Accelerating Transformer-Based Scene Text Detection and Recognition via Token Pruning

Sergi Garcia-Bordils ✉[1,2][0000−0002−4222−8367], Dimosthenis Karatzas[1][0000−0001−8762−4454], and Marçal Rusiñol[2][0000−0002−1734−2205]

[1] Computer Vision Center, Universitat Autonoma de Barcelona, Barcelona, Spain
{sergi.garcia, dimos}@cvc.uab.cat
[2] AllRead MLT

Abstract. Scene text detection and recognition is a crucial task in computer vision with numerous real-world applications. Transformer-based approaches are behind all current state-of-the-art models and have achieved excellent performance. However, the computational requirements of the transformer architecture makes training these methods slow and resource heavy. In this paper, we introduce a new token pruning strategy that significantly decreases training and inference times without sacrificing performance, striking a balance between accuracy and speed. We have applied this pruning technique to our own end-to-end transformer-based scene text understanding architecture. Our method uses a separate detection branch to guide the pruning of uninformative image features, which significantly reduces the number of tokens at the input of the transformer. Experimental results show how our network is able to obtain competitive results on multiple public benchmarks while running at significantly higher speeds.

Keywords: Scene Text Detection · Scene Text Recognition · Transformer Acceleration.

1 Introduction

Joint text detection and recognition has become a popular topic in the field of computer vision for its wide range of applications. Text is omnipresent in man-made environments, and it plays a crucial role in different computer vision tasks such as visual-question answering [2,46] or cross-modal retrieval [34], and in many computer vision applications like autonomous navigation [43] or industrial automation [8].

Early deep-learning based systems for text detection and recognition were based on two-stage pipelines, a detection network that extracted regions of interest (RoI) and a recognition network that recognized the cropped regions. The two tasks were treated as separate problems, with no gradient flowing between the two networks. [25,26]. More recent works attempted to jointly optimize both parts of the pipeline, allowing end-to-end trainable architectures [11,24,29,33,50]. A common drawback of these networks is that they needed to explicitly rectify

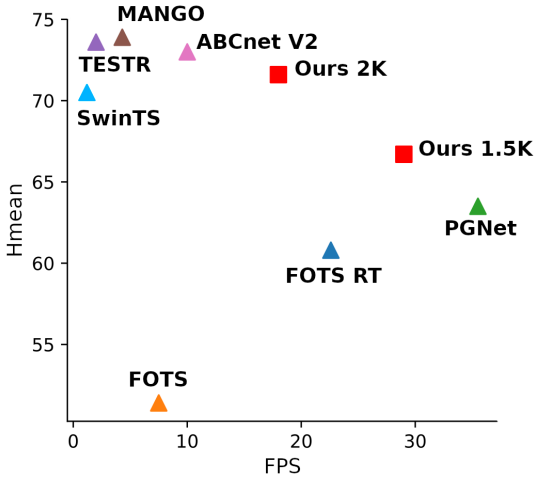

Fig. 1: Comparison between inference speed (in frames per second) and Hmean on ICDAR15 for different state-of-art scene text detection and recognition models. Our approach offers a balance between performance and inference speed thanks to our novel token pruning. The reported results use two different image scales (1500 and 2000).

the RoI before they can be fed into the recognizer, which usually reads from left to right. For example, the authors of FOTS [28], a network that detects rotated bounding boxes, rectified the rotation of the RoI with their proposed RoIRotate operation. Other models, such as AbcNet [29] or Mask TextSpotter [24] proposed more complex de-warping techniques that were able to rectify heavily distorted text, such as curved text.

More recently, different one-stage methods have started appearing that do not require corrective operations on the detected areas [13,18,19,40,52]. Many of these approaches are based on the transformer architecture proposed by Vaswani et al. [48]. The common approach is to pass the features extracted by a CNN to the transformer, where the powerful self-attention mechanism performs detection and recognition. The fully connected topology of the self-attention removes the need to use RoI corrective operations. One of the drawbacks of the transformer is the quadratic complexity of the self-attention mechanism with respect to the number of input tokens. Increasing the input image resolution results in significantly slower training and inference times and higher memory usage.

The $O(n^2)$ complexity of the transformer has motivated multiple works that attempt to improve the efficiency of architecture. Some NLP approaches [10, 51] have attempted modifying the fully connected self-attention with simpler topologies that reduce the complexity to $O(n)$. On vision and ViT-based [5] models, a popular approach is to reduce the number of tokens by employing

a sampling/pruning mechanism to progressively discard uninformative tokens [6, 23, 36, 42]. Many of these strategies such as ATS [6], or EViT [23] have been specially tailored for classification tasks on the ViT [5] architecture, and can not be directly applied to the object detection. Approaches like DynamicViT [42] or IA-RED^2 [36] need to train a specific component of the network to remove tokens, which often employs complex strategies such as reinforcement learning.

In this paper we introduce a novel token-pruning mechanism that has been specifically designed for scene text detection and recognition models. Our pruning approach works under the assumption that visual information of text is very local, while most of the background area is non-informative. The pruning mechanism reduces the complexity of the model and allows more efficient training parallelization and lower inference times. This pruning strategy has been applied to our own transformer encoder-based architecture, which is capable of reading text in multiple orientations, curvatures and distortions without needing to perform RoI corrections. Figure 2 shows an example of how our network performs pruning over the visual features before they go into the recognition branch. Our network has been designed to achieve a balance between quantitative performance and high inference speeds. As seen in Figure 1, our model manages to get competitive results with the state-of-art while running at higher FPS. The contributions of this paper are the following:

- A novel token-pruning mechanism that allows the architecture to reduce the size of the input to the recognizer branch, a transformer-encoder network. We show how our strategy yields lower training and inference times than cropping and recognizing the detected areas independently.
- An efficient end-to-end text detection and recognition architecture where both tasks happen independently of each other. The model does not require any type of RoI corrective operations over the detected areas thanks to the fully connected attention mechanism.
- We show that our method manages to balance performance and speed, allowing us to reach competitive results with the state-of-the-art at significantly higher inference speeds.

2 Related Work

2.1 End-To-End Scene Text Recognition

Scene text detection and recognition is a challenging topic that has been an active area of research for many years. The complexity and sophistication of the architectures increased as different datasets and annotation styles started emerging. The different scenarios for text detection feature horizontal bounding boxes [17], incidental text with 4-point annotations [16, 35, 47, 49] and, more recently, arbitrarily shaped text [3, 30] such as curved text, which often feature complex polygonal annotations.

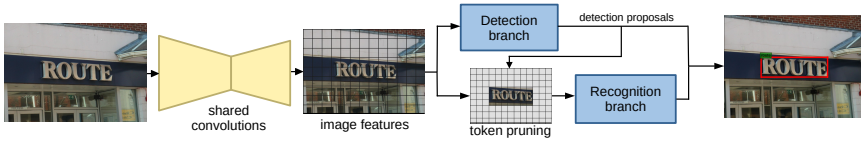

Fig. 2: The proposed architecture uses a shared convolutional backbone that extracts visual features from the image and then up-scales the output feature map. Two separate branches perform detection and recognition. We use the detection branch bounding boxes to guide the pruning mechanism, reducing the number of tokens at the input of the transformer. During training, we use the ground truth localization of the text to prune the image features.

Earlier models, such as Textboxes [26], directly cropped the detected horizontal boxes from the image, and performed no rectification to the crop. This was a problem for the most commonly used recognition architectures, such as the encoder-decoder [21, 22] and the CTC-based [9] networks, because they require the text to be horizontal and from left to right. As the complexity of annotations increased, models started to feature two-stage architectures that were fully end-to-end trainable and performed complex RoI rectifications to the detected areas.

More complex annotations and RoI corrective operations allowed recognition of text in different shapes and orientations. For example, FOTS [28] uses a text detection branch to predict oriented text boxes, and uses RoIRotation to obtain axis-aligned feature maps. The text recognition branch uses a bidirectional LSTM [12, 45] and a CTC [9] network to recognize the crop. Mask TextSpotterV3 [24] extracts rectangular crops from the segmentation and then masks-out the area outside of the region of interest. ABCNet [29] uses a more unconventional approach by fitting Bezier curves to the text instances, which helps to obtain smoother boundaries around the words. The curves are rectified with their proposed BezierAlign, which uses the control points of the curves to warp the word into a rectangular shape. TextDragon [7] predicts a series of quadrangles that follow the shape of the words and use their own RoISlide to rectify the text.

A recent trend in the community is to use transformer-based [48] networks. The fully connected topology of the self-attention mechanism avoids having to use any RoI rectifying operations at all. Some of them are capable of using simple annotations such as central keypoints [18, 39]. For example, TTS [19] uses a shared transformer encoder-decoder and different decoder heads to perform word recognition, detection and segmentation. This method can be trained by either providing the ground truth polygon annotations, the bounding box, or only the text in the image. TESTR [53] also employs an encoder-decoder approach to perform text detection and recognition. The authors use two transformer decoder networks to extract the detection and the recognition. More recently, DEER [18] uses a transformer encoder to perform detection-agnostic detection

and recognition. SwinTextSpotter [13] uses diverse transformer-encoder networks to improve the synergy between the detected areas and the recognizer.

2.2 Transformer Acceleration

The $O(n^2)$ complexity of the transformer architecture proposed by Vaswani et al. [48] has motivated multiple efforts to reduce its time and space complexity. On the NLP domain, different approaches have exploited the sparsity of the attention mechanism to reduce its complexity. For example, the Star-Transformer [10] replaces the fully connected attention with a star-shaped topology, reducing the complexity from quadratic to linear. Sparse Transformers [14] introduce multiple novel architectures that use sparse attention layers that perform faster un-batched decoding. The authors of the Linformer [51] also achieve linear complexity by approaching the self-attention with a low-rank matrix. Other networks such as TinyBERT [15] use distillation to transfer knowledge from a larger teacher BERT [4] network into a smaller one.

On the vision domain, numerous works have approached the problem by reducing the number of tokens on the input of the standard Vision Transformer (ViT) [5] architecture. The Hierarchical Vision Transformer [37] proposes an architecture that fuses tokens using a pooling operation after every transformer block, similar to the down-sampling of a convolutional network. EViT [23] progressively reduces the number of tokens along the different attention layers. The model uses the attention over the classification token to fuse uninformative tokens. DynamicViT [42] proposes a prediction module that estimates the importance of each token and discards tokens that are uninformative. Similar to EViT, the authors of ATS [6] propose a an adaptive token sampling method that uses the attention over the classification token to discard tokens. Unlike EViT, the method proposed is plug-and-play and does not need to be retrained. IA-RED^2 [36] employs a similar strategy to EViT, but they use a reinforcement algorithm to train the pruning algorithm.

3 Methodology

Transformer acceleration methods for vision are mostly focused on object classification with ViT-based models. By contrast, our method has been specifically designed with scene text understanding in mind. We test our proposed acceleration approach on our own transformer encoder-based model, which has been designed for fast inference speeds. The architecture performs detection and recognition using a shared convolutional backbone and two separate branches for detection and recognition. The detection branch is based on the CenterNet [54] architecture, which we use to predict the location of the text. These predicted locations are used to guide the pruning of uninformative image features before feeding them to the recognition branch, which uses the transformer encoder. This branch is capable of reading text in multiple orientations as well as curved text without performing RoI cropping or corrective operations. The self-attention mechanism

of the transformer combines local information and encodes latent representations of the words in a grid (Figure 5 shows an example of the recognition grid).

Our token pruning mechanism allows faster training and inference speeds without compromising the accuracy of the network, in addition of using less memory. By making use of large, publicly available datasets for scene-text detection and recognition, our approach is able to balance competitive quantitative results with fast inference speeds. Figure 3 shows a more detailed overview of the proposed architecture.

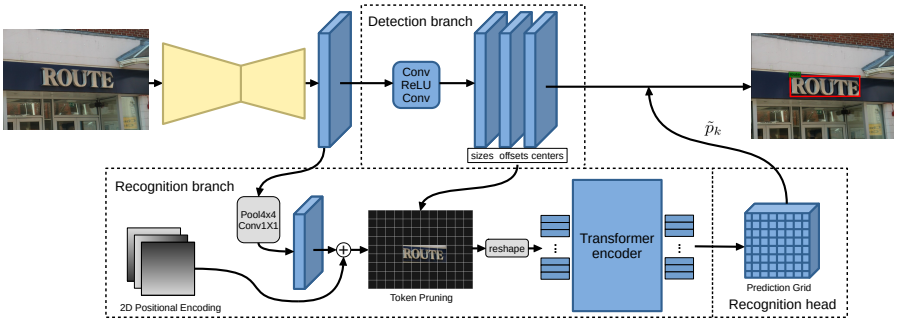

Fig. 3: A more detailed overview of the detection and recognition branches of our proposed architecture. The CenterNet-based detection branch generates detection proposals while the transformer-based recognition branch encodes a latent representation of each word in a grid. A separate recognition head generates a dense prediction map. The pruning mechanism uses the proposals from the detection branch to reduce the number of tokens at the input of the transformer encoder.

3.1 Architecture

Our model uses a ResNet-34 and a series of transposed convolutional operators as the backbone of the network. More specifically, we apply two transposed convolutions that expand the feature maps up to 1/8 of the original resolution. Inspired by U-Net [44], we combine the lower-level feature maps of the ResNet with up-scaled feature maps of the expanding path. In the original U-Net, the higher-level feature maps are cropped and then concatenated to the feature maps of the expansive path. Instead, our architecture applies a 1×1 convolutional operator over the ResNet feature maps to match the number of channels of the corresponding up-sampled maps, which are then added. The backbone outputs a feature map $f = \mathbb{R}^{\frac{W}{S} \times \frac{H}{S} \times D}$, where H and W are the height and width of the original image and S is the stride, in our case $S = 8$. The output feature map is shared by two different branches that perform detection and recognition.

Text Detection Branch The text detection branch is based on the CenterNet architecture, an efficient object detection framework that predicts axis-aligned bounding boxes. CenterNet uses central keypoint estimation to predict the center of the bounding boxes by generating in a heatmap $\hat{Y} \in [0,1]^{\frac{W}{S} \times \frac{H}{S}}$. During training, the ground truth heatmap Y is generated by drawing a Gaussian kernel at the center of each object, which reduces the penalty around the ground truth keypoints. In this heatmap, a value $Y_{X,Y} = 1$ represents a keypoint, while $Y_{X,Y} = 0$ is background. The loss for the heatmap L_k is the modified focal loss [27] introduced by [20]. Following [54] and [20], we set $\alpha = 2$ and $\beta = 4$.

The stride of the feature map introduces a discretization error in the keypoint estimation. To overcome this, CenterNet introduces a local offset $\hat{O} \in \mathbb{R}^{\frac{W}{S} \times \frac{H}{S} \times 2}$ that helps to adjust each center. Like in the original CenterNet paper, the loss of the offsets L_{off} is the L1 loss at the keypoint locations. CenterNet predicts the widths and the heights of the bounding boxes by regressing both components at the center of each point, the output has the same form as the local offset $\hat{R} \in \mathbb{R}^{\frac{W}{S} \times \frac{H}{S} \times 2}$. The loss of the offsets L_{size} is again the L1 loss in each one of the ground truth keypoints.

Token Pruning Our architecture introduces a token pruning strategy that reduce the number of tokens before the recognition transformer encoder. Since the space and time complexity of the attention mechanism of the transformer is quadratic with the number of input tokens, reducing the size of the input can yield more efficient training and inference times. This approach to reduce the number of tokens is based on the assumption that the features relevant to recognize text are local for each text instance, while the surrounding areas are uninformative. This strategy employs the detected text areas from the detection branch to discard part the visual features that come from the CNN. Any visual features from z_1 that does not overlap with a bounding box are discarded, since they probably do not contain textual information (Figure 4 shows how features that are not overlapping any text detections get discarded). The pruning is applied after adding a 2D positional encoding [1,38] because we need to preserve the relative position of the tokens on the original feature map. Our experimental results show how the pruning mechanism does not affect the performance of the network, since all the relevant information for the recognition head is being preserved. This process is also fully end-to-end trainable.

This joint pruning and recognition strategy can be seen akin to two-stage architectures such as FOTS [28] or Mask TextSpotter [24, 33], where each text detection is used to crop and recognize the RoI in the feature map. Our approach differs in that we do not crop and recognize the RoIs one by one. Instead, we remove the non-informative areas of the image features and perform the recognition in parallel. For a more fair comparison between both approaches, we also implemented a classic Two-Stage version of our model that performs RoI cropping and recognition with a transformer encoder. In our experimental section we show how the pruning approach yields faster training and inference times w.r.t. the Two-Stage version.

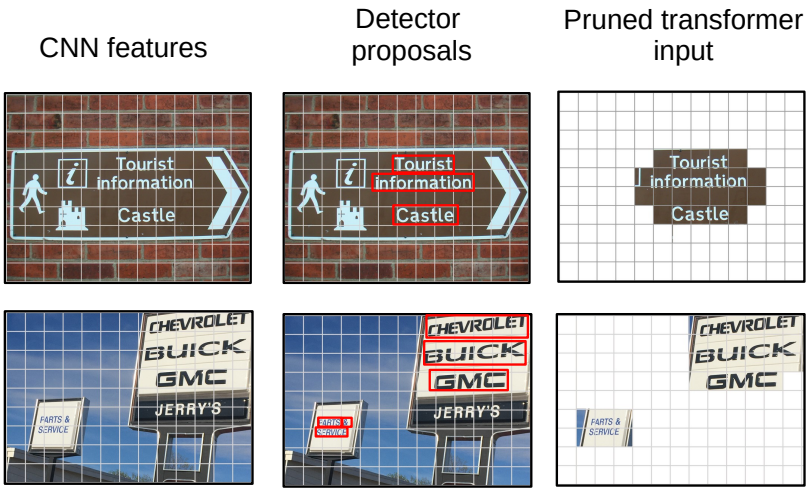

Fig. 4: Our pruning strategy discards tokens of the image that do not contain text information. Using the bounding boxes detected as a guide, we create a mask that has the size of the output feature map (which we abstractly represent as a grid over the original image), and use it to discard features that are outside of text regions. If the detector fails to localize a text instance, it will not get recognized.

Text Recognition Branch The recognition head is principally composed of a transformer encoder [48]. In our experiments we have trained two different versions of the network, a Small version with 4 encoder layers and a Base version with 8 layers.

The inputs to the transformer are the up-scaled features outputted by the convolutional backbone in the form of a flattened vector of tokens. After applying the token pruning described in the previous section, the visual features are inputted into the transformer encoder. The self-attention mechanism of the transformer flexibly combines information around each cell of the input features to generate a latent representation grid.

After applying the self-attention, the recognition head generates a text prediction for every token, where the output is encoded as a maximum of M characters. We apply a softmax activation function over the character dimension to generate per-character confidences, obtaining final predictions of size $M \times C$, where C is the size of our alphabet. The loss of the recognition branch L_{recog} is the cross-entropy loss between the predicted character confidences and the ground truth one-hot vector of each character. Like the offset regression $\hat{O}$ and bounding box height and width regression, the loss for each word is calculated at the

center of the words, while the predictions around it do not contribute to the loss.

The final optimization objective of the network is defined by the addition of the four previous losses:

$$L = L_k + L_{off} + \lambda_{size} L_{size} + L_{recog} \tag{1}$$

where λ_{size} is used to scale the bounding box regression loss, we set $\lambda_{size} = 0.1$ like in the original CenterNet paper.

Fig. 5: Visualization of the predicted word grid generated by the recognition head. The overlaid mask shows the confidence for each one of the predicted words (in this example, the prediction in blue represents the final recognized word). During training, the loss is only taken into account in the center of the word, while the areas around it are ignored.

3.2 Training Details

The model was trained using two NVIDIA A40, the resolution during training was 1024×1024 with a batch size of 32. The optimizer used in all the cases was AdamW [32], with a gradual learning-rate warm-up of 1000 iterations. The model is pre-trained with SynthText for two epochs, with an initial learning rate of 1e−4 with no learning rate decay. Next, the model is trained using a combined dataset of ICDAR13 [17], ICDAR15 [16], ICDAR17 [35], COCO-text [49] and TextOCR [47] for 40 epochs at the same learning rate. After 20 epochs we decay the learning rate to 1e−5. Some datasets, such as ICDAR17, include text in different scripts than Latin, we do not take into account these text instances. During training, we use the ground truth detections to guide the token pruning.

4 Experiments

4.1 Text Detection and Recognition Datasets

We have evaluated our model on ICDAR13, ICDAR15, and Total-Text. On ICDAR13 we use the standard evaluation protocol proposed by the authors. The

datasets ICDAR15 and Total-Text feature rotated quadrilaterals and irregular polygonal annotations respectively. To be able to compare our method in these datasets, we adopt the evaluation protocol proposed by TTS [19], where they propose to use the horizontal bounding box version of the ground truth annotation to evaluate the predictions. Their experimental results show that this evaluation strategy has a minor effect on the final results.

In the ICDAR15 and Total-Text datasets we used a single input resolution of 1400 and 2000 respectively. For ICDAR13 we use two different scales of 2000 and 500 pixels to better deal with text at different sizes.

Results Table 1 shows end-to-end detection and recognition in quantitative results on ICDAR15. On this dataset our model manages to perform on par with the latest state-of-art models, which shows how the recognition branch is able to successfully deal with oriented text. Our model is also able of significantly higher inference speeds than the latest models and it is only surpassed by FOTS, which our model manages to widely surpass on accuracy.

Method	IC15			FPS
	S	W	G	
FOTS [28]	81.1	75.9	60.8	**22.6**
Boundary [50]	79.7	75.2	64.1	-
TextPerceptron [41]	80.5	76.6	65.1	8.8
ABCnet [31]	82.7	78.5	73.0	10.0
TextDragon [7]	**86.2**	**82.0**	68.1	-
MANGO [40]	85.4	80.1	73.9	4.3
DEER [18]	82.7	79.0	75.6	-
SwinTextSpotter [13]	83.9	77.3	70.5	1.2
TESTR [53]	85.2	79.4	73.6	2.0
TTS [19]	85.0	81.5	**77.3**	-
Ours	84.6	80.2	71.6	18.0

Table 1: End-To-End results on ICDAR15. The results reported were obtained using the Small version of our network.

In Table 2 we show the results for the datasets ICDAR13 and Total-Text. Our model obtains good results using two scales, achieving competitive results with the latest models. When using a single scale the performance drops, but still maintains good results. Despite never seeing the training set of Total-Text (a dataset mainly focused on rotated text), our model also obtains good results on it.

Finally, figure 4.1 shows qualitative examples of word detection and recognition with different types of distortions. Our model is capable of dealing with different types of distortions such as rotations or curvatures thanks to the transformer encoder-based recognizer.

Method	IC13 S	W	G	Total-Text None	full
FOTS [28]	88.8	87.1	86.0	-	-
Boundary [50]	88.2	87.7	84.1	65.0	-
TextPerceptron [41]	91.4	90.7	85.8	69.7	78.3
ABCnet [31]	-	-	-	70.4	78.1
TextDragon [7]	-	-	-	75.8	84.4
MANGO [40]	**93.4**	**92.3**	**88.7**	72.9	83.6
DEER [18]	-	-	-	74.8	81.3
SwinTextSpotter [13]	-	-	-	74.3	84.1
TESTR [53]	-	-	-	73.3	83.9
TTS_{poly} [19]	-	-	-	75.6	84.4
TTS_{box} [19]	-	-	-	**75.9**	**84.5**
Ours	85.2	83.4	78.3	61.5	72.1
Ours TS	92.3	89.2	87.2	64.2	74.6

Table 2: End-To-End results on ICDAR13 and Total-Text.

(a) Examples of successful detection and recognition with the Small model.

Fig. 6: Our transformer-based approach manages to successfully perform detection and recognition with horizontal bounding boxes. The models successfully recognizes the text with different types of distortions.

4.2 Token Pruning

In this section we evaluate the performance gains of our token pruning approach. We also compare it with an alternative Two-Stage variation of our model. In this version, the locations of the detection branch are used to crop the image features overlapping the bounding boxes. The transformer encoder performs recognition for each one of the cropped regions, but unlike our architecture this does not happen in parallel. The three versions of our architecture were trained using the Small (with 4 attention layers) and Base (with 8 layers) sizes of the transformer encoder.

In Table 3 we compare the three variants of our architecture trained under the same configuration. As seen in the table, the three variants offer similar quantitative results on ICDAR15. The token pruning and Two-Stage variants considerably reduce the number of MAC operations (one multiplication and one addition) with respect the model with no pruning. At a resolution of 2000 pixels, the pruning mechanism removes an average of 91% of non-informative tokens, which reduces 95% of the operations in the transformer encoder. The number of operations remains similar between the Small and Base versions, which allows our pruning approach to maintain almost the same number of FPS. In the Base version of our model, the reduction represents almost half of the overall operations of the network (from 510 to 279 GMACs).

Since the recognition happens in parallel in the fully connected attention mechanism of the transformer encoder, the proposed pruning version is slightly more computationally expensive than the Two-Stage version. However, the parallelization of the transformer operations allows the pruning version to obtain faster inference speeds. The Base version of our model has almost the same number of operations as the Small one, and reaches similar FPS during inference.

Training The benefit of this parallel approach is that is easier to batch the recognition during training, which results in reduced training times. In the rightmost column of Table 3 we can see the number of iterations per second for all the variants of our model using the same configuration (an input resolution of 1024×1024 and a batch size of 32). Using token pruning reduces 40% the training times for the Small model while in the Base model the reduction is 76%.

Image Resolution A bigger image size has a quadratic impact on the number of operations of the encoder head. In Figure 7 we can see the effect of input image size on the FPS of the three variations during inference. Thanks to the great reduction in number of operations of the pruning, our proposed strategy achieves faster inference speeds than the two variations of our model. The gap between the Small and Base non-pruning version (red lines) is considerably bigger than the gap between the two versions of our Pruning approach, which is between 1 and 2 FPS depending on the size.

Recognition	Layers	S	W	G	FPS	GMACs	it/s
No Pruning	4	79.3	77.1	74.1	12	394	0.84
Two-Stage	4	**79.5**	**77.2**	74.5	15	**278**	1.06
Pruning	4	79.4	77.1	**74.7**	**18**	282	**1.18**
No Pruning	8	79.1	77.2	74.2	8	510	0.65
Two-Stage	8	79.1	**77.4**	74.4	12	**279**	1.02
Pruning	8	**79.3**	77.3	**75.1**	**17**	284	**1.15**

Table 3: Performance comparison between using token pruning, no pruning and Two-Stage for inference. The models were evaluated on ICDAR13 at a resolution of 2000 pixels. The MACs count includes the operations of the convolutional backbone (which totals 276 GMACs for the used image size). The rightmost column also shows the number of iterations per second during training. The batch size is 32 and the resolution used is 1024×1024.

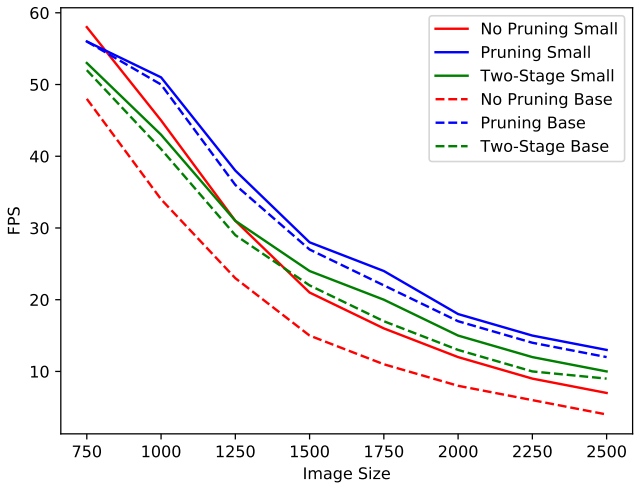

Fig. 7: Effects of the input image size on the FPS for the different variants of our model. The solid line shows the performance using the Small transformer (4 encoder layers) while the dashed line shows the performance of the Base model (8 encoder layers).

5 Conclusions

In this paper we have introduced a novel strategy to improve the efficiency of transformer-based architectures for scene text recognition. Our token pruning mechanism, which has been specially designed for scene-text detection and recognition, effectively decreases training and inference times of the network.

We have tested this approach on our own transformer-based architecture, which has been tailored to achieve a balance between speed and accuracy. Thanks to the proposed pruning mechanism, our model achieves fast inference speeds while being competitive with the state of the art.

Acknowledgements This work has been supported by grants PDC2021-121512-I00, PID2020-116298GB-I00 and PLEC2021-007850 funded by the European Union NextGenerationEU/PRTR and MCIN/AEI/10.13039/501100011033; the EU Lighthouse on Safe and Secure AI - ELSA funded by European Union's Horizon Europe programme under grant agreement No 101070617; the Spanish Project NEOTEC SNEO-20211172 from CDTI; grant Torres Quevedo PTQ2019-010662; and the Industrial Doctorate programme of the Catalan Government (2020 DI 058).

References

1. Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.
2. Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019.
3. Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017.
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
5. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
6. Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, pages 396–414. Springer, 2022.
7. Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9076–9085, 2019.
8. Luís Gómez, Marçal Rusinol, and Dimosthenis Karatzas. Cutting sayre's knot: reading scene text without segmentation. application to utility meters. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 97–102. IEEE, 2018.

9. Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.
10. Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. *arXiv preprint arXiv:1902.09113*, 2019.
11. Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5020–5029, 2018.
12. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
13. Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4593–4603, 2022.
14. Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Lukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva. Sparse is enough in scaling transformers. *Advances in Neural Information Processing Systems*, 34:9895–9907, 2021.
15. Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.
16. Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.
17. Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013.
18. Seonghyeon Kim, Seung Shin, Yoonsik Kim, Han-Cheol Cho, Taeho Kil, Jaeheung Surh, Seunghyun Park, Bado Lee, and Youngmin Baek. Deer: Detection-agnostic end-to-end recognizer for scene text spotting. *arXiv preprint arXiv:2203.05122*, 2022.
19. Yair Kittenplon, Inbal Lavi, Sharon Fogel, Yarin Bar, R Manmatha, and Pietro Perona. Towards weakly-supervised text spotting using a multi-task transformer. *arXiv preprint arXiv:2202.05508*, 2022.
20. Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.
21. Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2231–2239, 2016.
22. Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2231–2239, 2016.

23. Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.
24. Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *European Conference on Computer Vision*, pages 706–722. Springer, 2020.
25. Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018.
26. Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
27. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
28. Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018.
29. Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9809–9818, 2020.
30. Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90:337–345, 2019.
31. Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *arXiv preprint arXiv:2105.03620*, 2021.
32. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
33. Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83, 2018.
34. Andres Mafla, Sounak Dey, Ali Furkan Biten, Lluis Gomez, and Dimosthenis Karatzas. Fine-grained image classification and retrieval by combining visual and locally pooled textual features. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2950–2959, 2020.
35. Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017.
36. Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34:24898–24911, 2021.
37. Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable vision transformers with hierarchical pooling. In *Proceedings of the IEEE/cvf international conference on computer vision*, pages 377–386, 2021.

38. Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.
39. Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Shenggao Zhu, Jing Li, Dahua Lin, Chunhua Shen, et al. Spts: Single-point text spotting. *arXiv preprint arXiv:2112.07917*, 2021.
40. Liang Qiao, Ying Chen, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Mango: a mask attention guided one-stage scene text spotter. *arXiv preprint arXiv:2012.04350*, 2020.
41. Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11899–11907, 2020.
42. Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.
43. Sangeeth Reddy, Minesh Mathew, Lluis Gomez, Marçal Rusinol, Dimosthenis Karatzas, and CV Jawahar. Roadtext-1k: Text detection & recognition dataset for driving videos. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11074–11080. IEEE, 2020.
44. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
45. Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.
46. Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.
47. Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8802–8812, 2021.
48. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
49. Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.
50. Hao Wang, Pu Lu, Hui Zhang, Mingkun Yang, Xiang Bai, Yongchao Xu, Mengchao He, Yongpan Wang, and Wenyu Liu. All you need is boundary: Toward arbitrary-shaped text spotting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12160–12167, 2020.
51. Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.
52. Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. Convolutional character networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9126–9136, 2019.

53. Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9519–9528, 2022.
54. Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.