"Motion-Attentive Transition for Zero-Shot Video Object Segmentation"

M.Sc. Data Science
Computer Vision

Aniket Santra (MDS202106)
Avik Das (MDS202112)
Meghna Mondal (MDS202123)

Instructor:
Dr Kavita Sutar

Lecturer, Chennai Mathematical Institute
ksutar@cmi.ac.in
https://www.cmi.ac.in/~ksutar

22nd April, 2023

Contents

Work contribution

Aniket Santra	Understanding the theoretical concepts with group Implemented part of the code using Matlab & Python Generated optical flow of DAVIS-16 dataset Presented part of the talk
Avik Das	Understanding the theoretical concepts and helping the group to understand the same Wrote the Modules part in the Report Helped in understanding the code Presented part of the talk
Meghna Mondal	Understanding the theoretical concepts and did research on the related works done previously Wrote the Introductory part in this report Helped in understanding the code Presented part of the talk

1 Introduction

The task of automatically segmenting primary object(s) from videos has gained significant attention in recent years, and has a powerful impact in many areas of computer vision, including surveillance, robotics and autonomous driving. However, due to the lack of human intervention, in addition to the common challenges posed by video data (e.g. appearance variations, scale changes, background clutter), the task faces great difficulties in accurately discovering the most distinct objects throughout a video sequence. Early nonlearning methods typically address this using handcrafted features, e.g. motion boundary , saliency and point trajectories. More recently, research has turned towards the deep learning paradigm, with several studies attempting to fit this problem into a zero-shot solution . These methods generally learn a powerful object representation from large-scale training data and then adapt the models to test videos without any annotations, I.e., Zero-shot video segmentation aims to automatically segment objects in a video without the need for explicit training on the specific objects or scene.

Object motion has always been considered as an informative cue for automatic video object segmentation. This is largely inspired by the remarkable capability of motion perception in the human visual system (HVS) , which can quickly orient attentions towards moving objects in dynamic scenarios. In fact, human beings are more sensitive to moving objects than static ones, even if the static objects are strongly contrasted against their surroundings.

By considering information flow from motion to appearance, we can alleviate ambiguity in object appearance (e.g. visually similar to the surroundings), thus easing the pressure in representation learning of objects. However, in the context of deep learning, most segmentation models do not leverage this potential.

Motivated by these observations, we propose a MotionAttentive Transition Network (MATNet) for zero-shot video object segmentation (ZVOS) within an encoder-bridge decoder framework which not only inherits the superiorities of two-stream models for multimodal feature learning, but also progressively transfers intermediate motion-attentive features to facilitate appearance learning. The transition is carried out by multiple MotionAttentive Transition (MAT) blocks.

1.1 Abstract

The paper present a novel Motion-Attentive Transition Network (MATNet) for zero-shot video object segmentation, which provides a new way of leveraging motion information to reinforce spatio-temporal object representation. We discussed about an asymmetric attention block, called Motion-Attentive Transition (MAT), which is designed within a two-stream encoder, which transforms appearance features into motion-attentive representations at each convolutional stage. In this way, the encoder becomes deeply interleaved, allowing for closely hierarchical interactions between object motion and appearance. This is superior to the typical two-stream architecture, which treats motion and appearance separately in each stream and often suffers from overfitting to appearance information. Additionally, a bridge network is proposed to obtain a compact, discriminative and scale-sensitive representation for multilevel encoder features, which is further fed into a decoder to achieve segmentation results. We analysed the performance of the model on datasets (i.e. DAVIS-16, FBMS and Youtube-Objects) and get that our model achieves compelling performance against the state-of-the-arts. We reviewed the architecture of the model and also reproduce the results for DAVIS-16 Dataset .

2 Related Work

2.1 Automatic Video Object Segmentation

Automatic, or unsupervised, video object segmentation aims to segment conspicuous and eye-catching objects without any human intervention. Traditional methods require no training data and typically design heuristic assumptions (e.g. motion boundary , objectness , saliency and long-term point trajectories) for segmentation. Now in many methods solve this task with zero-shot solutions and utilize motion because of its complementary role to object appearance. They typically adopt heuristic methods to fuse motion and appearance cues or use two-stream networks to learn spatio-temporal representations in an end-to-end fashion. However, a major drawback of these approaches is that they fail to consider the importance of deep interactions between appearance and motion in learning rich spatiotemporal features. To address this issue, we propose a deep interleaved two-stream encoder, in which a motion transition module is leveraged for more effective representation learning.

2.2 Neural Attention

Neural attention has been widely used in recent neural networks for various tasks, such as object recognition , re-identification ,visual saliency and medical imaging . It allows the networks to focus on the most informative parts of the inputs. In our model, neural attention is used in two ways: first, in the encoder network, soft attention is applied independently to intermediate appearance or motion feature maps, and motion attention is further transferred to enhance the appearance attention. Second, in the bridge network, a scale-sensitive attention module is designed to obtain more compact features.

3 Proposed Method

Our proposed model MATNet is an end-to-end deep neural network for ZVOS, consisting of three concatenated networks, i.e. an interleaved encoder, a bridge network and a decoder.

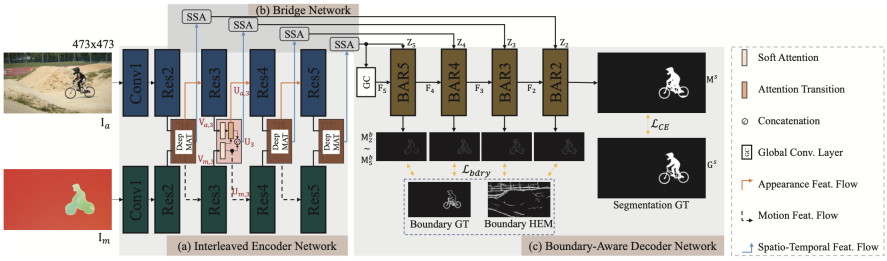

Figure 1: Pipleline of MATNet. The frame $\mathbf{I}_a$ and flow $\mathbf{I}_m$ are first input into the interleaved encoder to extract multi-scale spatio-temporal features $\mathbf{U}_i$. At each residual stage, we break the original information flow in ResNet. Instead, a deep MAT block is proposed to create a new interleaved information flow by simultaneously considering motion $\mathbf{V}_{m,i}$ and appearance $\mathbf{V}_{a,i}$. $\mathbf{U}_i$ is further fed into the decoder via the bridge network to obtain boundary results $\mathbf{M}_2^b \sim \mathbf{M}_5^b$ and the segmentation $\mathbf{M}^s$.

3.1 Interleaved Encoder Network

Our encoder relies on a two-stream structure to jointly encode object appearance and motion, which has been proven effective in many related video tasks. Unlike previous works that treat the two streams equally, our encoder includes a MAT block at each network layer, which offers a motion-to-appearance pathway for information propagation. Here, we take the first five convolutional blocks of ResNet-101 as the backbone for each stream. Given an RGB frame I_a and its optical flow map I_m, the encoder extracts intermediate

features $V_{a,i} \in R^{W\times H\times C}$ and $V_{m,i} \in R^{W\times H\times C}$, respectively, at the time $i-th, (i \in 2,3,4,5)$ residual stage. The MAT block $\mathcal{F}_{MAT}$ enhances these features as follows:

$$U_{a,i}, U_{m,i} = \mathcal{F}_{MAT}(V_{a,i}, V_{m,i})$$

where $U_{,i} \in R^{W\times H\times C}$ indicates the enhanced features. We then obtain the spatio-temporal object representation U_i at the $i-th$ stage as

$$U_i = Concat(U_{a,i}, U_{m,i}) \in R^{W\times H\times 2C}$$

which is further fed into the down-stream decoder via a bridge network.

3.2 Bridge Network

The bridge network is expected to selectively transfer encoder features to the decoder. It is formed by SSA modules, each of which takes advantage of the encoder feature U_i at the $i-th$ stage and predicts an attention-aware feature Z_i . This is achieved by a two-level attention scheme, wherein the local-level attention adopts channel-wise and spatial-wise attention mechanisms to focus input features on the correct object regions as well as suppress possible noises existing in the redundant features, while the global-level attention aims to re-calibrate the features to account for objects of different sizes.

3.3 Decoder Network

The decoder network takes a coarseto-fine scheme to carry out segmentation. It is formed by four BAR modules, i.e. $BAR_i, i \in 2,3,4,5$, each corresponding to the $i-th$ residual block. From BAR_5 to BAR_2, the resolution of feature maps gradually increases by compensating for high-level coarse features with more low-level details. The BAR_2 produces the finest feature map, whose resolution is 1/4 of the input image size. It is processed by two additional layers, conv(3×3,1) → sigmoid, to obtain the final mask output $M^s \in R^{W\times H}$. . As follows, we will introduce the three proposed modules (i.e. MAT, SSA, BAR) in detail. For simplicity, we omit the subscript i.

4 Modules

4.1 Motion-Attentive Transition Module

The MAT module is comprised of two units:

- Soft Attention (SA) unit
- Attention Transition (AT) unit

4.1.1 Soft Attention

This unit softly weights the input feature map V_m (or V_a) at each spatial location by using an 1x1 convolution kernel.

$$A_m = softmax(w_m * V_m) \tag{1}$$

After this, we can obtain the attention enhanced feature by performing element-wise multiplication to each channel of V_m (or V_a)

$$\tilde{U_m^c} = A_m \odot V_m^c \tag{2}$$

Similarly, we can obtain $\tilde{U_a^c}$
Thus, from SA unit, we get Motion-attentive feature $\tilde{U_m}$ & Appearance-attentive feature $\tilde{U_a}$ as output.

4.1.2 Attention Transition

In this unit, we wanted to transfer motion-attentive features $\tilde{U_m}$ by seeking the affinity between $\tilde{U_m}$ & $\tilde{U_a}$ in a non-local manner using the following multi-modal bilinear model:

$$S = \tilde{U_m^T} W \tilde{U_a} \tag{3}$$

The affinity matrix S can effectively capture pairwise relationships between the two feature spaces $\tilde{U_m}$ & $\tilde{U_a}$.
However, it also introduces a huge number of parameters, which increases the computational cost and creates the risk of over-fitting. To overcome this problem, W is approximately factorized into two low-rank matrices P and Q in the following manner-

$$W = PQ^T; P, Q \in R^{C \times \frac{C}{d}}$$

where d ($d > 1$) is a reduction ratio. Then, Eq. 3 can be rewritten as:

$$S = \tilde{U_m^T} P Q^T \tilde{U_a} = (P^T \tilde{U_m})^T (Q^T \tilde{U_a}) \tag{4}$$

This operation is equal to applying channel-wise feature transformations to $\tilde{U_m}$ & $\tilde{U_a}$ before computing the similarity. This not only significantly reduces the number of parameters by 2/d times, but also generates a compact channel-wise feature representation for each modal. Then, we normalize S row-wise to derive an attention map S^r conditioned on motion features and achieve enhanced appearance features U_a
Motion conditioned attention:

$$S^r = softmax^r(S) \tag{5}$$

Attention-enhanced feature:

$$U_a = \tilde{U_a} S^r \tag{6}$$

4.1.3 Deep MAT

We know, Deep network structures have achieved great success due to their powerful representational ability. Therefore, the MAT module get extended into a deep structure consisting of L MAT layers cascaded in depth (denoted by $F_{MAT}^{(1)}, F_{MAT}^{(2)}, \cdots, F_{MAT}^{(L)}$).
Input & Output for $F_{MAT}^{(\ell)}$:

$$U_a^{(\ell)}, U_m^{(\ell)} = F_{MAT}^{(\ell)}(U_a^{(\ell-1)}, U_m^{(\ell-1)}) \tag{7}$$

where $U_a^{(\ell)}$ computed as in Eq. 6 and $U_m^{(\ell)} = \tilde{U_m}^{(\ell-1)}$ following Eq. 2.

$$U_a^{(0)}, U_m^{(0)} = V_a, V_m$$

Since the performance dropped by directly stacking the MAT modules, we stacked multiple MAT modules in a residual form in the following manner:

$$U_a^{(\ell)} = U_a^{(\ell-1)} + \tilde{U_a}^{(\ell-1)} S^r = U_a^{(\ell-1)} + (A_a^{(\ell-1)} \odot V_a^{(\ell-1)}) S^r \tag{8}$$

$$U_m^{(\ell)} = U_m^{(\ell-1)} + \tilde{U_m}^{(\ell-1)} = U_m^{(\ell-1)} + A_m^{(\ell-1)} \odot V_m^{(\ell-1)} \tag{9}$$

Through this procedure, we keep the significance of previous layer output to have a better result.

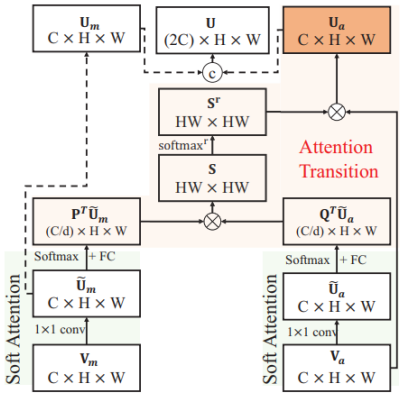

Figure 2: Computational graph of the MAT block. ⊗ and ⓒ indicate matrix multiplication and concatenation operations, respectively.

4.2 Scale-Sensitive Attention Module

SSA module is extended from a simplified CBAM (F_{CBAM}) by adding a global-level attention F_g CBAM(Convolutional Block Attention Module) consists of two sequential sub-modules:

- Channel attention
- Spatial attention

4.2.1 Channel attention

- Squeeze Operator (F_s):
  It Uses Global Average Pooling (GAP), which computes the average of each channel of the feature maps over the entire spatial region. The output of the GAP operation is a 1D vector s of length equal to the number of channels in the feature maps, which represents the global spatial information of the feature map U.
  $$s = F_s(U) \tag{10}$$

- Excitation Operator (F_e):
  Once the global spatial information is obtained using the squeeze operator, it is passed through two fully connected (FC) layers with a ReLU activation function to learn the channel-wise attention weights.
  $$fc(\frac{2C}{16}) \rightarrow ReLU \rightarrow fc(2C) \rightarrow sigmoid$$
  $$e = F_e(s) \tag{11}$$
  The output of the FC layers e is a channel descriptor that contains the attention weights for each channel, which is then multiplied with the original feature map U to compute the attended feature map Z_c.
  $$Z_c = e \star U \tag{12}$$

4.2.2 Spatial attention

It learns channel-wise attention weights for each spatial location of the input feature map U. The attention weights are computed using two fully connected (FC) layers with a sigmoid activation function, which takes as input a spatial descriptor that summarizes the information of each spatial location across all channels.

The spatial descriptor is obtained by applying a 1D convolutional filter with a kernel size of 7x7 to the input feature maps, which captures the contextual information of each spatial location.

$$conv(7 \times 7, 1) \rightarrow sigmoid$$

The output of the convolutional filter is then passed through a ReLU activation function and a global average pooling (GAP) operation, resulting in a matrix p that summarizes the spatial information of each location across all channels.

$$p = F_p(Z_c) \tag{13}$$

The attention weights p obtained from the FC layers are then multiplied with the original feature map Z_c, resulting in the attended feature maps.

$$Z_{CBAM} = p \odot Z_c \tag{14}$$

4.2.3 Global level attention

The global-level attention F_g shares a similar spirit to the channel attention layer in Eq. 10, in that it shares the same squeeze layer but modifies the excitation layer as

$$fc(\frac{2C}{16}) \rightarrow fc(1) \rightarrow sigmoid$$

It takes Z_c as input and gives a scale-selection factor g as output.
Finally we obtain scale-sensitive features Z as follows:

$$Z = g * Z_{CBAM} + U \tag{15}$$

Identity mapping has been used to avoid losing important information on the regions with attention values close to 0.

4.3 Boundary-Aware Refinement Module

In the decoder network, each BAR module, e.g. BAR_i, receives two inputs, i.e. Z_i from the corresponding SSA module and F_i from the previous BAR. To obtain a sharp mask output.
the BAR first performs object boundary estimation using an extra boundary detection module F_{bdry}, which compels the network to emphasize finer object details. The predicted boundary map is then combined with the two inputs to produce finer features for the next BAR module. It can be formulated as:

$$M_i^b = F_{bdry}(F_i) \Rightarrow F_{i-1} = F_{BAR_i}(Z_i, F_i, M_i^b) \tag{16}$$

BAR module gets benefited from two key factors

- ASPP(Atrous Spatial Pyramid Pooling)
- Introducing off-the-shelf HED model

F_{bdry}: It consists of a stack of convolutional layers and a sigmoid layer and takes F_i as input from the previous BAR Module and produces a boundary map M_i^b as output.

4.3.1 ASPP

The ASPP module operates on the output of the final convolutional layer in a neural network and generates a set of feature maps at different scales. It uses multiple parallel convolutional filters with different dilation (atrous) rates, which control the effective receptive field size of the filters.

It effectively captures features at different scales, without requiring additional computation or introducing spatial distortion. The ASPP module then combines the feature maps generated by the different filters using a concatenation operation, resulting in a multi-scale feature representation of the input image.

4.3.2 HED model

The HED model is based on a fully convolutional neural network (FCN) architecture, which allows it to process images of arbitrary sizes and produce output maps that are of the same size as the input image. The model consists of several convolutional and pooling layers, followed by a series of fully-connected convolutional layers.

The model simultaneously predicts edges at multiple scales, rather than predicting edges at a single scale and then up-sampling the output map to match the input size. This approach allows the model to capture edges of different scales and orientations, leading to more accurate and robust edge detection.

The procedure goes as follows:
The model predicts a boundary map $E \in [0,1]^{W \times H}$
Then it automatically mines hard negative pixels to support the training of F_{bdry}, where a pixel k is regarded as a hard negative pixel if it has a high edge probability (e.g. $E_k > 0.2$) and falls outside the dilated ground-truth region.

$$w_k = \begin{cases} 1 + E_k, \text{if k is a hard pixel} \\ 1, \text{otherwise} \end{cases}$$

This w_k is then used to weight the following boundary loss:

$$L_{bdry}(M^b, G^b) = -\sum_k w_k((1 - G_k^b)log(1 - M_k^b) + G_k^b log(M_k^b)) \tag{17}$$

where M^b and G^b are the boundary prediction and ground truth, respectively.

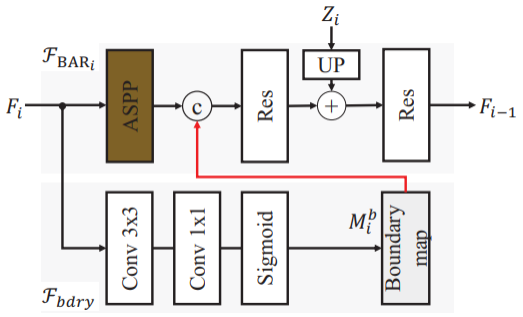

Figure 1: Computational graph of the BAR_i block. ⓒand $\oplus$ indicate concatenation and element-wise addition operations, respectively.

Experiments

Experimental Setup

We've carried out the comprehensive experiments on the popular dataset DAVIS-16. It's a dataset for video object segmentation consists of 50 high-quality video sequences(30 for training and 20 for validation) in total. Each frame contains pixel-accurate annotations for foreground objects.

Ablation Study

We've done the ablation analysis of our MATNet on DAVIS-16. Here, we've explored the effectiveness of MAT Block, SSA Block, HEM, ResNet-101. For quantitative evaluation, we've used two standard metrics namely region similarity $\mathcal{J}$ and boundary accuracy $\mathcal{F}$.
Region Similarity:- It's used to evaluate the performance of an image segmentation algorithm. It compares the segmentation result to a ground truth segmentation, which is a manually annotated segmentation that provide the true label of each pixel in the image.
Boundary Accuracy:- It evaluates how accurately an image segmentation algorithm can detect the boundaries between different objects or regions in an image. (It's typically computed as the percentage of boundary pixels that are correctly labelled as either boundary or non-boundary pixels.)

Table 1: Ablation study of the proposed network on DAVIS-16 measured by the Mean $\mathcal{J}$ and Mean $\mathcal{F}$.

Network Variant	Mean $\mathcal{J}\uparrow$	$\Delta\mathcal{J}$	Mean $\mathcal{F}\uparrow$	$\Delta\mathcal{F}$
MATNet *w/o* MAT	79.5	−2.9	77.3	−3.4
MATNet *w/o* SSA	80.7	−1.7	79.7	−1.0
MATNet *w/o* HEM	81.4	−1.0	78.4	−2.3
MATNet *w/* Res50	81.1	−1.3	79.3	−1.4
MATNet *w/* Res100	**82.4**	–	**80.7**	–

Table 2: Performance comparisons with different numbers of MAT blocks cascaded in each MAT layer on DAVIS-16.

Metric	$L=0$	$L=1$	$L=3$	$L=5$	$L=7$
Mean $\mathcal{J}\uparrow$	79.5	80.6	81.6	82.4	82.2
Mean $\mathcal{F}\uparrow$	77.3	80.3	80.7	80.7	80.6

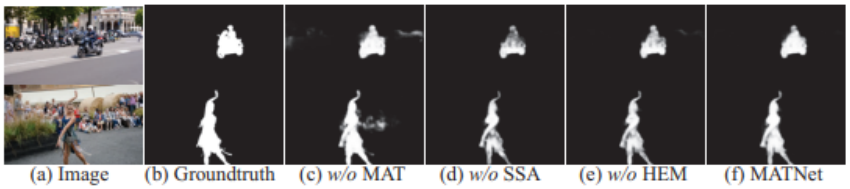

Figure 2: Qualitative results of ablation study.

MAT Block:- We first studied the effects of the MAT block by comparing our full model to one of the same architecture without MAT, denoted as MATNet *w/o* MAT. The encoder in this network is thus equivalent

to a standard two-stream model, where convolutional features from the two streams are concatenated at each residual stage for object representation. As shown in Table 1, this model encountered a huge performance degradation (2.9% in Mean $\mathcal{J}$ and 3.4% in Mean $\mathcal{F}$), which demonstrates the effectiveness of the MAT block.

Moreover, we've also evaluated the performance of MATNet with a different number of MAT blocks in each deep residual MAT layer. As shown in Table 2, the performance of the model gradually improved as L increased, reaching saturation at L = 5. Based on this analysis, we've used L = 5 as the default number of MAT blocks in MATNet.

SSA Block:- To measure the effectiveness of the SSA block, we designed another network variant, MATNet *w/o* SSA, by replacing the SSA block with a simple skip layer. As can be observed, its performance was 1.7% lower than our full model in terms of Mean $\mathcal{J}$, and 1.0% lower in Mean $\mathcal{F}$. The performance drop is mainly caused by the redundant spatio-temporal features from the encoder. Our SSA block aims to eliminate the redundancy by only focusing on the features that are beneficial to segmentation.

Effectiveness of HEM:- We've also studied the influence of using HEM or Hard Example Mining during training. HEM is expected to facilitate the learning of more accurate object boundaries, which should further boost the segmentation procedure. The results in Table 1 (see MATNet *w/o* HEM) indicate the importance of HEM. By directly controlling the loss function in Eq. 12, HEM helps to improve the contour accuracy by 2.3%.

Impact of Backbone:- To verify that the high performance of our network is not mainly due to the powerful backbone, we replaced ResNet-101 with ResNet-50 to construct another network, i.e. MATNet *w/o* Res50. We saw that the performance slightly degraded, but it still outperformed AGS in terms of both Mean $\mathcal{J}$ and Mean $\mathcal{F}$. This further confirms the effectiveness of MATNet.

Qualitative Comparison:- Figure 1 shows visual results of the above ablation studies on two sequences. We can see that all of the network variants produce worse results compared with MATNet. It should also be noted that the MAT block has the greatest impact on the performance.

Conclusion

- We presented a novel model, MATNet, for ZVOS, which introduces a new way of learning rich spatio-temporal object features. The MATNet model is a powerful convolutional neural network architecture that can be used for various computer vision tasks including spatio-temporal object segmentation. The advantage of using MATNet for segmentation is that it can automatically learn features from the input images without the need for manual feature engineering.

- One of the key strengths of MATNet is its ability to learn representations that are invariant to variations in lighting, contrast, and scale. This makes it well-suited for tasks such as object recognition and segmentation, where the appearance of objects can vary greatly depending on the lighting and other environmental factors.

- The MATblocks extract increasingly complex features within a two-stream interleaved encoder, which allow the transition of attentive motion features to enhance appearance learning at each convolution stage. The encoder features are further processed by a bridge network to produce a compact and scale-sensitive representation which provides several advantages in terms of feature extraction, computational efficiency, generalization, fine-grained segmentation.

- The output from the bridge network then fed into a decoder, which obtains accurate segmentation in top-down fashion. The decoder network in the MATNet model provides several advantages in terms of high-resolution output, feature refinement, flexibility, regularization, and reduced memory requirements, making it a powerful tool for image segmentation tasks in computer vision.
- The proposed interleaved encoder is a novel two-stream framework for spatio-temporal representation learning in videos. It can be easily extended to other video analysis tasks, such as action recognition and video classification.
- Extensive experimental results indicate that MATNet achieves favorable performance against current state-of-the-art methods. There are many video object segmentation algorithm like COSNet (CO-attention Siamese Network), AGNN (Attentive Graph Neural Network), LSMO (Layered Sequential Models), LVO (Localizing Video Objects) but it is observed in study that MATNet's performance is better than this methods.