Exploiting Temporal State Space Sharing for Video Semantic Segmentation

Syed Ariff Syed Hesham[1,2], Yun Liu[3*], Guolei Sun[4*], Henghui Ding[5], Jing Yang[6],
Ender Konukoglu[4], Xue Geng[2], Xudong Jiang[1]
[1] Nanyang Technological University [2] Institute for Infocomm Research, A*STAR
[3] VCIP, CS, Nankai University [4] CVL, ETH Zurich [5] Fudan University [6] Guizhou University

Abstract

Video semantic segmentation (VSS) plays a vital role in understanding the temporal evolution of scenes. Traditional methods often segment videos frame-by-frame or in a short temporal window, leading to limited temporal context, redundant computations, and heavy memory requirements. To this end, we introduce a ***Temporal Video State Space Sharing (TV3S)*** *architecture to leverage Mamba state space models for temporal feature sharing. Our model features a selective gating mechanism that efficiently propagates relevant information across video frames, eliminating the need for a memory-heavy feature pool. By processing spatial patches independently and incorporating shifted operation, TV3S supports highly parallel computation in both training and inference stages, which reduces the delay in sequential state space processing and improves the scalability for long video sequences. Moreover, TV3S incorporates information from prior frames during inference, achieving long-range temporal coherence and superior adaptability to extended sequences. Evaluations on the VSPW and Cityscapes datasets reveal that our approach outperforms current state-of-the-art methods, establishing a new standard for VSS with consistent results across long video sequences. By achieving a good balance between accuracy and efficiency, TV3S shows a significant advancement in spatiotemporal modeling, paving the way for efficient video analysis. The code is publicly available at* `https://github.com/Ashesham/TV3S.git`.

1. Introduction

Semantic Segmentation has achieved substantial progress through the introduction of convolutional neural networks [5, 40, 48, 49, 72] and, more recently with vision transformers [4, 14, 39] due to their ability to capture spatial patterns and contextual information. However, these techniques are mainly designed for static images and do not leverage the temporal dynamics present in videos [2, 11, 12, 23, 57, 67].

*Corresponding authors: Yun Liu and Guolei Sun

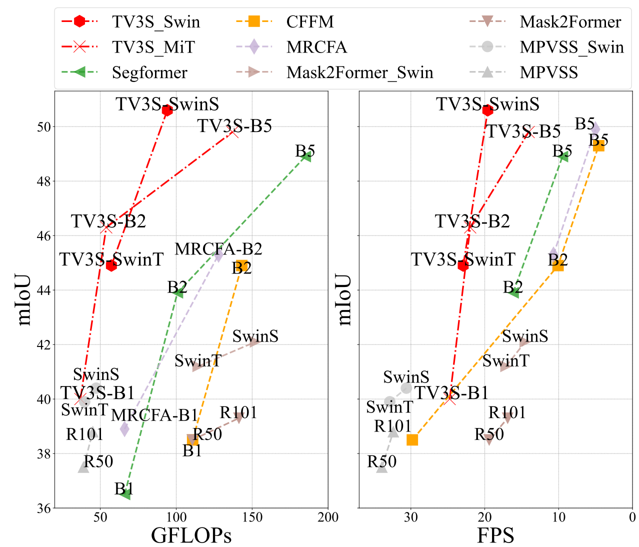

Figure 1. Comparison of the proposed TV3S with baseline models for VSS. By enhancing temporal information, our TV3S demonstrates superior performance over the baselines.

Recently, research attention has increasingly shifted toward video semantic segmentation (VSS) [36, 53, 54, 61], due to its potential to benefit a wide range of practical applications. Unlike static image segmentation, VSS requires models to track motion, adapt to changes in appearance, and handle interactions among objects across consecutive frames. These challenges necessitate sophisticated techniques to accurately interpret dynamic environments where objects may occlude one another, shift positions, or exhibit varying states of motion and stillness [7, 47].

The contemporary approaches of VSS fall into two main categories. The first leverages *optical flow*, modeling pixel movement between consecutive frames to align features and support object tracking, thereby enhancing temporal coherence. These methods effectively propagate temporal information, making them useful for fine-grained temporal alignment [8, 13, 17, 50, 67]. However, they have significant drawbacks, including high computational costs due to the complexity of estimating accurate flow fields, especially in scenes with occlusions or sudden changes. Additionally, optical flow relies on precise motion estimation, where inaccuracies in dynamic scenes can propagate and lead to reduced accuracy in video segmentation.

The second category involves *feature aggregation*, where information from multiple frames are combined to improve segmentation accuracy [28, 70]. The researchers relied on recurrent neural networks (RNNs) particularly Convolutional Long Short-Term Memory (ConvLSTM) [52] to introduce temporal structures to the models [15, 43, 46, 62, 64]. Recently, VM-RNN [60] has advanced this area by applying an efficient LSTM model to work with Vision Mamba [74] to capture spatiotemporal dynamics. While these methods can capture both spatial and temporal information, they face challenges when scaling to handle long video sequences, largely due to the high computational and memory demands of recurrent networks. Furthermore, the sensitivity of LSTMs to sequence length often leads to training instability, requiring careful tuning and optimization [46, 52, 64]. These limitations underscore the need for more scalable architectures.

Due to such challenges in past feature aggregation methods, the focus in this domain has shifted towards the usage of transformer-based models [25, 29–31, 57–59, 65] [63]. However, these approaches still face challenges, including high memory costs and limited scalability, particularly with long video sequences or high-resolution frames. Efficient information sharing across numerous frames continues to be a challenge, making real-time applications difficult.

To address the challenges of integrating spatial and temporal information in VSS, we present **Temporal State Space Sharing (TV3S)**, designed to overcome the limitations of existing models. Unlike traditional optical flow methods, which struggle with inaccuracies in dynamic scenes, and RNN- and Transformer-based architectures, which are constrained by expensive feature pooling and high computational costs, TV3S takes a more efficient approach by processing spatial patches in parallel while dynamically sharing temporal information across video frames. Specifically, a spatially encoded input frame is split into discrete patches, which are processed independently through a series of TV3S blocks, each containing **Temporal State Space (TSS)** modules with a selective gating mechanism that effectively integrates and propagates spatiotemporal information across frames, ensuring minimal computational overhead. To further enhance motion handling at the edges of patches, we introduce a shifted window-based approach that works with un-shifted and shifted encoded features, enabling the model to capture movements near the boundaries while still maintaining efficient temporal information sharing. These components work jointly to integrate both local and long-range temporal features, significantly improving VSS performance while avoiding excessive computational costs and improving overall efficiency as evident in Fig. 1.

In summary, the contributions of our paper include:

- We present the Temporal Video State Space Sharing (TV3S) architecture, a novel framework that shares and propagates temporal information across video frames efficiently and effectively.
- We process spatial patches independently with a selective gating mechanism efficiently, thus enabling parallel computation during both training and inference and supporting scalability for long, high-resolution video sequences.
- We design a shifted window-based approach within the TV3S block, enhancing temporal state space sharing and capturing long-range spatial context effectively.

Through extensive experiments on the VSPW [42] and Cityscapes [9] datasets, we demonstrate that our approach surpasses existing state-of-the-art methods, establishing new benchmarks for efficiency and accuracy in VSS.

2. Related Work

Semantic segmentation began with natural images, with the introduction of fully convolutional networks (FCNs) [40] pioneering an end-to-end pixel-wise classification framework. Building on this foundation, subsequent works enhanced segmentation accuracy by adopting atrous convolutional layers [5, 6], employing pyramid architectures [5, 6, 72], leveraging encoder-decoder architectures [10, 45, 48], and incorporating attention mechanisms [16, 24, 66]. More recently, transformer-based architectures like SegFormer [69], Segmenter [56], SETR [73], and Mask2Former [7], have further advanced the field by learning global dependencies. While these developments have significantly improved image semantic segmentation, transitioning them to VSS introduces new challenges related to temporal coherence and efficient exploitation of temporal information.

2.1. VSS Based on Recurrent Neural Networks

Early works in this field explored recurrent neural networks (RNNs) for temporal modeling. Valipour *et al.* [62] incorporated a recurrent unit between the encoder and decoder, significantly improving video segmentation performance. Evaluations by [15] on different RNN structures such as ConvRNN [55], ConvGRU [3], and ConvLSTM [52] on the KITTI dataset [18] demonstrated the superiority of ConvLSTM in handling video sequences. Further explorations have combined ConvGRU with optical flow to represent pixel displacements and maintain temporal continuity [44], while bidirectional ConvLSTM has been applied to merge temporally adjacent features, enhancing stability across frames [43]. Despite these advancements, the high computational and memory cost of RNNs for processing longer video sequences poses notable challenges, highlighting the need for more efficient models capable of handling

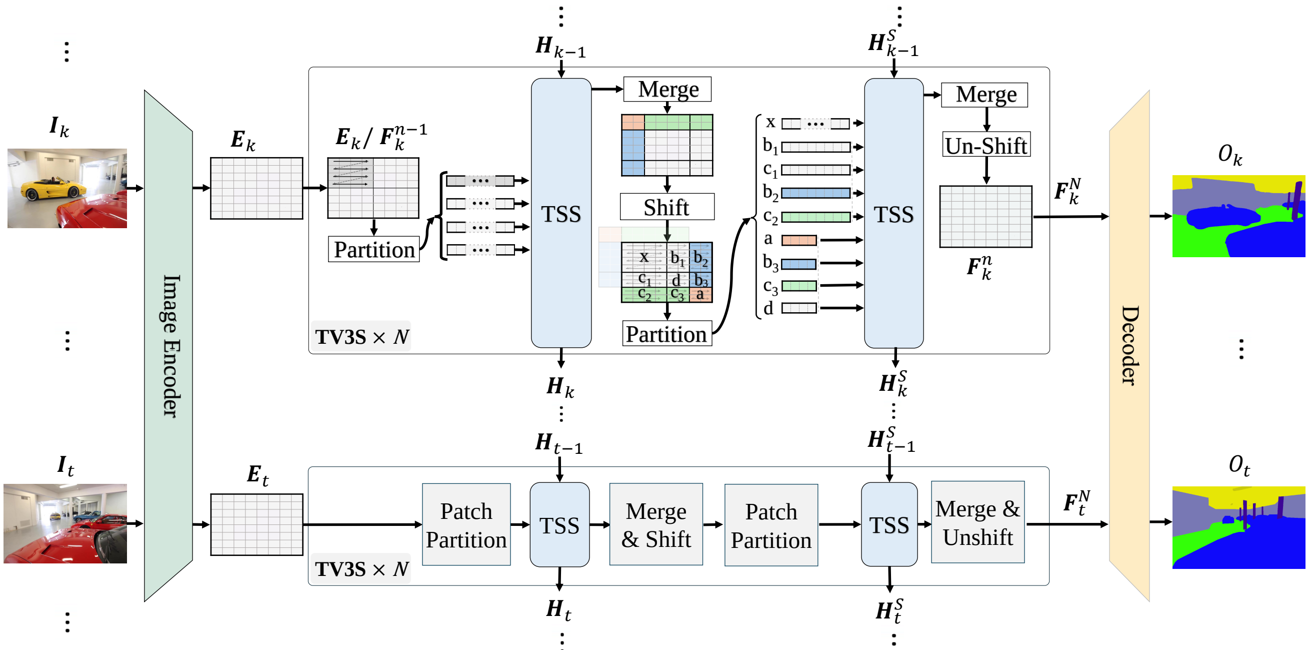

Figure 2. Overview of the proposed TV3S architecture, illustrating the encoder-decoder framework that employs state space models and our TSS module based on Mamba [19] for independent spatial and temporal processing.

long-range temporal dependencies [46, 52].

2.2. VSS Based on CNNs and Transformers

To address RNN limitations, CNN-based methods were developed for temporal modeling. DFF [75] and Accel [26] utilized optical flow to propagate features between frames, reducing redundancy. ClockNet [51] and LLVS [35] introduced adaptive feature reuse to exploit semantic similarity across frames for efficiency and temporal coherence.

Recognizing the limitations of optical flow and CNNs for long-range temporal modeling (see Sec. 1), recent advancements [1, 31, 33, 34] have seen a shift towards using transformers [63] in VSS. Transformers, with their self-attention mechanism, can capture global dependencies across frames, making them suited for temporal feature aggregation. Among notable approaches, MPVSS [67] proposes a memory-augmented transformer framework to capture multi-frame dependencies, enabling efficient temporal aggregation across longer video sequences. Similarly, CFFM [57] and MRCFA [58] employ multi-resolution cross-frame attention to handle temporal variations by disentangling static and dynamic contexts within video frames, allowing for a refined segmentation process that distinguishes between stationary and moving elements.

Despite advancements, transformer-based methods face high computational costs due to the quadratic complexity of self-attention, especially with high-resolution frames or long videos [63]. Additionally, they are usually designed to learn temporal information in a short video sequence due to the high complexity. This gap highlights the need for more efficient and holistic models that can simultaneously manage computational costs while effectively modeling long-range temporal information in VSS.

2.3. State Space Models

State space models (SSMs) present a promising alternative for temporal modeling, addressing the shortcomings of RNNs and transformers. Unlike RNNs, which suffer from scaling challenges with sequence length, SSMs like S4 [21, 22] exhibit with linear complexity by imposing diagonal structures on state matrices, making them more efficient for long data sequences. Enhanced through HiPPO [20] initialization, SSMs handle extensive dependencies while requiring less memory, making them well-suited for tasks requiring to store long temporal context. Recently, Mamba [19] introduced a selective-scan mechanism to process temporal data efficiently. This advancement has spurred adaptations in vision-specific models, such as Vision Mamba [74] and VMamba [38], which incorporate Mamba blocks in a hierarchical structure to overcome directional sensitivities and maintain scalability across high-resolution inputs. Additionally, Vim [74] refined the scanning techniques to prevent overfitting. Furthermore, U-Mamba [41] has explored a hybrid network architecture that combines SSMs with convolutional layers. VideoMamba [32] represents one of the early frameworks to leverage SSM-based modules for video understanding, focusing on video clip classification. However, it is limited by its offline processing, which increases memory and computational demands, restricts dense semantic segmentation, and hinders its ability

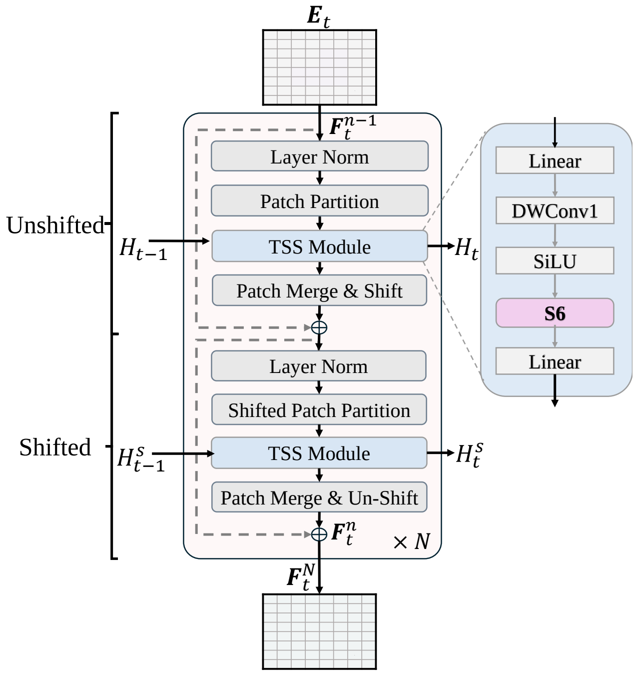

Figure 3. Internal structure of the TV3S block, illustrating the flow of internal operations, and the propagation of hidden states for efficient spatiotemporal integration.

to capture the global context of sequences.

However, despite these advancements, current vision-specific models largely rely on hierarchical designs fail to fully exploit varying temporal resolutions and context lengths, highlighting a critical need for improved mechanisms for temporal state sharing. To this end, our proposed TV3S architecture integrates parameter-efficient SSMs to enhance effective temporal state sharing, thus proving advantageous for VSS. This integration not only underscores the computational efficiency and scalability of SSMs in VSS but also emphasizes their suitability for tasks requiring long-range temporal coherence, representing a significant novelty in the field.

3. Methodology

Fig. 2 illustrates the proposed TV3S architecture, which effectively captures and integrates temporal dependencies within video sequences using a state space modeling framework. It aims to leverage both temporal and spatial information to achieve accurate and consistent segmentation results.

3.1. Overall Architecture

The architecture processes a sequence of video frames denoted as $\{\boldsymbol{I}_{t-l}, ..., \boldsymbol{I}_{t-k}, ..., \boldsymbol{I}_{t-1}, \boldsymbol{I}_t\}$, where I_t represents the frame at time t, and $\{l, ..., k, ..., 1\}$ are temporal offsets capturing frames from the past relative to t. These frames are fed into an image encoder such as MiT [69] and Swin [39]. The encoder produces a corresponding set of encoded feature $\{\boldsymbol{E}_{t-l}, ..., \boldsymbol{E}_{t-k}, ..., \boldsymbol{E}_{t-1}, \boldsymbol{E}_t\}$, capturing rich spatial context.

The TV3S architecture processes these feature maps sequentially through a series of N TV3S blocks (with a default value of $N = 4$), each containing two TSS modules that independently process spatial patches of the feature map, facilitating parallelization. Layer normalization is applied before each TSS module, and residual connections are included after each module to stabilize training. Each TSS module consists of operations including a linear projection, a 1-dimension depthwise convolution (DWConv1), a SiLU activation feeding into the state space model (S6), and finally ending with another linear projection to project back to the input dimension, as detailed in Fig. 3.

3.2. Temporal Video State Space Sharing Block

The TV3S block is the core unit of the TV3S architecture, which enables seamless integration of temporal information across frames. Each block consists of two TSS modules that employ a state space model to capture and propagate temporal dependencies within frames. The encoded feature map $\boldsymbol{E}_t$ is partitioned into non-overlapping patches of size $w \times w$. Each patch $\boldsymbol{P}_t^{i,j}$ is then flattened, transforming the spatial dimensions into a linear sequence suitable for parallel processing, with i and j representing the patch indices. Mathematically, $\boldsymbol{P}_t^{i,j}$ can be expressed as

$$\begin{aligned} \boldsymbol{P}_t^{i,j} &= \boldsymbol{E}_t[i : i + w, j : j + w, :] \in \mathbb{R}^{C_E \times w \times w}, \\ \hat{\boldsymbol{P}}_t^{i,j} &= \text{Flatten}(\boldsymbol{P}_t^{i,j}) \in \mathbb{R}^{C_E \times (w \times w)}, \end{aligned} \tag{1}$$

where C_E denotes the number of channels in $\boldsymbol{E}_t$ after encoding $\boldsymbol{I}_t$. In contrast to traditional Mamba variants [38, 41, 74], which process sequences sequentially, our approach parallelizes the handling of each window or patch, offering two main advantages. First, the encoder effectively encodes complex spatial information across channels, eliminating for further processing of spatial relationships. Second, minimal changes between consecutive frames allow window-based processing to efficiently capture temporal dynamics, enabling a highly parallelized and computationally efficient model.

State space model for temporal aggregation.

Each TSS module within a TV3S block utilizes a state space model to capture temporal dependencies between frames. Given encoded feature map $\boldsymbol{E}_t$, the hidden state $\boldsymbol{H}_t$ is updated using the current flattened patch $\hat{\boldsymbol{P}}_t^{i,j}$ and the previous hidden state $\boldsymbol{H}_{t-1}$:

$$\begin{aligned} \boldsymbol{H}_t^{i,j} &= f_A(\Delta, \boldsymbol{A}_s)\boldsymbol{H}_{t-1}^{i,j} + f_B(\Delta, \boldsymbol{A}_s, \boldsymbol{B}_s)\hat{\boldsymbol{P}}_t^{i,j}, \\ \boldsymbol{F}_t^{i,j} &= \boldsymbol{C}_s \boldsymbol{H}_t^{i,j}. \end{aligned} \tag{2}$$

Here, $\boldsymbol{A}_s \in \mathbb{R}^{N_s \times N_s}$, $\boldsymbol{B}_s \in \mathbb{R}^{N_s \times C_E}$, and $\boldsymbol{C}_s \in \mathbb{R}^{C_E \times N_s}$ are learnable state space matrices. As seen in Eq. (2), the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are discretized using the time scale parameter $\boldsymbol{\Delta}_s$ with fixed discretization functions $f_A(.)$ & $f_B(.)$, referred as *discretization rules*, as described in [19] due to their many advantages. Here, N_s denotes the hidden states dimensionality of the state space model and the aggregated feature output is denoted as $\boldsymbol{F}_t^{i,j}$. This mechanism enables each patch to independently capture and propagate its temporal dependencies across two dimensions (i, j).Consequently, temporal data storage consists of a total of $\frac{W}{w} \times \frac{H}{w}$ hidden states, promoting effective temporal consistency and high parallelization.

Shifting and edge handling. To enhance spatial context sharing, the reshaped feature maps undergo a shifting operation inspired by Swin Transformer [39]. This shift is parameterized as $s \times s$, where $s < w$ (typically $s = w/2$). The shifting rearranges the blocks so that edge blocks receive information from adjacent regions, thereby enriching spatial-temporal representation. The shifted feature maps are then re-partitioned into patches, with those on the right and bottom edges further subdivided into smaller sub-blocks of dimensions $2 \times w \times \frac{w}{2}$, $2 \times \frac{w}{2} \times w$, and $4 \times \frac{w}{2} \times \frac{w}{2}$, as visualized in Fig. 2. These re-partitioned patches are processed through a second TSS module in the same TV3S block, integrating the newly acquired spatial context with temporal information.

In the proposed architecture, each TV3S block consists of a pair of TSS modules, with one module processing unshifted blocks and the other handling shifted blocks. By stacking N such TV3S blocks, the architecture forms a deep hierarchical structure termed TV3S architecture, which is capable of capturing intricate temporal dependencies across frames. The final aggregated feature representation, $\boldsymbol{F}^N$, is passed through a linear projection layer to map it to C classes, followed by an interpolation to produce the final segmentation output.

3.3. Training Strategy

The training strategy for the TV3S architecture focuses on optimizing both temporal feature aggregation and robust spatial feature extraction to enable efficient learning of short- and long-term dependencies within video sequences, facilitating high VSS performance.

During training, sequential input frames are taken at the intervals $\{\boldsymbol{I}_{t-9}, \boldsymbol{I}_{t-6}, \boldsymbol{I}_{t-3}, \boldsymbol{I}_t\}$ the same as [42]. Each of the frames in set $\{\boldsymbol{I}_{t-k}\}$ with $k = \{9, 6, 3, 0\}$, is passed through the encoder to generate encoded features $\boldsymbol{E}_{t-k}$ encapsulating rich spatial information. An intermediate feature $\hat{\boldsymbol{O}}_{t-k}$ is extracted straight from $\boldsymbol{E}_{t-k}$ with the use of linear projection that aligns the channel dimensions of $\boldsymbol{E}_{t-k}$ with the number of classes C in the dataset. Concurrently, the encoded features $\boldsymbol{E}_{t-k}$ are fed into the TV3S blocks to integrate temporal information, producing aggregated features $\boldsymbol{F}_{t-k}^N$.

Algorithm 1 Inference Procedure for TV3S Architecture

Input: Frame sequence $\{\boldsymbol{I}_t\}$
Output: Segmentation map $\boldsymbol{O}_t$
Initialize: Hidden states $\boldsymbol{H}$ for all TV3S blocks
for each frame $\boldsymbol{I}_t$ in the sequence **do**
  $\boldsymbol{E}_t \leftarrow \text{Encoder}(\boldsymbol{I}_t)$ // Extract features
  for each patch (i, j) in $\boldsymbol{E}_t$ **do**
    $\hat{\boldsymbol{P}}^{i,j} \leftarrow \text{Flatten}(\boldsymbol{E}_t[i : i + w, j : j + w, :])$
    for each TV3S block n from 1 to N **do**
      //Update Features and Hidden State.
      $\boldsymbol{F}_t^{n,i,j}, \boldsymbol{H}_t^{n,i,j} \leftarrow \text{TV3S}^n(\hat{\boldsymbol{P}}^{i,j}, \boldsymbol{H}^{n,i,j})$
    end for
  end for
  $\boldsymbol{O}_t \leftarrow \text{Interpolate}(\text{Linear}(\boldsymbol{F}_t^N))$ // Segmentation
  $\boldsymbol{H} \leftarrow \{\boldsymbol{H}_t\}$ // Store for next frame
end for

With both output predictions extracted, a weighted cross-entropy between these output predictions $\boldsymbol{O}$ and the ground-truth segmentation masks $\boldsymbol{M}$ is computed as training loss for all frames in the input sequence. The training loss is computed as:

$$\mathcal{L} = \lambda \sum_{k=\{9,6,3,0\}} \mathcal{L}_{\text{CE}}(\hat{\boldsymbol{O}}_{t-k}, \boldsymbol{M}_{t-k}) + \mathcal{L}_{\text{CE}}(\boldsymbol{O}_t, \boldsymbol{M}_t). \quad (3)$$

Loss formulation presented in Eq. (3) employs a dual-loss strategy to ensure accurate segmentation of the final frame $\boldsymbol{O}_t$ while preserving spatial relationships in the intermediate features $\hat{\boldsymbol{O}}_{t-k}$ from the encoder. By applying cross-entropy losses $\mathcal{L}_{\text{CE}}$, with intermediate losses weighted at $\lambda = 0.5$, the model effectively balances spatial feature learning and temporal coherence, enhancing its capability to extract temporally consistent spatial information for improved segmentation across consecutive frames.

3.4. Inference Procedure

During inference, the TV3S architecture processes each frame sequentially while leveraging stored hidden states to maintain temporal coherence across the video sequence, as detailed in Algorithm 1. For each frame $\boldsymbol{I}_t$, the encoder first extracts the encoded features $\boldsymbol{E}_t$, which are then partitioned into non-overlapping spatial patches indexed by (i, j) and flattened into $\boldsymbol{P}_{i,j}$. Each flattened patch is processed through the TV3S blocks, updating the current hidden states $\boldsymbol{H}_t$ using the previous states $\boldsymbol{H}_{t-1}$. After processing all patches, the aggregated feature map $\boldsymbol{F}_t^N$ is reconstructed from the updated hidden states. This feature map is then passed through a linear projection layer and interpolated to match the original image resolution, producing the final segmentation map $\boldsymbol{O}_t$. The updated hidden states $\boldsymbol{H}_t$ are then stored back in $\boldsymbol{H}$ for use with subsequent frames, ensuring continuous temporal integration and consistent segmentation results.

Methods	Backbones	mIoU↑	mVC_8↑	mVC_{16}↑	GFLOPs↓	Params(M)↓	FPS↑
Segformer† [69]	MiT-B1	36.5	84.7	79.9	26.6	13.8	58.7
CFFM [57]	MiT-B1	38.5	88.6	84.1	-	15.5	29.8
MRCFA [58]	MiT-B1	38.9	88.8	84.4	-	16.2	40.1
TV3S (Ours)	MiT-B1	**40.0**	**90.7**	**87.0**	36.9	17.3	24.7
Segformer [69]	MiT-B2	43.9	86.0	81.2	100.8	24.8	16.2
CFFM [57]	MiT-B2	44.9	89.8	85.8	143.2	26.5	10.1
MRCFA [58]	MiT-B2	45.3	90.3	86.2	127.9	27.3	10.7
TV3S (Ours)	MiT-B2	**46.3**	**91.5**	**88.35**	53.9	28.3	21.9
Mask2Former† [7]	R50	38.5	81.3	76.4	110.6	44.0	19.4
MPVSS [67]	R50	37.5	84.1	77.2	38.9	84.1	33.9
Mask2Former† [7]	R101	39.3	82.5	77.6	141.3	63.0	16.9
MPVSS [67]	R101	38.8	84.8	79.6	45.1	103.1	32.3
DeepLabv3+† [6]	R101	34.7	83.2	78.2	379.0	62.7	9.2
UperNet† [68]	R101	36.5	82.6	76.1	403.6	83.2	16.0
PSPNet† [72]	R101	36.5	84.2	79.6	401.8	70.5	13.8
OCRNet† [71]	R101	36.7	84.0	79.0	361.7	58.1	14.3
TCB† [42]	R101	37.8	87.9	84.0	1692	-	-
ETC† [37]	OCRNet	37.5	84.1	79.1	361.7	-	-
Segformer [69]	MiT-B5	48.9	87.8	83.7	185.0	82.1	9.4
CFFM [57]	MiT-B5	49.3	90.8	87.1	413.5	85.5	4.5
MRCFA [58]	MiT-B5	**49.9**	90.9	87.4	373.0	84.5	5.0
TV3S (Ours)	MiT-B5	<u>49.8</u>	**91.7**	**88.7**	137.0	85.6	14.0
Mask2Former† [7]	Swin-T	41.2	84.5	80.0	114.4	47.4	17.1
MPVSS [67]	Swin-T	39.9	85.9	80.4	39.7	114.0	32.8
TV3S (Ours)	Swin-T	**44.9**	**88.0**	**83.5**	57.3	31.7	22.9
Mask2Former† [7]	Swin-S	42.1	84.7	79.3	152.2	68.9	14.5
MPVSS [67]	Swin-S	40.4	86.0	80.7	47.3	108.0	30.6
TV3S (Ours)	Swin-S	**50.6**	**89.6**	**85.8**	94.1	53.1	19.5

Table 1. Quantitative comparison of our model with existing methods on the VSPW dataset [42]. Our model achieves a strong balance between *accuracy*, *model complexity*, and *operational speed*. FPS and FLOPs are calculated with an input resolution of 480 × 853. (†Frame-by-Frame processing)

4. Experiments

4.1. Experimental Setup

Implementation details. The implementation of our approach is based on `MMSegmentation` codebase and all the experiments including training and inference were conducted with 2 A100 NVIDIA GPUs. The main experiments are done with the backbone same as the SegFormer (Variants with MiT and Swin) which are pre-trained with ImageNet. While the model is aimed to work with any number of frame sequences, during training, the model is trained with just three reference frames, $\{k_1, k_2, k_3\} = \{-9, -6, -3\}$. More information on training is found in Sec. 3.3. To improve receptive field when processing the features from the backbone, the window size w and the shift s are set to 20 and 10 with number of TV3S blocks N set to 4 (more information on these at Sec. 4.3) and the decoder made with mamba following its default parameters following [19] with the input embedding dimension of 256 matching the SegFormer implementation. The images from VSPW dataset [42] are cropped down to 480×480 and are augmented using various augmentation methods during training, including cropping, resizing, flipping and addition of photometric distortion during training. Optimization of the model is done with the use AdamW optimizer and a ”poly” learning rate schedule initializing the learning rate at $6e-5$. Testing is performed using the context of full video with the frame receiving the context of all the past frames within the video through the hidden states $\boldsymbol{H}$ due to its high efficiency & effectiveness following Sec. 3.4. Note that for all cases there was no use of post-processing on the obtained output like in [27].

Datasets. The experiments were mainly conducted with the use of VSPW dataset [42] which stands as one of the largest VSS benchmark. The dataset consists of training, validation and test subsets containing 2,806 clips (198,244 frames), 343 clips (24,502 frames) and 387 clips (28,887 frames) accordingly. The dataset consisting of a rich 124 categories with a dense annotation of frame rate of 15fps contrasts itself from the previously available datasets, which had very sparse annotation with just one frame being annotated every 10s of frames. Furthermore, the dataset cover-

Methods	Backbones	mIoU↑	GFLOPs↓	Params(M)↓	FPS↑
ETC† [37]	R18	71.1	434.1	–	–
SegFormer† [69]	MiT-B0	71.9	-	3.7	58.5
CFFM [57]	MiT-B0	74.0	80.7	4.6	15.8
MRCFA [58]	MiT-B0	72.8	77.5	4.2	16.6
SegFormer† [69]	MiT-B1	74.1	-	13.8	46.8
CFFM [57]	MiT-B1	75.1	158.7	15.4	11.7
MRCFA [58]	MiT-B1	75.1	145.0	14.9	13.0
TV3S (ours)	MiT-B1	**75.6**	83.6	17.3	25.1

Table 2. Quantitative comparison of our method with efficient alternative approaches on the Cityscapes dataset [9], using a resolution of 512 × 1024. (†Frame-by-Frame processing)

Models	Evaluation (mIoU)		
	Add	Concat	Direct
1 TSS (No Shift)	37.6	37.8	38.0
TV3S (No Shift)	38.1	38.4	38.9
TV3S (Shift)	**38.5**	**39.3**	**39.5**

Table 3. Evaluation on the implication of the shifted mechanism and the output method.

ing various different scenarios of both outdoor and indoor scenes makes it suitable for training models to well verify the adaptability of the performance standing as the best benchmark in the field of VSS. While most experiments and training were done with the VSPW dataset [42], the proposed method was also evaluated on the Cityscapes dataset [9] which annotates one frame out of every 30 frames for benchmarking the results.

Evaluation metrics. We use mean IoU (mIoU) and mean video consistency (mVC) as key metrics. The metric mVC evaluates the smoothness of predicted segmentation maps over time, assessing performance in the temporal domain. More formally, given a video clip $\{\boldsymbol{I}_c\}_{c=0}^{C_v}$ with ground truth masks $\{\boldsymbol{M}_c\}_{c=1}^{C_v}$ and predicted outputs $\{\boldsymbol{O}_c\}_{c=1}^{C_v}$, VC_n is computed as follows:

$$\mathrm{VC}_n = \frac{1}{C_v - n + 1} \sum_{i=1}^{C_v - n + 1} \frac{(\cap_i^{i+n-1} \boldsymbol{M}_i) \cap (\cap_i^{i+n-1} \boldsymbol{O}_i)}{\cap_i^{i+n-1} \boldsymbol{M}_i}, \tag{4}$$

where $\mathrm{C_v} \geq \mathrm{n}$. Once the VC_n of all the videos are computed, their mean is computed to obtain the $\mathrm{mVC_n}$. Eq. (4) shows that $\mathrm{mVC_n}$ finds the common areas of the predicted masks among frames which indicates the level of consistency of the prediction masks across time. More information on the metric can be found at [42].

4.2. Comparison with State-of-the-art Models

Our model is compared against state-of-the-art models on the VSPW dataset [42], as shown in Tab. 1. The table is divided into three groups based on model size and backbone, offering insights at varying scales. In the first group with small models (<30M parameters), our method outperforms the baselines, demonstrating efficiency even with limited model capacity. In the second group with larger models (>30M parameters), TV3S achieves near-state-of-the-art performance by effectively capturing rich contextual information. The third group focuses on the Swin Transformer backbone, where our model excels with over 8 mIoU ahead of the next best, highlighting its ability to preserve temporal correlations and ensure spatial accuracy. Overall, our method demonstrates superior visual consistency across all groups, further illustrated by Fig. 4 showcasing temporally consistent segmentation.

We also present results on the Cityscapes dataset [9] in Tab. 2, using smaller model variants at an input resolution of 1024 × 512. Due to the dataset's annotation structure, metrics such as $\mathrm{mVC_8}$ and $\mathrm{mVC_{16}}$ are not applicable, but our model still achieves top performance in mIoU, underscoring its strong generalizability across datasets.

Window sizes	mIoU	$\mathrm{mVC_8}$	$\mathrm{mVC_{16}}$
4	38.5	89.0	84.6
6	38.6	89.2	84.7
12	39.3	89.3	85.0
16	39.2	89.3	85.0
20	**40.0**	**90.7**	**87.0**
28	40.0	88.8	84.2
36	39.9	89.1	84.9

Table 4. Impact of the window size on model performance.

Backbones	TV3S Blocks	mIoU	$\mathrm{mVC_8}$	$\mathrm{mVC_{16}}$
MiT-B1	1	36.2	88.1	83.6
	2	37.4	88.6	84.3
	3	37.6	88.5	83.5
	4	**40.0**	**90.7**	**87.0**

Table 5. Performance metrics based on the number of TV3S blocks in the model.

4.3. Ablation Study

All ablation studies were conducted on the VSPW dataset [42] using the MiT-B1 backbone and following the same training and inference strategies as previously described.

Impact of shifted representations and feature integration. We evaluated the effectiveness of different TSS/TV3S decoder (see Tab. 3), including one TSS layer, unshifted TV3S, and shifted TV3S. Despite the only difference between unshifted and shifted TV3S being the feature representation, the results clearly show that shifted representation significantly outperforms its unshifted counterparts, highlighting their effectiveness. We further tested adding feature maps through addition and concatenation before the final prediction, but both methods reduced performance, suggesting that reintroducing spatial information after temporal modeling disrupts temporal coherence.

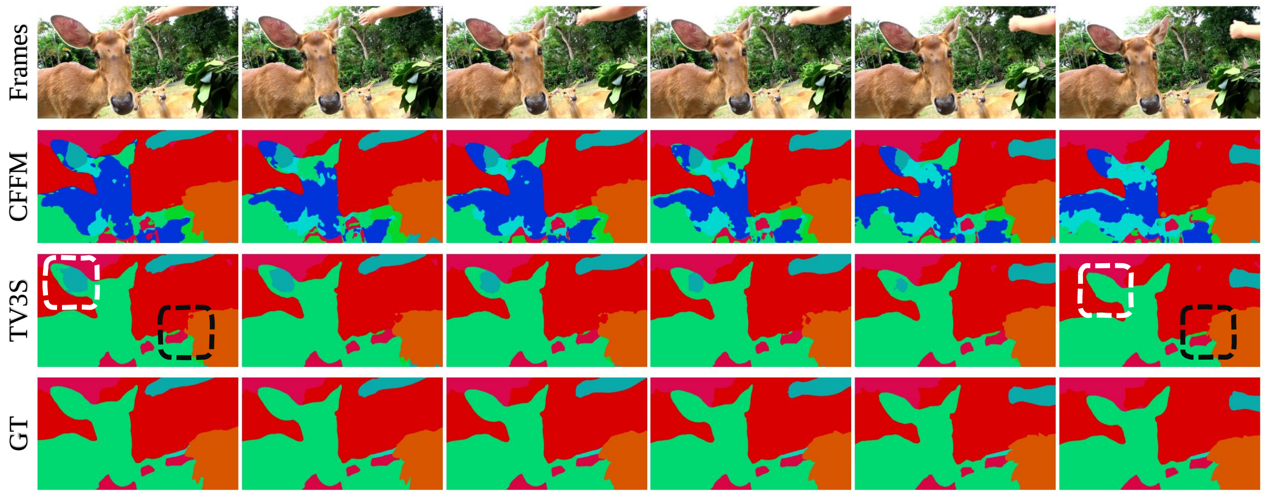

Figure 4. Qualitative example of our TV3S architecture compared with the current baseline. This displays both improved performance in performing spatial predictions and utilizing the temporal information to produce temporally consistent segmentation results.

Number of Frames	mIoU	mVC_8	mVC_{16}
1	37.9	84.5	79.1
2	39.2	86.9	81.8
4	39.5	88.9	84.2
8	39.7	90.4	85.9
16	39.7	91.2	87.2
32	**39.7**	**91.5**	**87.8**

Table 6. Performance based on the amount of temporal information available during inference

Impact of window size. TV3S establishes temporal relationships with window size controlling the spatial context during processing. Larger windows help summarize stable scenes but may overload the model in dynamic settings, while smaller windows enhance understanding in fast-changing environments but may miss correlations in slower scenes. As shown in Tab. 4, the optimal performance is achieved with a 20×20 window size, striking a balance between spatial context and model efficiency.

Effect on the number of TV3S blocks. In our experiments, we assessed the impact of varying the number of TV3S blocks, as shown in Tab. 5. Increasing the number of blocks improved mIoU from 36.18 with one block to 40.00 with four blocks, along with enhanced temporal consistency, as shown by the mVC_8 and mVC_{16} scores. This upward trend suggests that stacking TV3S blocks better aggregates temporal features, capturing complex relationships without introducing excessive redundancy.

4.4. Temporal Context

Importance of long temporal context. Given Mamba's RNN-like structure, we conducted an evaluation of its performance on the VSPW dataset [42] with varying temporal context sizes that ranged from 1 to 32 frames. In our approach, the videos that were shorter than 32 frames were excluded from the and at the same time the first 32 frames were omitted during inference to ensure a fair and unbiased evaluation of the model's capabilities. The findings presented in Tab. 6 indicate that there is a notable improvement in performance as the temporal context sizes increase. However, this enhancement reaches a plateau once the temporal information saturates. This saturation effect is likely attributed to the limited changes that can be captured within smaller temporal windows, which causes diminishing returns in performance improvements beyond a certain threshold.

5. Conclusion

The work proposes a TV3S architecture that addresses the challenges of VSS by capturing temporal dynamics. The proposed structure is designed to achieve good computational efficiency and scalability. By leveraging independent processing of spatial blocks through state space models enhanced with Mamba [19], our approach enables parallel computation during both training and inference. This design mitigates the time delay and high memory demand typically associated with sequential processing in traditional state space and recurrent models, making it highly suitable for long, high-resolution video sequences. Extensive experiments on the VSPW [42] and Cityscapes [9] datasets show that the TV3S architecture not only surpasses existing methods in segmentation accuracy but also significantly improves temporal prediction consistency.

Future works. Our model enhances the encoder's focus on key temporal features, integrating spatial and temporal information for improved video frame segmentation. Its adaptability extends to tasks like object detection and action recognition, while its multi-modal data integration offers new research opportunities in audio-visual learning, emphasizing temporal synchronization.

Acknowledgment

This work is supported by the Ministry of Education Singapore Tier 1 project funding grant No. MOE Tier 1 RG98/24 and by Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021).

References

[1] Zhaochong An, Guolei Sun, Zongwei Wu, Hao Tang, and Luc Van Gool. Temporal-aware hierarchical mask classification for video semantic segmentation. In *BMVC*, 2023. 3

[2] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys*, 51(4):1–41, 2018. 1

[3] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI*, 40(4): 834–848, 2017. 1, 2

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2, 6

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 1, 2, 6

[8] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. SegFlow: Joint learning for video object segmentation and optical flow. In *ICCV*, pages 686–695, 2017. 1

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 2, 7, 8

[10] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, pages 2393–2402, 2018. 2

[11] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. MeViS: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, pages 2694–2703, 2023. 1

[12] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV*, pages 20224–20234, 2023. 1

[13] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *AAAI*, pages 10713–10720, 2020. 1

[14] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[15] Ekrem Emre Yurdakul and Yucel Yemez. Semantic segmentation of rgbd videos with recurrent fully convolutional neural networks. In *ICCV workshops*, pages 367–374, 2017. 2

[16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 2

[17] Yuan Gao, Zilei Wang, Jiafan Zhuang, Yixin Zhang, and Junjie Li. Exploit domain-robust optical flow in domain adaptive video semantic segmentation. In *AAAI*, pages 641–649, 2023. 1

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[19] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3, 5, 6, 8

[20] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. In *NeurIPS*, pages 1474–1487, 2020. 3

[21] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2021. 3

[22] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *NeurIPS*, pages 572–585, 2021. 3

[23] Shunan Guo, Zhuochen Jin, David Gotz, Fan Du, Hongyuan Zha, and Nan Cao. Visual progression analysis of event sequence data. *IEEE TVCG*, 25(1):417–426, 2018. 1

[24] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2

[25] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, pages 8818–8827, 2020. 2

[26] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR*, pages 8866–8875, 2019. 3

[27] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NeurIPS*, pages 109–117, 2011. 6

[28] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, pages 3168–3175, 2016. 2

[29] Jiangwei Lao, Weixiang Hong, Xin Guo, Yingying Zhang, Jian Wang, Jingdong Chen, and Wei Chu. Simultaneously short-and long-term temporal modeling for semi-supervised video semantic segmentation. In *CVPR*, pages 14763–14772, 2023. 2

[30] Jiangyun Li, Yikai Zhao, Jun Fu, Jiajia Wu, and Jing Liu. Attention-guided network for semantic video segmentation. *IEEE Access*, 7:140680–140689, 2019.

[31] Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. Video semantic segmentation via sparse temporal transformer. In *ACM MM*, pages 59–68, 2021. 2, 3

[32] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. In *ECCV*, pages 237–255, 2025. 3

[33] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video K-Net: A simple, strong, and unified baseline for video segmentation. In *CVPR*, pages 18847–18857, 2022. 3

[34] Xiangtai Li, Haobo Yuan, Wenwei Zhang, Guangliang Cheng, Jiangmiao Pang, and Chen Change Loy. Tube-Link: A flexible cross tube framework for universal video segmentation. In *ICCV*, pages 13923–13933, 2023. 3

[35] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *CVPR*, pages 5997–6005, 2018. 3

[36] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017. 1

[37] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, pages 352–368, 2020. 6, 7

[38] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. VMamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024. 3, 4

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 1, 4, 5

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 2

[41] Jun Ma, Feifei Li, and Bo Wang. U-Mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024. 3, 4

[42] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. VSPW: A large-scale dataset for video scene parsing in the wild. In *CVPR*, pages 4133–4143, 2021. 2, 5, 6, 7, 8

[43] Seyed Shahabeddin Nabavi, Mrigank Rochan, and Yang Wang. Future semantic segmentation with convolutional lstm. *CoRR*, 2018. 2

[44] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *CVPR*, pages 6819–6828, 2018. 2

[45] Pedro O O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NeurIPS*, pages 1441–1447, 2015. 2

[46] Andreas Pfeuffer, Karina Schulz, and Klaus Dietmayer. Semantic segmentation of video sequences with convolutional LSTMs. In *IEEE Intelligent Vehicles Symposium*, pages 1441–1447, 2019. 2, 3

[47] Rui Qian, Weiyao Lin, John See, and Dian Li. Controllable augmentations for video representation learning. *Visual Intelligence*, 2(1):1, 2024. 1

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 1, 2

[49] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 1

[50] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J Black. Optical flow with semantic segmentation and localized layers. In *CVPR*, pages 3889–3898, 2016. 1

[51] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *ECCV*, pages 852–868, 2016. 3

[52] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, pages 802–810, 2015. 2, 3

[53] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *CVPR Workshop*, pages 587–597, 2018. 1

[54] Mennatullah Siam, Alex Kendall, and Martin Jagersand. Video class agnostic segmentation benchmark for autonomous driving. In *CVPR*, pages 2825–2834, 2021. 1

[55] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852, 2015. 2

[56] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 2

[57] Guolei Sun, Yun Liu, Henghui Ding, Thomas Probst, and Luc Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *CVPR*, pages 3126–3137, 2022. 1, 2, 3, 6, 7

[58] Guolei Sun, Yun Liu, Hao Tang, Ajad Chhatkuli, Le Zhang, and Luc Van Gool. Mining relations among cross-frame affinities for video semantic segmentation. In *ECCV*, pages 522–539, 2022. 3, 6, 7

[59] Guolei Sun, Yun Liu, Henghui Ding, Min Wu, and Luc Van Gool. Learning local and global temporal contexts for video semantic segmentation. *IEEE TPAMI*, 46(10):6919–6934, 2024. 2

[60] Yujin Tang, Peijie Dong, Zhenheng Tang, Xiaowen Chu, and Junwei Liang. VMRNN: Integrating vision mamba and

LSTM for efficient and accurate spatiotemporal forecasting. In *CVPR Workshop*, pages 5663–5673, 2024. 2

[61] Vassilios Tsakanikas and Tasos Dagiuklas. Video surveillance systems-current status and future trends. *Computers & Electrical Engineering*, 70:736–753, 2018. 1

[62] Sepehr Valipour, Mennatullah Siam, Martin Jagersand, and Nilanjan Ray. Recurrent fully convolutional networks for video segmentation. In *IEEE Winter Conference on Applications of Computer Vision*, pages 29–36, 2017. 2

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2, 3

[64] Bowen Wang, Liangzhi Li, Yuta Nakashima, Ryo Kawasaki, Hajime Nagahara, and Yasushi Yagi. Noisy-LSTM: Improving temporal awareness for video semantic segmentation. *IEEE Access*, 9:46810–46820, 2021. 2

[65] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *ICIP*, pages 2254–2258, 2021. 2

[66] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2

[67] Yuetian Weng, Mingfei Han, Haoyu He, Mingjie Li, Lina Yao, Xiaojun Chang, and Bohan Zhuang. Mask propagation for efficient video semantic segmentation. In *NeurIPS*, pages 7170–7183, 2024. 1, 3, 6

[68] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 6

[69] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, pages 12077–12090, 2021. 2, 4, 6, 7

[70] Chenliang Xu, Caiming Xiong, and Jason J Corso. Streaming hierarchical video segmentation. In *ECCV*, pages 626–639, 2012. 2

[71] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190, 2020. 6

[72] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 1, 2, 6

[73] 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 *CVPR*, pages 6881–6890, 2021. 2

[74] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 2, 3, 4

[75] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, pages 2349–2358, 2017. 3