TransBlast: Self-Supervised Learning Using Augmented Subspace with Transformer for Background/Foreground Separation

Islam Osman
University of British Columbia
3333 University Way, Kelowna, BC
islam.osman@ubc.ca

Mohamed Abdelpakey
University of British Columbia
3333 University Way, Kelowna, BC
mohamed.abdelpakey@ubc.ca

Mohamed S. Shehata
University of British Columbia
3333 University Way, Kelowna, BC
mohamed.sami.shehata@ubc.ca

Abstract

Background/Foreground separation is a fundamental and challenging task of many computer vision applications. The F-measure performance of state-of-the-art models is limited due to the lack of fine details in the predicted output (i.e., the foreground object), and the limited labeled data. In this paper, we propose a background/foreground separation model based on a transformer that has a higher learning capacity than the convolutional neural networks. The model is trained using self-supervised learning to leverage the limited data and learn a strong object representation that is invariant to changes. The proposed method, dubbed TransBlast, reformulates the background/foreground separation problem in self-supervised learning using the augmented subspace loss function. The augmented subspace loss function consists of two components: 1) the cross-entropy loss function and 2) the subspace that depends on Singular Value Decomposition (SVD). The proposed model is evaluated using three benchmarks, namely CDNet, DAVIS, and SegTrackV2. The performance of TransBlast outperforms state-of-the-art background/foreground separation models in terms of F-measure.

1. Introduction

Background/Foreground separation (also known as foreground segmentation) is the task of separating the moving foreground from its background. This task has gained much attention in the last few years due to its importance in many computer vision applications such as autonomous driving [22], object tracking [1, 2, 3], and surveillance [46].

Most research efforts in this area have been devoted to relying on building deep architectures which heavily depend on supervised or unsupervised Convolutional neural networks (ConvNet). For example, [39] uses an end-to-end ConvNet based on correlation learning-based edge extraction. However, these architectures usually produce objects that lack fine details. Moreover, the learning-based techniques are more powerful for extracting low-level, mid-level, and high-level features from image or video frames. However, from literature, it is observed that all ConvNet-based methods are not able to give good performance in terms of fine details in addition to the need for heavy fine-tuning. This paper introduces a novel design that is motivated by the success of Transformer architectures in Natural Language Processing (NLP) and computer vision. The proposed architecture is designed in Self-supervised learning using Barlow twins [58] to leverage the limited data for background/foreground separation.

The proposed architecture (TransBlast) consists of five components: 1) convolutional encoder, 2) transformer, 3) convolutional decoder, 4) multiple-output combiner block (MOC-block), and 5) subspace learning module. The convolutional encoder embeds the input into feature maps. The feature maps are then divided into patches and used as an input to the transformer. The output of the transformer is patches with the same number of input patches. These patches are reshaped to form feature maps. The reshaped feature maps are then used as an input to the convolutional decoder. An output is produced from each block in the decoder. Finally, all outputs are scaled and combined using the MOC-block to produce multi-scale semantic feature maps. These feature maps are then used during the optimization to train TransBlast based on subspace learning module that help producing a high fidelity segmented object. Moreover,

the same feature maps are used to produce the final output.

TransBlast combines Convolutional Neural Network (CNN) Encoder and Decoder in addition to Transformer Encoder and Decoder in the same architecture. Consequently, TransBlast inherits a strong inductive bias from CNN, which allows efficient training and convergence. On the other hand, TransBlast relaxes the inductive bias from the transformer, which increases the overall network generalization. Moreover, TransBlast augments the loss function with a subspace that is learned based on Singular Value Decomposition (SVD). This subspace controls the weights of the MLP network as shown in Figure 1 to produce the Eigenvalues that are maximized during the training to force the MLP network to learn the subspace. The learned subspace forces the network to predict the foreground fine details. It is worth mentioning that TransBlast is trained using self-supervised learning to leverage the limited data and learn strong object representation. This representation is robust against different transformations.

To summarize, the main contributions of this paper are:

- TransBlast is a novel self-supervised architecture with a transformer for background/foreground separation. The proposed architecture is designed such that the segmented foreground has fine details. The fine details are inherited from the augmented loss with the subspace learning.

- A novel augmented loss function that uses the subspace learning as a component in the loss function to augment the cross-entropy component. The augmentation is done by maximizing the Eigenvalues of the low-rank of the subspace during the optimization.

- Using self-supervised learning to leverage the limited data and learn strong object representation that is robust to different challenging scenarios.

2. Related work

In this section, we review different moving object detection and segmentation techniques. The techniques are categorized into three different categories: 1) Unsupervised learning techniques (i.e., Statistical-based and subspace learning methods), 2) supervised learning techniques (i.e., Deep learning-based techniques), and 3)semi-supervised learning.

Unsupervised learning techniques: This category uses holistic methods to filter foreground objects from the background without labels as in [8, 50]. In [17] null space was used to present the image in the null feature space, thus maximizing the distance between the foreground and background features and enabling the segmentation of the foreground objects. In [24, 12, 38] optical flow was used to detect foreground objects. While these methods can provide good accuracy in foreground segmentation, they heavily depend on hyper-parameter fine-tuning that have to be adjusted for each dataset. Consequently, making them less adaptable and robust to changes from the datasets they were initially designed based on.

Supervised learning methods: The main advantage of deep learning-based techniques is their ability to learn crucial features that help distinguish between the foreground and background classes as in [36, 9]. This category can be further divided into two different sets: 1) Convolution-based networks and 2) Transformer-based networks.
In Convolution-based networks, the network consists of convolution, pooling, batch normalization, etc. By stacking the layers to form a pyramid, with the higher-level layers learning features from wider receptive fields. In these methods [13, 21, 41, 42, 43, 44, 7, 35], convolution-based network is used as the backbone for the architecture. Overall, the performance in this sub-category is relatively good compared to the statistical-based techniques. However, the predicted foreground usually lacks the fine details due to large receptive field and pooling layers.

Transformers-based networks have recently raised a lot of interest in solving computer vision tasks [15, 37, 14, 60], In DETR an end-to-end object detector was introduced in which the image is represented through a sequence of spatial features enabling the use of traditional transformer architectures that are usually used in NLPs. This method simplifies the traditional detection pipeline while achieving comparable results with CNN-based architectures. However, the method suffers from slow convergence. In a follow-up work, Deformable DETR [60] was proposed to improve on DETR by adding a deformable attention module and applying a pre-filter to extract key elements out of the feature map. SETR [59] introduced a new segmentation model treating the input image as a sequence of image patches with patch embedding. In [57] another end-to-end object segmentation method was introduced namely VisTR that is based on transformers. VisTR introduced a new strategy for sequence matching and segmentation. [16] a new approach uses sparse attention operator was introduced to solve the computational complexities in other transformer-based methods, thus enabling the processing of high-resolution videos.

Semi-supervised learning: Leverages the usage of few-labeled data and maintain good performance. In [19], they proposed an algorithm that is composed of segmentation, background initialization, graph construction, unseen sampling, and a semi-supervised learning method showing one bound for the sample complexity in semi-supervised learning, and two bounds for the condition number of the Sobolev norm. In [20], two architectures are proposed for moving object segmentation (MOS) using semi-supervised

learning and a new evaluation procedure for GSP-based MOS algorithms.

Even though the previous methods achieve a good performance, they lack the fine details in the predicted outputs. Consequently, the performance is degraded over time. TransBlast achieves high performance by maintaining the fine details of the segmented foreground over time.

3. TransBlast architecture

The proposed TransBlast is a novel Transformer-based network architecture for background/foreground separation. The loss function of TransBlast is designed to be augmented by subspace learning. The architecture of TransBlast is shown in Figure 1. TransBlast is an encoder-decoder architecture. The encoder is divided into a convolutional encoder and a Transformer encoder. The same goes for the decoder. TransBlast uses transformer due to its learning capacity is better than the convolutional neural network (CNN). Hence, it can learn a stronger object representation than that of a CNN.

TransBlast input is a single RGB frame with a fixed dimensions $256 \times 256 \times 3$. A convolutional encoder is used to transform the input from the image space to the feature space. The CNN encoder consists of 3 residual blocks with a max-pooling layer between every two residual blocks. Each residual block has 2 convolutional layers with ReLU activation function followed by skip connection from input to the output of the second convolutional layer. Then, batch normalization is added after skip connection.The number of filters and the filter size is fixed in all convolutional layers of the network, which are 64 and 3×3, respectively. The output feature maps of the CNN encoder have dimensions $64 \times 64 \times 64$ corresponding to height, width, and channel (i.e., $H \times W \times C$). The feature maps are divided into patches with the size $4 \times 4 \times 64$. This will result in $16 \times 16 = 256$ patches. Each patch is flattened to form $1 - D$ vector of size 1024. All flatten patches are used as an input to the transformer encoder.

The transformer encoder consists of 6 encoder block. In each encoder block, a layer normalization is applied to the input patches x. Then, the input is transformed to three inputs q, k, and v as following:

$$q = x + p, k = x + p, v = x \tag{1}$$

where q, k, and v are the query, key, and value, respectively. p is the position encoding which is a learnable vector. An multi-head attention layer A_{mh} is applied on q, k, and v, such that:

$$A(q, k) = \frac{q \times k^T}{\sqrt{d_k}}, \tag{2}$$

$$A(q, k) = softmax(A(q, k)), \tag{3}$$

where d_k is a scaling factor which is the dimension of k.

$$A_{mh}(q, k, v) = proj(A(q, k) \times v), \tag{4}$$

where $proj$ is a linear projection layer. A skip connection from before the A_{mh} to after it is added. Then, a layer normalization followed by feed-forward network ffn with another skip connection is applied as following:

$$x = x + A_{mh}(q, k, v), \tag{5}$$

$$x = layer_{norm}(x), \tag{6}$$

$$x = x + ffn(x). \tag{7}$$

where ffn is the feed-forward network consists of two linear projections layers. Then x is feed as an input to the next encoder block along with the position encoding p.

The transformer decoder also consists of 6 decoder blocks. Each decoder block has the same structure as the encoder block but has two A_{mh} layers. The decoder output is a set of flattened patches with the same dimensions as the input flattened patches of the encoder. These output patches are unflattened to form $4 \times 4 \times 64$ patches. Then, the unflattened patches are reshaped to form feature maps of dimensions $64 \times 64 \times 64$.

The convolutional decoder uses the reshaped transformer output as an input and decodes it to output a binary image where black pixel means background pixel and white pixel means foreground object pixel. The architecture of the CNN decoder is 2 residual blocks with an up-sampling layer between every two residual blocks. The two residual blocks have the same number of filters, which is 64, and ReLU activation function. A convolutional layer with a 2 filter and sigmoid activation function is added to each block. This convolutional layer is used to output 2 binary images from each block—one binary image for the segmented foreground and another one for the predicted boundaries of the foreground. An extra up-sampling layer is added before the convolutional layer of the first block to make the output match the final output size.

The multiple-output combiner block (MOC-block) consists of a residual block that uses the output feature maps of the convolutional decoder and the 2 outputs as an input. This residual block learns to combine the feature maps and different outputs to produce feature maps that highlight the moving objects and their boundaries. At the end of MOC-block a single convolutional layer with one filter and sigmoid activation function to produce the final output, which is a binary image with dimensions $256 \times 256 \times 1$.

The subspace module augments the cross-entropy loss function by calculating the SVD for the output tensor of the MOC-block. Then, the decomposed matrix S_1 is extracted to produce the low-rank approximation matrix S_2, as shown

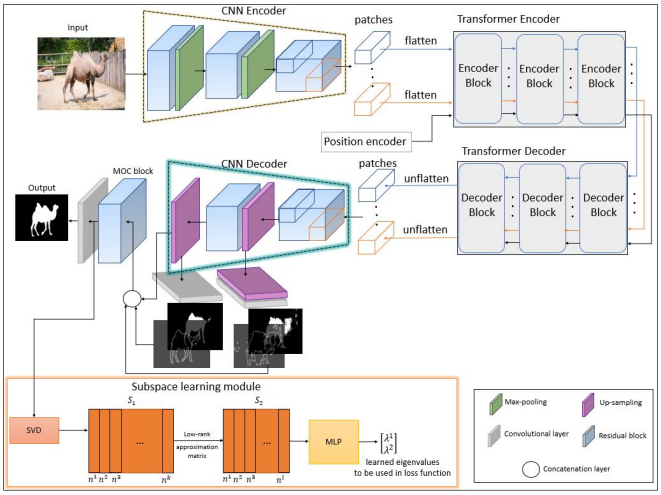

Figure 1. TransBlast architecture.

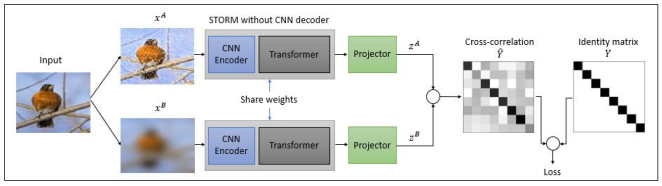

Figure 2. TransBlast architecture during training using Barlow twins.

in Figure 1. The low-rank approximation matrix S_2 is calculated from S_1 as follows:

$$S_1 = U\Sigma V^T \tag{8}$$

$$S_2 = U_k \Sigma_k V_k^T \tag{9}$$

where U and V are orthogonal matrices, and Σ is a diagonal matrix with non-negative entries. Note that, the diagonal in the Σ matrix is ordered descending with respect to the singular values. The k is the top singular values (i.e., low-rank). The low-rank approximation matrix S_2 is fed into a multi-layer perception (MLP), which is 4-layers network. The output of the MLP is two Eigenvalues. The objective is to maximize the Eigenvalues during the training when added to the cross-entropy as described in the next section.

4. TransBlast training

TransBlast is trained using a self-supervised learning technique that allows the model to learn a strong object representation for the input image using unlabeled data. The learning process is divided into two phases: 1) Pre-text task (learning representation using Barlow twins [58]) and 2) Down-stream task (use the trained model to learn to segment foreground objects).

In the pre-text task phase, the network architecture is changed to use Barlow twins as a representation learning technique. The CNN decoder is replaced by 3 linear projection layers. Each projection layer has 4096 neurons. The CNN decoder is removed because it is a task-specific part (i.e., related to the foreground segmentation task only), and in self-supervised learning, we focus on learning a strong object representation that is independent of any task (i.e., task-agnostic representation). Then, TransBlast is transformed to Siamese architecture [11], where two branches of the same network with shared weights but use different inputs. The modified TransBlast for using Barlow twins is shown in Figure 2. The different inputs are augmented from the original input image. TransBlast uses data augmentation techniques such as random image crop, horizontal flipping, color jitter, grayscale, gaussian blur, and solarization. Each data augmentation technique is applied with a probability. The two branches extract embedding vectors z^A and z^B for the different inputs x^A and x^B. A cross-correlation is applied on these two embedding vectors, which is computed as:

$$c_{ij} = \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b (z^A_{b,i})^2} \sqrt{\sum_b (z^B_{b,j})^2}}. \tag{10}$$

If the resulted cross-correlation c is an identity matrix I, this means both embedding vectors are the same, and this is the target of Barlow twins technique to force the network

to extract embedding vector that is invariant to the different data augmentation techniques. This is done by computing a loss function that minimizes the difference between the computed cross-correlation and the identity matrix. This loss function ℓ_{BT} is computed as:

$$\ell_{BT} = \sum_i (I_{ii} - c_{ii})^2 + \alpha \sum_i \sum_{i \neq j} (I_{ij} - c_{ij})^2. \quad (11)$$

Since, the identity matrix I has ones on the diagonal I_{ii} and zeros on the rest of the matrix, the loss function can be rewritten as:

$$\ell_{BT} = \sum_i (1 - c_{ii})^2 + \alpha \sum_i \sum_{i \neq j} {c_{ij}}^2. \quad (12)$$

In down-stream task phase, the CNN decoder is added to the network trained by Barlow twins. All parameters in the network that have been updated using Barlow twins will be kept unchanged (i.e., frozen parameters) during this phase. Only the parameters in CNN decoder will be updated. This CNN decoder transforms the embedding representation that is learned by Barlow twins to segmenting objects in foreground domain. The input in this phase is a single image and the output is the binary image as mentioned in previous section. The loss function in this phase is computed as:

$$\ell_{BFS} = \ell(y, \hat{y}) + \sum_{i=1}^{2} (\ell(y, \hat{y}^i) + \ell(y_b, \hat{y}_b^i)). \quad (13)$$

where ℓ_{BFS} is the loss function for the background/foreground separation, ℓ is a cross-entropy loss function, y is the ground-truth, $\hat{y}$ is the output of the network, $\hat{y}^i$ is the output of i^{th} decoder block, y_b is the boundaries ground-truth, and $\hat{y}_b^i$ is the predicted boundaries of the i^{th} decoder block.

The loss function of the subspace learning module (SLM) that augments the cross-entropy is given by:

$$\ell_{SLM} = -\beta \sum_{i=1}^{2} (\lambda_i). \quad (14)$$

Where β is an empirical strength value that is set to 0.2, λ is the Eigenvalue that is produced by MLP. The final augmented loss function is calculated as follows:

$$\ell_{final} = \ell_{BFS} - \ell_{SLM}. \quad (15)$$

5. Experiments

Two different versions of the proposed model are evaluated in this section. The first version is TransBlast-LA, which is the network without loss augmentation. The purpose of this version is to show the strength of the proposed network alone and how loss augmentation will affect the performance of the network. The second version is TransBlast, which is the full proposed model. Evaluation of the proposed TransBlast is done using three benchmarks CDNet, DAVIS, and SegTrackV2. Another dataset is used in the Barlow twins training phase, which is Mini-ImageNet.

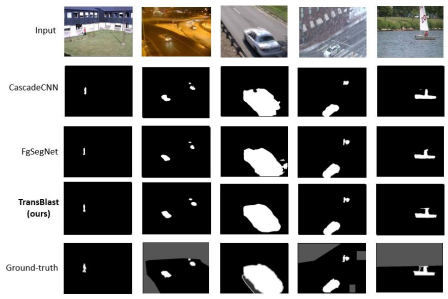

Figure 3. Sample results from CDNet dataset.

5.1. Benchmarks

Mini-ImageNet [51] is a subset from the large scale dataset ImageNet [45]. Mini-ImageNet consists of 100 different classes, with overall 60,000 colored images, 600 per class. This dataset is used in the pretext task of self-supervised learning to learn a strong object representation using the Barlow twins technique.

ChangeDetection.Net (CDNet) 2014 [55] dataset consists of 11 different challenges. Each challenge has a number of videos ranges from 4 to 6, the number of frames in each video ranges from 1000 to 8000. The total number of videos in the dataset is 53 videos.

Densely Annotated VIdeo Segmentation (DAVIS) [40] is a video object segmentation dataset. DAVIS consists of 50 videos. Each video has a number of frames ranges from 50 to 104. In each video, a single object is annotated, which is the object of interest in this video.

SegTrackV2 [30] is a video multiple objects segmentation dataset. The number of videos in the dataset is 14, and the number of frames ranged from 21 to 279 in each video.

5.2. Results

The proposed model TransBlast is pre-trained using Barlow twins to learn strong object representation from the Mini-ImageNet dataset. Then, in the downstream phase, the convolutional decoder is added, and its parameters are updated using the foreground segmentation datasets. In this phase, the rest of the network parameters is also fine-tuned.

In Table 1, the proposed model is trained on 200 frames from each video, and testing is done on the rest of the frames. The proposed model outperforms state-of-the-art models even without loss augmentation. This proves that transformer-based network with self-supervised initializa-

Table 1. Results of TransBlast compared to state-of-the-art methods on CDNet, methods with * means the results are reproduced.

Method	FPR	FNR	Re	Pr	F-Measure
IUTIS-5 [6]	0.005	0.215	0.789	0.808	0.771
SemanticBGS [10]	0.004	0.211	0.789	0.830	0.789
BSUV-Net [49]	0.005	0.179	0.820	0.811	0.786
BSUV-Net-SBGS [49]	0.005	0.182	0.817	0.831	0.798
DeepBS [4]	0.009	0.245	0.754	0.833	0.745
SuBSENSE [47]	0.009	0.187	0.812	0.751	0.741
WisenetMD [29]	0.009	0.182	0.817	0.766	0.753
PAWCS [48]	0.005	0.228	0.771	0.785	0.740
FgSegNetV2* [33]	0.005	0.080	0.893	0.741	0.801③
CascadeCNN* [56]	0.007	0.097	0.821	0.771	0.786
TransBlast-LA (ours)	0.004	0.086	0.841	0.797	0.815②
TransBlast (ours)	0.004	0.083	0.867	0.811	**0.831**①

Table 2. Results of TransBlast compared to state-of-the-art models in DAVIS. * means the results are reproduced.

Model	PT	OF	F-Measure
CNIM [5]	Yes	Yes	0.850②
LUCID [23]	Yes	No	0.820
STRCF [28]	No	No	0.816
DSRFCN3D [27]	No	No	0.766
DCL [31]	No	No	0.711
DHS [34]	No	No	0.758
RFCN [52]	No	No	0.740
LGFOGR [54]	No	No	0.601
RST [26]	No	No	0.645
SAG [53]	No	No	0.548
UOVOS [61]	No	No	0.772
FgSegNet* [33]	No	No	0.847③
CascadeCNN* [56]	No	No	0.814
TransBlast-LA (ours)	No	No	0.845
TransBlast (ours)	Yes	No	**0.859**①

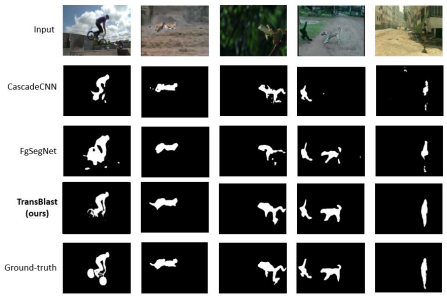

Figure 4. Sample results from SegTrackV2 dataset.

Table 3. Results of TransBlast compared to state-of-the-art models in SegTrackV2. * means the results are reproduced.

Model	F-Measure
DCL [31]	0.730
UOVOS [61]	0.643
SSAV [18]	0.801
MBNM [32]	0.716
DSRFCN3D [27]	0.878
RFCN [52]	0.737
DHS [34]	0.762
LGFOGR [54]	0.614
RST [26]	0.677
SAG [53]	0.646
GDHF [25]	0.868
STRCF [28]	0.899②
FgSegNet* [33]	0.780
CascadeCNN* [56]	0.767
TransBlast-LA (ours)	0.882③
TransBlast (ours)	**0.904**①

Figure 5. Sample results from DAVIS dataset.

tion outperforms the CNN-based networks when the model is trained using the augmented loss function, the performance increase, even more, achieving 0.831 in terms of F-measure.

Results in Tables 2 and 3 are produced by training the model using 25 frames from each video in the dataset. If the video has less than 25 frames, as in some videos of SegTrackV2, the model is trained on half of the frames.

Results of TransBlast against state-of-the-art models in

Table 4. Effect of each component in TransBlast.

Transformer	MOC-block	Boundary optimization	Self-supervised	Loss augmentation	F-Measure
No	No	No	No	No	0.702
Yes	No	No	No	No	0.836
Yes	Yes	No	No	No	0.857
Yes	Yes	Yes	No	No	0.873
Yes	Yes	Yes	Yes	No	0.882
Yes	Yes	Yes	Yes	Yes	0.904

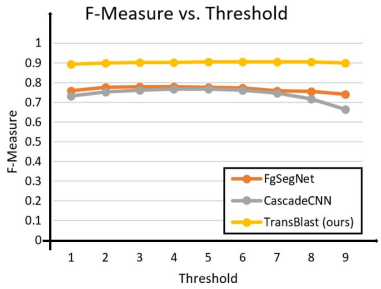

Figure 6. Results of TransBlast against three other models in SegTrackV2 using different threshold values.

the DAVIS dataset are shown in Table 2. TransBlast-LA achieved the 4^{th} place. Training the model using the proposed loss function caused the performance to be significantly improved from the 4^{th} place to the 1^{st} place. In this table, (PT) means the model requires pre-training, and (OF) means the model applies online fine-tuning.

Table 3 shows the results of the proposed model against state-of-the-art models in SegTrackV2. TransBlast-LA is on the 3^{rd} place compared to state-of-the-art, and when applied the loss augmentation, the model outperformed state-of-the-art and achieved 1^{st} place. The results of the two models FgSegNet and CascadeCNN are produced by training the models using the same training set. The reason for producing the results of other models is to compare the outputs visually as shown in Figures 3, 4, and 5. Also, to compare the output of different models against different threshold values as shown in the Ablation study subsection 5.3.

5.3. Ablation study

A study is made to show the effect of different threshold values on the proposed model against other models. As shown in Figure 6, the performance of all CNN-based models is affected by the threshold value. The performance keeps increasing until a certain threshold value then starts to decrease again. On the other hand, the performance of the proposed TransBlast does not get affected by the threshold value. This is due to the strong learned object representation, and also because the TransBlast is based on the transformer, which is a well-known for its ability to learn a representation that is better than the CNN in many different computer vision tasks (e.g., Object detection in [14]).

The effect of different components in TransBlast is shown in Table 4. The results in this table are produced using the SegTrackV2 dataset. The components presented in the table are transformer, MOC-Block, boundary optimization (i.e., optimized to extract the boundaries), self-supervised learning using Barlow twins, and loss augmentation. As shown in Table 4, when the network had only the convolutional encoder/decoder, the performance was 0.702 in F-measure. After adding the transformer, the performance becomes 0.836. Adding MOC-block, which combines multiple outputs at different scales, improved the performance by 2.1%. When the network is trained to extract boundaries using the proposed loss function in Eq. 13 in Section 4, the performance improved by almost 1.6%. Pre-training the network self-supervised learning improved the performance by 0.9%. Finally, the full proposed TransBlast achieved 0.904 F-Measure, which is the first place in this dataset.

6. Conclusion

The proposed TransBlast is a transformer-based architecture for background/foreground separation. TransBlast takes advantage of Convolutional Neural Network (CNN), where the training benefits from the strong inductive bias. Moreover, TransBlast relaxes the inductive bias in the Transformer leading to generalization and fine details in the predicted output.

TransBlast is trained using self-supervised learning to leverage the limited labeled data and learn strong object representation. The loss function of background/foreground separation in TransBlast is augmented by the subspace that is based on SVD. The augmented loss function uses MLP to produce Eigenvalues which are used during the optimization process to learn the subspace. To make the loss more computationally efficient, TransBlast uses the subspace-based on the low-rank of the matrix produced by SVD.

TransBlast is evaluated using three benchmarks, namely CDNet, DAVIS, and SegTrackV2. The performance of TransBlast outperforms state-of-the-art background/foreground separation models, achieving 0.831, 0.859, and 0.904 on DAVIS and SegTrackV2, respectively.

References

[1] Mohamed H Abdelpakey and Mohamed S Shehata. Domainsiam: Domain-aware siamese network for visual object tracking. In *ISVC*, pages 45–58. Springer, 2019.

[2] Mohamed H Abdelpakey and Mohamed S Shehata. Dpsiam: Dynamic policy siamese network for robust object tracking. *IEEE Transactions on Image Processing*, 29:1479–1492, 2019.

[3] Mohamed H Abdelpakey, Mohamed S Shehata, and Mostafa M Mohamed. Denssiam: End-to-end densely-siamese network with self-attention model for object tracking. In *Advances in Visual Computing: 13th International Symposium, ISVC 2018, Las Vegas, NV, USA, November 19–21, 2018, Proceedings*, volume 11241, page 463. Springer, 2018.

[4] Mohammadreza Babaee, Duc Tung Dinh, and Gerhard Rigoll. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition*, 76:635–649, 2018.

[5] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5977–5986, 2018.

[6] Simone Bianco, Gianluigi Ciocca, and Raimondo Schettini. Combination of video change detection algorithms by genetic programming. *IEEE Transactions on Evolutionary Computation*, 21(6):914–928, 2017.

[7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[8] Thierry Bouwmans. Recent advanced statistical background modeling for foreground detection-a systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011.

[9] Thierry Bouwmans, Sajid Javed, Maryam Sultana, and Soon Ki Jung. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Networks*, 117:8–66, 2019.

[10] Marc Braham, Sébastien Piérard, and Marc Van Droogenbroeck. Semantic background subtraction. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4552–4556. IEEE, 2017.

[11] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

[12] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010.

[13] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1483–1498, 2021.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[15] 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.

[16] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5912–5921, 2021.

[17] Agwad ElTantawy and Mohamed S Shehata. Local null space pursuit for real-time moving object detection in aerial surveillance. *Signal, Image and Video Processing*, 14(1):87–95, 2020.

[18] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8554–8564, 2019.

[19] Jhony H Giraldo, Sajid Javed, and Thierry Bouwmans. Graph moving object segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2020.

[20] Jhony H Giraldo, Sajid Javed, Maryam Sultana, Soon Ki Jung, and Thierry Bouwmans. The emerging field of graph signal processing for moving object segmentation. In *International Workshop on Frontiers of Computer Vision*, pages 31–45. Springer, 2021.

[21] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[22] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 5390–5399, 2019.

[23] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. *International Journal of Computer Vision*, 127(9):1175–1197, 2019.

[24] Jens Klappstein, Tobi Vaudrey, Clemens Rabe, Andreas Wedel, and Reinhard Klette. Moving object segmentation using optical flow and depth information. In *Pacific-Rim Symposium on Image and Video Technology*, pages 611–623. Springer, 2009.

[25] Hieu Le, Vu Nguyen, Chen-Ping Yu, and Dimitris Samaras. Geodesic distance histogram feature for video segmentation. In *Asian Conference on Computer Vision*, pages 275–290. Springer, 2016.

[26] Trung-Nghia Le and Akihiro Sugimoto. Contrast based hierarchical spatial-temporal saliency for video. In *Image and Video Technology*, pages 734–748. Springer, 2015.

[27] Trung-Nghia Le and Akihiro Sugimoto. Deeply supervised 3d recurrent fcn for salient object detection in videos. In *BMVC*, volume 1, page 3, 2017.

[28] Trung-Nghia Le and Akihiro Sugimoto. Video salient object detection using spatiotemporal deep features. *IEEE Transactions on Image Processing*, 27(10):5002–5015, 2018.

[29] Sang-ha Lee, Gyu-cheol Lee, Jisang Yoo, and Soonchul Kwon. Wisenetmd: Motion detection using dynamic background region analysis. *Symmetry*, 11(5):621, 2019.

[30] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013.

[31] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 478–487, 2016.

[32] Siyang Li, Bryan Seybold, Alexey Vorobyov, Xuejing Lei, and C-C Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 207–223, 2018.

[33] Long Ang Lim and Hacer Yalim Keles. Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications*, 23(3):1369–1380, 2020.

[34] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 678–686, 2016.

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[36] Murari Mandal and Santosh Kumar Vipparthi. An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[37] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[38] Etienne Mémin and Patrick Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719, 1998.

[39] Prashant W Patil, Kuldeep M Biradar, Akshay Dudhane, and Subrahmanyam Murala. An end-to-end edge aggregation network for moving object segmentation. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8149–8158, 2020.

[40] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[42] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[46] Nicolas Scheiner, Florian Kraus, Fangyin Wei, Buu Phan, Fahim Mannan, Nils Appenrodt, Werner Ritter, Jurgen Dickmann, Klaus Dietmayer, Bernhard Sick, and Felix Heide. Seeing around street corners: Non-line-of-sight detection and tracking in-the-wild using doppler radar. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[47] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, 2014.

[48] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. A self-adjusting approach to change detection based on background word consensus. In *2015 IEEE winter conference on applications of computer vision*, pages 990–997. IEEE, 2015.

[49] Ozan Tezcan, Prakash Ishwar, and Janusz Konrad. Bsuv-net: A fully-convolutional neural network for background subtraction of unseen videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2774–2783, 2020.

[50] Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE signal processing magazine*, 35(4):32–55, 2018.

[51] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[52] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *European conference on computer vision*, pages 825–841. Springer, 2016.

[53] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Salient object detection with recurrent fully convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1734–1746, 2018.

[54] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Transactions on Image Processing*, 24(11):4185–4196, 2015.

[55] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An

expanded change detection benchmark dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 387–394, 2014.

[56] Yi Wang, Zhiming Luo, and Pierre-Marc Jodoin. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, 96:66–75, 2017.

[57] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021.

[58] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[59] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[61] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. Unsupervised online video object segmentation with motion property understanding. *IEEE Transactions on Image Processing*, 29:237–249, 2019.