Best Practices for a Handwritten Text Recognition System

George Retsinas, Giorgos Sfikas, Basilis Gatos and Christophoros Nikou
National Technical University of Athens, NCSR Demokritos & University of Ioannina
gretsinas@central.ntua.gr, sfikas@cs.uoi.gr, bgat@iit.demokritos.gr, cnikou@cs.uoi.gr

Motivation

Task: line-level/word-level Handwritten Text Recognition (HTR)
Motivation: Revisit basic concepts/practices of typical HTR systems

3 Directions:
- Preprocessing Steps
- Architectural Choices
- Training Procedure

Covering the complete pipeline of a modern DNN-based system

Proposed modifications are orthogonal to the majority of existing approaches

Preprocessing

Three simple steps:

1. Input images should have a fixed resolution (e.g. 128×1024 for text-line images)

    If *image size* > *fixed resolution*: **Pad** images with background color

    else: **Resize** image to fixed resolution

Preprocessing

Why padding? *Fully-utilize GPU capabilities*

Three simple steps:

1. Input images should have a <u>fixed resolution</u> (e.g. 128×1024 for text-line images)

   If *image size < fixed resolution*: **Pad** images with background color

   else: **Resize** image to fixed resolution

Preserve aspect-ratio if possible!

Preprocessing

Why padding? *Fully-utilize GPU capabilities*

Three simple steps:

1. Input images should have a fixed resolution (e.g. 128×1024 for text-line images)

   If *image size* < *fixed resolution*: **Pad** images with background color

   else: **Resize** image to fixed resolution

Preserve aspect-ratio if possible!

2. Augmentations:

   small affine deformations & Gaussian noise

Typical augmentation step

Preprocessing

Why padding? *Fully-utilize GPU capabilities*

Three simple steps:

1. Input images should have a fixed resolution (e.g. 128×1024 for text-line images)

   If *image size < fixed resolution*: **Pad** images with background color

   else: **Resize** image to fixed resolution

   Preserve aspect-ratio if possible!

2. Augmentations:

   small affine deformations & Gaussian noise

   Typical augmentation step

3. Text padding:

   add space before and after of each transcription

   Corresponds to image padding
   We expect to find spaces!

Preprocessing

Why padding? *Fully-utilize GPU capabilities*

Three simple steps:

1. Input images should have a <u>fixed resolution</u> (e.g. 128×1024 for text-line images)

   If *image size < fixed resolution*: **Pad** images with background color

   else: **Resize** image to fixed resolution

   Preserve aspect-ratio if possible!

2. Augmentations:

   small affine deformations & Gaussian noise

   Typical augmentation step

3. Text padding:

   add space before and after of each transcription

   The extra spaces are removed during evaluation

   Corresponds to image padding
   We expect to find spaces!

Network Architecture

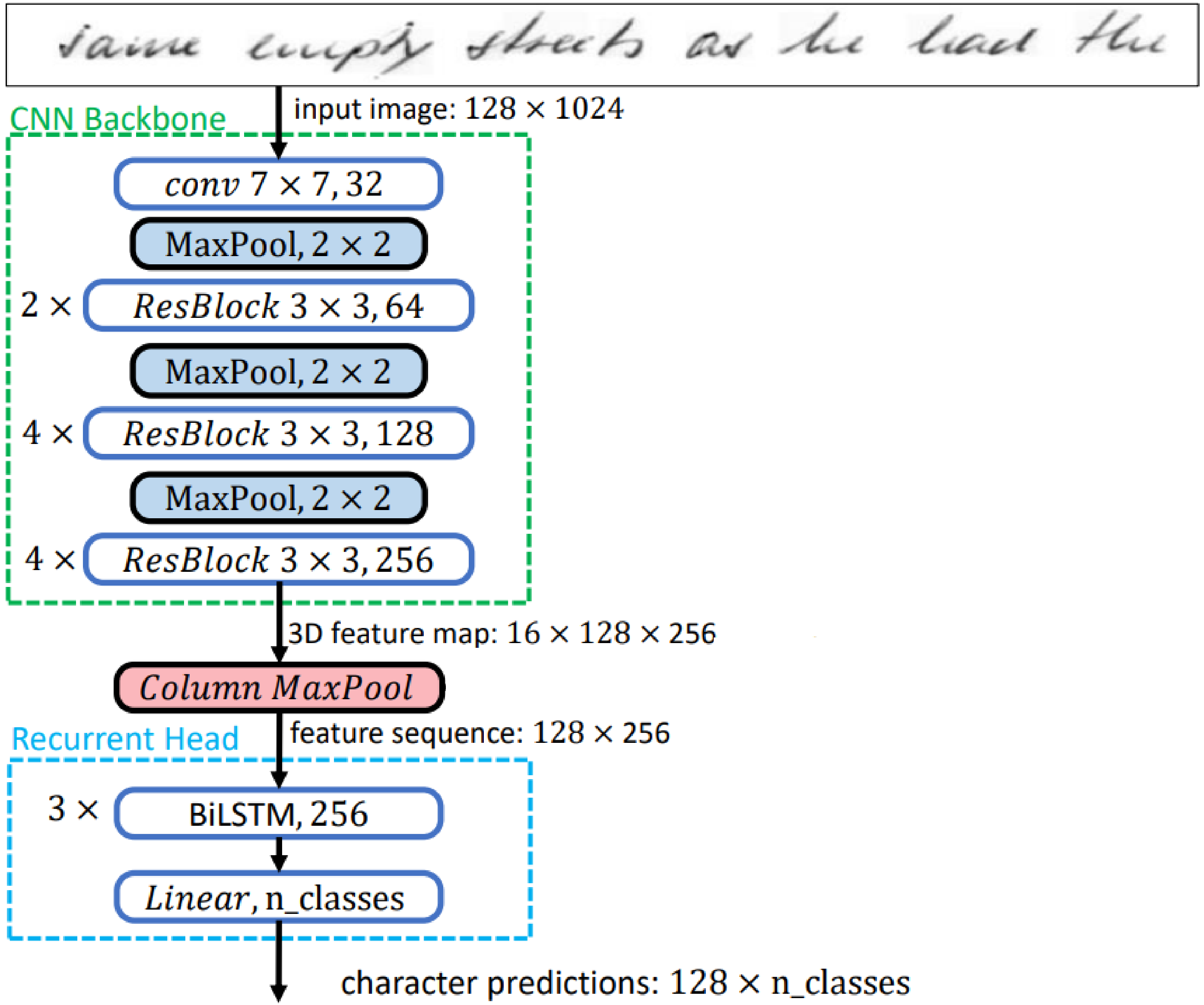
Convolutional-Recurrent Architecture:

- Convolutional Backbone
- Flattening Operation
- Recurrent Head

Network Architecture

Convolutional-Recurrent Architecture:

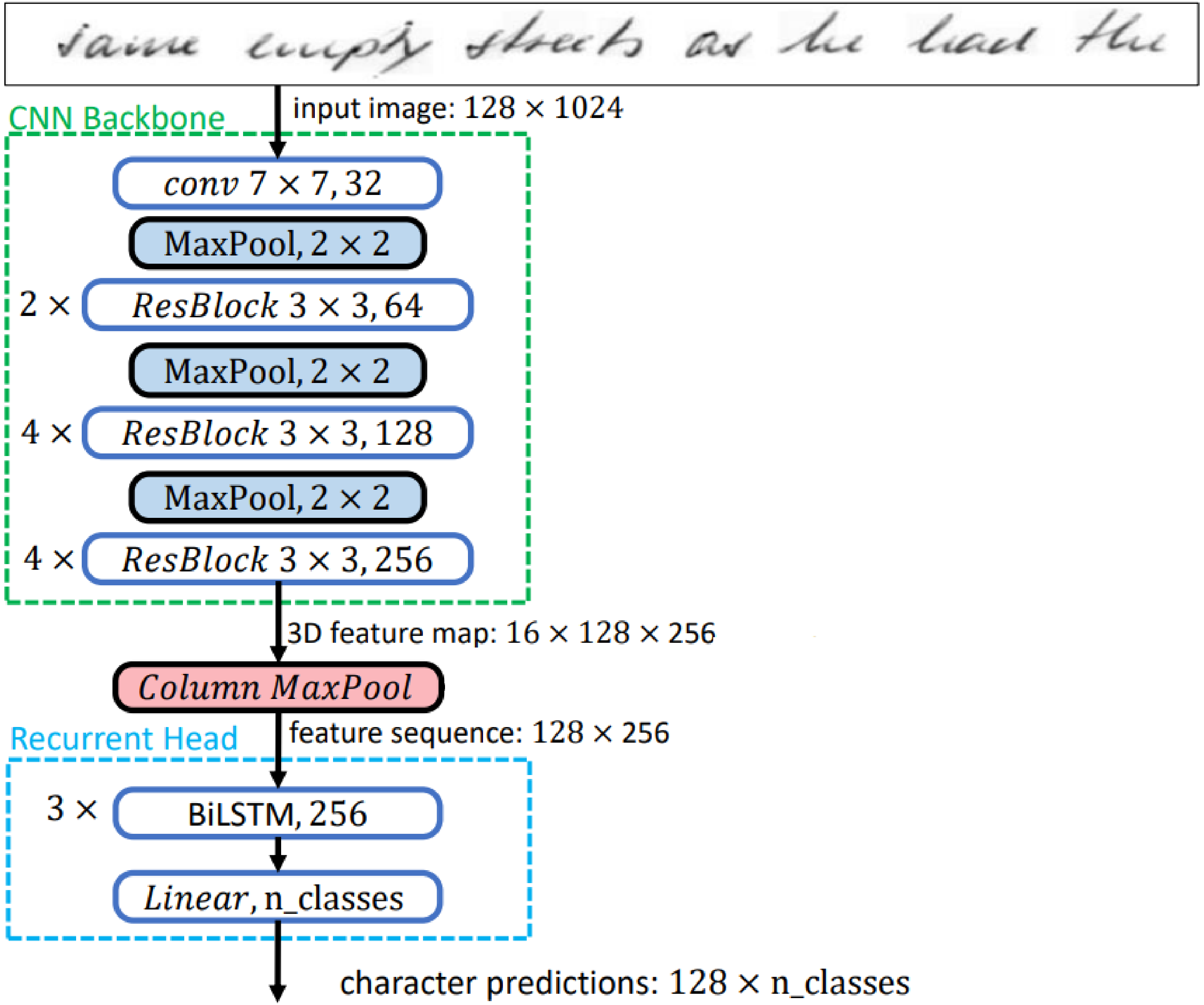
- Convolutional Backbone
- Flattening Operation
- Recurrent Head



Network Architecture

Convolutional-Recurrent Architecture:

- Convolutional Backbone
- Flattening Operation
- Recurrent Head



✓ ResNet-like CNN backbone
✓ BiLSTM head of 3 layers
✓ Training with CTC loss

Network Architecture

Convolutional-Recurrent Architecture:

- Convolutional Backbone
- Flattening Operation
- Recurrent Head

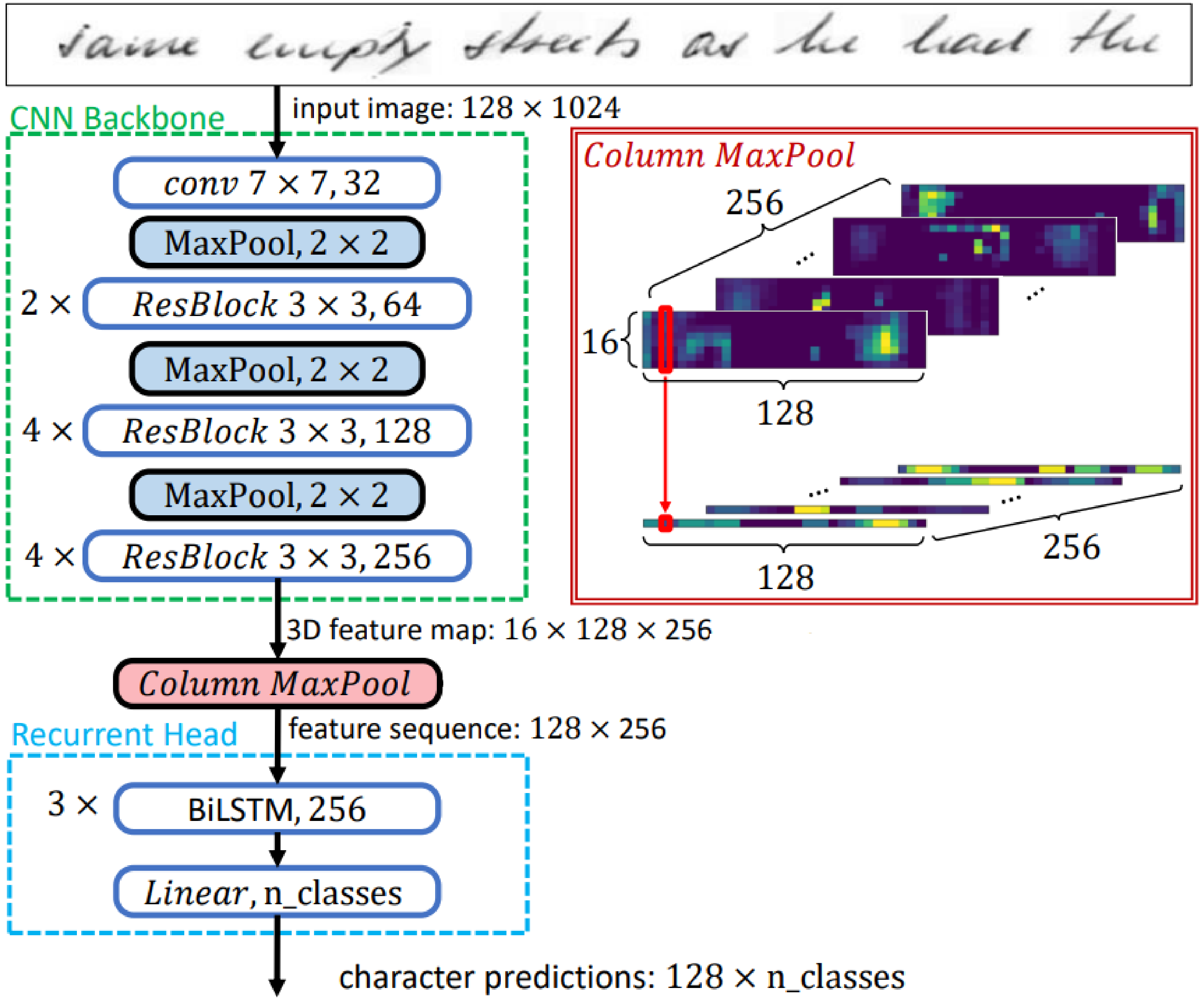
typical flattening operation!

column-wise concatenation

vs

column-wise max-pooling

proposed flattening operation!



Convolutional-Recurrent Architecture:

- Convolutional Backbone
- Flattening Operation
- Recurrent Head

typical flattening operation!

column-wise concatenation

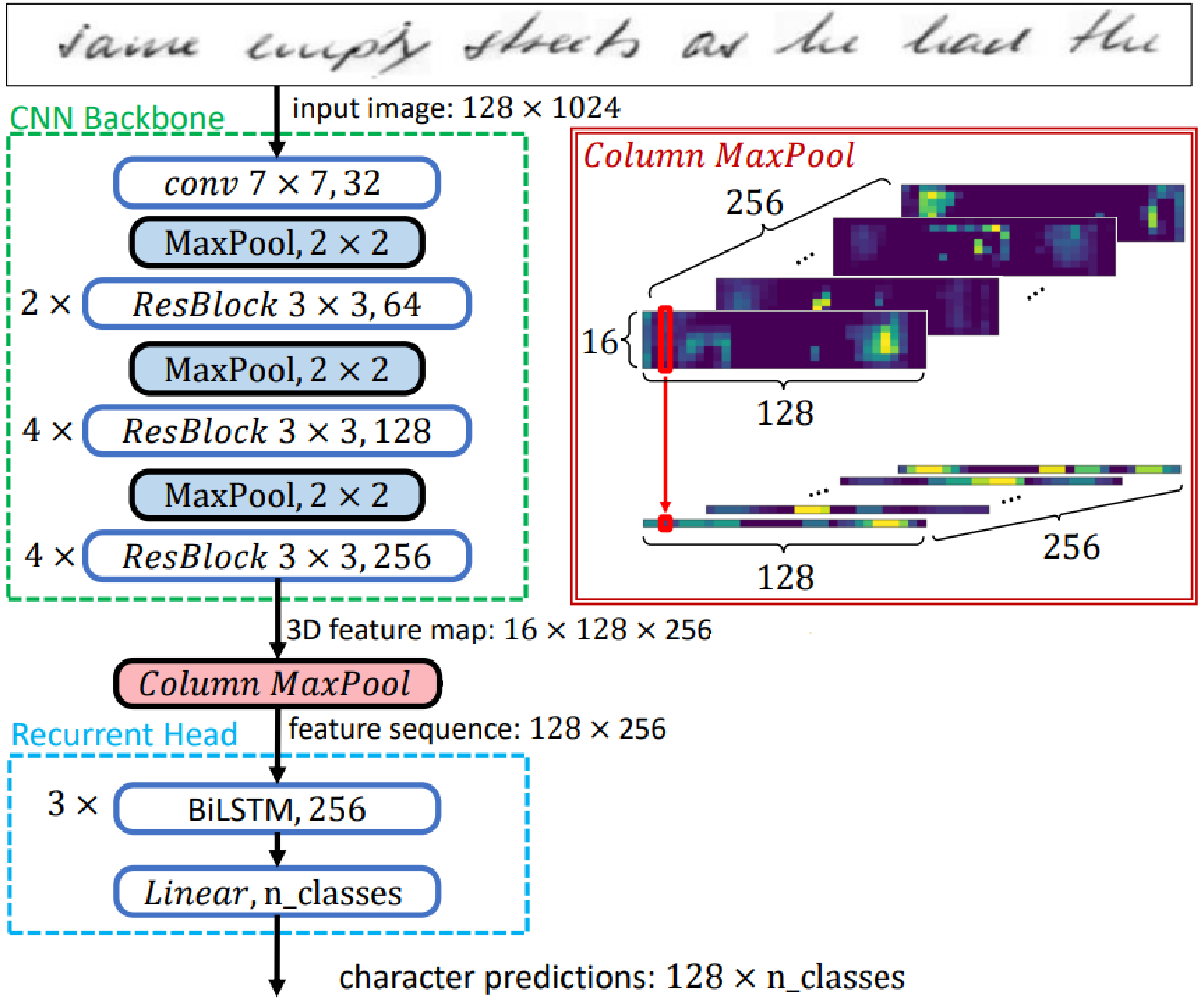
vs

column-wise max-pooling

proposed flattening operation!

Why?

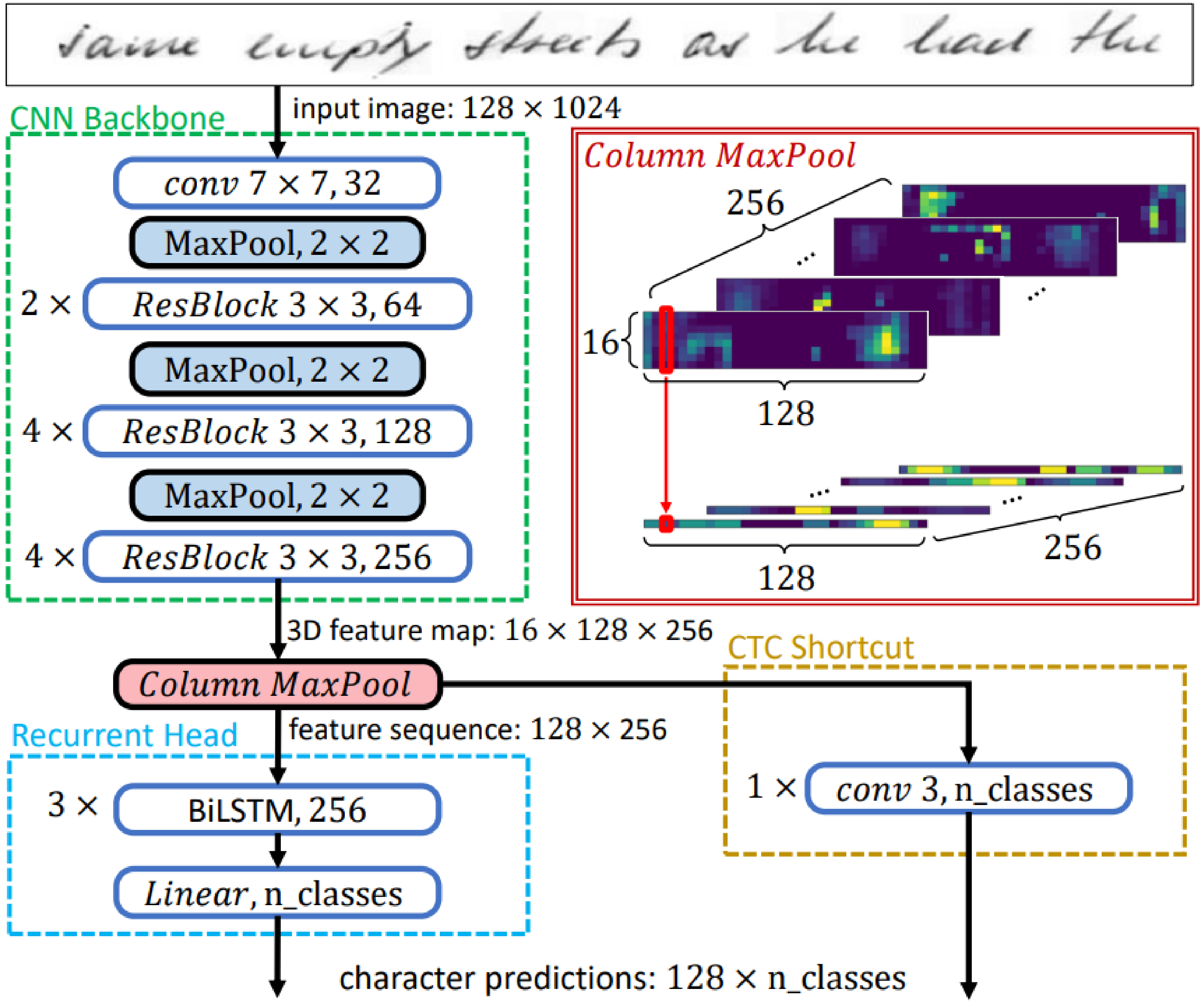
- CNN has already found features of higher receptive fields.
- Character position in the y-axis does not affect HTR performance
- Cheaper!



Training Procedure

Proposed Modification: **CTC shortcut**

Intuition: assist the training of the recurrent module by providing an alternative (simple) decoding path

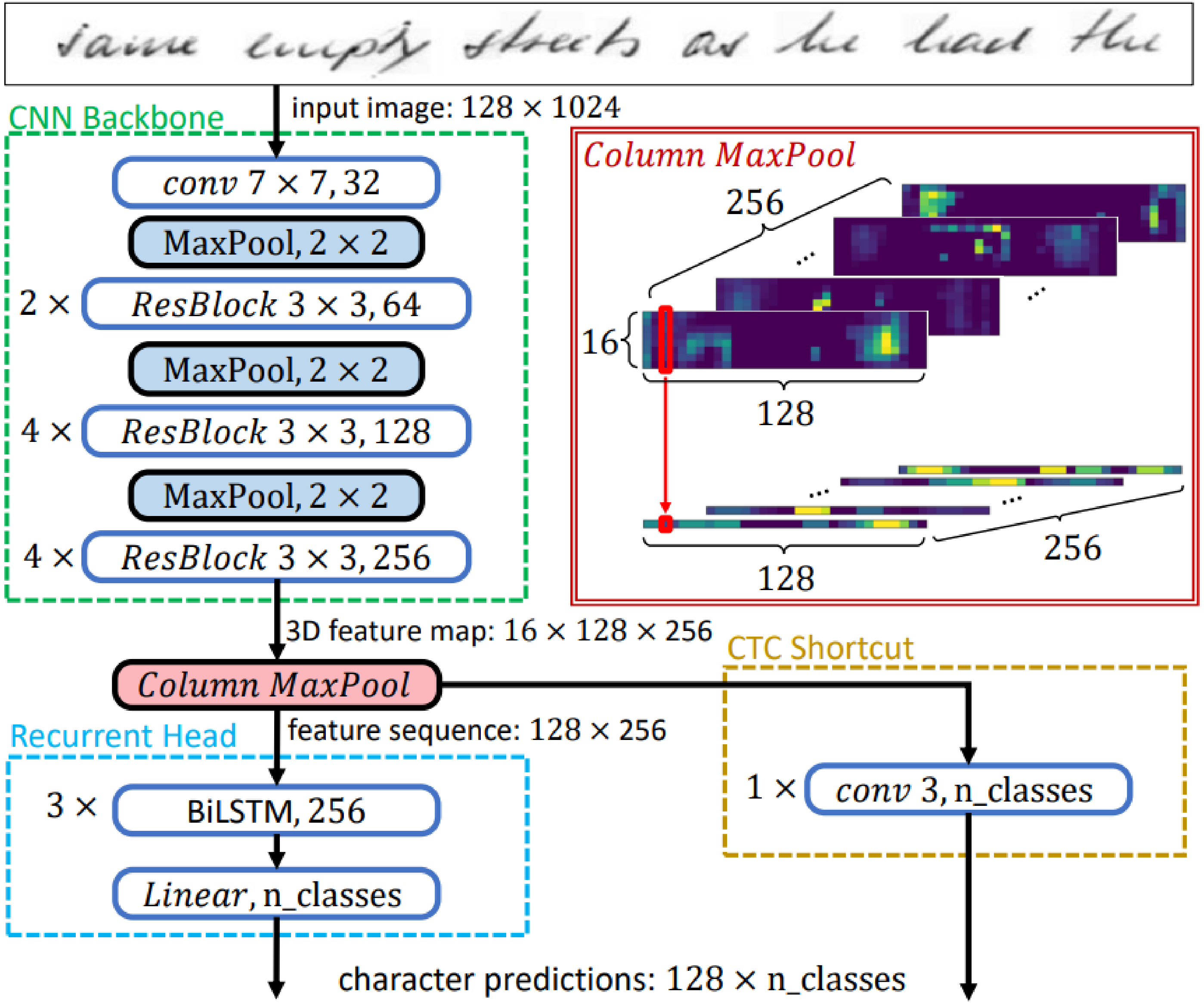


Training Procedure

Proposed Modification: **CTC shortcut**

Intuition: assist the training of the recurrent module by providing an alternative (simple) decoding path

CTC shortcut module consists only of
a single 1D convolutional layer, with kernel size 3



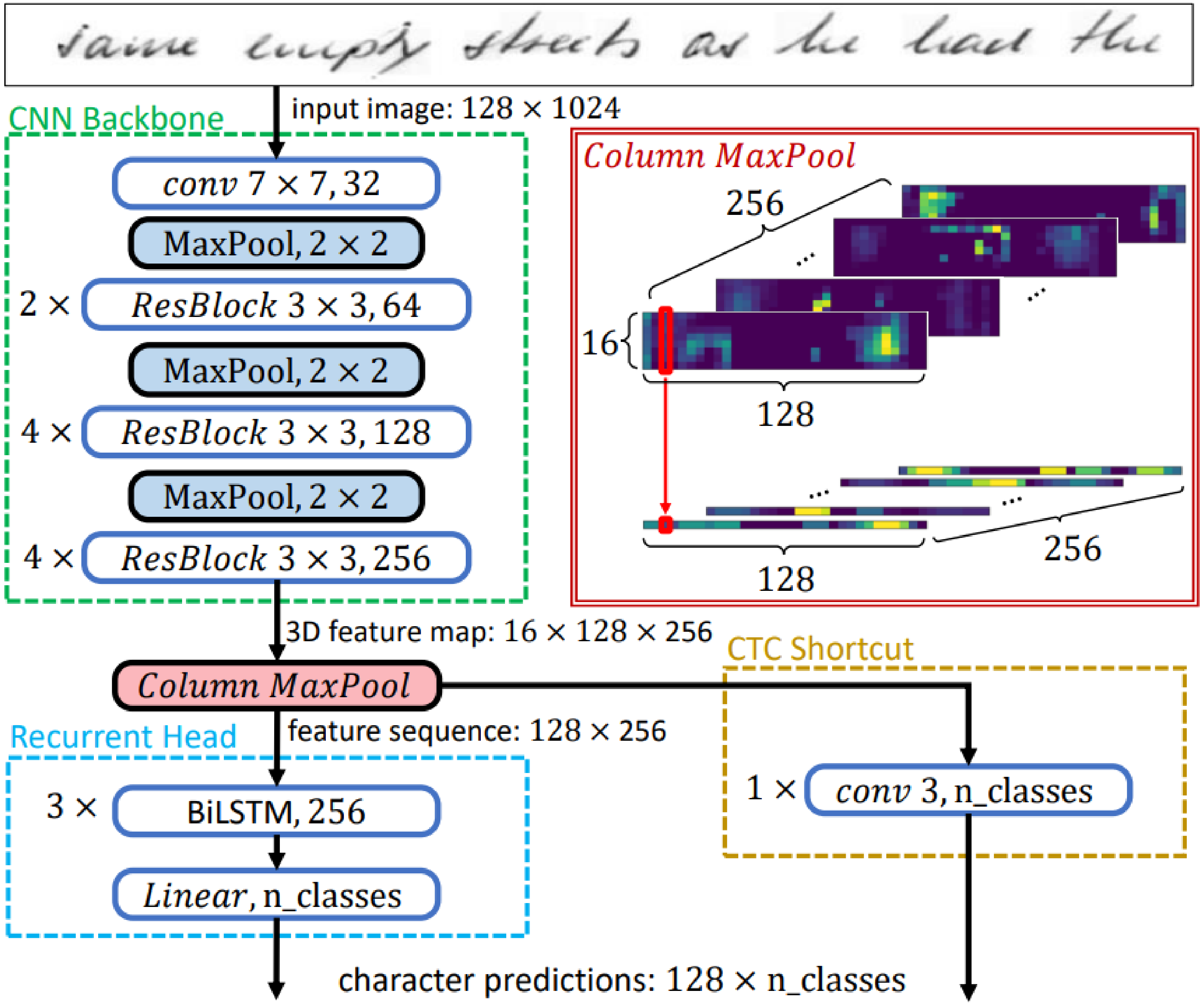
Training Procedure

Proposed Modification: **CTC shortcut**

Intuition: assist the training of the recurrent module by providing an alternative (simple) decoding path

CTC shortcut module consists only of a single 1D convolutional layer, with kernel size 3

Quickly generate discriminative features at the top of the CNN backbone through the straightforward 1D convolutional path, simplifying the training task for the recurrent part.



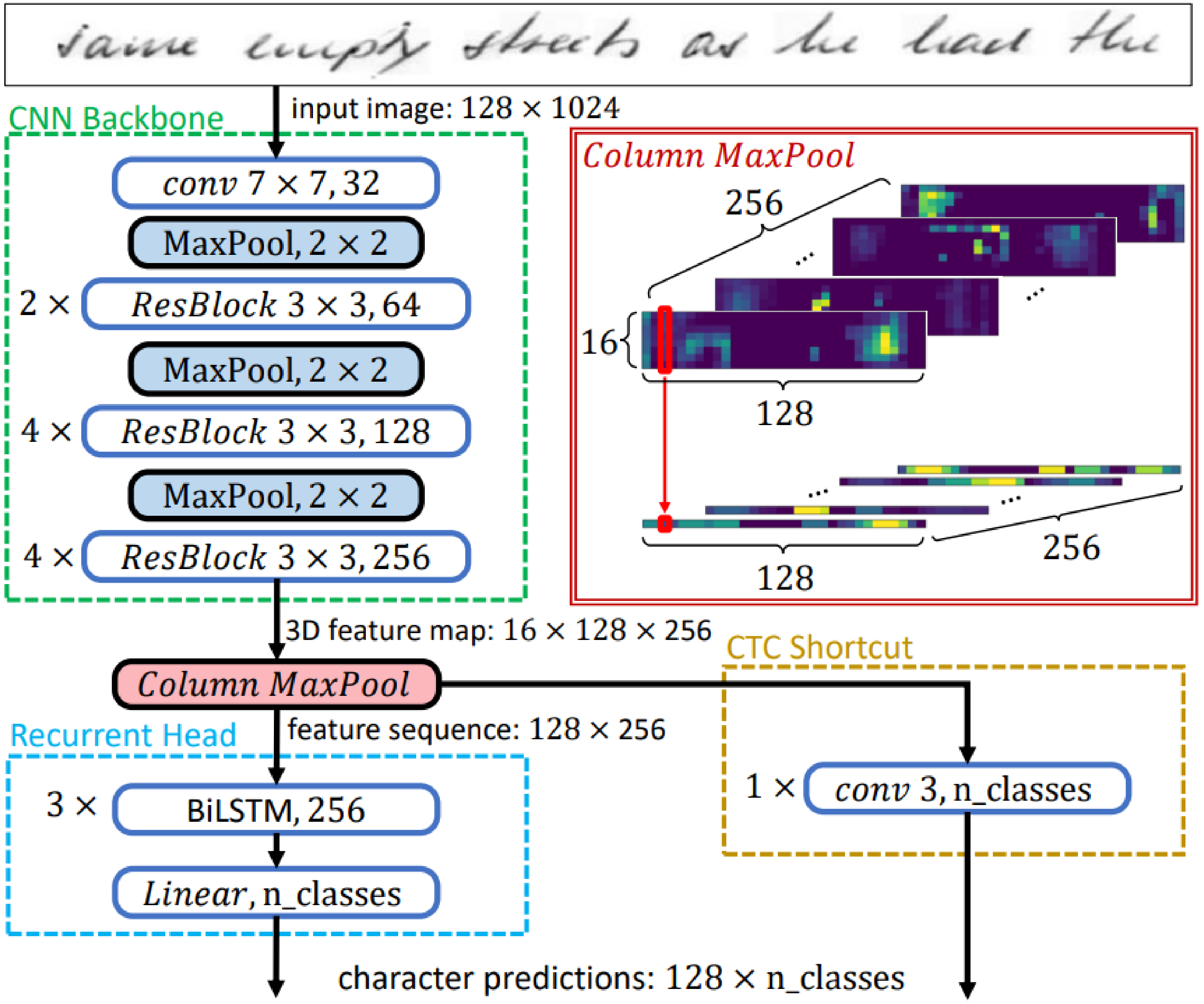
Training Procedure

Proposed Modification: **CTC shortcut**

Intuition: assist the training of the recurrent module by providing an alternative (simple) decoding path

CTC shortcut module consists only of a single 1D convolutional layer, with kernel size 3

Quickly generate discriminative features at the top of the CNN backbone through the straightforward 1D convolutional path, simplifying the training task for the recurrent part.



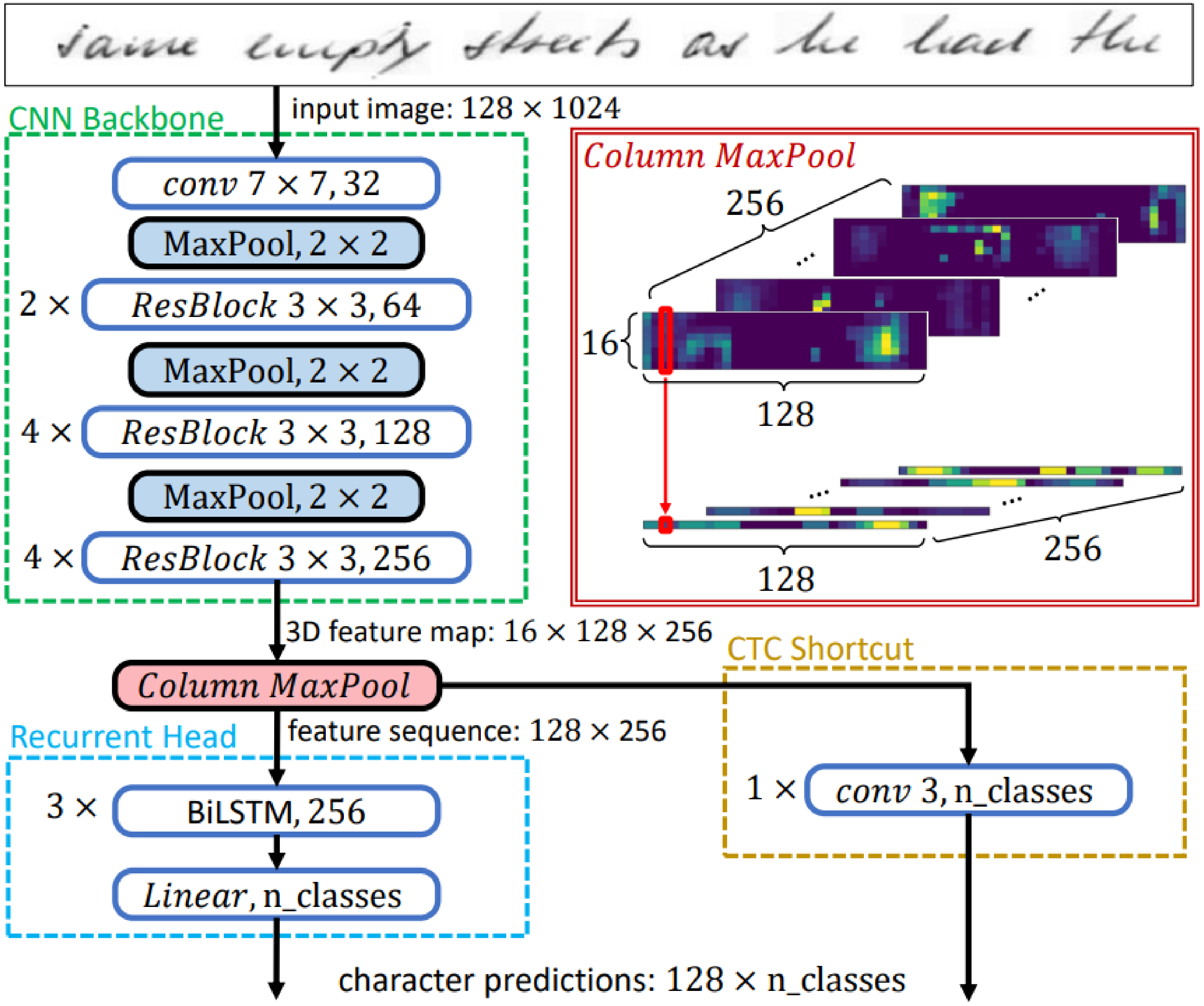
Trained with Multi-task loss:

$$L_{CTC}(f_{rec}(f_{cnn}(I)); s) + 0.1\, L_{CTC}(f_{shortcut}(f_{cnn}(I)); s)$$

Training Procedure

CTC shortcut only assists training!
Omitted during evaluation!

Inference time is not affected!



$$L_{CTC}(f_{rec}(f_{cnn}(I)); s) + 0.1\, L_{CTC}(f_{shortcut}(f_{cnn}(I)); s)$$

Ablation

Line-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.28	15.29	5.93	19.57
		yes	3.72	13.18	5.11	16.96
resized	max-pooling	no	3.73	13.54	5.28	17.77
		yes	3.47	12.77	4.85	16.19
padded	concatenation	no	4.06	14.40	5.54	18.60
		yes	3.37	12.22	4.71	15.94
padded	max-pooling	no	3.46	12.55	4.93	16.81
		yes	**3.21**	**11.89**	**4.62**	**15.89**

Word-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.35	12.55	5.58	15.46
		yes	4.27	12.02	5.46	15.13
resized	max-pooling	no	4.25	12.17	5.69	15.87
		yes	4.09	11.65	5.23	14.40
padded	concatenation	no	4.17	11.99	5.66	15.66
		yes	3.98	11.50	5.37	14.98
padded	max-pooling	no	4.00	11.25	5.43	15.06
		yes	**3.76**	**10.76**	**5.14**	**14.33**

- Adam optimizer
- 1e-3 initial lr
- 240 epochs
- Multistep scheduler

×0.1 @ epochs 120 & 180

Ablation

Line-level (IAM):

			Validation		Test	
Preprocessing	Flattening	CTC Shortcut	CER(%)	WER(%)	CER(%)	WER(%)
resized	concatenation	no	4.28	15.29	5.93	19.57
		yes	3.72	13.18	5.11	16.96
resized	max-pooling	no	3.73	13.54	5.28	17.77
		yes	3.47	12.77	4.85	16.19
padded	concatenation	no	4.06	14.40	5.54	18.60
		yes	3.37	12.22	4.71	15.94
padded	max-pooling	no	3.46	12.55	4.93	16.81
		yes	**3.21**	**11.89**	**4.62**	**15.89**

Word-level (IAM):

			Validation		Test	
Preprocessing	Flattening	CTC Shortcut	CER(%)	WER(%)	CER(%)	WER(%)
resized	concatenation	no	4.35	12.55	5.58	15.46
		yes	4.27	12.02	5.46	15.13
resized	max-pooling	no	4.25	12.17	5.69	15.87
		yes	4.09	11.65	5.23	14.40
padded	concatenation	no	4.17	11.99	5.66	15.66
		yes	3.98	11.50	5.37	14.98
padded	max-pooling	no	4.00	11.25	5.43	15.06
		yes	**3.76**	**10.76**	**5.14**	**14.33**

- ✓ Padding > Resizing

Ablation

Line-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.28	15.29	5.93	19.57
		yes	3.72	13.18	5.11	16.96
resized	max-pooling	no	3.73	13.54	5.28	17.77
		yes	3.47	12.77	4.85	16.19
padded	concatenation	no	4.06	14.40	5.54	18.60
		yes	3.37	12.22	4.71	15.94
padded	max-pooling	no	3.46	12.55	4.93	16.81
		yes	**3.21**	**11.89**	**4.62**	**15.89**

✓ Padding > Resizing

Not in word-level test set

Word-level resizing not as critical

Word-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.35	12.55	5.58	15.46
		yes	4.27	12.02	5.46	15.13
resized	max-pooling	no	4.25	12.17	5.69	15.87
		yes	4.09	11.65	5.23	14.40
padded	concatenation	no	4.17	11.99	5.66	15.66
		yes	3.98	11.50	5.37	14.98
padded	max-pooling	no	4.00	11.25	5.43	15.06
		yes	**3.76**	**10.76**	**5.14**	**14.33**

Ablation

Line-level (IAM):

			Validation		Test	
Preprocessing	Flattening	CTC Shortcut	CER(%)	WER(%)	CER(%)	WER(%)
resized	concatenation	no	4.28	15.29	5.93	19.57
		yes	3.72	13.18	5.11	16.96
resized	max-pooling	no	3.73	13.54	5.28	17.77
		yes	3.47	12.77	4.85	16.19
padded	concatenation	no	4.06	14.40	5.54	18.60
		yes	3.37	12.22	4.71	15.94
padded	max-pooling	no	3.46	12.55	4.93	16.81
		yes	**3.21**	**11.89**	**4.62**	**15.89**

Word-level (IAM):

			Validation		Test	
Preprocessing	Flattening	CTC Shortcut	CER(%)	WER(%)	CER(%)	WER(%)
resized	concatenation	no	4.35	12.55	5.58	15.46
		yes	4.27	12.02	5.46	15.13
resized	max-pooling	no	4.25	12.17	5.69	15.87
		yes	4.09	11.65	5.23	14.40
padded	concatenation	no	4.17	11.99	5.66	15.66
		yes	3.98	11.50	5.37	14.98
padded	max-pooling	no	4.00	11.25	5.43	15.06
		yes	**3.76**	**10.76**	**5.14**	**14.33**

- ✓ Padding > Resizing
- ✓ Max-Pooling > Concat

Ablation

Line-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.28	15.29	5.93	19.57
		yes	3.72	13.18	5.11	16.96
resized	max-pooling	no	3.73	13.54	5.28	17.77
		yes	3.47	12.77	4.85	16.19
padded	concatenation	no	4.06	14.40	5.54	18.60
		yes	3.37	12.22	4.71	15.94
padded	max-pooling	no	3.46	12.55	4.93	16.81
		yes	**3.21**	**11.89**	**4.62**	**15.89**

Word-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.35	12.55	5.58	15.46
		yes	4.27	12.02	5.46	15.13
resized	max-pooling	no	4.25	12.17	5.69	15.87
		yes	4.09	11.65	5.23	14.40
padded	concatenation	no	4.17	11.99	5.66	15.66
		yes	3.98	11.50	5.37	14.98
padded	max-pooling	no	4.00	11.25	5.43	15.06
		yes	**3.76**	**10.76**	**5.14**	**14.33**

- ✓ Padding > Resizing
- ✓ Max-Pooling > Concat
- ✓ ***CTC shortcut consistently improves performance!***

Ablation

Line-level (IAM):

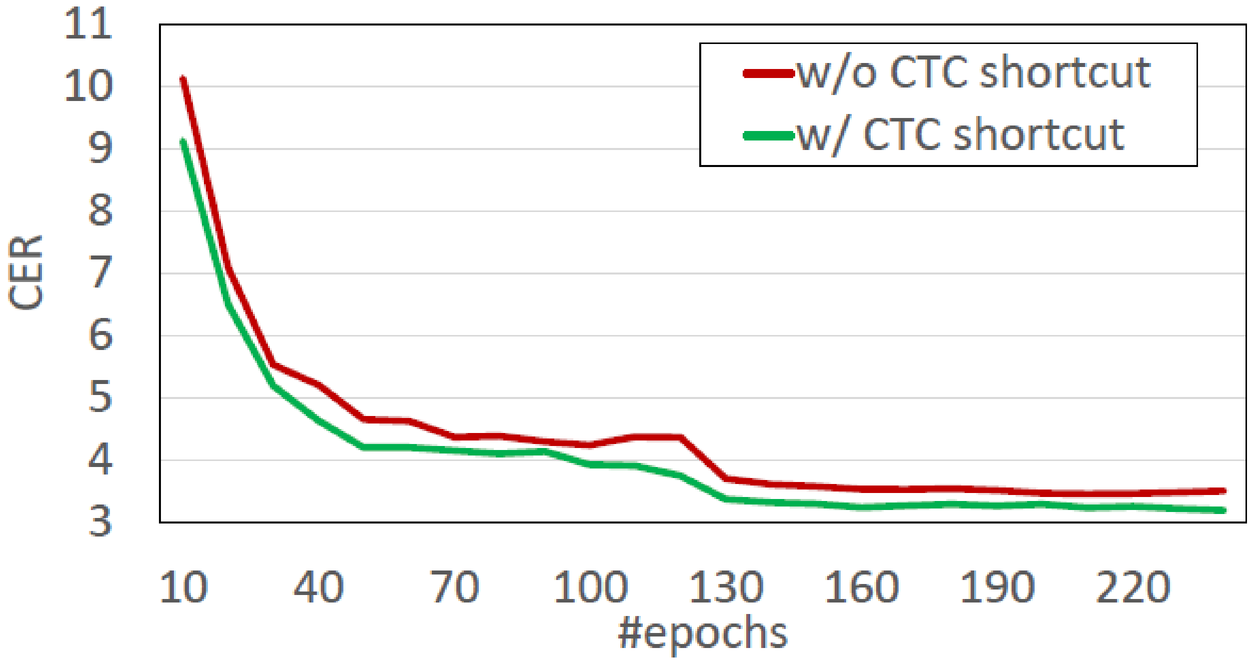
Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.28	15.29	5.93	19.57
		yes	3.72	13.18	5.11	16.96
resized	max-pooling	no	3.73	13.54	5.28	17.77
		yes	3.47	12.77	4.85	16.19
padded	concatenation	no	4.06	14.40	5.54	18.60
		yes	3.37	12.22	4.71	15.94
padded	max-pooling	no	3.46	12.55	4.93	16.81
		yes	**3.21**	**11.89**	**4.62**	**15.89**

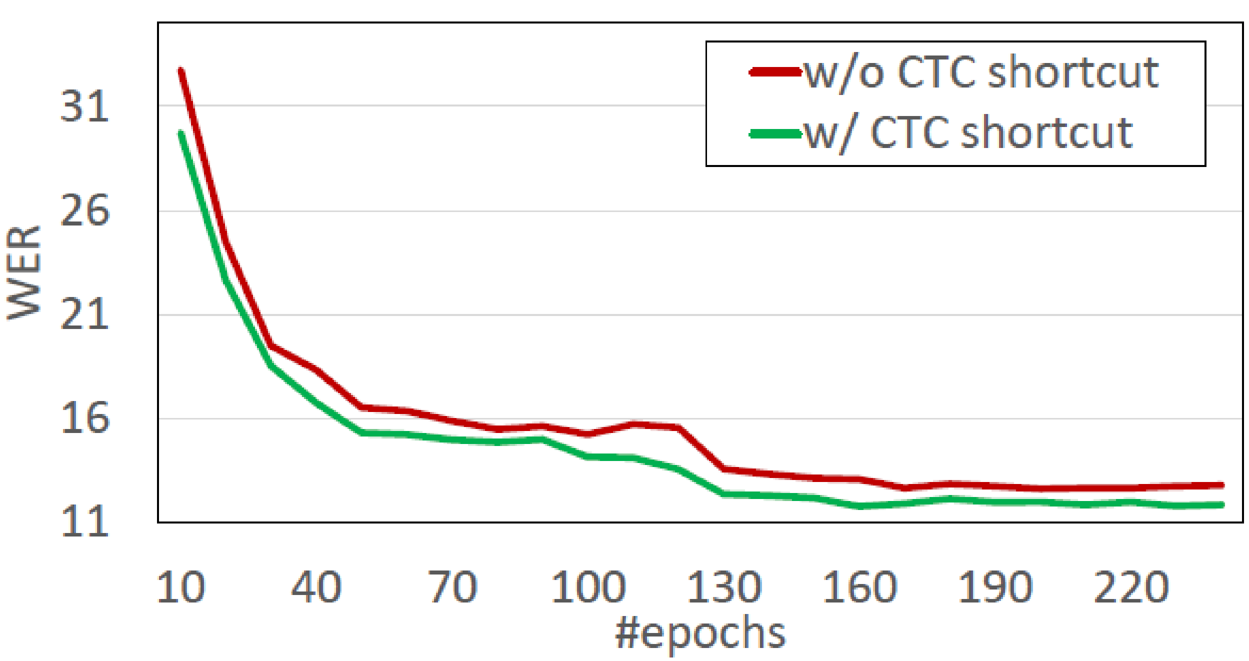
Word-level (IAM):

Preprocessing	Flattening	CTC Shortcut	Validation CER(%)	Validation WER(%)	Test CER(%)	Test WER(%)
resized	concatenation	no	4.35	12.55	5.58	15.46
		yes	4.27	12.02	5.46	15.13
resized	max-pooling	no	4.25	12.17	5.69	15.87
		yes	4.09	11.65	5.23	14.40
padded	concatenation	no	4.17	11.99	5.66	15.66
		yes	3.98	11.50	5.37	14.98
padded	max-pooling	no	4.00	11.25	5.43	15.06
		yes	**3.76**	**10.76**	**5.14**	**14.33**

- ✓ Padding > Resizing
- ✓ Max-Pooling > Concat
- ✓ ***CTC shortcut consistently improves performance!***
- ✓ *Overall gain is over 3.5% @ WER (line-level recognition)*

Ablation – CTC shortcut





Comparison to SOTA

Line-level recognition of recent SOTA approaches on both IAM and RIMES datasets

	IAM		RIMES	
Method	CER(%)	WER(%)	CER(%)	WER(%)
Chen et al.	11.15	34.55	8.29	30.5
Pham et al.	10.8	35.1	6.8	28.5
Khrishnan et al.	9.78	32.89	-	-
Chowdhury et al.	8.10	16.70	3.59	9.60
Puigcerver	6.2	20.2	2.60	10.7
Khrishnan et al.	9.78	32.89	-	-
Markou et al.	6.14	20.04	3.34	11.23
Dutta et al.	5.8	17.8	5.07	14.7
Wick et al.	5.67	-	-	-
Michael et al.	5.24	-	-	-
Tassopoulou et al.	5.18	17.68	-	-
Yousef et al.	4.9	-	-	-
Retsinas et al.	4.55	16.08	3.04	10.56
Proposed	4.62	15.89	2.75	9.93

Proposed modifications are orthogonal to the majority of existing approaches

Conclusions

Instead of adding more complex components, first assist the system to learn!

Thank you

Acknowledgements

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the calls: “RESEARCH - CREATE - INNOVATE”, project Culdile, and “OPEN INNOVATION IN CULTURE”, project Bessarion.

Comparison to SOTA

Indicative result for word-level recognition (IAM test-set):

Our method achieves 5.14% CER / 14.33% WER

vs

Luo et al. achieve 5.13% CER / 13.35% WER

complex augmentation scheme along an STN component

Luo et al., "Learn to augment: Joint data augmentation and network optimization for text recognition", CVPR, 2020