High-Performance Transformers for Table Structure Recognition Need Early Convolutions

ShengYun (Anthony)
Peng

Seongmin
Lee

Xiaojing
Wang

Rajarajeswari
Balasubramaniyan

Polo
Chau

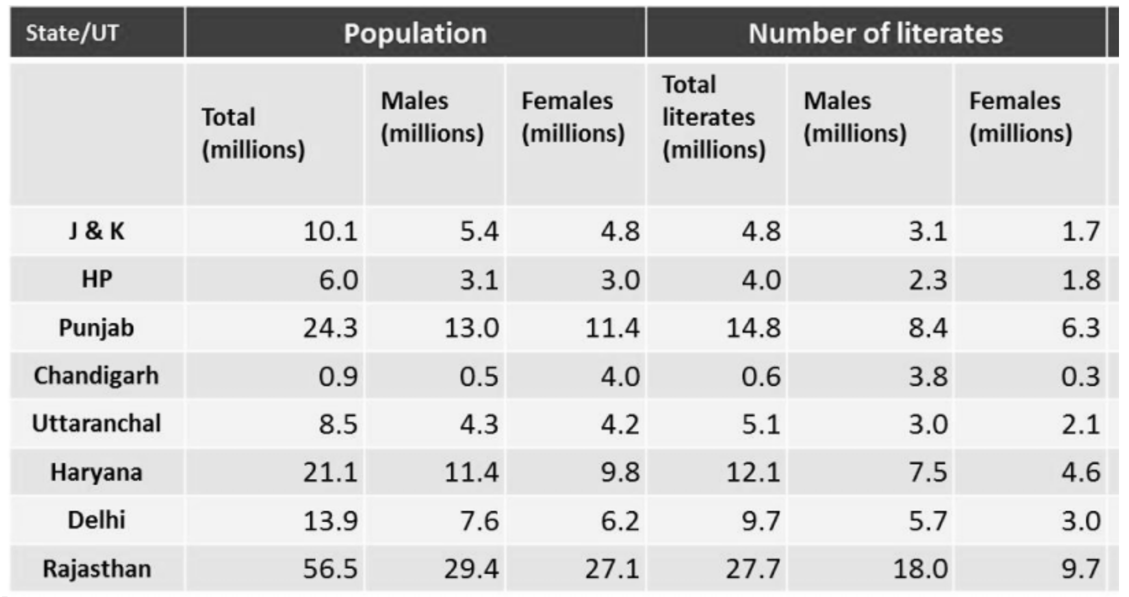
Georgia Tech

ADP

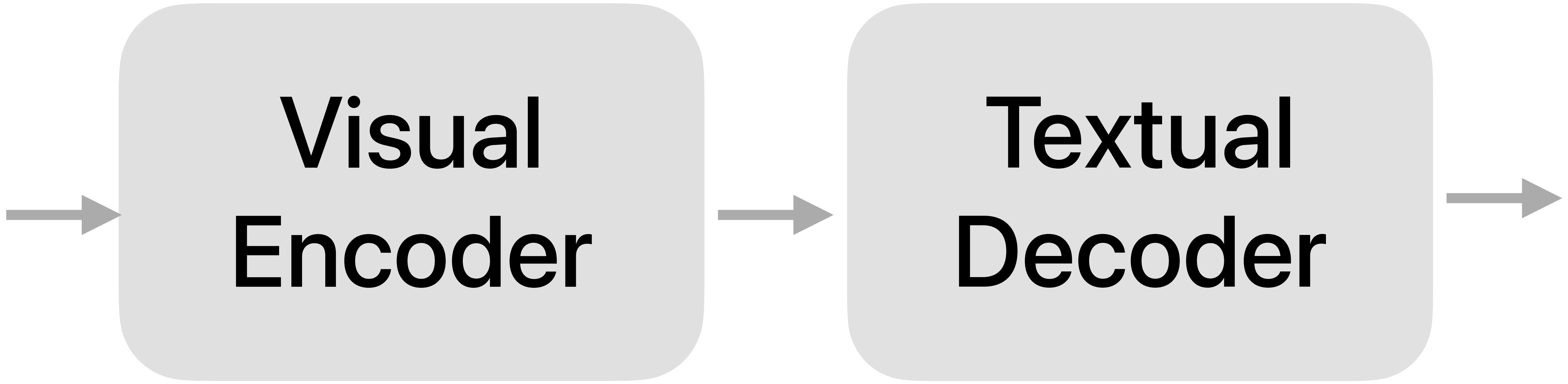
Oral Presentation at NeurIPS'23 Workshop on Table Representation Learning

Existing table structure recognition research treats task as image-to-text generation

Input Image



State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7



Visual Encoder

Textual Decoder

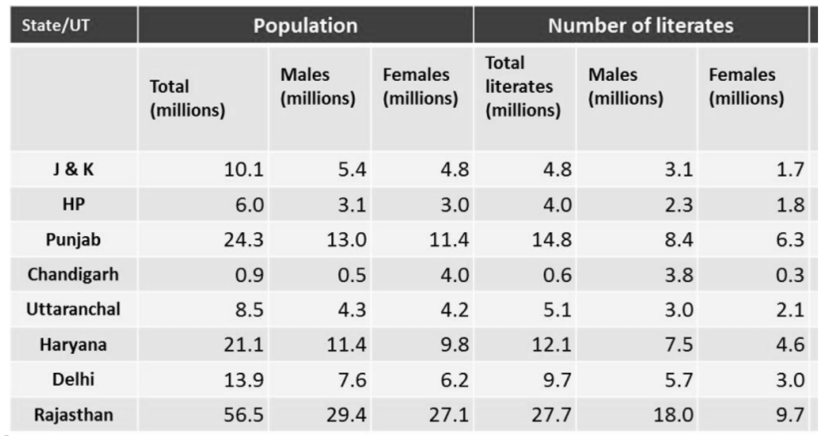
HTML Tags

<thead><tr>
<td></td>…

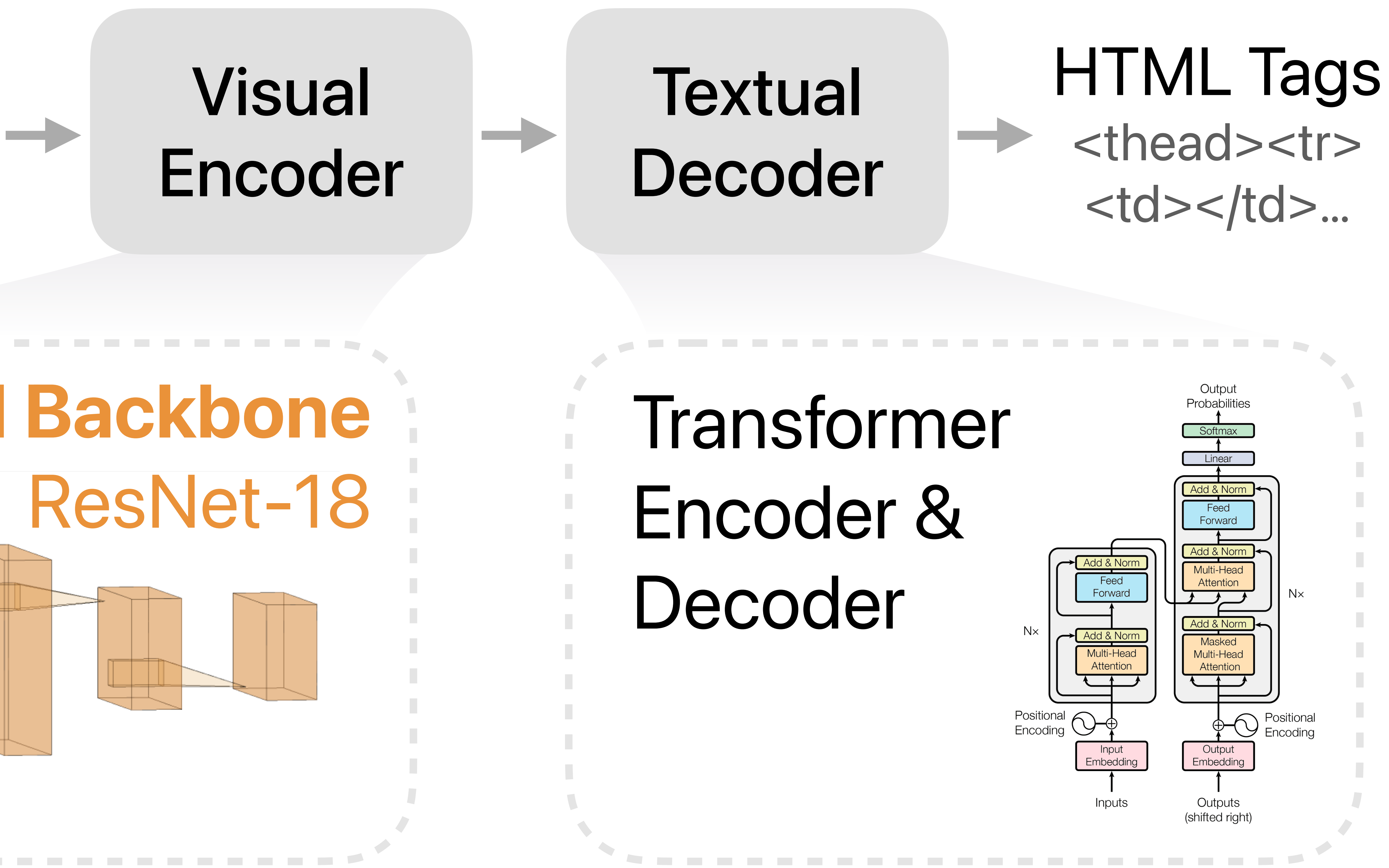
Model architecture in existing method:
Hybrid CNN-Transformer

- CNN backbone takes up ~50% of the total model parameters
- Significantly reduces both training and inference speed

Input Image



State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7



Visual Encoder

Textual Decoder

HTML Tags
<thead><tr>
<td></td>...

CNN Backbone
ResNet-18

Transformer Encoder & Decoder

But hybrid CNN-Transformers seldomly used:

Vision Transformers use simple linear projection instead of CNN backbone

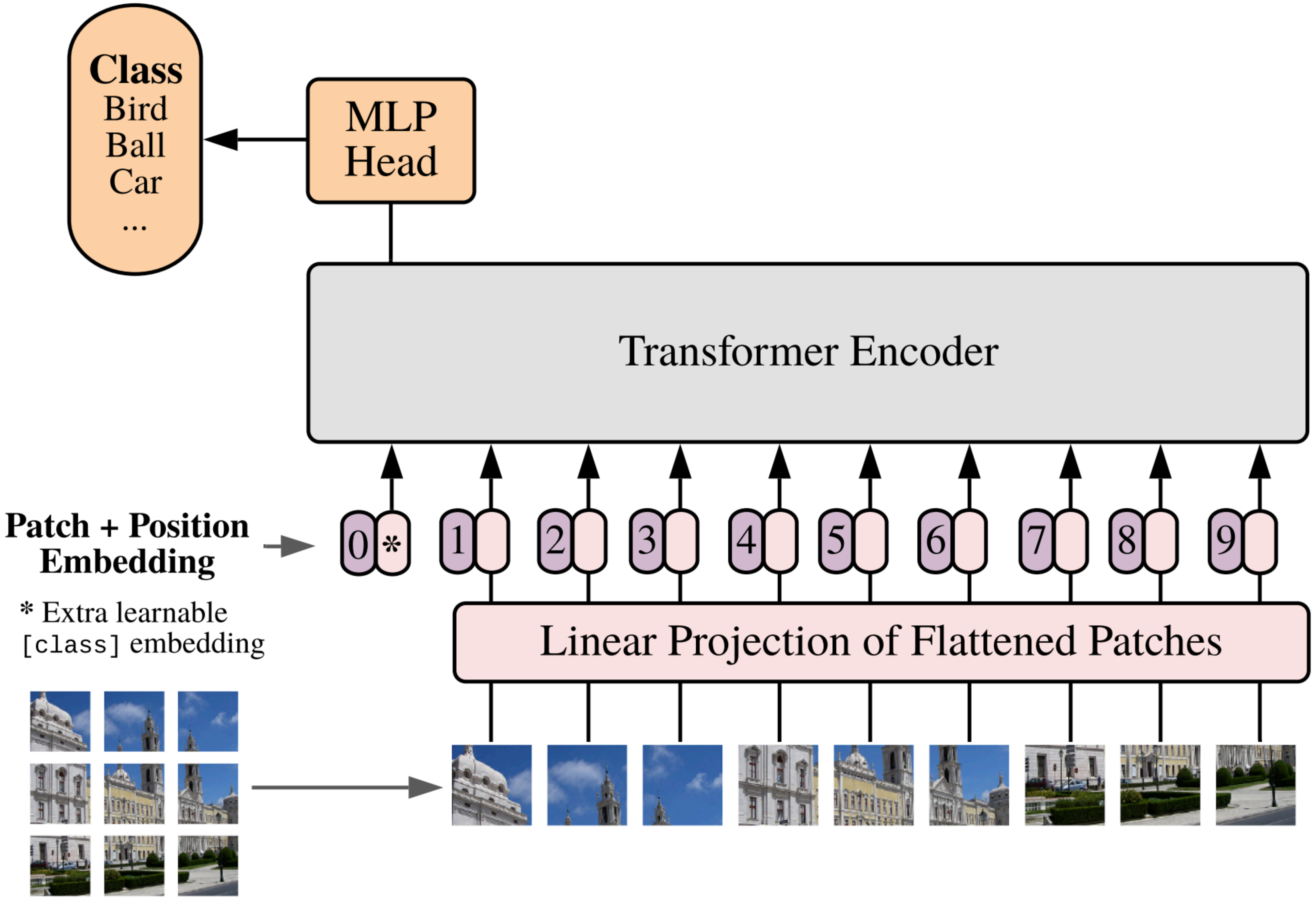
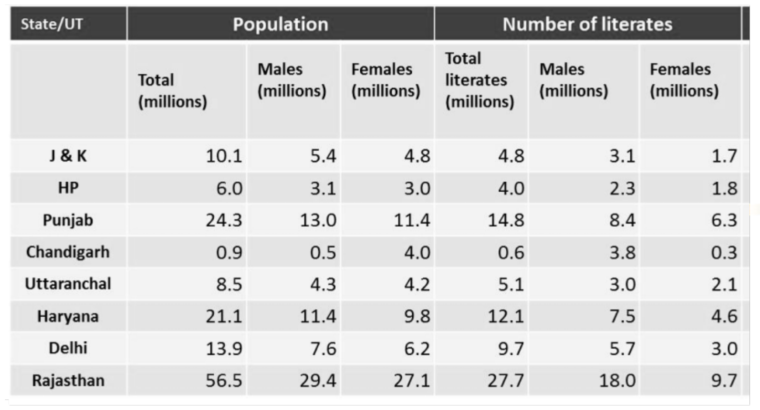

Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Image credit: A. Dosovitskiy, et. al.

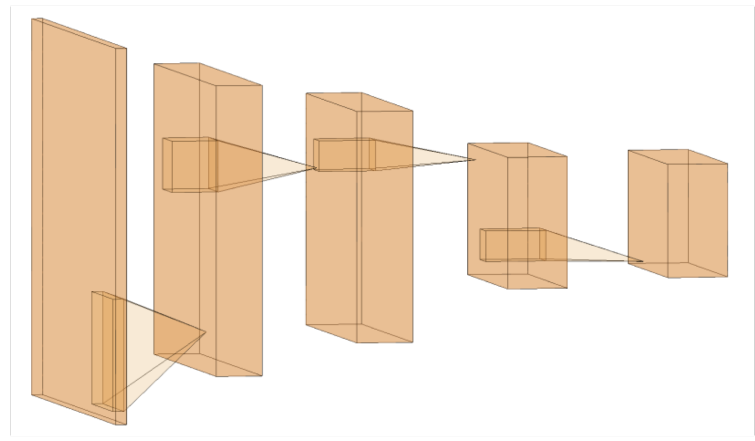
Can we simply employ the linear projection?

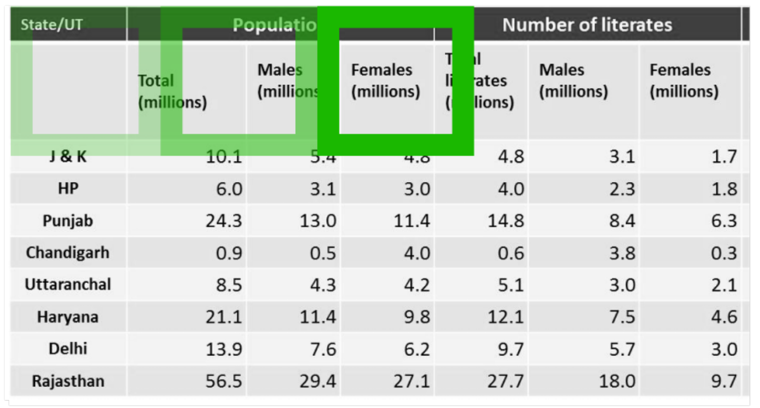
Input image



State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7

	Performance	Model Complexity
ResNet-18 CNN Backbone	High ✓ 94.50% Complex TEDS	High ✗ 28.70M #Param. 42.22G MAC
LinearProj	Low ✗ 86.62%	Lowest ✓ 22.67M 10.28G





State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7

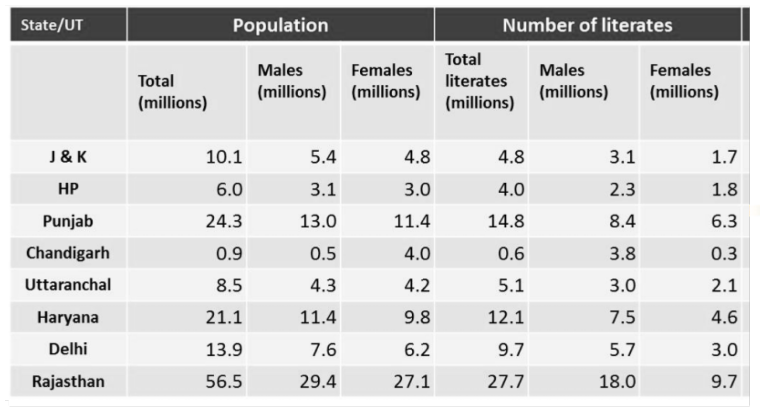
Textual Decoder

Visual Encoder Options for Table Structure Recognition

Can we simply employ the linear projection?

No, performance suffers

Input image

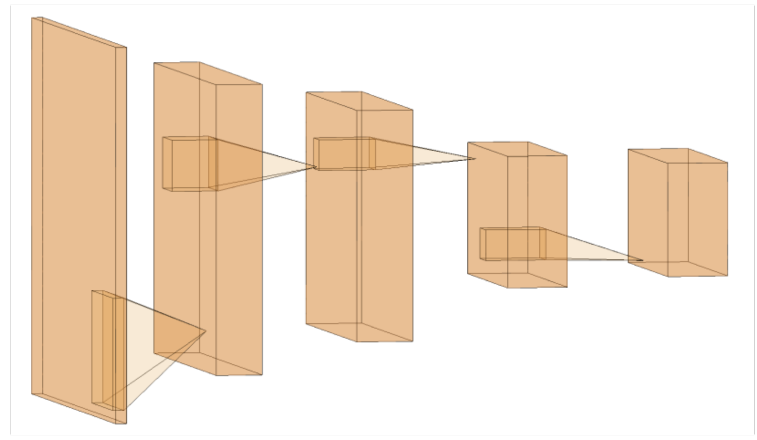


State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7

Performance

Model Complexity

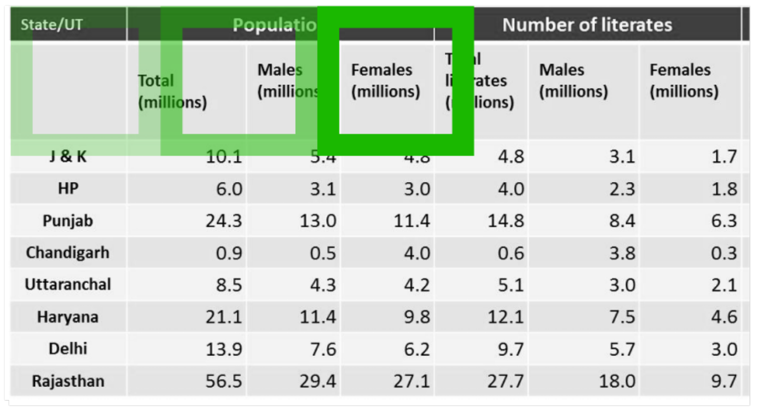
ResNet-18
CNN Backbone



High ✓
94.50%
Complex TEDS

High ✗
28.70M #Param.
42.22G MAC

LinearProj



State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7

Low ✗
86.62%

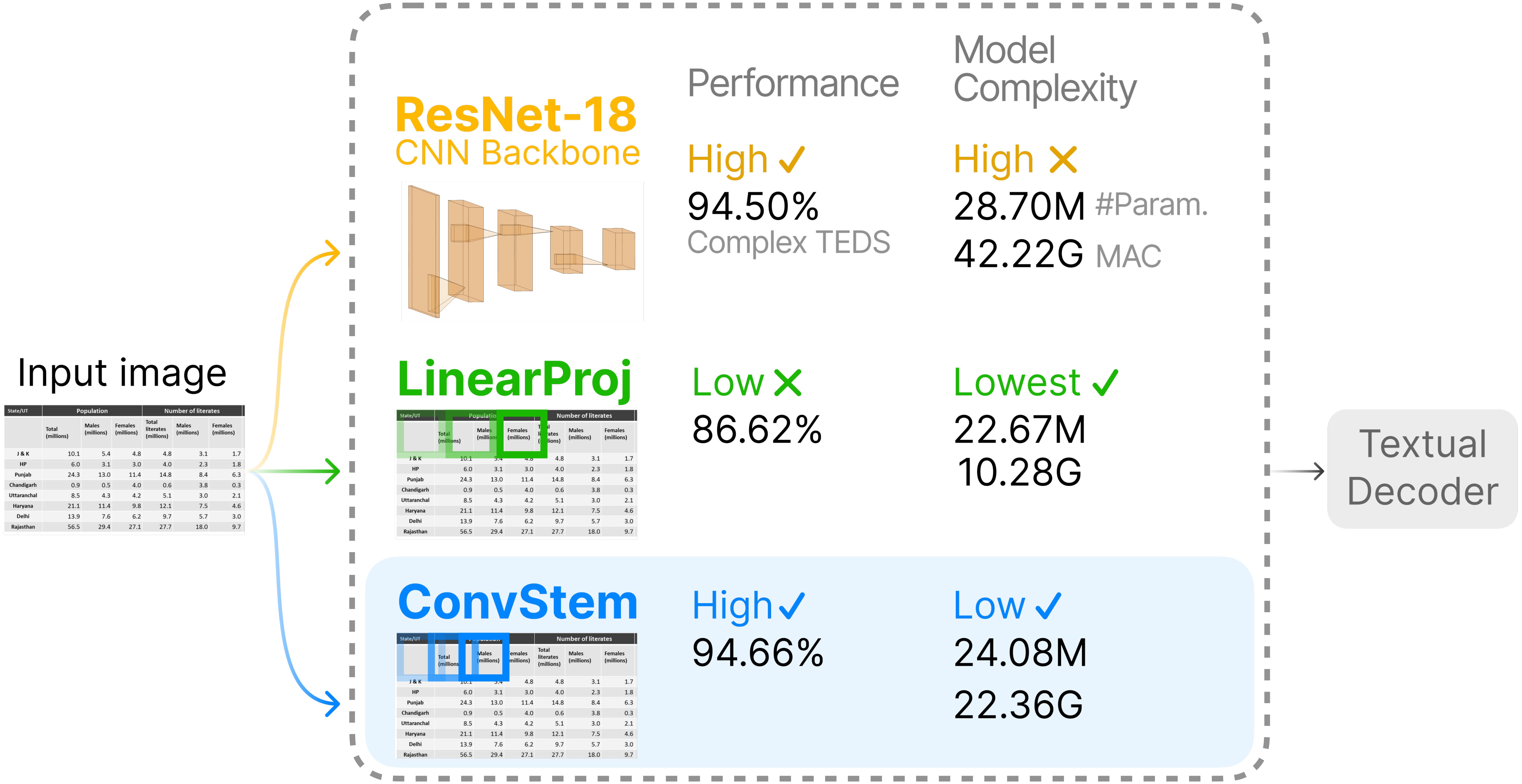
Lowest ✓
22.67M
10.28G

Textual Decoder

Visual Encoder Options for Table Structure Recognition

Our Key Contribution & Discovery

ConvStem matches CNN backbone performance with a simpler model



Input image

State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7

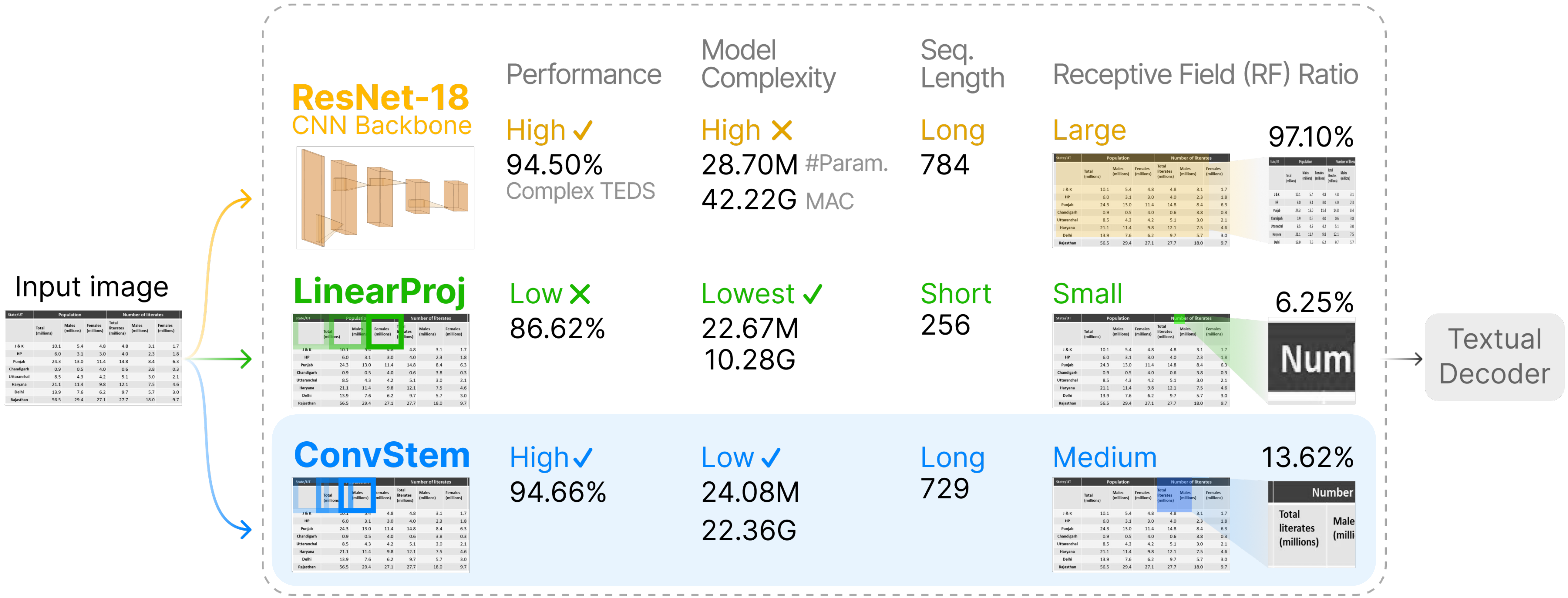
	Performance	Model Complexity
ResNet-18 CNN Backbone	High ✓ 94.50% Complex TEDS	High ✗ 28.70M #Param. 42.22G MAC
LinearProj	Low ✗ 86.62%	Lowest ✓ 22.67M 10.28G
ConvStem	High ✓ 94.66%	Low ✓ 24.08M 22.36G

Textual Decoder

Visual Encoder Options for Table Structure Recognition

Why is convolutional stem effective?

Higher receptive field ratio & longer sequence length



Input image

State/UT	Population			Number of literates		
	Total (millions)	Males (millions)	Females (millions)	Total literates (millions)	Males (millions)	Females (millions)
J & K	10.1	5.4	4.8	4.8	3.1	1.7
HP	6.0	3.1	3.0	4.0	2.3	1.8
Punjab	24.3	13.0	11.4	14.8	8.4	6.3
Chandigarh	0.9	0.5	4.0	0.6	3.8	0.3
Uttaranchal	8.5	4.3	4.2	5.1	3.0	2.1
Haryana	21.1	11.4	9.8	12.1	7.5	4.6
Delhi	13.9	7.6	6.2	9.7	5.7	3.0
Rajasthan	56.5	29.4	27.1	27.7	18.0	9.7

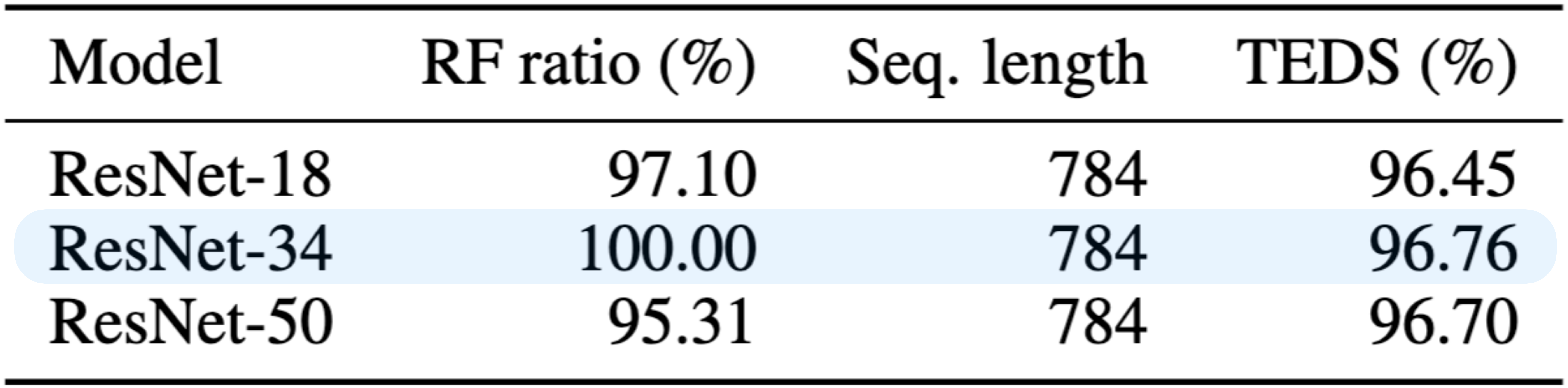
	Performance	Model Complexity	Seq. Length	Receptive Field (RF) Ratio
ResNet-18 CNN Backbone	High ✓ 94.50% Complex TEDS	High ✗ 28.70M #Param. 42.22G MAC	Long 784	Large 97.10%
LinearProj	Low ✗ 86.62%	Lowest ✓ 22.67M 10.28G	Short 256	Small 6.25%
ConvStem	High ✓ 94.66%	Low ✓ 24.08M 22.36G	Long 729	Medium 13.62%

Textual Decoder

ConvStem matches CNN backbone performance with a simpler model

CNN Backbone

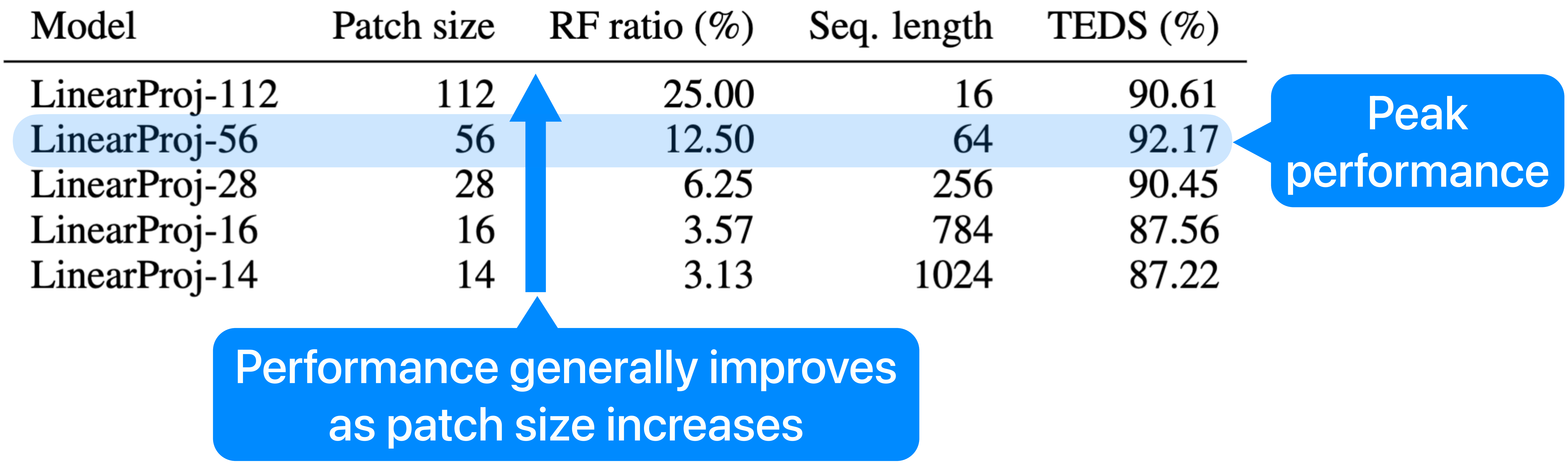
ResNet-34 has the highest TEDS due to its high RF ratio



Model	RF ratio (%)	Seq. length	TEDS (%)
ResNet-18	97.10	784	96.45
ResNet-34	100.00	784	96.76
ResNet-50	95.31	784	96.70

Linear Projection

As the patch size increases, performance generally improves, reaching its peak at a patch size of 56



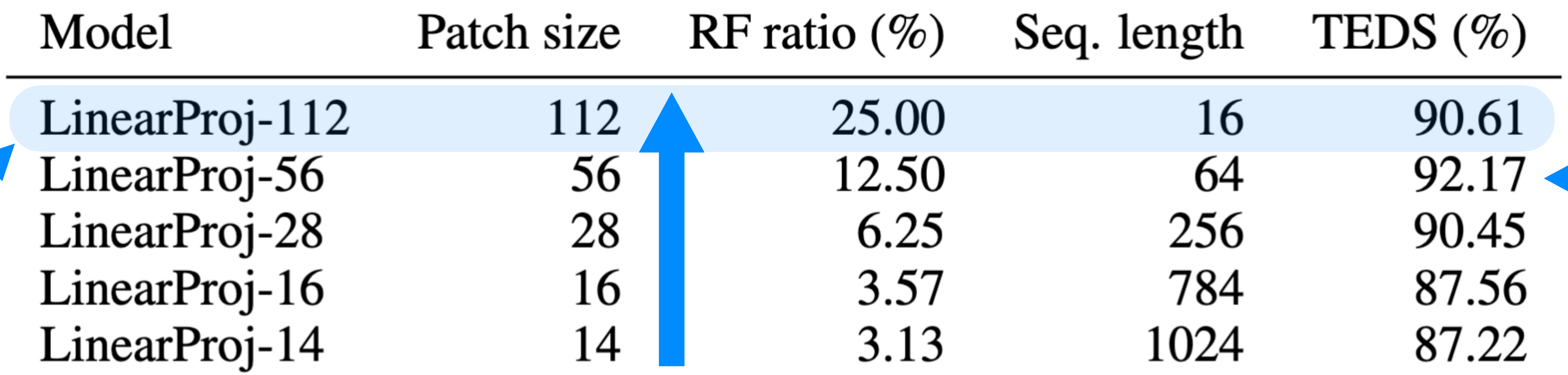
Model	Patch size	RF ratio (%)	Seq. length	TEDS (%)
LinearProj-112	112	25.00	16	90.61
LinearProj-56	56	12.50	64	92.17
LinearProj-28	28	6.25	256	90.45
LinearProj-16	16	3.57	784	87.56
LinearProj-14	14	3.13	1024	87.22

Peak performance

Performance generally improves as patch size increases

Linear Projection

As the patch size increases, performance generally improves, reaching its peak at a patch size of 56



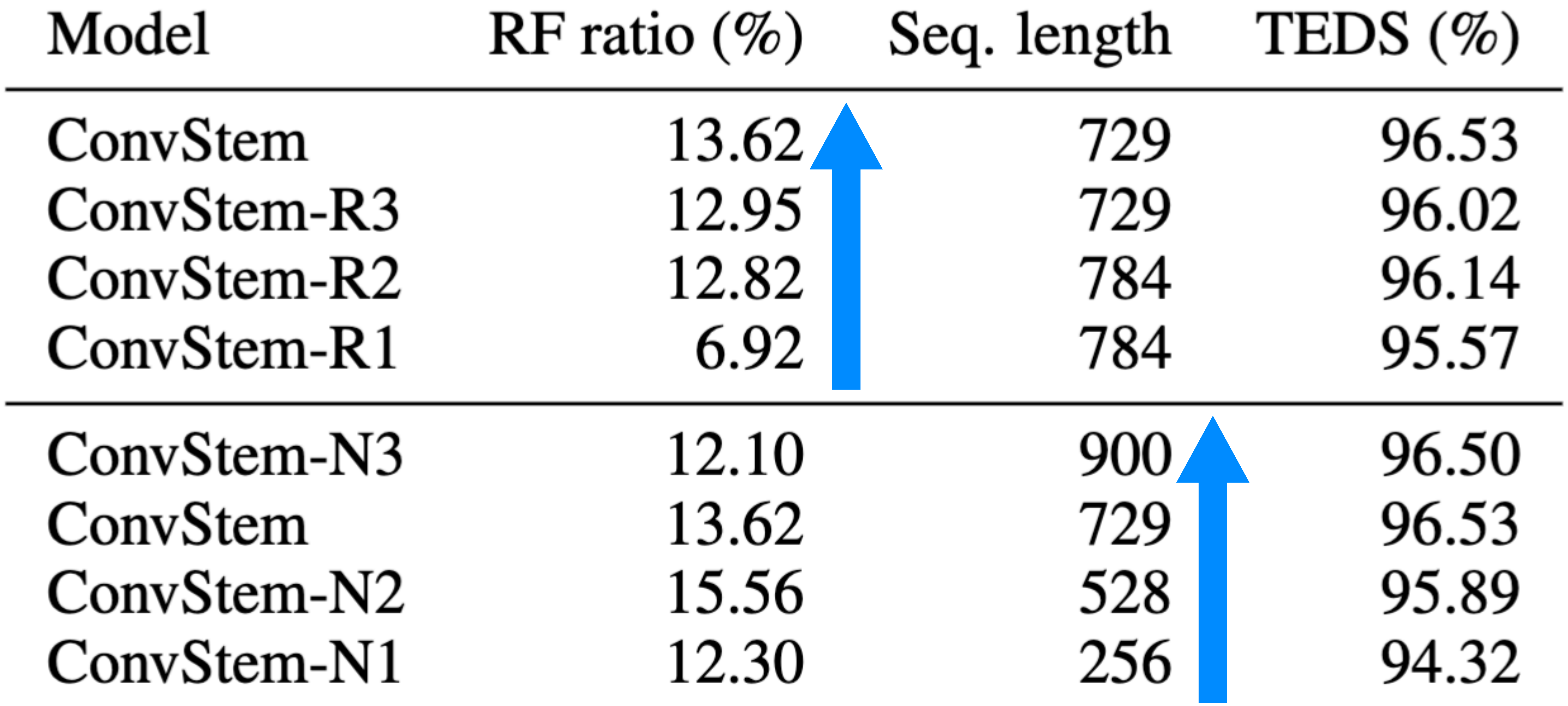
Model	Patch size	RF ratio (%)	Seq. length	TEDS (%)
LinearProj-112	112	25.00	16	90.61
LinearProj-56	56	12.50	64	92.17
LinearProj-28	28	6.25	256	90.45
LinearProj-16	16	3.57	784	87.56
LinearProj-14	14	3.13	1024	87.22

Peak performance

Patch size too large ➡ sequence length too small ➡ worse performance
(Patch size & sequence length inversely correlated)

Convolutional Stem

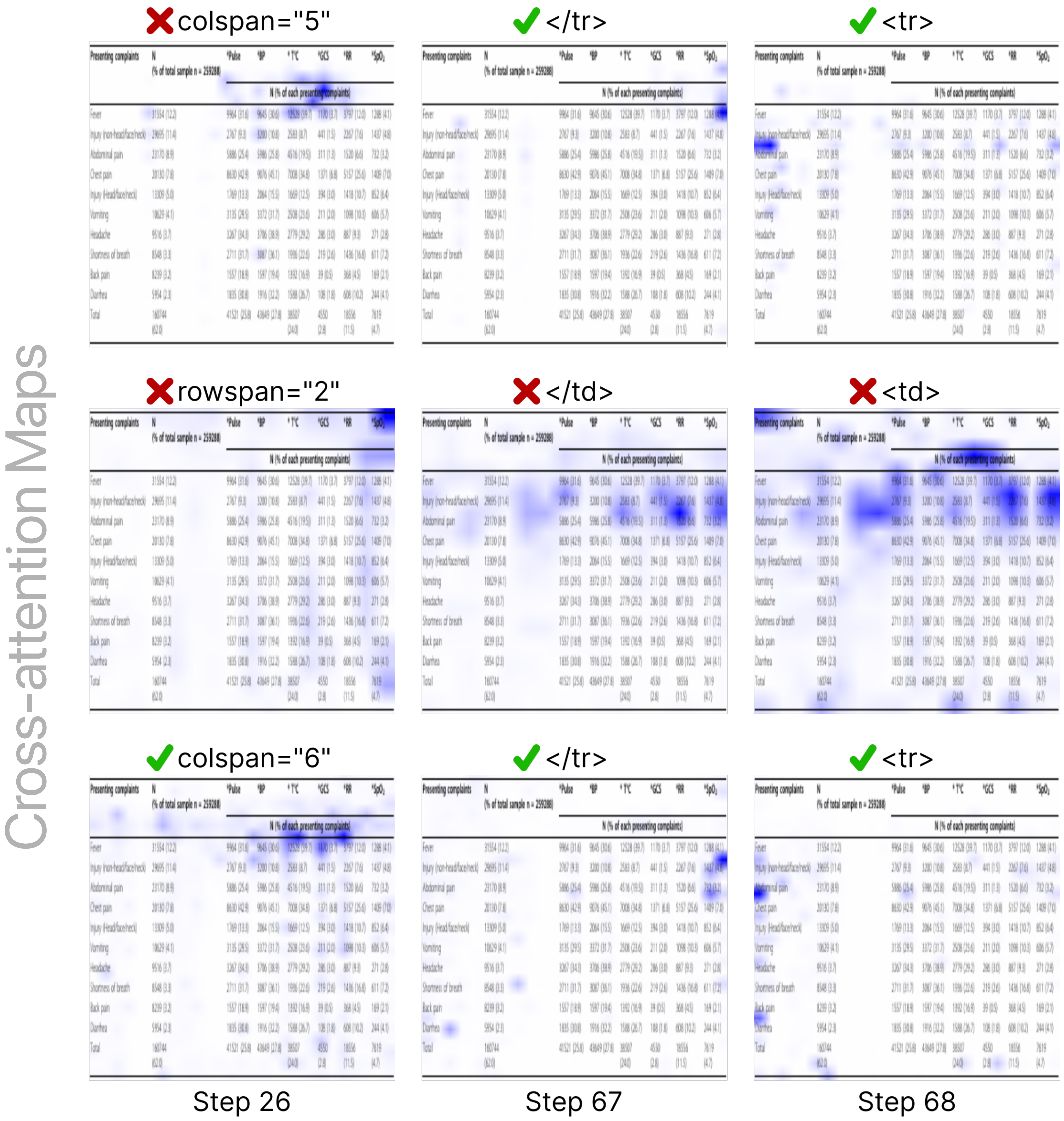
Optimal balance of RF ratio & sequence length



Model	RF ratio (%)	Seq. length	TEDS (%)
ConvStem	13.62	729	96.53
ConvStem-R3	12.95	729	96.02
ConvStem-R2	12.82	784	96.14
ConvStem-R1	6.92	784	95.57
ConvStem-N3	12.10	900	96.50
ConvStem	13.62	729	96.53
ConvStem-N2	15.56	528	95.89
ConvStem-N1	12.30	256	94.32

Higher RF ratio increases TEDS

Longer sequence length increases TEDS



CNN Backbone (ResNet-18)

Miscounts columns in header spanning cell

TEDS: 98.28%

Linear Projection (LinearProj-28)

Mis-predict html tags due to scattered attention

TEDS: 86.07%

ConvStem

Accurately reconstructs table

TEDS: 100.00%

Easy-to-Use Open-Source Research

github.com/poloclub/tsr-convstem

Step 1. Configure experiment

```
EXP_r18_e2_d4_adamw := $(PUBTABNET) $(MODEL_r18_e2_d4) $(OPT_adamw)
```

Step 2. Train & evaluate the model

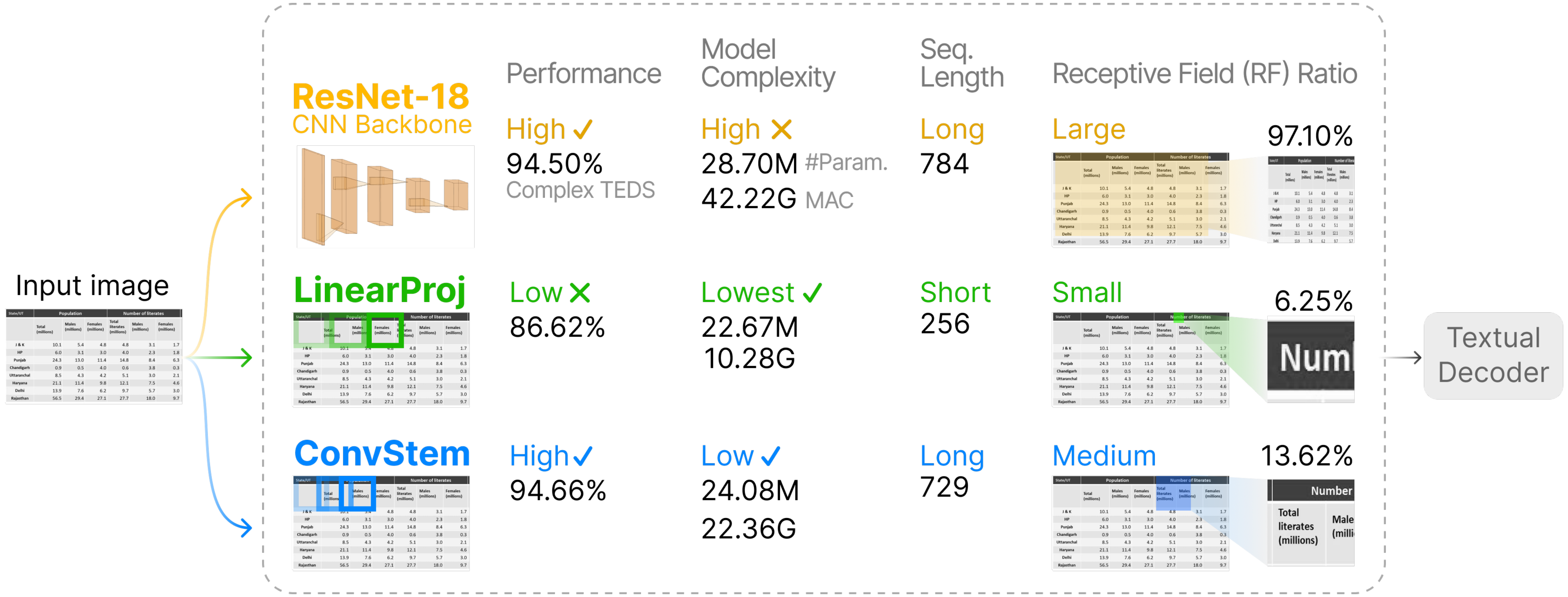
```
$ make experiments/r18_e2_d4_adamw/.done_teds_structure
```

Thanks!

github.com/poloclub/tsr-convstem

High-Performance Transformers for Table Structure Recognition Need Early Convolutions

ConvStem matches classic CNN backbone performance, with a much simpler model.



	Performance	Model Complexity	Seq. Length	Receptive Field (RF) Ratio	
ResNet-18 CNN Backbone	High ✓ 94.50% Complex TEDS	High ✗ 28.70M #Param. 42.22G MAC	Long 784	Large	97.10%
LinearProj	Low ✗ 86.62%	Lowest ✓ 22.67M 10.28G	Short 256	Small	6.25%
ConvStem	High✓ 94.66%	Low ✓ 24.08M 22.36G	Long 729	Medium	13.62%

Visual Encoder Options for Table Structure Recognition

ShengYun (Anthony) Peng
shengyun-peng.github.io

Seongmin Lee

Xiaojing Wang

Rajarajeswari Balasubramaniyan

Polo Chau

Georgia Tech

ADP