Video Salient Object Detection Using Multi-scale Self-attention

A Thesis Submitted to the Department of Computer Science and Communications Engineering,
the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: July 24th, 2023

Jiahao Liu
(5121FG56-6)

Advisor: Prof. Hiroshi Watanabe
Research guidance: Research on Audiovisual Information Processing

Acknowledgments

Foremost, I wish to convey my profound appreciation to Professor Hiroshi Watanabe for the support and mentorship he has provided, not solely in my research endeavors but also in shaping the trajectory of my life and career. His guidance has been instrumental in both my academic pursuits and personal development. Additionally, thanks to the device and good environment the professor provided, I can successfully conduct my experiments and finish my dissertation.

Furthermore, I extend my gratitude to all the esteemed members of Watanabe Lab for their unwavering assistance in both my academic pursuits and daily life. Especially I will express my gratitude to Liu Haoyuan, because he helped me a lot when I was not in Japan.

Thirdly, I want to express my gratitude to everyone who gives me advice on my research, especially for the members of IP seminar.

Finally, I would like to extend my heartfelt appreciation to my parents for their unwavering support and encouragement throughout my academic journey. I often feel so lucky for myself to have such good parents. I wish I could become better in the future, living up to my parents' expectations and cultivation.

Abstract

Videos as one of the most engaging mediums strike a deep connection with humans. Video salient object detection (VSOD) aims at discovering and locating the most visually distinctive parts of a video clip. Compared with still-image based Salient Object Detection (SOD) tasks, VSOD does not only suffer from processing a huge amount of data but also is directly affected by temporal dynamics. For instance, both moving foreground and background objects in a video clip make some existing image-based methods less effective.

How to effectively model both spatial information and temporal dynamics is crucial to this task. Recently, there are some works using self-attention mechanism to capture the spatiotemporal information due to its ability of modeling long-range dependencies of patch tokens. However, these models designate similar receptive fields of the spatiotemporal feature maps, which limits the ability of the models in handling the frames with multiple salient objects of different scales. To address this issue, we propose a Multi-Scale Self-Attention (MSSA) operation to better model the spatiotemporal features of salient objects with different scales. The experimental results demonstrate that our method achieves better performance on challenge datasets by using MSSA operation.

Keywords: Computer Vision, Video Processing, Video Salient Object Detection

List of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Research Background

Videos as one of the most common mediums strike an important role in our daily life. Video Salient Object Detection (VSOD) constitutes a fundamental undertaking within the realm of video processing, with the primary objective of identifying and segmenting the most visually distinctive regions within a given video clip. This task originates from the investigation of human visual attention mechanisms, as humans exhibit the ability to swiftly focus on the most informative aspects of visual scenes. The practical implications of Video Salient Object Detection (VSOD) span a broad spectrum of real-world applications, including video captioning [1], video compression [2], visual tracking [3], and autonomous driving [4].

However, compared with the still image-based tasks, video SOD has an important difference, that is, the motion information between the adjacent frames. When events occurring in the real world are temporally condensed into a few seconds of video footage, the pixel values across different frames may exhibit temporal inconsistencies on the time dimension. How to effectively take such dynamic information into consideration makes VSOD very challenging.

1.2. Existing Methods

In recent years, VSOD task has witnessed a notable surge in the application of learning-based techniques, prominently featuring convolutional neural network (CNN) based methodologies as the prevailing paradigm. Some previous graph-based methods [5] intend to utilize the motion information based on spatio-temporal coherence, but due to the limitation of handcrafted features such methods fail to accurately model the spatial and temporal information in complicated scenes. Some fully convolutional network based methods [6], [7] utilize previous frame or predicted saliency map

concatenated with the current frame as the input to the network to model the temporal coherence. For the lack of employing explicit motion estimation, such CNN based methods are more easily affected by temporal dynamics from a video clip.

To better utilize both spatial and temporal information for detection, some works [8], [9] employ 3D convolutions as a means to capture and incorporate temporal information into their models. Compared to the 2D convolution, an additional dimension is added to the 3D convolution kernel which also brings additional computation cost.

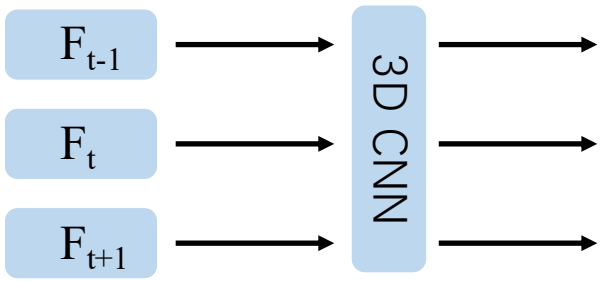


Fig. 1 Illustration for 3D CNN based methods. F_{t-1} denotes the last frame, F_{t+1} denotes the next frame.

In recent, more and more works [10], [11] intend to fuse optical flow information to help the network learn more about representations of object motion. Incorporating additional prior information, such as optical flow or depth cues, may deviate the network from being a truly end-to-end architecture.

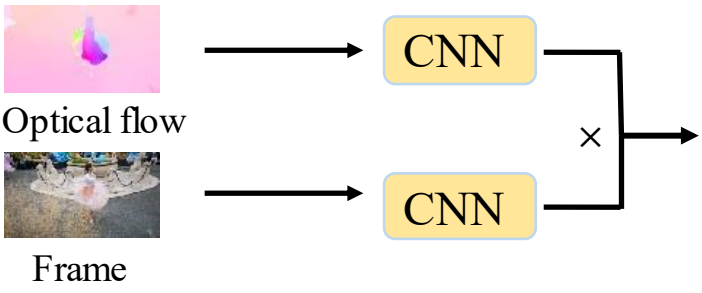


Fig. 2 Illustration for two-stream network based methods.

Similar to Recurrent Neural Network (RNN), the Long Short-term Memory (LSTM) network could consider the information of the last step. Such method uses a CNN

module to learn the static feature of the images, and fed to ConvLSTM layer then give the final saliency prediction. Therefore, it can be used to model the temporal coherence. Fan et al. [12] introduced a foundational model incorporating ConvLSTM and introduced a comprehensively annotated dataset for VSOD.

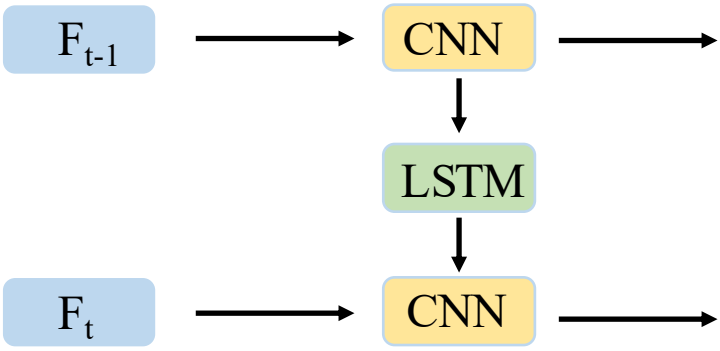


Fig. 3 Illustration for ConvLSTM based methods. F_{t-1} and F_t denotes the last frame and current frame.

In more recent studies, several approaches have been developed with the aim of leveraging non-local attention-based mechanisms to effectively explore pairwise relations among adjacent frames. Specifically, Fan et al. [12] and Gu et al. [13] introduced saliency-aware attention modules and Constrained Self-Attention (CSA) operations, respectively, to enhance the model's ability to capture motion cues in the video data. And Su et al [14] first introduce Transformer block to capture the long-range dependencies through self-attention mechanism. But the original vision transformer [15] blocks rely on static receptive fields of the tokens and is incapable of capturing features at different scales.

Fig. 4 Illustration of Gu el 's method and Illustration of Su el 's method.

1.3 Research Objectives

Taking inspiration from the above observations, this paper presents a multi-scale self-attention guided VSOD network, which could not only capture the long-range

dependencies between adjacent video frames but also improve the model's ability of handling images with multiple objects in different scales. The proposed method utilizes multi-scaled self-attention within different Vision Transformer [15] blocks. Specifically, the multi-scaled self-attention intends to merge different receptive field sizes into tokens, thereby representing larger object features, while concurrently retaining certain tokens to preserve fine-grained features. In short, the main contributions of our paper are summarized as follows:

- We propose a network which could consider both spatial and temporal information of a video clip.
- Incorporating diverse Transformer blocks, our approach is designed to effectively capture long-range dependencies among adjacent frames. And we use different scaled self-attention operations to merge multi-scaled features within a self-attention layer.
- We evaluate our method on video salient object detection benchmark. Experimental results show that the proposed multi-scaled self-attention operations makes the network performance better than that with original Transformer blocks in ViT.

1.4 Outline of Thesis

The structure of this thesis is as follows:

Chapter 1: We give the introduction of Video Salient Object Detection task and the background of this research, which contains the challenging part of this task. We also introduce the existing methods in VSOD task, which contains 3D convolution based method, optical-flow based method, ConvLstm based method and attention based method.

Chapter 2: We produce an introduction to the related technologies in our work. In 2.1, we give a brief introduction to Convolutional Neural Network, including classic

VGG backbone which is used in our model. In 2.2, we introduce self-attention mechanism in vision field. Due to its ability of capturing long-range dependencies of patch tokens, we introduce it to our work for modelling context information between adjacent video frames. In 2.3, we mainly talk about the improvements of fixed-scale self-attention mechanisms. In this part, we mainly introduce two works, swin-Transformer [18] and PVT (Pyramid Vision Transformer) [19]. One splits the feature maps into regions and perform self-attention within them locally, the other utilizes spatial reduction to merge tokens of key and value.

Chapter 3: In this chapter, we mainly talk about the proposed architecture for this task. In 3.1, we introduce the detailed design for our model. In 3.2, we mainly talk about the multi-scale self-attention operation in this proposed architecture. In 3.3, we introduce the training scheme of our work, which includes the loss functions and the training details.

Chapter 4: The experimental parts are shown in chapter 4. Firstly we introduce the four benchmark datasets in this task which includes $DAVIS_{16}$ [22], FBMS [23] SegTrack-V2 [24] and ViSal [25]. Secondly, we give an introduction to the evaluation metrics which help to evaluate the quality of the proposed model. Then we give the experimental results of our work and comparisons to other models.

Chapter 5: In this chapter, we give the conclusion to our work and the predicted improvements in the future work.

Chapter 2 Related Work

2.1 Introduction of Convolution Neural Network

2.1.1 Construction of Convolutional Neural Network

Input layer: The input to the entire network typically comprises a three-dimensional matrix, representing the pixel information of an image. The dimensions of this matrix correspond to the image size, where the length and width indicate the spatial dimensions, while the depth denotes the number of color channels in the image. In the case of black-and-white images, the depth is 1, whereas for images in RGB color mode, the depth is 3.

Convolution layer: The convolution layer is a critical component of CNNs. Unlike the fully connected layer, each node in the convolution layer receives input from a localized region in the neural network's preceding layer, typically represented by small blocks of size 3x3 or 5x5. As a result of the convolution operation, the node matrix undergoes an increase in depth, contributing to the extraction of deeper features within the network.

Pooling layer: The pooling layer in a CNN is responsible for spatially downsampling the input 3D matrix while maintaining its depth. This downsampling process can be likened to transforming a high-resolution image into a lower-resolution representation. Consequently, the pooling layer facilitates a reduction in the number of nodes within the final fully connected layer, effectively diminishing the overall parameter count of the neural network. It is important to note that the pooling layer itself lacks trainable parameters.

Fully connected layer: Following several iterations of convolution and pooling operations, the final classification outcome is typically obtained through one to two fully connected layers at the end of the CNN architecture. These convolution and

pooling operations play a vital role in abstracting information from the input image, transforming it into higher-level features with increased information content. Consequently, we can perceive convolution and pooling as automated image extraction processes. Subsequent to feature extraction, the task of classification is accomplished using fully connected layers. In the case of multi-class classification, the softmax activation function is often employed as the final layer, enabling the derivation of the probability distribution for each sample belonging to various categories.

Next, I will introduce the two most important parts in CNN, Convolution layers and Pooling layers.

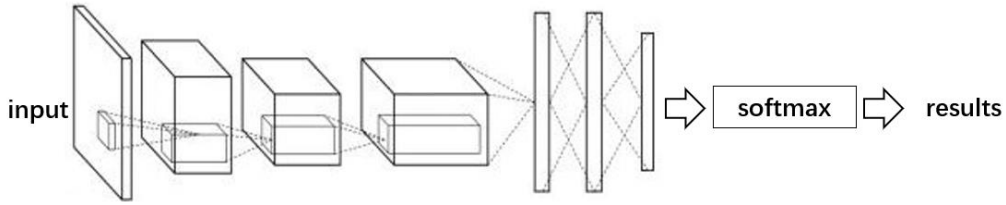


Fig. 5 Convolutional neural network.

Convolution layer: The focal element within the convolutional neural network structure is the filter, depicted as a yellow and orange 3x3x3 matrix in Fig.6. The filter plays a crucial role in transforming a sub-node matrix from the current layer of the neural network into a unit node matrix within the subsequent layer. The unit node matrix is characterized by a length and width of 1, while its depth remains unrestricted. The convolution operation demands careful consideration of several parameters, including the number of filters, the size of the filter, the convolution stride, and the padding size.

The commonly used filter sizes are 3 × 3 or 5 × 5, the first two dimensions in the yellow and orange matrices in Fig.6, which are artificially set. The depth of the node matrix of the filter, that is, the last dimension of the yellow and orange matrices in Fig.6 (last dimension of the size of the filter), is determined by the depth of the node matrix in the current layer of the neural network (the depth of the node matrix of the RGB image is 3). The depth of the convolution layer output matrix (also known as the depth

of the filter) is determined by the number of filters in the convolution layer. This parameter is also artificially set, and generally increases with the convolution operation.

Pooling layer: First of all, I would like to introduce the concept of pooling. Owing to the inherent spatial correlation in an image, neighboring pixels often exhibit similar values, resulting in the convolution layer's output pixels in close proximity also possessing comparable values. Consequently, a considerable portion of the information within the output of the convolution layer tends to be redundant.

The pooling layer efficiently reduces the spatial dimensions of the matrix by downsampling the input data (mainly reduce the length and width of the matrix, generally do not reduce the depth of the matrix), so as to reduce the parameters in the final full connection layer.

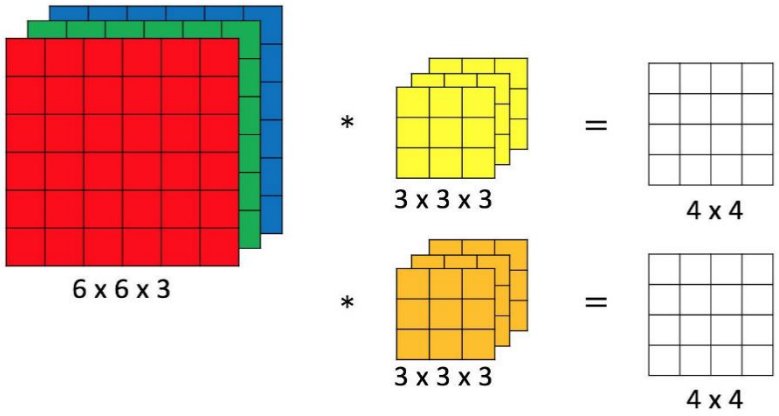


Fig. 6 Convolution operation.

2.1.2 Introduction of VGG-16

VGG [16] is a deep neural network architecture proposed by the Visual Geometry Group at Oxford University in 2014. The VGG architecture utilizes continuous small convolutional kernels (3x3) and pooling layers to construct a deep neural network. The depth of the deep neural network can reach 16 or 19 layers. Therefore, VGG-16 and VGG-19 are the two most commonly used network structures. The VGG-16 and VGG-19 are very similar, both composed of VGG blocks. The VGG blocks are composed of

convolutional layers and pooling layers. The difference is that VGG-19 has 3 additional convolutional layers and 1 dense layer, which improves the model's ability to fit more complex datasets.

The VGG blocks utilize small 3x3 convolution kernel to increase the number of channels while the pooling layers cut down height and width of feature maps. There are several advantages to using 3x3 small convolutional kernel: (1) Reduce model parameters. A 3x3 convolutional kernel has 9 weight parameters, while a 5x5 convolutional kernel requires 25 weight parameters. Therefore, using a 3x3 convolutional kernel can significantly reduce the number of network parameters, thereby reducing the risk of overfitting. (2) Improve the nonlinear ability of the model. Multiple 3x3 convolution kernels connected in series can form a convolution kernel with a larger Receptive field, and this combination has stronger nonlinear ability. In VGG, using 3x3 convolutional kernels multiple times is equivalent to using larger convolutional kernels, which can improve the network's feature extraction ability. (3) Reduce computational complexity. VGG network uses multiple 3x3 convolution kernels, which can increase the Receptive field without increasing the amount of computation, but improve the network performance.

The constructure of VGG-16.

Input: 224x224 RGB image. Block1: Contains 2 convolutional layers [64x3x3]. Block2: Contains 2 convolutional layers [128x3x3]. Block3: Contains 3 convolutional layers [256x3x3]. Block4: Contains 3 convolutional layers [512x3x3]. Block5: Contains 3 convolutional layers [512x3x3]. The shape of input tensors is 224x224x3, after Block1, the shape of the tensors will be 112x112x64 (height*width*channels). The convolution layers in the VGG block do not change the spatial size of the feature maps, but the pooling operations will cut down the spatial sizes of them. The function of convolution layers is to extract different features from different levels. That’s why the number of channels will be gradually increase. Different types of features are stored

in different channels, and the increase in the number of channels helps the backbone network better extract features from input images. After five max-pooling operations, the shape of the tensors will be 7x7x512 (height*width*channels). The number of channels of 512 will enable the backbone network to learn features at different levels. Then tensors will be flattened and after several dense layers, the number of channels will be cut down to 4096 and then to 1000 (ImageNet challenge contains 1000 categories).

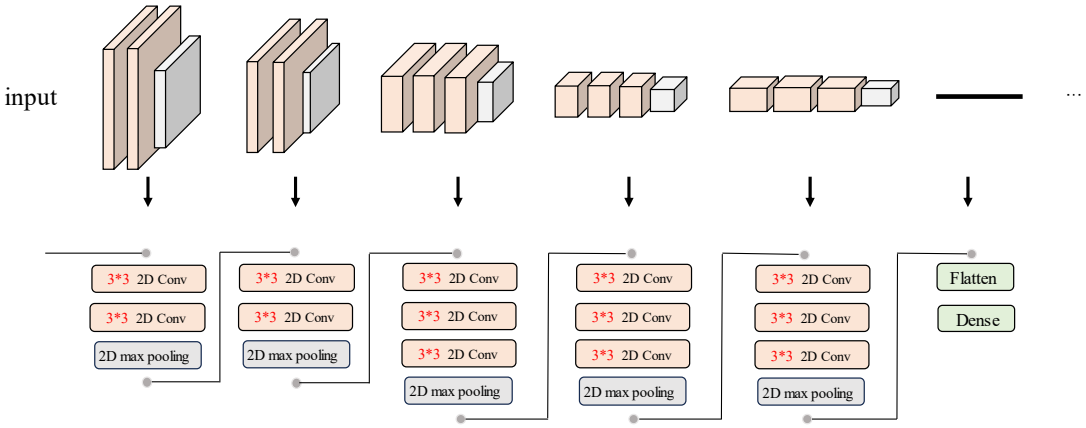


Fig. 7. Introduction to the VGG blocks in VGG-16 architecture.

2.2 Self-Attention Mechanism in Vision Field

Self-attention mechanism is proposed to capture long-range dependencies in machine translation. As a sequence model, it works by measuring pair-wise relationships of all patch tokens. Recently, Vision Transformer (ViT) [15] have shown that such self-attention mechanism also has superior performance in visual tasks. In Vision Transformer, an image is viewed as fixed-size patch tokens. Thus, we divide the input image into patch tokens and linearly embed each of them. Then the feature sequence will be fed into self-attention layer to do multi-head self-attention operation. After self-attention layer and LayerNorm, the feature sequence will be fed into FeedForward layer to do Multilayer Perceptron (MLP) operation. Because ViT is designed and first applied to classification tasks, an there is an additional token in the

input sequence, and the corresponding output of this token is result of the predicted category. The architecture of Vision Transformer (ViT) is shown below.

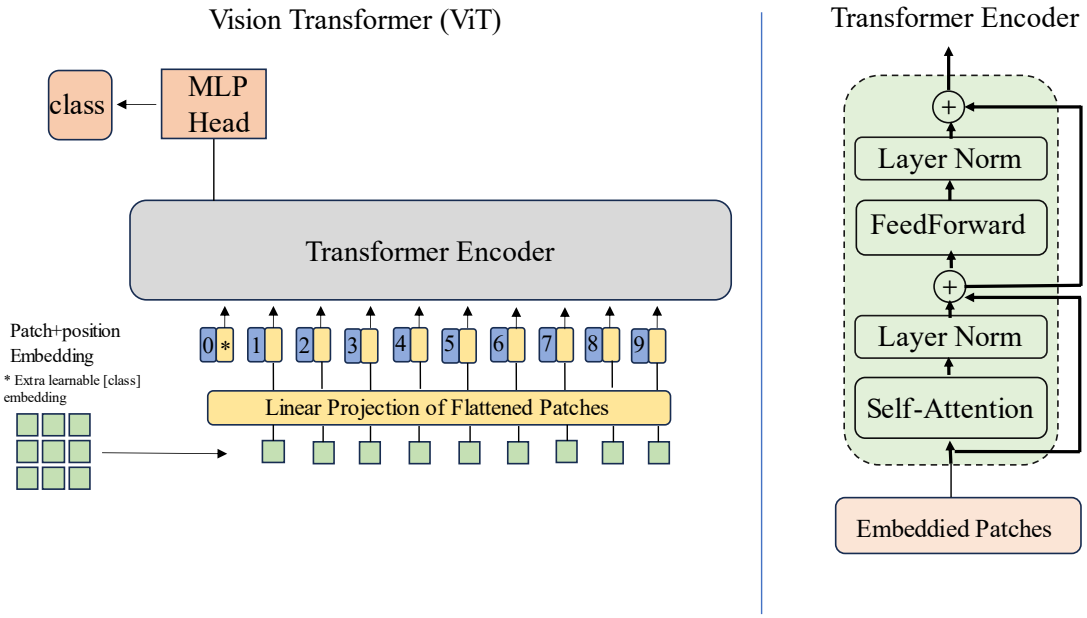


Fig. 8. Introduction to Vision Transformer (ViT) and Transformer block.

ViT only utilizes the encoder of Transformer architecture and the ViT block contains the following parts.

Patch Embedding. If the height and width of the input image are 224x224, dividing the image into size 16x16 will generate 196 patches per image. The dimension of each patch is 16x16x3, after linear projection (the dimension of linear projection is 768 x n, n=768), the dimension of the input image will be 196x768. The number of patches is 196 and the dimension of each token is 768. However, there is an additional cls token which is for classification. Therefore, the final dimension is 197x768. Up to now, the vision task is changed to a seq2seq problem.

Positional encoding. The positional encoding in ViT could be considered as a table which contains N rows (N is the length of input pathch sequence). Each row of this table represents a tensor, whose dimension is same as the input after patch embedding (768). The operation of positional encoding is sum operation, not the concat operation. Thus, after adding additional positional encoding, the dimension is still 197x768.

Layer-Normalization and Multi-head attention. The dimension of tensors after Layer-Normalization is still 197x768. When doing multi-head self-attention to the tensors, the input will be projected to q, k, v. If the parameter of the heads is n, the dimension of q, k, v will be 197x(768/n). The tensors will do self-attention operations under each head respectively, and finally be concatenated to the original shape of 197x768. Then after the Layer-Normalization, the shape of the output is still 197x768.

MLP operation. In each ViT block, there are two linear layers which firstly increase the dimension to 197x3072 and then reduce it to 197x768. The aim of MLP operation is to increase the representation ability of backbone network by introducing more non-linear transformations.

After each block, the shape of the tensors will be unchanged. When the backbone network goes deeper, finally the cls token Z_l will give the predicted result for the network.

2.2.1 Multi-head Self-Attention Mechanism

Transformer is proposed in the paper Attention is All You Need [17]. It was initially utilized in Natural Language Processing (NLP) task. Next, I will briefly introduce self-attention mechanism in Transformer.

As shown in Fig.9, the input at the bottom of the figure (X^1, X^2, X^3, X^4) represents the input sequence data. For example, tensors X^1 - X^4 could represent a sentence which contains four words, that is 'I like playing basketball'. Then after embedding, they will become a^1-a^4. Next there will be three matrices multiplied by each of them to obtain q^i, k^i, v^i, $i \in (1, 2, \dots, T)$:

$$qi = W^Q a_i, \tag{1}$$

$$ki = W^k a_i, \tag{2}$$

$$vi = W^V a_i, \tag{3}$$

The following Fig. 9 shows how the output b1 corresponding to the input X1 is obtained. Firstly, calculate the vector dot product of q1 with different k values, and calculate the $\hat{a}1,i$,which is a number between 0 and 1 through Softmax operation. Then multiply the $\hat{a}1,1$, $\hat{a}1,2$, $\hat{a}1,3$, and $\hat{a}1,4$ obtained in the previous step with the v1, v2, v3, and v4, and then sum them to obtain the output b1. Similarly, the calculation process for b2, b3, and b4 is basically the same as b1.

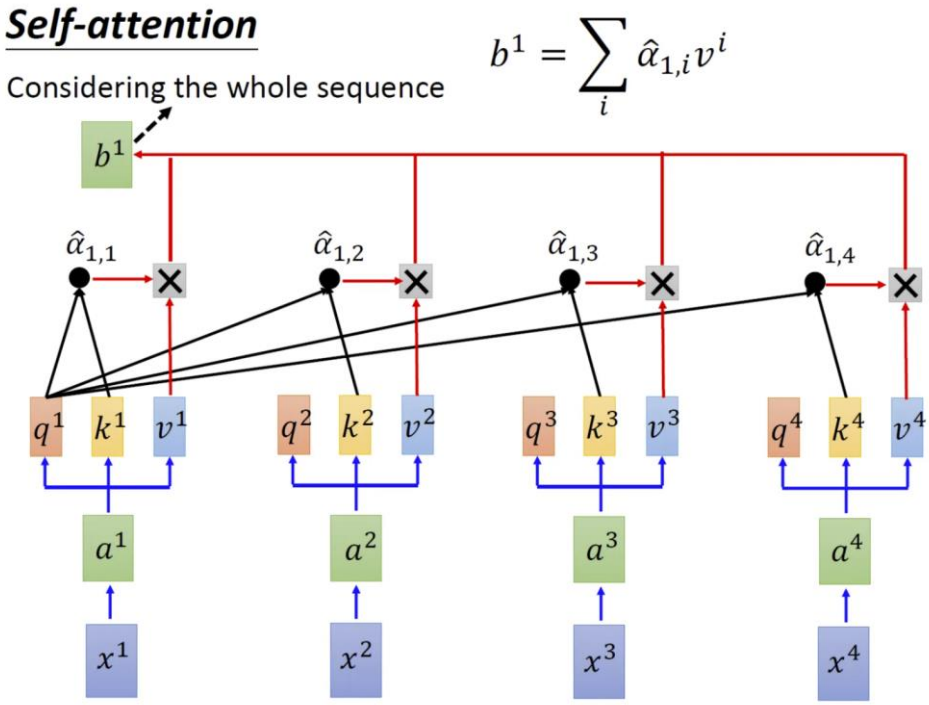


Fig. 9. The calculation process of self-attention operation.

For the input sequence, x1, x2, x3, x4…, unlike the processing of RNN/LSTM, the Self-attention mechanism can perform parallel calculations on x1, x2, x3, x4. This greatly improves the model's speed for feature extraction.

Why the Transformer needs multi-head self-attention?

When encoding information about the current location, the model excessively focuses its attention on its own location. The usage of multi-head self-attention mechanism can provide the output of the attention layer with encoded representation

information belonging to different subspaces, thereby enhancing the representation ability of the model. The multi-head self-attention operation is illustrated below.

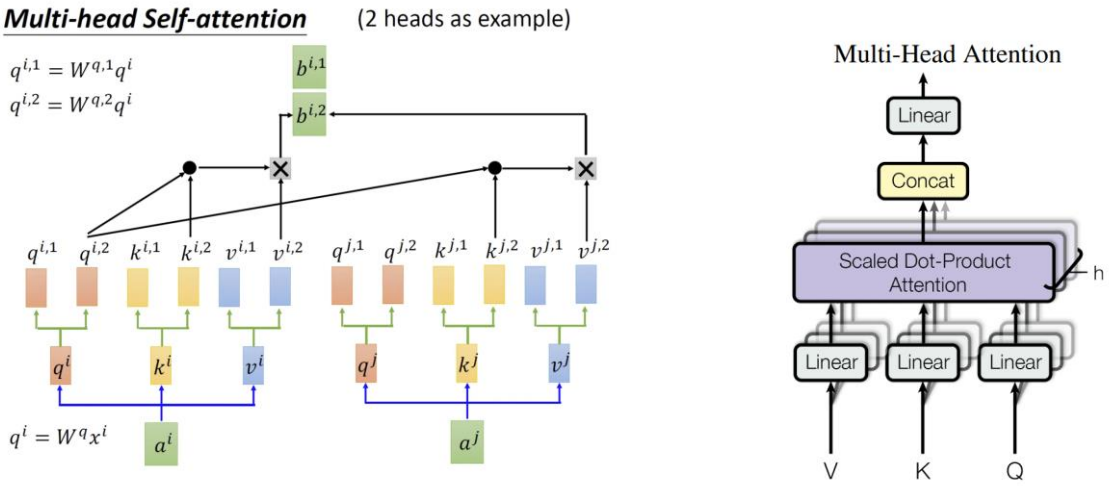


Fig. 10. The illustration of multi-head self-attention.

2.2.2 Introduction of Layer Normalization

Layer Normalization is a normalization function used in deep neural networks. It can normalize the output of each neuron in the network, so that the output of each layer in the network has similar distribution.

Different from Batch Normalization, Layer Normalization is not performed on the input of each mini batch. Instead it is performed on the output of each neuron. Specifically, for the output of each layer of network, we can perform Layer Normalization on the output of each neuron:

$$LayerNorm(xi) = \frac{xi - \mu}{\sqrt{\sigma^2 + \epsilon}} , \tag{4}$$

where μ represent the mean and σ is standard deviation of the output of the neuron in the mini batch, respectively. ϵ is a small constant number used to prevent the denominator from being zero. Through Layer Normalization, the network can converge faster and also improve its generalization ability. In tasks such as Natural Language Processing, Layer Normalization could achieve better performance. Since its

normalization for each neuron does not depend on the size of the mini batch, it can also be used in smaller mini batches.

2.3 Improvement of Fixed-scale Multi-head Self-attention

In recent years, Transformer based models have not only been fully applied in the field of Natural Language Processing (NLP), but also achieved astonishing performance in the field of computer vision, i.e., object detection and semantic segmentation… At the same time, more and more new improvements have been made to address the shortcomings of the original Vision Transformer model.

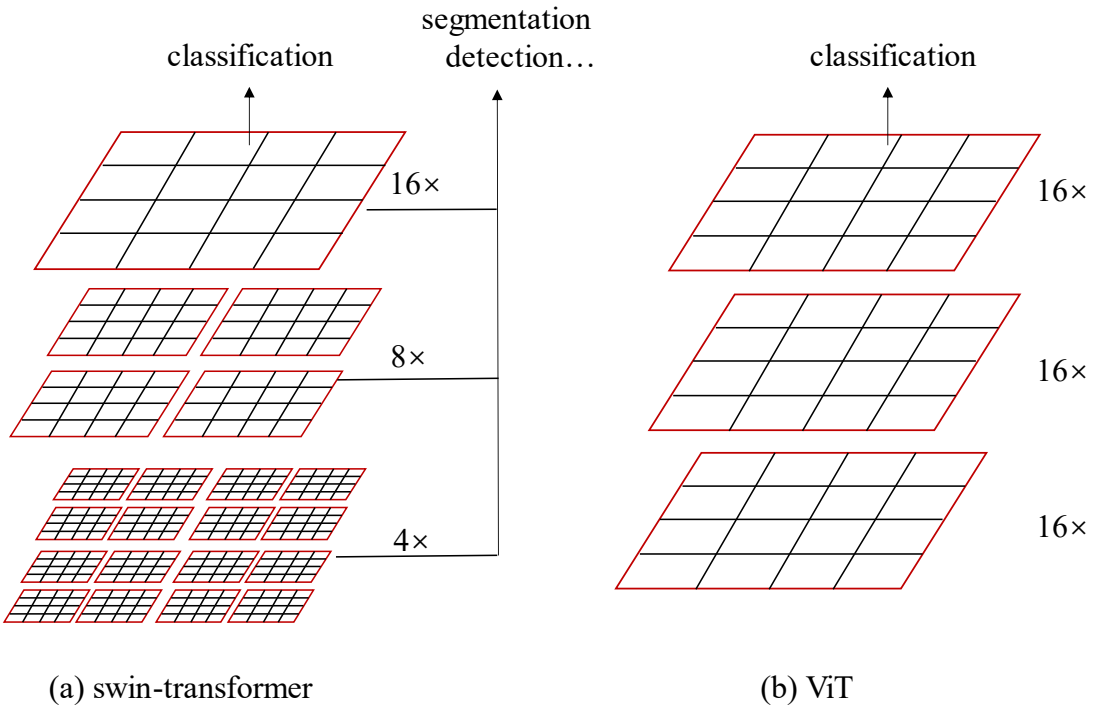
2.3.1 Swin-Transformer

Swin-Transformer [18] is published by Microsoft on ICCV in 2021. Once published, this paper has already been the top on multiple visual tasks. Fig.11 shows the difference between Swin-Transformer and original ViT. From the figure, we could summarize at least two differences:

Firstly, Swin-Transformer adopts a hierarchical structure, which facilitates the handling of varying feature map sizes. For example, there are downsampling of 4 times, 8 times, and 16 times for the image. This type of backbone architecture proves to be well-suited for tasks like object detection and semantic segmentation. In the original Vision Transformer, the downsampling rate was fixed at 16 times from the outset, with subsequent feature maps preserving this unaltered downsampling rate. We could see this difference in the Fig.11.

Secondly, Swin-Transformer incorporates the concept of Windows Multi-Head Self-Attention (W-MSA). In this approach, for instance, in the 4x downsampling and 8x downsampling scenarios depicted in the subsequent figure, the feature map is partitioned into multiple non-intersecting regions (referred to as Windows), within which the Multi-Head Self-Attention mechanism operates independently. By utilizing this localized attention strategy, as opposed to performing Multi-Head Self-Attention

on the entire (Global) feature map in Vision Transformer, the computational complexity is effectively reduced. Although this approach reduces computational complexity, it also isolates feature fusion between different windows. Therefore, the author proposed the concept of Shifted-Windows Multi-Head Self-Attention (SW-MSA).



(a) swin-transformer (b) ViT

Fig. 11. Illustration of Swin-Transformer and ViT.

2.3.2 Pyramid Vision Transformer

The motivation proposed in this paper [19] is to enhance the performance of original Vision Transformer. The illustration of comparisons with CNN, ViT and PVT is shown in Fig. 12. The proposal of ViT is creative, but there is also much space for improvement. For example, it is difficult to directly apply ViT backbone to some downstream tasks. There are two main reasons. On the one hand, it has network design issues, and on the other hand, it has high memory usage on GPU.

(1) Network design issue. We know that ViT divides an input image into patch tokens, thus a 16-stride or 32-stride feature map will be obtained. The size of the feature

map in the network remains unchanged, and overall it is a columnar structure as shown

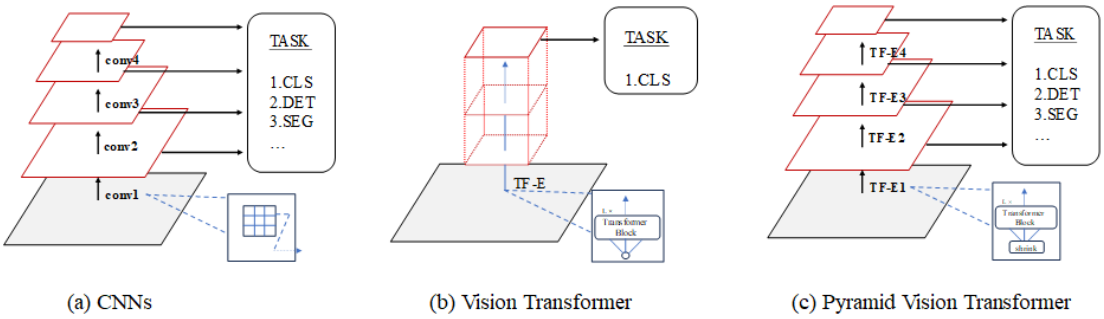


Fig. 12. Comparisons with CNNs, Vision Transformer, and Pyramid Vision Transformer.

in Fig.12 (b). From the figure, we can see that CNN is a pyramid structure, and the deeper the network, the smaller the feature map and the more channels there are. However, ViT is a columnar structure that maintains a single size of the feature maps throughout the network.

The question is whether the choice of 16-stride or 32-stride is reasonable for different tasks? The answer is clearly no. The problem of the input ViT feature tokens: The resolution of ViT feature maps depends on the setting of the input patch-token size, however, the resolution requirements vary for different images or tasks. The more complicated the image or task, the higher the demand for resolution. An apple with a resolution of 2x2 is sufficient, a seagull with a resolution of 3x3, and a palace with a resolution of 4x4.

(2) Memory usage issues. Another issue with ViT is its high GPU memory usage, which makes it difficult to process high-resolution input image. For Classification task, a resolution of 224x224 may be sufficient for input images, but for tasks such as Semantic Segmentation and Object Detection, we often need higher resolution input images. The high GPU memory usage of ViT is largely due to the need for Attention computation. The higher the resolution of the input image, the higher the memory usage for Attention computation, so it is difficult to process high-resolution images with ViT.

Chapter 3 Proposed Method

In this chapter, we mainly introduce the proposed method and its overall architecture, and the main proposal Multi-Scaled Self-Attention(MSSA) branch for this task. We give the detailed process of computation of Multi-Scaled Self-Attention operation. Then in 3.3, we give the introduction of the network training. In this part, we introduce the loss function and detailed training schemes.

As mentioned in Chapter 1 Introduction, in order to capture both spatial information and temporal dynamics, which is the challenge of this task, we decided to utilize VGG to extract spatial features and utilize self-attention mechanism to capture long-range dependencies. That's because after down sampling by CNN backbone, the length of patch tokens will be much shorter than before. As known to us, the computational cost for ViT is based on the input size of images. Thus, for the lightweight task VSOD, such operation is necessary and could cause a reduction of calculation.

Since the input is adjacent N frames of video clips, after patch embedding, the feature tokens which contains N frames information, will be fed into self-attention layer. VSOD task aims at locating and segmenting the common foreground objects of a video clip. Therefore, such self-attention operation will capture long-range dependencies from these consecutive frames. And makes the model pay more attention to the common foregrounds of the adjacent frames. The illustration of this process is shown in Fig.13. That's why we use self-attention mechanism to replace the sequence model. Another reason is that the traditional sequence model like LSTM is serial, which means that it needs to know the step of last time and then calculate the next step. Such operation will slow down the computational speed of the model. However, self-attention mechanism could do matrix multiplication in parallel, which could accelerate on GPU.

Finally, we concatenate the spatial feature maps with the enhanced temporal feature maps, and use a convolution layer to give the final predicted masks.

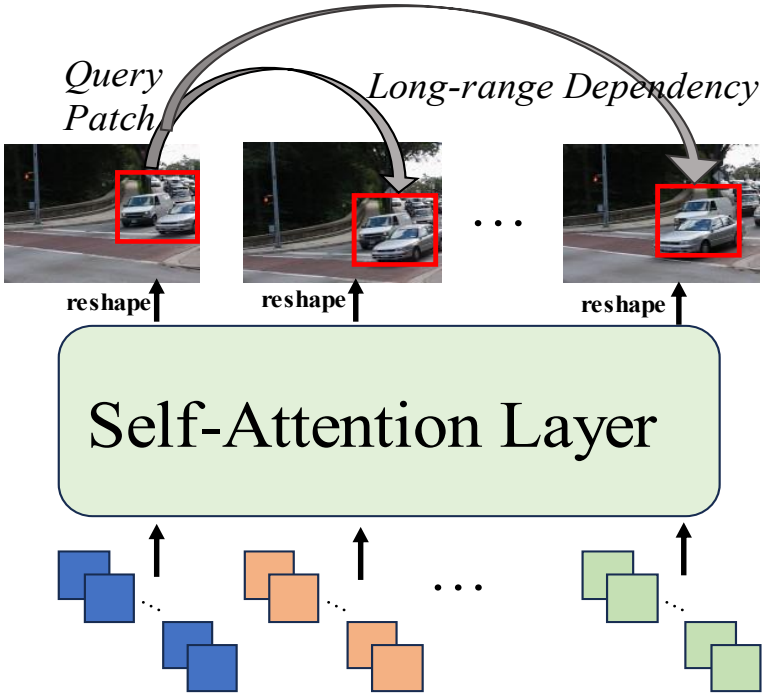


Fig. 13. Illustration of capturing long-range dependency using self-attention mechanism.

3.1 Network Architecture

The overview of our proposed method is shown in Fig.14. Given an input of N frames of a video clip, they are firstly fed into CNN backbone to extract multi-scale spatial features. The encoder of our network is built upon VGG16 which I introduced in the related work part. The VGG blocks utilize small 3*3 convolution kernel to increase the number of channels while the pooling layers cut down height and width of feature maps. Such operation makes the encoder learn more deep features while the increase of the calculation amount continues to slow down.

We encapsulate different self-attention schemes into different transformer blocks. For this part of network, we call it MSSA. MSSA is designed to capture spatial temporal information of adjacent frames. They are utilized in the top two layers in the Fig.14, because the computational cost of the self-attention mechanism depends on the length of feature sequence. To avoid expensive memory consumption and computational cost, we use it on the spatial reduced feature layers to produce the multi-scale self-attention

enhanced feature maps. For decoder, we reference the strategies of feature pyramid network [20]. The MSSA enhanced spatiotemporal feature maps are upsampled using bilinear interpolation to match the low-level feature size. Then we concatenate the static spatial features with the spatiotemporal features. Finally, a convolution layer is utilized to give the predicted mask.

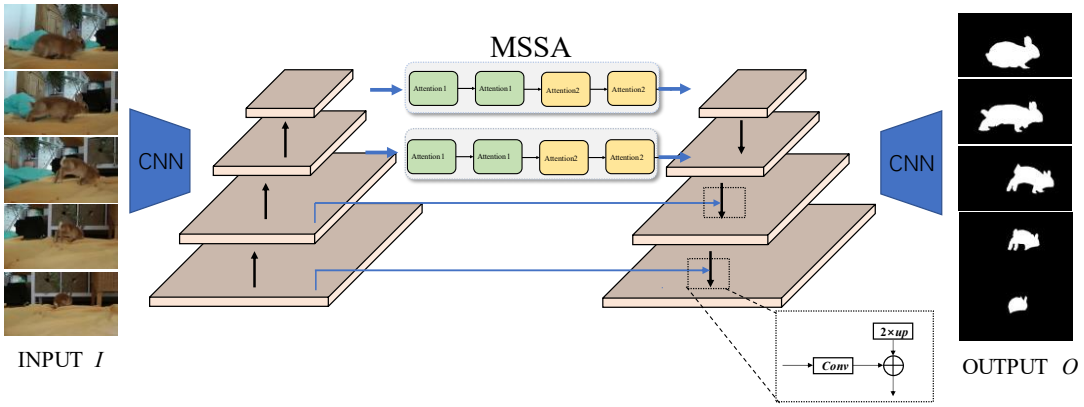


Fig. 14. Overview of our proposed method.

The video clip, consisting of N frames (e.g., N = 5), is initially processed through a CNN backbone to extract static spatial features. Subsequently, the designed Multi-Scaled Self-Attention (MSSA) module is applied to the top two feature layers. This module is specifically designed to capture long-range dependencies among the frames, thereby enhancing the modeling of spatio-temporal correlations. The output of this module yields enhanced spatiotemporal feature maps. To achieve consistency between spatio-temporal features and static spatial feature size, bilinear interpolation is employed for upsampling. Subsequently, the low-level feature and spatio-temporal feature maps are concatenated to yield the final predicted mask.

3.2 Multi-scale Self-Attention

Motivation of proposing multi-scale self-attention for this task.

In VSOD task, because the form of data is a continuous frame of video, the distance between the objects and the camera in the dataset determines their sizes in the frames. Thus, the size of the same saliency object in consecutive frames of the video is different,

as shown in Fig.15. The different sizes of the same salient object will bring difficulties to such process. Therefore, we proposed multi-scale self-attention in order to make the model fuse more different scaled feature information.

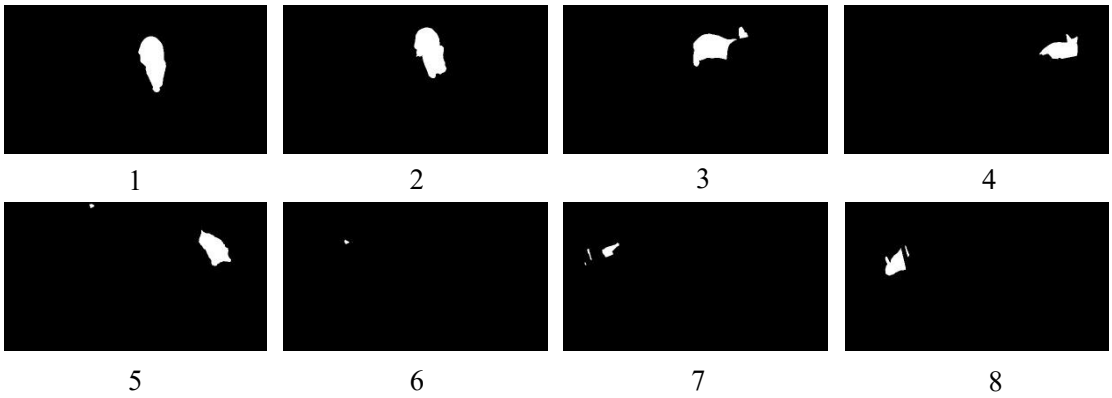


Fig. 15. Illustration of the same salient object with different sizes in consecutive frames.

In our work, since the input is *N* frames of a video clip, the shape of feature sequence has an additional dimension, that is, number of frames. In order to handle the frames with multiple salient foreground objects, we merge multiscale tokens within one attention layer. The input sequence $F \in R^{h \times w \times frames \times c}$ are firstly projected into query (Q), key (K), and value (V). Different from Shunted Transformer [21], we do not choose a series of different down-sampling ratios which is specially designed as backbone network. We only use one down-sampling ratio 2 which means reduction of the feature sequence's length by half. The illustration of our multi-scale self-attention operation is shown in Fig.16.

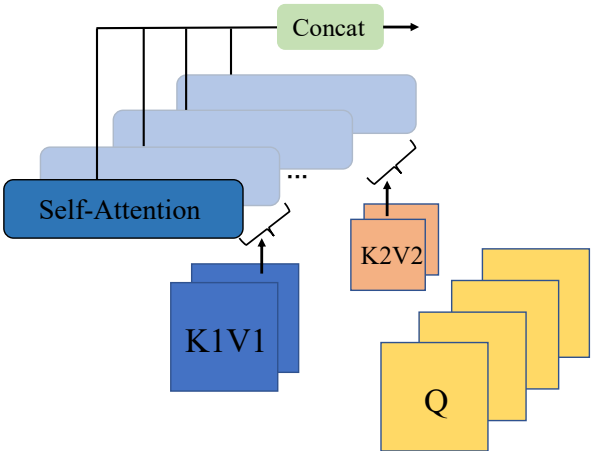


Fig. 16. Illustration of multi-scale self-attention.

Integrating the keys with different scales within one attention layer enables the network to capture multi-granularity spatio-temporal features. Specifically, we split the attention process into two parts. The sizes of keys K and values V are different in each part for different heads indexed by i:

$$\begin{aligned} Q_i &= XW_i^Q \\ K_i &= spatial\ reduced(X, r_i)W_i^K \\ V_i &= spatial\ reduced(X, r_i)W_i^V. \end{aligned} \tag{5}$$

In this work, we use a 3D convolution layer with kernel size and stride of $\boldsymbol{r_i}$ to implement the spatial reduction operations. In practice, we set $\boldsymbol{r_i}$ to 2 or 1 to prevent incurred feature information loss since our feature maps have already down sampled by CNN encoder. The shunted self-attention is calculated by:

$$h_i = Softmax(\frac{Q_i K_i^T}{\sqrt{d_h}})V_i \tag{6}$$

Then we replace the original self-attention layer in Transformer block to form the new blocks. The specific design for our MSSA branch is shown Fig.17. The green blocks denote the Transformer blocks with original self-attention layer, and the yellow blocks are with the multi-scale self-attention layer. Then we utilize the MSSA branch to capture the spatio-temporal features.

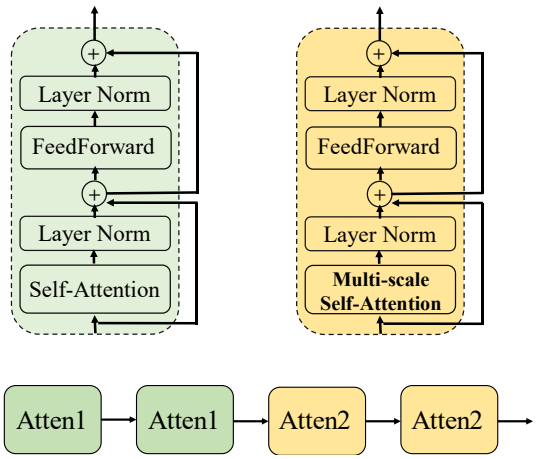


Fig. 17. Illustration of proposed MSSA branch.

3.3 Network Training

3.3.1 Loss Function

Similar to PCSA network [13] and UFO [14], we use a weighted Binary Cross-Entropy loss and a IoU loss for pixel-wise segmentation. We denote the predicted probability as *P (i, j)* and denote the groundtruth as *G(i, j)*. The ratio of all positive pixels to the total number of pixels in the image is calculated as γ. Then the weight Binary Cross-Entropy loss can be defined as:

$$L_{wbce} = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\gamma\, G(i,j)log\big(p(i,j)\big) - (1-\gamma)\big(1-G(i,j)\big)\Big(log\big(1-p(i,j)\big)\Big), \quad (7)$$

where H and W denote the height and width of the image.

Besides we also use IoU loss to evaluate segmentation accuracy as follows:

$$L_{iou} = 1 - \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} P(i,j)G(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W}[p(i,j)+G(i,j)-P(i,j)G(i,j)]}. \quad (8)$$

The total loss of the framework is formulated as follows:

$$L_{total} = L_{wbce} + L_{iou}. \quad (9)$$

3.3.2 Training Scheme

In accordance with the procedure adopted by various studies, we initially conduct pre-training on a static image dataset, and then finetune the whole network on the video dataset. The input contains $N = 5$ frames, which has fixed size of 224 × 224 for training and testing. The numbers of multi-heads of two different Transformer blocks are set to 4 and 8 respectively, and the hidden dimensions of FeedForward layer are set to 782. All experiments are conducted on two GTX 1080Ti GPUs.

Pretrain phase. We firstly pre-trained the network on static image dataset COCO-SEG [22] which contains 200,000 images and each image has pixel-wise annotations. The Adam optimizer is employed with an initial learning rate of 1e-5, which is reduced by half every 20,000 epochs. The pretraining process takes about 21 hours for 100,000

epochs.

Finetune phase. After pretraining, we finetune the network on the video datasets $DAVIS_{16}$ [23] and FBMS [24], which contains 59 video clips in total. We set the batchsize to 4 and use the same learning rate schedule as the pretrain phase. The data augmentation methods contain random rotation, random flip, random crop and color jitter, which are the common practice for data augmentation. The finetune phase takes about 12 hours for 100,000 epochs.

Chapter 4 Experimental Results

4.1 Datasets

We evaluate video salient object detection methods on DAVIS16 [23], FBMS [24] SegTrack-V2 [25] and ViSal [26] benchmarks. DAVIS16 has 50 high-quality video sequences (30 for training and 20 for testing) which have pixel-wise manually created segmentation in the form of binary mask for each frame. ViSal and SegTrack-V2 datasets are used to evaluate the model because all VSOD methods are not trained with any subsets of them.

4.2 Evaluation Metrics

Three main metrics are used to evaluate the VSOD method, including mean absolute error MAE [27], F-measureF_β[28], and Structural measurement (S-measure) [29]. MAE measures the absolute pixel errors between the predicted mask and groundtruth:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(G_i - S_i)| \tag{10}$$

where N denotes the number of all pixels, S_i represents the predicted saliency map value and G_i represents the groud truth value. F-measure is computed as a weighted mean of precision and recall and it defined as follows:

$$F_\beta = \frac{(1+\beta^2)\,Precision \times Recall}{\beta \times Precision + Recall} \tag{11}$$

S-measure assesses the structural similarity between the real-valued saliency map and the binary ground truth. It takes into consideration both object-aware (So) and region-aware (Sr) structural similarities:

$$S = \alpha \times S_o + (1-\alpha) \times S_r \tag{12}$$

where α is set to 0.5.

4.3 Experimental Results and Comparisons

4.3.1 Visual Results

The Visual Results could be seen in this part.

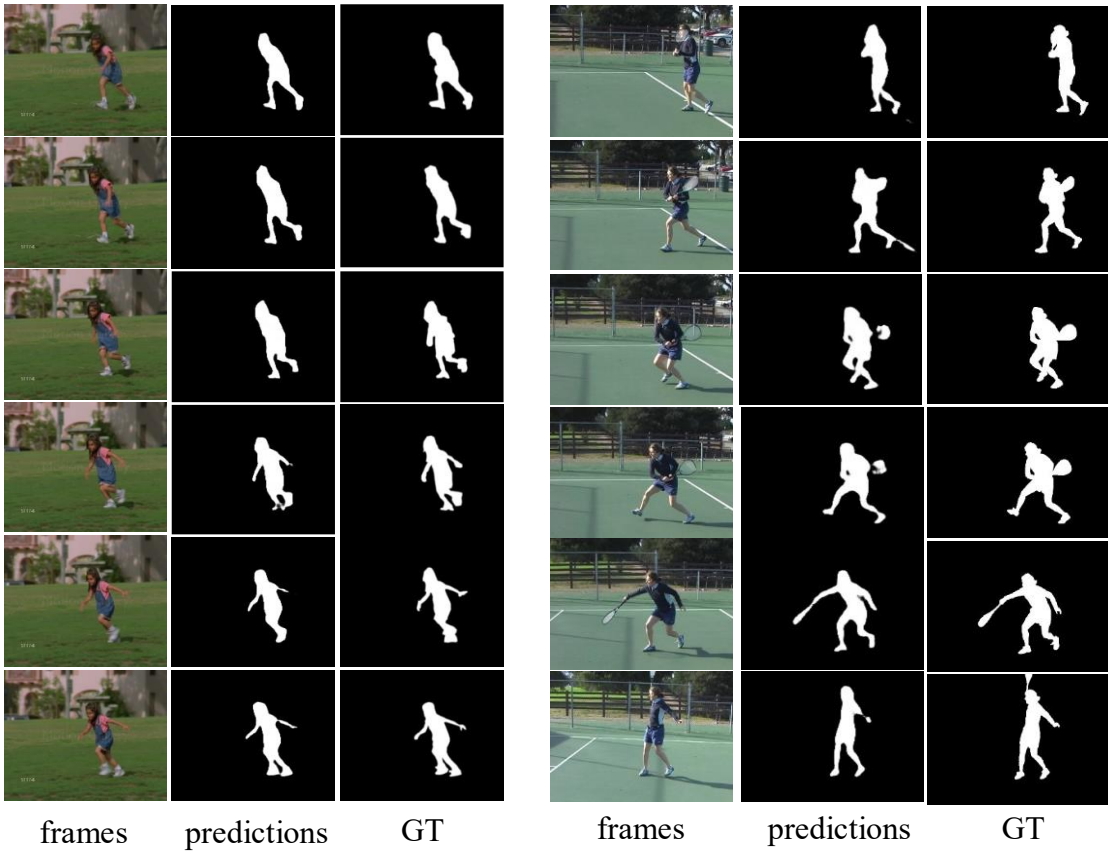


Fig. 18. Visual results for our proposed method.

These frames are from the public datasets, FBMS (tennis), and SegTrack-V2 (girl). The first column represents the original video clip, the second column displays the predicted salient mask, and the third column showcases the ground truth. From the depicted figure, it is evident that the predicted salient mask generated by our proposed method yields results comparable to the ground truth mask.

4.3.2 Ablation Study

We investigate the effect of the proposed MSSA branch by making comparisons with the original transformer blocks. The results can be seen in Tab.1, under the same training scheme, the MAE of the baseline with MSSA branch is lower than that with

original Transformer blocks.

Table. 1 Analysis of effectiveness of MSSA branch.

Method	DAVIS	SegTrack-V2
	MAE↓	*MAE*↓
VGG+Tranformer block	0.041	0.045
VGG+MSSA(ours)	**0.033**	**0.034**

In table.1, we compare the main metrics MAE between the VGG+Transformer and our proposed VGG+MSSA. From this table, we can see that our proposed MSSA gives the better performance than that with original Transformer blocks, which shows the effectiveness of MSSA branch.

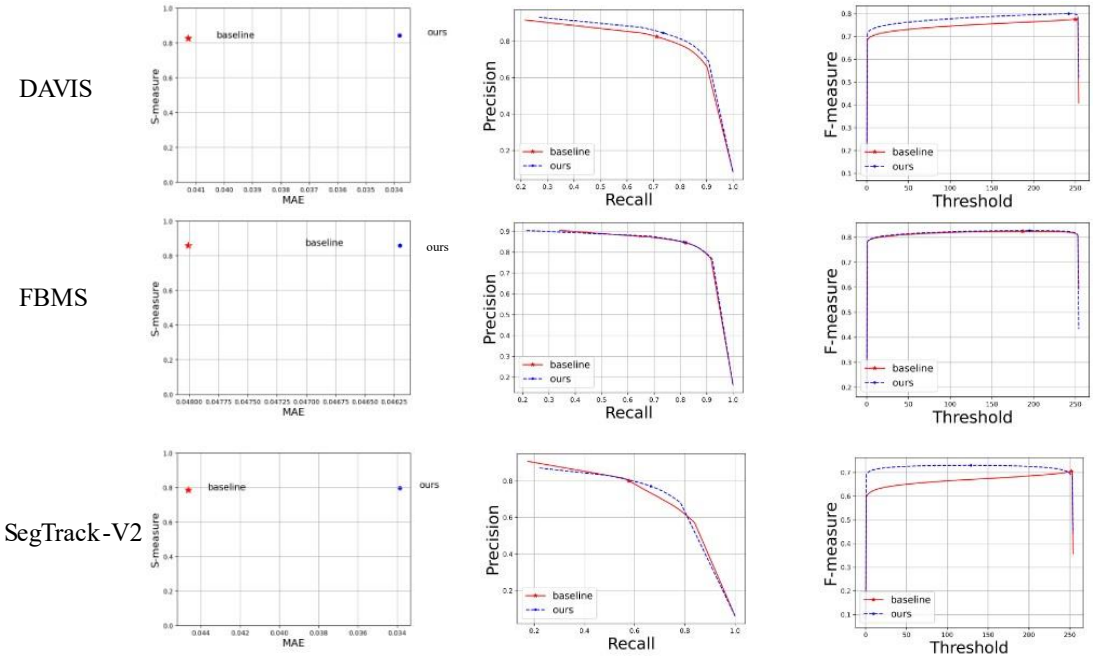


Fig. 19. Analysis of effectiveness of proposed MSSA method.

In Fig.19, we compare different metrics on the three main datasets. The first figure gives the comparisons with S-measure and MAE, and our proposed MSSA gives the better performance on these datasets. The Precision-Recall Curves of our proposed method also higher than that with original Transformer blocks. F-measure curves also give the same performance, which proves that our proposed MSSA is effective.

Then we compare our method with several previous SOTA methods, which includes two conventional methods: MSTM [30], STBP [31] and three deep-learning based methods: SCOM [32], SCNN [33], FGRN [34]. We use three evaluation metrics: MAE [27], Fmearsure [28], and Smeasure [29] for a fair comparison. Quantitative comparison results are shown in Tab.2. This shows that our method can perform better than some previous SOTA methods due to our proposed multi-scaled self-attention branch.

Table. 2 Comparisons of our method with other state-of-the arts on four VSOD datasets.

Methods	DAVIS			SegV2			FBMS			ViSal		
	$MAE\downarrow$	$S_m\uparrow$	$F_\beta^{max}\uparrow$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta^{max}\uparrow$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta^{max}\uparrow$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta^{max}\uparrow$
MSTM[28]	0.174	0.566	0.395	0.114	0.643	0.500	0.177	0.613	0.500	0.095	0.749	0.673
STBP[29]	0.096	0.677	0.544	0.061	0.735	0.640	0.152	0.627	0.595	0.163	0.629	0.622
SCOM[30]	0.055	0.814	0.746	**0.030**	**0.815**	**0.764**	0.079	0.794	0.797	0.122	0.762	0.831
SCNN[31]	0.077	0.761	0.679	-	-	-	0.095	0.794	0.762	0.071	0.847	0.831
FGRN[32]	0.043	0.838	0.783	0.035	0.770	0.694	0.088	0.809	0.767	0.045	0.861	0.848
Ours	**0.033**	**0.845**	**0.800**	0.034	0.799	0.730	**0.046**	**0.861**	**0.831**	**0.029**	**0.900**	**0.875**

4.4 Experimental Results of Our Method with Optical-flow

However, there are some cases which are difficult to locate and segment the accurate saliency maps from the background. For instance, it is hard for the model to segment a dancing girl form the audience in DAVIS dataset, because they are all belong to the same object “people”. As we can see in the Fig.21, the images in the first column are the RGB images. A dancing girl is dancing in front of a crowd of people. Even for people, it seems hard to locate and segment the salient foreground dancing girl. Thus, introducing additional optical flow information is useful in such condition.

Using additional optical flow information could alleviate this issue by making our network pay more attention to the moving foreground objects. Therefore, we use the same backbone network to generate the flow feature maps to enhance the image features. The enhanced feature maps are then fed into the subsequent networks. The test results on DAVIS of our network with additional optical flow can be seen in Tab.3. The additional optical flow information alleviates the mistake in the dataset to some extent. However not all the datasets have rich optical flow information and using such optical flow information could make our network inconvenient.

The architecture of our utilized encoder is shown below, we use the same backbone VGG-16 to generate optical flow features and then multiply with the RGB image feature maps. The subsequent network is unchanged and use the same way to extract spatial-temporal information. Finally gives the predicted salient mask.

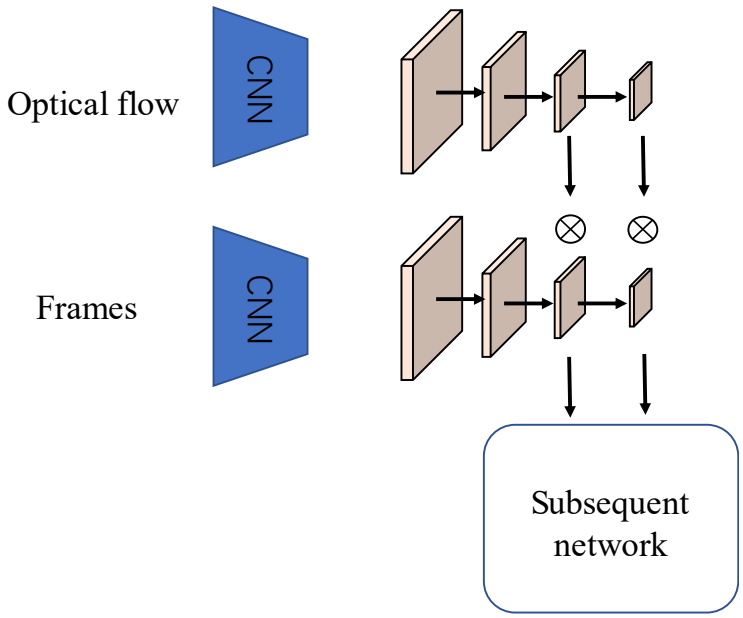


Fig. 20. Illustration of our method with optical flow.

The utilized optical flow information is generated by PWC-Net [35]. In practical operation, we manually used this optical flow estimation method to obtain the optical information from the video dataset DAVIS, as supplementary information for the network.

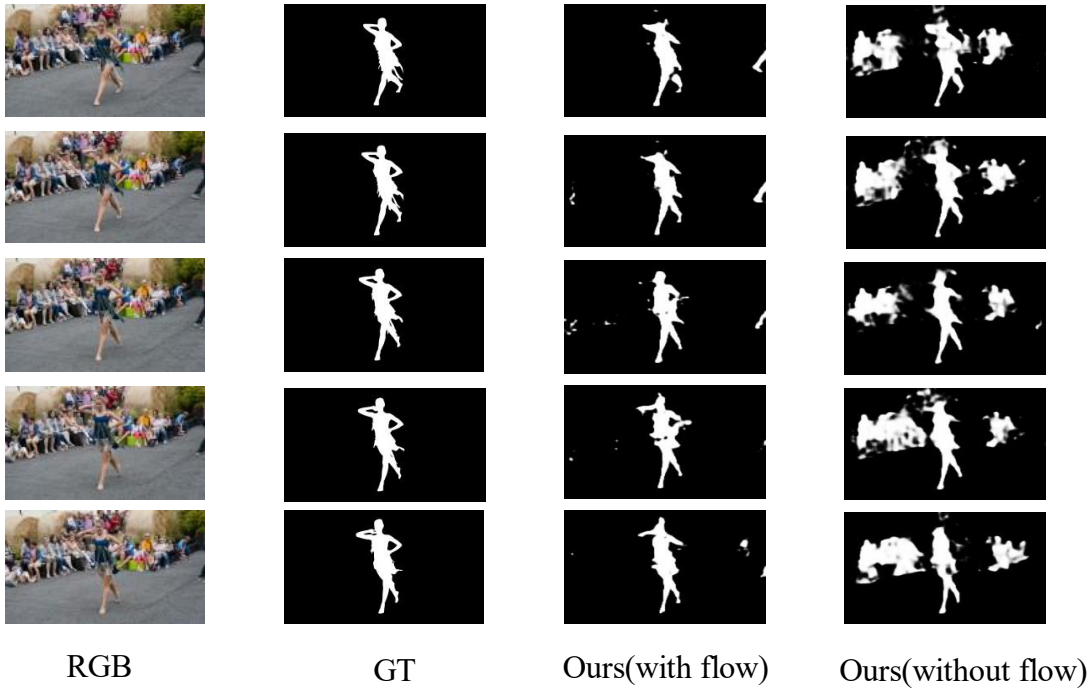


Fig. 21. Visual comparisons for our model with additional optical flow and without optical flow.

From Fig.21, we can see that adding additional optical flow information helps the network better locate and segment the foreground salient objects. However, not all the video datasets contain the sufficient optical flow information. Some datasets only contain key frames of a video clip. Therefore, we conduct the experiments mainly on the large dataset DAVIS. And the comparison results could be seen in the following Table 3. When there is sufficient optical flow, the prior information will improve the accuracy of segmentation.

Table. 3 Comparisons of our method with optical flow

Methods	DAVIS		
	$MAE\downarrow$	$S_m\uparrow$	$F_\beta^{max}\uparrow$
Ours (without flow)	0.033	0.845	0.800
Ours (with flow)	**0.023**	**0.881**	**0.853**

Chapter 5 Conclusion and future works

5.1 Conclusion

In this paper, we proposed a multi-scale self-attention (MSSA) module for VSOD, which could not only capture the spatiotemporal information but also make the network effectively model the salient objects with different scales. Experiments show that our MSSA achieve better performance than the original self-attention operation in ViT for VSOD task, and the test results on four benchmarks also demonstrate the effectiveness of our method.

How to effectively capture spatial information and temporal dynamics is key to VSOD task. In the introduction part, we analyzed the challenge of this task and existing methods. According to my observation of the data in the video datasets, I found that the size of the same saliency object varies in different frames. Therefore, we proposed Multi-scaled Self-attention branch together with the CNN backbone to better adapt to the characteristics of the datasets. In Chapter 3, we give the architecture of our proposed model and related mathematical derivation. Finally, we give the experimental results and analysis of results. In this part, we have demonstrated that the proposed Multi-scaled Self-attention is effective, which is the main contribution of this paper.

5.2 Future works

In this paper, we proposed that using Multi-scaled Self-attention to integrating features from different scales while capturing long-range dependencies. Such work has not been done in VSOD task before. And we improve the effectiveness of our proposed method. However, due to the training scheme and insufficient design of some network feature fusion modules, there is still room for improvement in the performance of the model.

Secondly, I have submitted the paper to IEEE Global Conference on Consumer

Electronics(GCCE) 2023. If the paper is accepted I will submit the four-page version.

Bibliography

[1] Y. Pan, T. Yao, H. Li, and T. Mei, “Video Captioning with Transferred Semantic Attributes,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.984-992, Jul. 2017.

[2] H. Hadizadeh, and I. V. Bajic, “Saliency-Aware Video Compression, IEEE Transactions on Image Processing,” vol.23, no.1, pp.19-33, Jan. 2014.

[3] H. Wu, G. Li, and X. Luo, “Weighted Attentional Blocks for Probabilistic Object tracking,” Springer, The Visual Computer, vol.30, pp.229-243, Feb. 2014.

[4] Z. Zhang, S. Fidler, and R. Urtasun, “Instance-Level Segmentation for Autonomous Driving with Deep Densely Connected MRFs,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.669–677, Jun.2016.

[5] Z. Liu, J. Li, L. Ye, G. Sun and L. Shen, “Saliency Detection for Unconstrained Videos Using Superpixel-Level Graph and Spatiotemporal Propagation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 12, pp. 2527-2542, Dec. 2017.

[6] W. Wang, J. Shen and L. Shao, “Video Salient Object Detection via Fully Convolutional Networks, ” IEEE Transactions on Image Processing, vol. 27, no. 1, pp. 38-49, Jan. 2018.

[7] M. Sun, Z. Zhou, Q. Hu, Z. Wang and J. Jiang, “SG-FCN: A Motion and Memory-Based Deep Learning Model for Video Saliency Detection, ” IEEE Transactions on Cybernetics, vol. 49, no. 8, pp. 2900-2911, Aug. 2019.

[8] T. -N. Le and A. Sugimoto, “Video Salient Object Detection Using Spatiotemporal Deep Features,” IEEE Transactions on Image Processing, vol. 27, no. 10, pp. 5002-5015, Oct. 2018.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” IEEE International Conference on Computer Vision (ICCV), pp. 4489–4497, Dec.2015.

[10] H. Li, G. Chen, G. Li, and Y. Yizhou, “Motion Guided Attention for Video Salient Object Detection,” IEEE International Conference on Computer Vision (ICCV), pp. 4489–4497, Oct.2019.

[11] P. Yan, G. Li, Y. Xie, Z. Li, C. Wang, T. Chen, and L. Lin, “Semi-supervised Video

Salient Object Detection Using Pseudo-labels," IEEE International Conference on Computer Vision (ICCV), pp. 7284–7293, Oct.2019.

[12] D. -P. Fan, W. Wang, M. -M. Cheng and J. Shen, "Shifting More Attention to Video Salient Object Detection," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8554–8564, Jun.2019.

[13] Y. Gu, L. Wang, Z. Wang, Y. Liu, M.-M. Cheng, and S.-P. Lu, "Pyramid Constrained Self-Attention Network for Fast Video Salient Object Detection", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, pp. 10869-10876, Apr. 2020.

[14] Y. Su, J. Deng, R. Sun, G. Lin, H. Su and Q. Wu, "A Unified Transformer Framework for Group-based Segmentation: Co-Segmentation, Co-Saliency Detection and Video Salient Object Detection," in IEEE Transactions on Multimedia, pp.1-13, Apr.2023.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale," International Conference on Learning Representations (ICLR), May. 2021.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations (ICLR), May.2015.

[17] A.Vaswani, N.Shazeer, N.Parmar, et al. "Attention is all you need[J] ," Advances in neural information processing systems, pp.5998-6008, Dec.2017.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. "Swin transformer: Hierarchical vision transformer using shifted windows," IEEE International Conference on Computer Vision (ICCV), pp.10012-10022, Oct.2021.

[19] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," IEEE International Conference on Computer Vision (ICCV), pp.568-578, Oct.2021.

[20] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2117–2125, Jul.2017.

[21] Sucheng Ren, Daquan Zhou, Shengfeng He, Jias-hi Feng, Xinchao Wang.

"Shunted Self-Attention via Multi-Scale Token Aggregation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10853-10862, Jun.2022.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision. Springer, pp. 740–755, Mar.2014.

[23] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. "A benchmark dataset and evaluation methodology for video object segmentation," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 724–732, Jun.2016.

[24] P. Ochs, J. Malik, and T. Brox, "Segmentation of Moving Objects by Long Term Video Analysis," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 6, pp. 1187–1200, Jun.2014.

[25] F. Li, T. Kim, A. Humayun, D. Tsai and J. M. Rehg, "Video Segmentation by Tracking Many Figure-Ground Segments," 2013 IEEE International Conference on Computer Vision (ICCV), pp. 2192–2199, Dec.2013.

[26] W. Wang, J. Shen and L. Shao, "Consistent Video Saliency Using Local Gradient Flow Optimization and Global Refinement," IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 4185-4196, Nov. 2015.

[27] F. Perazzi, P. Krahenb ¨ uhl, Y. Pritch, and A. Hornung, "Saliency filters: " Contrast based filtering for salient region detection," in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 733–740. 6, Jun.2012.

[28] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in 2009 IEEE conference on computer vision and pattern recognition. IEEE, 2009, pp. 1597–1604. 6.

[29] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4548–4557. 6.

[30] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2334–2342, Jun.2016.

[31] T. Xi, W. Zhao, H. Wang and W. Lin, "Salient Object Detection With Spatiotemporal Background Priors for Video," in IEEE Transactions on Image

Processing, vol. 26, no. 7, pp. 3425-3436, July 2017.

[32] Y. Chen, W. Zou, Y. Tang, X. Li, C. Xu and N. Komodakis, “SCOM: Spatiotemporal Constrained Optimization for Salient Object Detection, ” in IEEE Transactions on Image Processing, vol. 27, no. 7, pp. 3345-3357, July 2018 .

[33] Y. Tang, W. Zou, Z. Jin, Y. Chen, Y. Hua and X. Li, “Weakly Supervised Salient Object Detection with Spatiotemporal Cascade Neural Networks,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 7, pp. 1973-1984, July 2019.

[34] G. Li, Y. Xie, T. Wei, K. Wang and L. Lin, “Flow Guided Recurrent Neural Encoder for Video Salient Object Detection,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 3243-3252.

[35] D. Sun, X. Yang, M. -Y. Liu and J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 8934-8943.