1901.03003)

MORAN

Recognition Module:
Replace CTC layer with an attention-based decoder.

Rectification Module:
Replace STN module with pixel-level rectification, which is free of geometric constraints like affine transformation.

Offset maps: 3 × 11 × 2
2 channels: x-coordinate and y-coordinate



Basic grid: normalize the coordinate of each pixel to [−1, 1].
top-left pixel coordinate: (−1, −1),
bottom-right pixel coordinate: (1, 1).

Figure 4. Overall structure of MORAN.

MORAN Rectification Samples

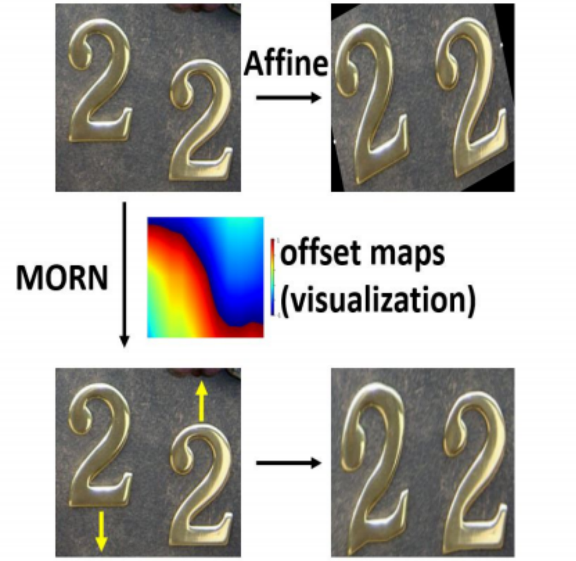


Figure 3. Comparison of the MORN and affine transformation. The MORN is free of geometric constraints. The main direction of rectification predicted by the MORN for each character is indicated by a yellow arrow. The offset maps generated by the MORN are visualized as a heat map. The offset values on the boundary between red and blue are zero. The directions of rectification on both sides of the boundary are opposite and outward. The depth of the color represents the magnitude of the offset value. The gradual-change in color indicates the smoothness of the rectification.

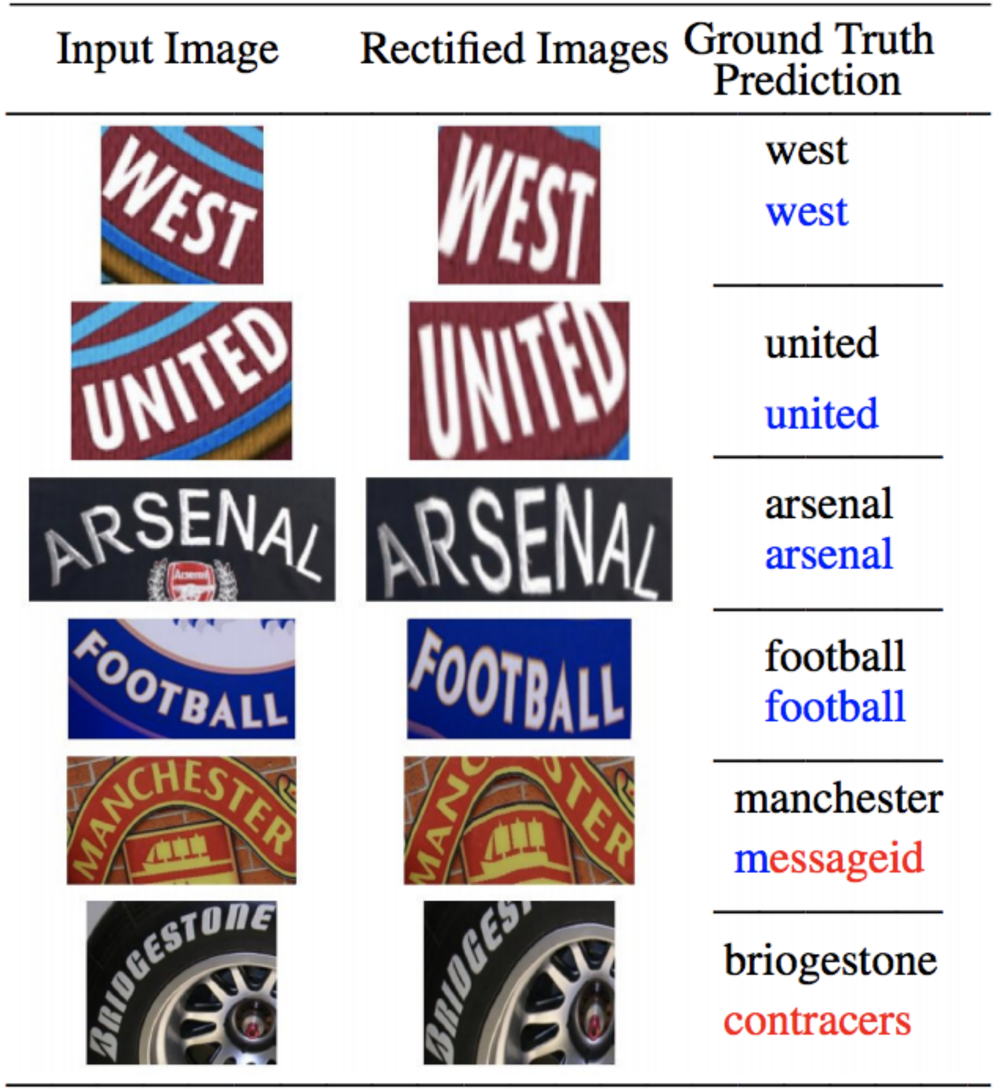
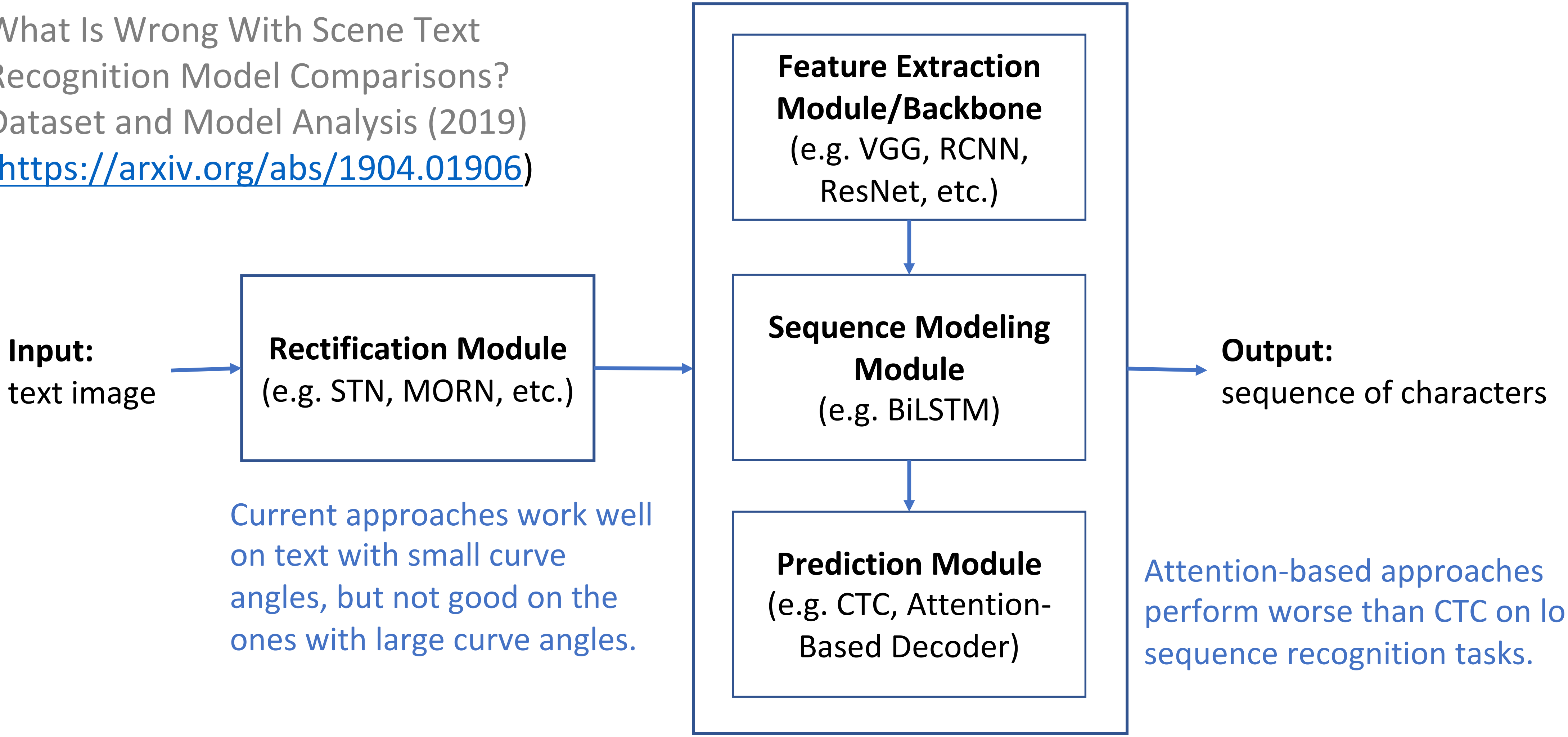


Figure 9. Effects of different curve angles of scene text. The first four rows are text with small curve angles and the last two rows are text with large curve angles. The MORAN can rectify irregular text with small curve angles.

Summary on Network Architecture

What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis (2019) (https://arxiv.org/abs/1904.01906)



Current approaches work well on text with small curve angles, but not good on the ones with large curve angles.

Attention-based approaches perform worse than CTC on long sequence recognition tasks.

A Comparative Study of Attention-based Encoder-Decoder Approaches to Natural Scene Text Recognition (2019) (https://ieeexplore.ieee.org/document/8978138)

TextScanner (2020)

TextScanner: Reading Characters in Order for Robust Scene Text Recognition (https://arxiv.org/abs/1912.12422)

Pretrain models with character-level annotation, and then do transfer learning on word-level or line-level annotation.

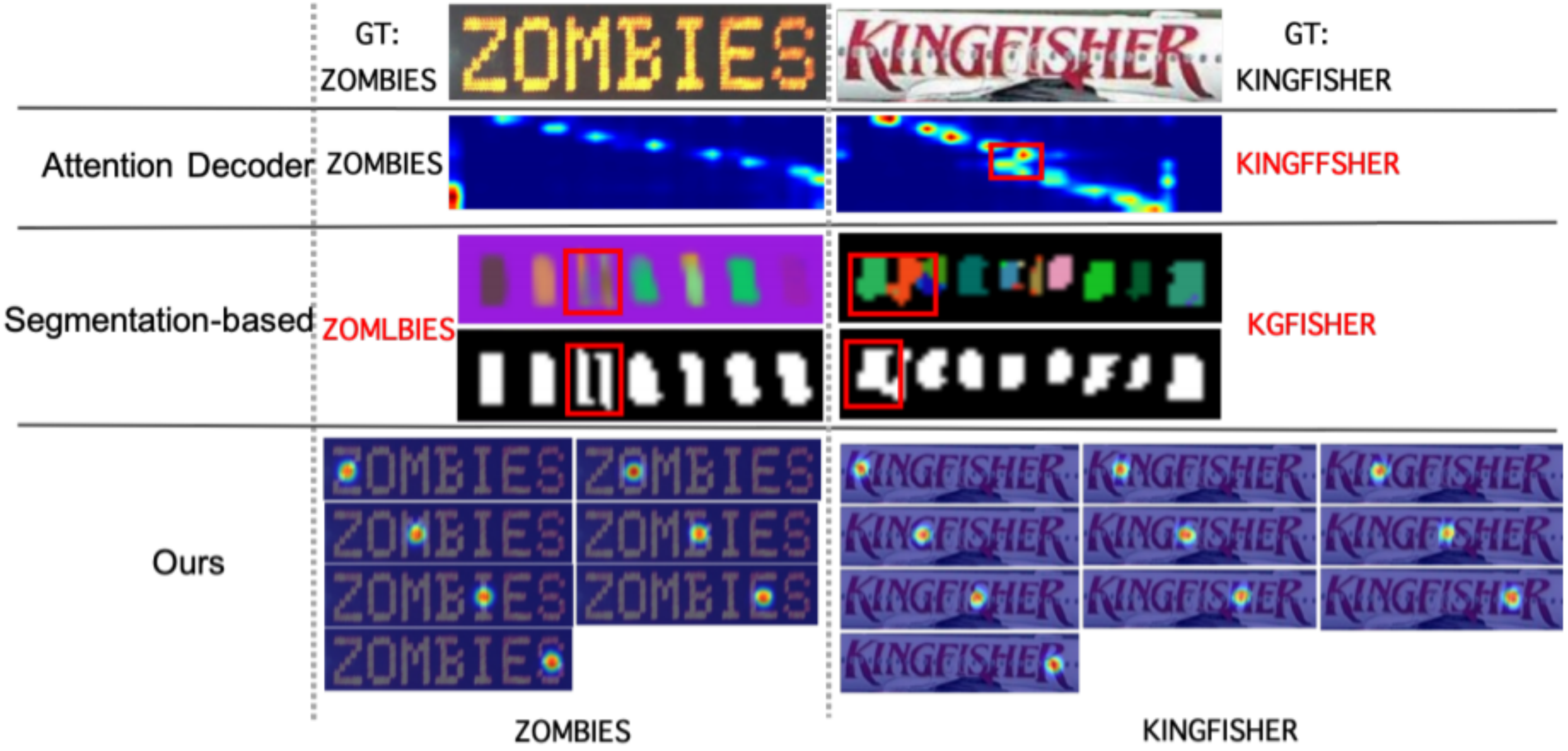


Figure 1: Our Motivation. RNN-attention-based methods may encounter the problem of *attention drift* (Cheng et al. 2017) (see the red rectangle), thus leading to incorrect prediction of character class. In semantic segmentation

TextScanner Architecture

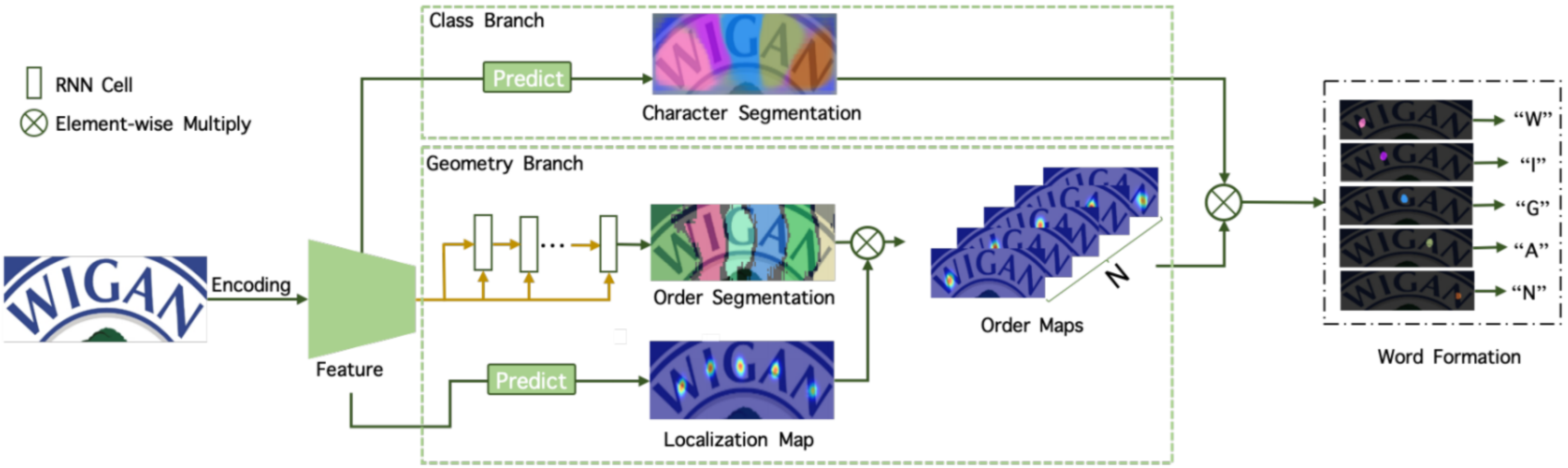


Figure 2: Schematic illustration of the proposed text recognition framework. Different colors in character segmentation map represent the values in different channels. The values in the localization map and order maps are visualized as heat maps. The predictions of the two branches are fused to extract characters (position, order, and class) and form the final output.

Geometry Branch Details

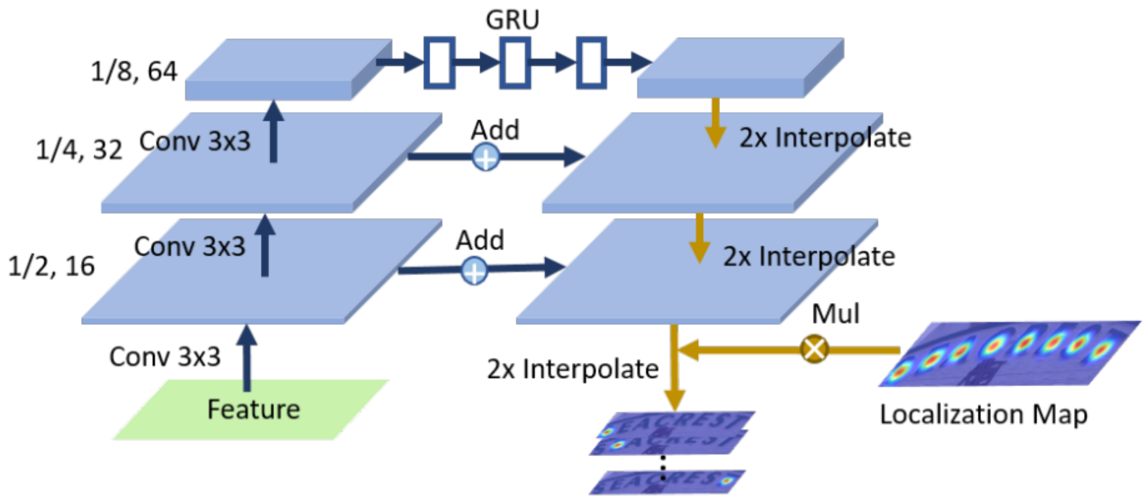


Figure 3: Illustration of the geometry branch. The feature maps are up-sampled and down-sampled by a pyramid architecture with skip connections. Features at the top layer is processed by an RNN module for context modeling.

Character-Level Annotation for Pre-Training

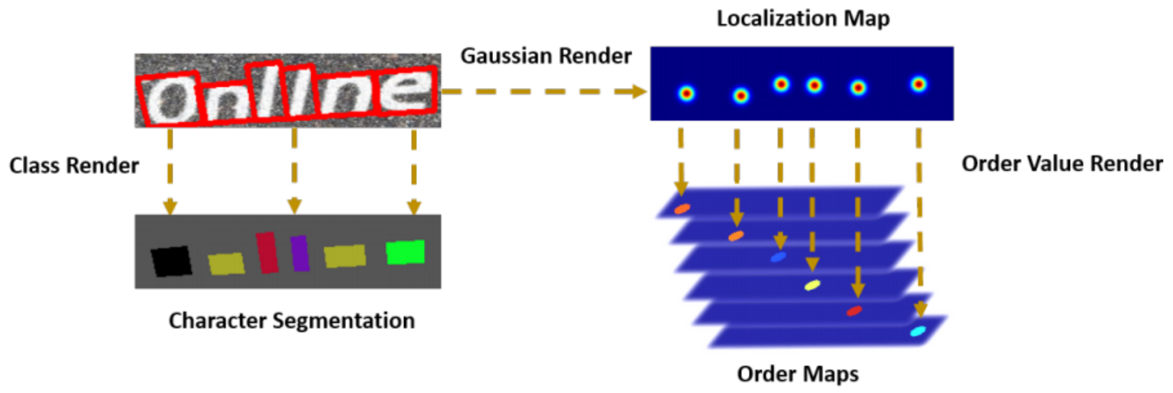


Figure 4: Ground truth generation for pre-training. Pixels outside shrunk boxes P' are represented as gray in character segmentation label, which are ignored in loss computation.

Mutual-Supervision Mechanism

After pre-training with character-level annotations, train models with word-level or line-level annotations with mutual-supervision mechanism.

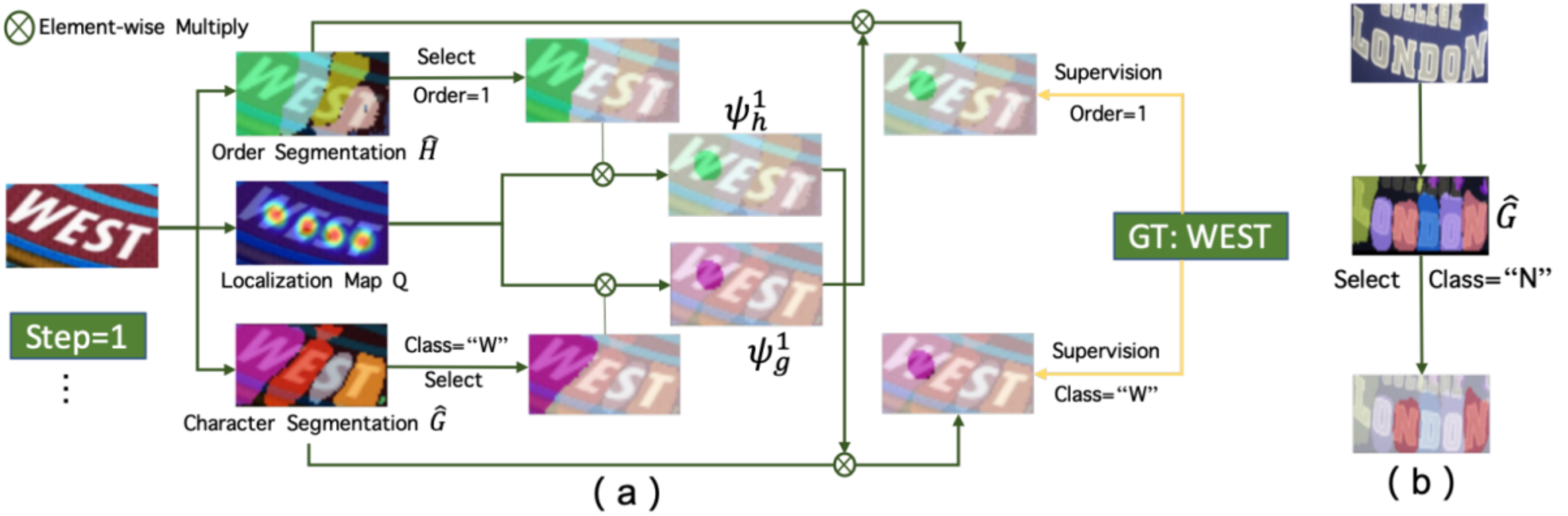


Figure 5: (a) Visualization of step 1 of mutual-supervision mechanism. The selected regions in $\hat{G}$ and $\hat{H}$ are refined using Q to get Ψ_g^1 and Ψ_h^1, which are then mapped into $\hat{H}$ and $\hat{G}$ separately. (b) Two regions in $\hat{G}$ are selected for 'N' in "LONDON".

References

Slides:

- Text Detection and Recognition (Megvii: Cong Yao): https://github.com/zsc/megvii-pku-dl-course/blob/master/slides/Lecture7(Text%20Detection%20and%20Recognition_20171031).pdf
- Spatial Transformer Networks (Victor Campos): https://www.slideshare.net/xavigiro/spatial-transformer-networks

Videos:

- Spatial Transformer Networks (DeepMind: arxivSTmovie.m4v): https://goo.gl/qdEhUu
- DB-Net and TextScanner (Megvii: Zhaoyi Wan, Speak in Mandarin): https://www.bilibili.com/video/av83837791

PP-OCR (2020)

PP-OCR: A Practical Ultra Lightweight OCR System (https://arxiv.org/abs/2009.09941) from Baidu Inc.

PP-OCR: A Practical Ultra Lightweight OCR System (2020, from Baidu Inc.)

PaddlePaddle

Image → Text Detection (db_mv3_slim, 1.4M) → Detection Boxes Rectify (dir_cls_mv3_slim, 0.5M) → Text Recognition (crnn_mv3_slim, 1.6M) → Output

Text Detection (db_mv3_slim, 1.4M)

1. Light Backbone
2. Light Head
3. Remove SE
4. Cosine Learning Rate Decay
5. Learning Rate Warm-up
6. FPGM Pruner

Detection Boxes Rectify (dir_cls_mv3_slim, 0.5M)

1. Light Backbone
2. Data Augmentation
3. Input Resolution
4. PACT Quantization

Text Recognition (crnn_mv3_slim, 1.6M)

1. Light Backbone
2. Data Augmentation
3. Cosine Learning Rate Decay
4. Feature Map Resolution
5. Regularization Parameters
6. Learning Rate Warm-up
7. Light Head
8. Pre-trained Model
9. PACT Quantization

PP-OCR Detector: DB (Differentiable Binarization)

- https://arxiv.org/pdf/1911.08947.pdf
- https://github.com/MhLiao/DB or https://github.com/WenmuZhou/DBNet.pytorch

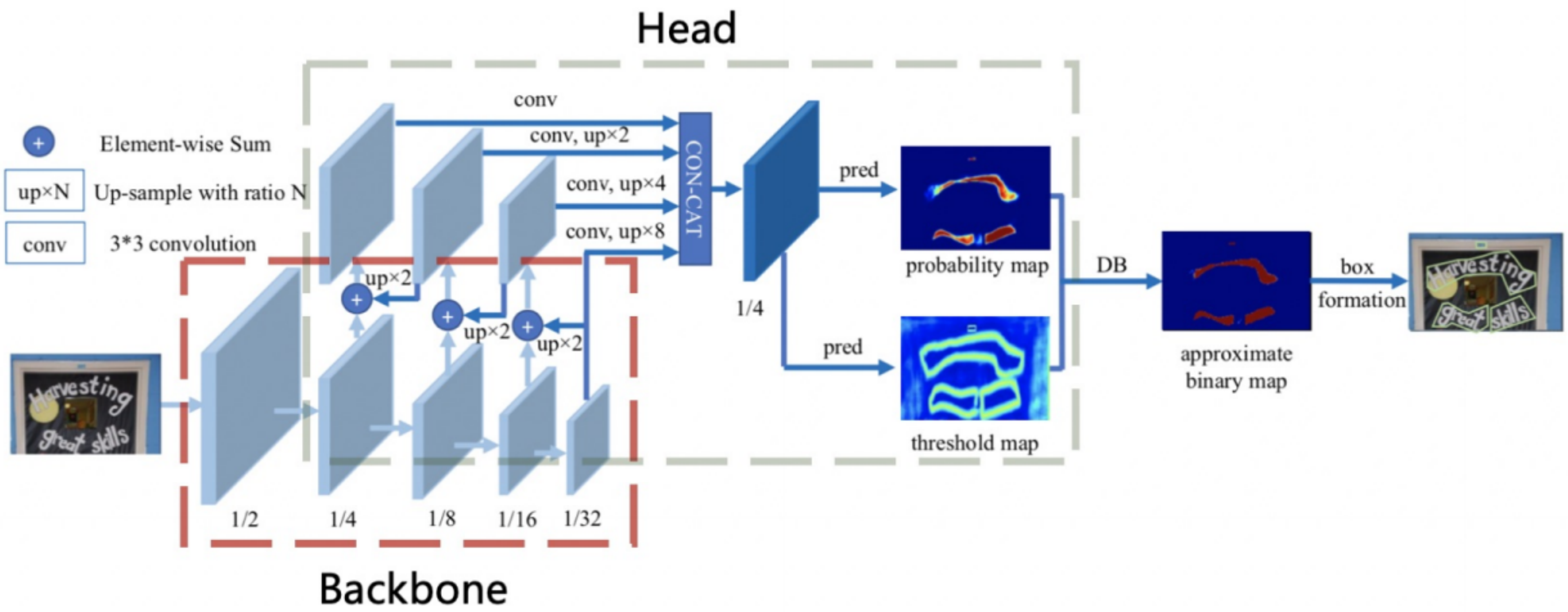


Figure 5: Architecture of the text detector DB. This figure comes from the paper of DB (Liao et al. 2020). The red and gray rectangles show the backbone and head of the text detector separately.

PP-OCR Inference Results on Reversed Images