Supplementary Material: Efficient Movie Scene Detection Using State-Space Transformers

Md Mohaiminul Islam[1*] Mahmudul Hasan[2] Kishan Shamsundar Athrey[2]
Tony Braskich[2] Gedas Bertasius[1]
[1]UNC Chapel Hill [2]Comcast Labs

Our supplementary material includes additional implementation details, more experimental results, and qualitative results.

1. Additional Implementation Details

TranS4mer Variants. We propose different variants of our model based on the configuration proposed by ViT [3]. In particular, we experiment with three variants of the TranS4mer model, i.e., TranS4mer-S/32, TranS4mer-B/32, and TranS4mer-S/16, as summarized in Table 1. The TranS4mer-S/32 has a hidden dimension of 384 and uses a patch size of 32. To construct the TranS4mer-B/32 variant, we increase the hidden dimension to 768 and use a patch size of 32. Finally, we develop the TranS4mer-S/16 model using a hidden dimension of 384 and a smaller patch size of 16, which produces more input tokens and, thus, is more costly. All models use 12 State Space Sequence Self-Attention (S4A) blocks. Unless otherwise noted, we use TranS4mer-S/32 as our default model for all experiments.

Training Details. For training, we use four data augmentation techniques: random crop, random flip, color jitter, and Gaussian blur. First, we crop the original frames of the video with a random size with a scale between [0.14, 1.0] and resize the cropped frames to a size of 224×224. Second, we flip the frame horizontally with a probability of 50%. Third, we apply random color jitter with a probability of 80% and drop the color to grayscale with a probability of 20%. Fourth, we apply Gaussian blur with a probability of 50%. Note that the same augmentation is applied to all input video frames.

We pretrain our model for 10 epochs with a base learning rate of 0.3. We reduce the learning rate with a cosine schedule after a linear warmup of 1 epoch. Afterward, we finetune the model for 20 epochs with a learning rate of 10^{-5}. We use the Adam optimizer [5] with a momentum of 0.9 and weight decay of 10^{-6}. Our default setting uses a batch size of 256, where each input video consists of 25 neighboring shots.

*Research done while MI was an intern at Comcast Labs.

Model	Patch #	Hidden #	MLP #	Params
TranS4mer-S/32	32	384	1536	32M
TranS4mer-B/32	32	768	3072	122M
TranS4mer-S/16	16	384	1536	32M

Table 1. Details of our TranS4mer model variants. All models use 12 S4A blocks. We use TranS4mer-S/32 as our default model.

Pseudo-boundary discovery using DTW algorithm. We use the pseudo-boundary discovery method proposed by Mun *et al.* [6] based on Dynamic Time Warping (DTW) [2] algorithm to pretrain our model using unlabeled data. The DTW algorithm finds a pseudo-boundary shot from an input video where the semantic transition is maximum. Note that pseudo boundaries might not represent an actual scene-level semantic transition. Nevertheless, the DTW algorithm divides the input video into two pseudo-scenes so that each scene contains semantically different shots.

Suppose we are given an input video as a sequence of shots, $\mathcal{V}_i = \{s_{i-m}, \ldots, s_i, \ldots, s_{i+m}\}$. The DTW algorithm divides the input video into two pseudo-scenes $\mathcal{V}_L = \{s_{i-m}, \ldots, s_{i^*}\}$ and $\mathcal{V}_R = \{s_{i^*+1}, \ldots, s_{i+m}\}$ by solving the following optimization problem using Dynamic Programming:

$$b^* = \underset{b=-m+1,\ldots,m-1}{\arg\max} \frac{1}{b+m} \sum_{j=-m+1}^{b} \mathcal{S}(\mathbf{r}_{i-m}, \mathbf{r}_{i+j}) + \frac{1}{m-b-1} \sum_{k=b+1}^{m-1} \mathcal{S}(\mathbf{r}_{i+m}, \mathbf{r}_{i+k}), \tag{1}$$

Here, $\mathbf{r}$ is a shot representation obtained by applying a linear layer on the CLS token associated with that shot, $i^* = i + b^*$ is the index of the optimal boundary shot, and $\mathcal{S}$ is the cosine similarity between two shot representations defined as $\mathcal{S}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$.

2. Additional Experimental Results

This section contains additional experimental results besides the ones described in the main draft. In particular,

Method	Backbone	AP	mIoU	AUC-ROC	F1
BaSSL	ViT-S/32	58.01	50.84	90.76	47.46
TranS4mer	ViT-S/32	**60.78**	**51.91**	**91.89**	**48.36**

Table 2. Comparison with prior works using the same backbone model (i.e., ViT-S/32). Besides achieving the best performance in all metrics, TranS4mer outperforms the prior state-of-the-art method (BaSSL [6]) by **+2.77**% AP.

in Subsection 2.1, we compare our model with the previous SOTA method, BaSSL [6], while using the same backbone network to ensure a fair comparison. Afterward, in Subsection 2.2, we perform ablation studies with different TranS4mer variants described in Table 1.

2.1. Comparison with Prior Works Using the Same Backbone Model

To ensure a fair comparison, we compare our model with the previous state-of-the-art method, BaSSL [6], using the same backbone model. In particular, we replace the backbone model of the BaSSL, i.e., ResNet50, with the backbone network used in our default model, i.e., ViT-S/32. Note that we use TranS4mer-S/32 as our default model, which uses the configuration of ViT-S/32 for the self-attention and MLP layers. We present the result in Table 2. We observe that TranS4mer outperforms BaSSL by **+2.77**% AP while using the same backbone network (ViT-S/32). Moreover, our model achieves the best performance in all metrics. This experiment suggests that the performance gain of the TranS4mer model does not come from the backbone model alone. Instead, the performance boost stems from our novel architecture that uses our proposed efficient intra-shot and inter-shot modules.

2.2. Ablation with Different TranS4mer Variants.

Next, we experiment with three different TranS4mer variants: TranS4mer-S/32, TranS4mer-B/32, and TranS4mer-S/16 (see Table 1). We present these results in Table 3, which suggest that increasing the hidden dimension and the MLP size (TranS4mer-B/32) boosts the performance compared to the TranS4mer-S/32 model by **0.13%** AP. Moreover, using a smaller patch size (16) and the same hidden dimensionality (TranS4mer-S/16) improves the performance by **0.28%** AP. However, both TranS4mer-B/32 and TranS4mer-S/16 models increase the runtime and GPU memory requirements, suggesting that with more computational resources, we can further enhance the performance of our default TranS4mer model.

3. Qualitative Results

We present several examples of correct and incorrect predictions made by our model and the previous SOTA method BaSSL [6] on the MovieNet dataset in Figure 1. Moreover, we show qualitative results on BBC and OVSD datasets in Figure 2 and Figure 3. In each of these figures (Figure 1, 2, and 3), we represent frames from the same scene by the border of the same color. The first row of each figure shows the ground truth scene boundaries; the second row shows the prediction made by the proposed TranS4mer model; and the third row shows the prediction of the prior state-of-the-art method, BaSSL [6].

Model	Mem.(GB) (↓)	Samples/s (↑)	Ap (↑)
TranS4mer-S/32	10.13	2.57	60.78
TranS4mer-B/32	17.32	1.51	60.91
TranS4mer-S/16	37.28	1.23	**61.06**

Table 3. **Model Variants.** Bigger models with larger hidden dimensions (TranS4mer-B/32) or smaller patch sizes (TranS4mer-S/16) result in better performance but incur higher computational costs.

3.1. Qualitative Results on the MovieNet Dataset

In Figure 1 (a) and (b), we present two examples of movie scenes from the MovieNet dataset [4]. In Figure 1 (a), we observe that our model can correctly identify the scene boundary where the BaSSL fails. Moreover, BaSSL produces more false positive boundaries, which might result from the lack of effective long-range modeling.

Furthermore, in Figure 1 (b), we also present a wrong prediction made by our model. However, we can see that, while our model misses the boundary shot by a small margin (one shot), BaSSL performs worse and produces more false positives.

3.2. Qualitative Results on the BBC Dataset

In Figure 2 (a) and (b), we present two examples of scenes from the BBC dataset [1]. First, in Figure 2 (a), we observe that both our method and BaSSL can correctly identify the ground-truth boundary. However, BaSSL produces more false positive predictions. Second, in Figure 2 (b), both TranS4mer and BaSSL make the wrong prediction.

3.3. Qualitative Results on the OVSD Dataset

In Figure 3 (a) and (b), we present two examples of scenes from the OVSD dataset [7]. In Figure 3 (a), our model can correctly recognize the scene boundary whereas BaSSL fails. On the other hand, in Figure 3 (b), both TranS4mer and BaSSL make the wrong predictions; however, BaSSL produces more false positive scene boundaries.

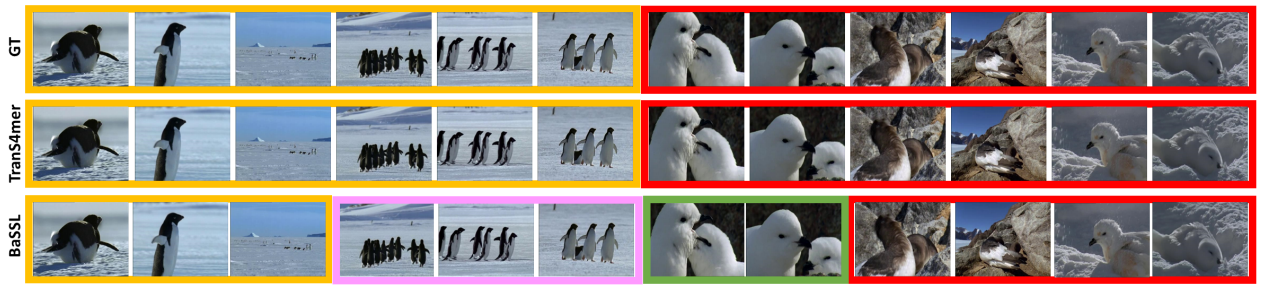
3.4. Summary of Qualitative Results

From these qualitative examples of Figure 1, 2 and 3, we observe that our proposed TranS4mer model performs better and produces less false positive predictions compared to the prior state-of-the-art method (BaSSL [6]). These observations suggest that TranS4mer can capture long-range dependencies and effectively identify scene boundaries in long

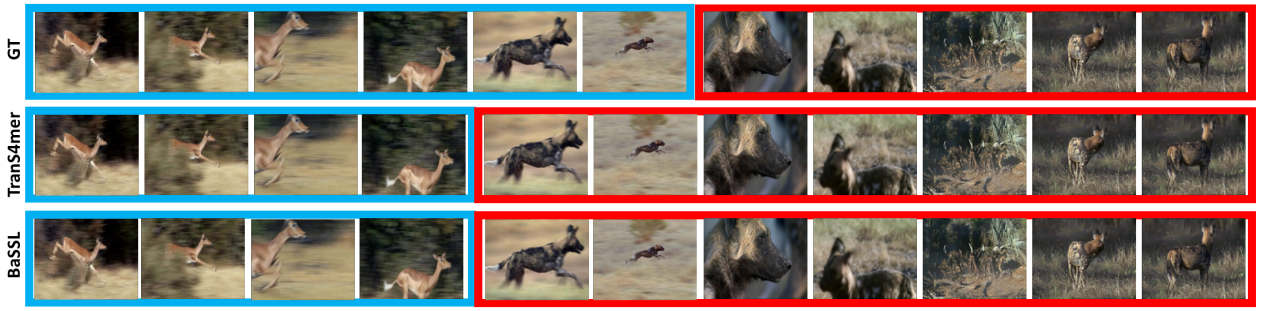
(a) TranS4mer can correctly identify the scene boundary, whereas BaSSL makes wrong prediction and also produces more false positives.

(b) Both TranS4mer and BaSSL fail to identify the scene boundary; however, BaSSL produces more false positive predictions.

Figure 1. Qualitative Results on MovieNet Dataset [4]. The border of the same color represents frames from the same scene. The first row of each figure shows the ground truth scene boundaries; the second and third rows show the predictions made by the proposed TranS4mer model and the previous SOTA method, BaSSL.



(a) Both TranS4mer and BaSSL can identify the ground-truth boundary; however, BaSSL produces more false positives.



(b) TranS4mer and BaSSL make the wrong prediction for the scene boundary.

Figure 2. Qualitative Results on the BBC Dataset [1]. The border of the same color represents frames from the same scene. The first row of each figure shows the ground truth scene boundaries; the second and third rows show the predictions made by the proposed TranS4mer model and the previous SOTA method, BaSSL.

movie videos. Moreover, these observations also match our initial hypothesis (Section 1 and Figure 1 of the main draft) that long temporal modeling is crucial for the movie scene boundary task, and models with short temporal con-

(a) TranS4mer can correctly identify the scene boundary where BaSSL fails.

(b) Both TranS4mer and BaSSL make wrong predictions, but BaSSL produces more false positives.

Figure 3. Qualitative Results on OVSD Dataset [1]. The border of the same color represents frames from the same scene. The first row of each figure shows the ground truth scene boundaries; the second and third rows show the predictions made by the proposed TranS4mer model and the previous SOTA method, BaSSL.

text might suffer from more false positive predictions. Since our model can effectively capture long-range dependencies, it performs much better at the movie scene detection task.

References

[1] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1199–1202, 2015. 2, 3, 4

[2] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994. 1

[3] 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. 1

[4] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*, pages 709–727. Springer, 2020. 2, 3

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[6] Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. Boundary-aware self-supervised learning for video scene segmentation. *arXiv preprint arXiv:2201.05277*, 2022. 1, 2

[7] Daniel Rotman, Dror Porat, and Gal Ashour. Robust and efficient video scene detection using optimal sequential grouping. In *2016 IEEE international symposium on multimedia (ISM)*, pages 275–280. IEEE, 2016. 2