

A UNIFIED MULTISCALE ENCODER-DECODER TRANSFORMER FOR VIDEO SEGMENTATION

Rezaul Karim

A Dissertation Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Graduate Program in
Electrical Engineering and Computer Science
York University
Toronto, Ontario, Canada

June 2024

Abstract

This dissertation presents an end-to-end trainable and unified multiscale encoder-decoder transformer for dense video estimation, with a focus on segmentation. We investigate this direction by exploring unified multiscale processing throughout the processing pipeline of feature encoding, context encoding and object decoding in an encoder-decoder model. Correspondingly, we present a Multiscale Encoder-Decoder Video Transformer (MED-VT) that uses multiscale representation throughout and employs an optional input beyond video (e.g., audio), when available, for multimodal processing (MED-VT++). Multiscale representation at both encoder and decoder yields three key benefits: (i) implicit extraction of spatiotemporal features at different levels of abstraction for capturing dynamics without reliance on additional preprocessing, such as computing object proposals or optical flow, (ii) temporal consistency at encoding and (iii) coarse-to-fine detection for high-level (e.g., object) semantics to guide precise localization at decoding. Moreover, we explore temporal consistency through a transductive learning scheme that exploits many-to-label propagation across time. To demonstrate the applicability of the approach, we provide empirical evaluation of MED-VT/MED-VT++ on three unimodal video segmentation tasks: (Automatic Video Object Segmentation (AVOS), actor-action segmentation, Video Semantic Segmentation (VSS)) and a multimodal task (Audio Visual Segmentation (AVS)). Results show that the proposed architecture outperforms alternative state-of-the-art approaches on multiple benchmarks using only video (and optional audio) as input, without reliance on additional preprocessing, such as object proposals or optical flow. We also document details of the model’s internal learned representations by presenting a detailed interpretability study, encompassing both quantitative and qualitative analyses.

*To my parents,
for their dedication and support.
And, to my family,
for their love and motivation.*

Acknowledgments

First and foremost, I want to extend my heartfelt gratitude to the Almighty Allah for granting me the strength and perseverance to complete my entire thesis without any interruptions.

I am immensely grateful to my supervisor, Dr. Richard P. Wildes, whose guidance and support have been invaluable throughout my Ph.D. journey. Without his initial encouragement and ongoing mentorship, I wouldn't have achieved what I have today. His unwavering belief in my abilities has been a driving force behind my academic and personal growth. I'll always cherish the moment when he recognized my potential as a researcher, inspiring me to aim higher and pursue my ambitions with confidence. I owe him a debt of gratitude that I can never fully repay.

I am immensely fortunate to have had a remarkable academic advisory committee consisting of Dr. Konstantinos G. Derpanis and Dr. Marcus Brubaker, whose dedication and unwavering support have significantly influenced my research career. Additionally, I extend my gratitude to my examination committee members, Dr. Laleh Seyyed-Kalantari, Dr. Faisal Qureshi, and Dr. Kohitij Kar, for their time and invaluable feedback on my dissertation.

My lab colleagues, Dr. Mennatullah Siam and Dr. He Zhao have greatly enriched and enhanced my thesis journey. Their camaraderie, insightful discussions, and unwavering support have consistently inspired me. To my dear friends, past and present, I extend heartfelt appreciation for their steadfast encouragement, making my Ph.D. journey truly rewarding and memorable. I'm also grateful to everyone who has supported me along this remarkable journey, as your belief in me has provided enduring motivation and strength.

Finally, I dedicate this work to my parents, Abdul Kadir and Rashida Akter, for their unwavering dedication to my education. I am deeply grateful to my wife, Tabassum Nijhum, for her continuous support, and to my son, Ayaan Muntasir, whose presence motivates my aspirations for the future. They are my driving force in overcoming challenges and persisting until the end.

Publications

Some of the ideas, materials and figures in this thesis have appeared previously in the following publications, pre-prints, and submitted manuscripts:

1. Karim, Rezaul, He Zhao, Richard P. Wildes, and Mennatullah Siam. "MED-VT: Multi-scale Encoder-Decoder Video Transformer with Application to Object Segmentation." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6323-6333. 2023.
2. Karim, Rezaul, and Richard P. Wildes. "Understanding Video Transformers for Segmentation: A Survey of Application and Interpretability." arXiv preprint arXiv:2310.12296 (2023).
3. Karim, Rezaul, He Zhao, Richard P. Wildes, and Mennatullah Siam. "A Unified Multi-scale Encoder-Decoder Transformer for Video Segmentation." arXiv preprint arXiv:2304.05930 (2024).

Contents

Abstract	ii
Dedication	iii
Acknowledgements	iv
Publications	v
Table of Contents	vi
List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	5
1.3 Dissertation outline	7
2 Related work	8
2.1 Video segmentation	8
2.1.1 Overview	8
2.1.2 Early approaches to video segmentation	9
2.2 Video segmentation tasks	11
2.2.1 Object segmentation	11
2.2.2 Actor-action segmentation	13
2.2.3 Scene segmentation	13
2.2.4 Multimodal segmentation	14
2.2.5 Tasks summary	15
2.3 Transformers for video segmentation	16
2.3.1 Encoder-decoder abstraction	16
2.3.2 Encoder	21
2.3.3 Decoder	27
2.3.4 Universal video segmentation transformers	35

2.3.5	Miscellaneous video transformers	36
2.4	Currently dominant video segmentation approaches	37
2.4.1	Objects	38
2.4.2	Actors and Actions	42
2.4.3	Scenes	42
2.4.4	Multimodal video segmentation	45
2.4.5	Training and testing efficiency	46
2.5	Video transformer interpretability	47
2.5.1	Internal representations in transformers	47
2.5.2	Attention interpretability	49
2.5.3	Understanding temporal modeling in video transformers	50
2.5.4	Inherently interpretable vision transformers	51
2.5.5	Interpretability for efficient transformer design	52
2.6	Summary	52
3	MED-VT Architecture	54
3.1	Preliminaries	54
3.1.1	Overview	54
3.1.2	Input format	55
3.1.3	Single-head attention	55
3.1.4	Multi-head attention	57
3.1.5	Self-attention	57
3.1.6	Cross-attention	58
3.2	MED-VT architecture components	59
3.2.1	Overview	60
3.2.2	Multiscale transformer encoder	60
3.2.3	Multiscale transformer decoder	61
3.2.4	Many-to-many temporal label propagation	64
3.3	Multimodal extension (MED-VT++)	67
3.4	Training and inference	70
3.5	Implementation details	71
3.6	Summary	71
4	Empirical evaluation	73
4.1	Automatic video object segmentation	74
4.1.1	Datasets and protocols	74
4.1.2	Task specific details	76

4.1.3	Results on MoCA	77
4.1.4	Results on DAVIS 2016	78
4.1.5	Results on YouTube-Objects	80
4.2	Actor-action segmentation	82
4.2.1	Datasets and protocols	82
4.2.2	Task specific details	83
4.2.3	Results on A2D	84
4.3	Video semantic segmentation	85
4.3.1	Datasets and protocols	85
4.3.2	Task specific details	86
4.3.3	Results on VSPW	86
4.4	Audio visual segmentation	87
4.4.1	Dataset and protocol	87
4.4.2	Task specific details	88
4.4.3	Results on AVSBench	89
4.5	Ablation study	90
4.6	Temporal properties	92
4.7	Training and inference efficiency	93
4.8	Summary	94
5	Interpreting MED-VT	95
5.1	Methodology	95
5.1.1	Quantitative analysis.	95
5.1.2	Qualitative analysis.	97
5.2	Component-wise results	98
5.2.1	Backbone	98
5.2.2	Encoder	100
5.2.3	Decoder	101
5.2.4	Label propagation	102
5.3	Summary	103
6	Conclusions	105
6.1	Summary and overall discussion	105
6.2	Societal impact	107
6.3	Future work	110
	Bibliography	113
	Appendices	146

List of Figures

1.1	Comparison of State-of-the-art Multiscale Video Transformers and Our Approach.	4
1.2	Multiscale Video Transformer (MED-VT, MED-VT++) on Different Unimodal and Multimodal Video Segmentation Tasks.	5
2.1	Overview of Video Segmentation Transformer Model Functional Components.	17
2.2	Pixel Decoder and Query Decoder in Pixel Level Estimation Model.	28
2.3	Recent Trends in Pixel Decoder Design.	30
3.1	Single Attention Head Mechanism.	55
3.2	Multi-head Attention Mechanism.	56
3.3	Self-attention Mechanism.	57
3.4	Cross-attention Mechanism.	58
3.5	Detailed (MED-VT) Architecture with Unified Multiscale Encoder-Decoder Transformer, Illustrated with Application to Automatic Video Object Segmentation (AVOS).	59
3.6	Decoder Stacked Coarse-to-fine Processing.	62
3.7	Detailed MED-VT++ Architecture for Multimodal Processing Illustrated with Application to Audio Visual Segmentation (AVS).	67
4.1	Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on MoCA Dataset.	78
4.2	Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on DAVIS 2016.	79
4.3	Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on YouTube Objects Dataset.	81

4.4	Qualitative Segmentation Results Comparing MED-VT to Baseline Algorithm on A2D Dataset.	84
4.5	Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on VSPW Dataset.	88
4.6	Qualitative Segmentation Results (red masks) Comparing MED-VT++ to Ground-truth and Baseline Algorithm on AVSBench Dataset.	91
4.7	Temporal Consistency Analysis Using IoU over Time on Dance-twirl Video of DAVIS'16 and Flounder Video of MoCA Dataset.	93
5.1	Example of Stylization Used for Interpretability Showing Stylized Frames from Both Stylized DAVIS 2016 and Stylized MoCA.	96
5.2	Layerwise Static vs. Dynamic Bias Analysis of MED-VT Compared to Our Single Scale Baseline on MoCA and DAVIS 2016 Datasets.	98
5.3	Unitwise Static vs. Dynamic Bias Analysis of MED-VT Compared to Our Single Scale Baseline on the MoCA and DAVIS 2016 datasets.	99
5.4	Activations (first principle component) at Different Backbone Layers as Heatmaps.	99
5.5	Activations (first principle component) at Different Encoder Layers as Heatmaps.	100
5.6	Activations (first principle component) Across Decoder Attention Heads as Heatmaps.	101
5.7	Receiver Operating Characteristic (ROC) Curve for the Label Propagator Encoding Comparing Three Scenarios.	103
5.8	Activations (first principle component) at Different Stages of Label Propagation as Heatmaps.	104

List of Tables

2.1	Summary of Video Segmentation Tasks.	16
2.2	Related approaches motivating MED-VT design choices.	53
4.1	Results of Moving Camouflaged Object Segmentation on MoCA Dataset. . .	77
4.2	Results on DAVIS 2016 Validation Set.	78
4.3	Results on YouTube-Objects.	80
4.4	State-of-the-art Comparisons for Actor-action Segmentation on A2D Dataset.	83
4.5	State-of-the-art Comparisons for Video Semantic Segmentation on VSPW Dataset.	87
4.6	State-of-the-art Comparison of Audio Visual Segmentation on AVSBench. . .	90
4.7	Multiscale Encoder, Decoder and Label Propagation Ablations.	91
4.8	Comparison of Many-to-One (Mto1) vs. Our Many-to-Many (MtoM) Label Propagation.	92
4.9	Analysis of Temporal Dynamics by Means of Dynamic Perturbations on Clips Using MoCA Dataset.	93
4.10	Comparison on Run Time and Memory Consumption.	94

Chapter 1

Introduction

1.1 Motivation

This dissertation focuses on multiscale video transformers for dense estimation. As a particular illustration of the developed approach, video segmentation is considered. In the following, we motivate the video transformer model, how it can be enhanced through multiscale processing, segmentation as an application area and the importance of interpretability of learning-based models.

Video transformers have shown tremendous ability to establish data relationships across space and time without the local biases inherent in convolutional and other similarly constrained approaches. This relationship is crucial for video understanding tasks as it helps in mining information regarding global or long-range context and temporal dynamics while maintaining local spatial dependency. The temporal information in video presents great potential to mine meaningful high-level information which is beyond the possibility of a single image. As examples: Motion allows segmentation of moving camouflaged objects in a scene, temporal order allows segmentation and recognition of actions such as putting down vs. picking up an object; long range temporal context allows segmentation of objects that are partially occluded or disappearing and reappearing in a scene. However, this great potential comes with great challenges including, but not limited to, the presence of motion blur, requiring temporal consistency as well as shape and scale variation of the same object as the viewing angle and distance varies when the object moves. Application of video transformers has shown tremen-

dous success in addressing these challenges as evident from the proposal of numerous transformer models for video segmentation, e.g., [80, 175, 153, 51, 124, 37, 35, 211, 103, 297, 96], as well as in other computer vision tasks [4, 54, 129, 140, 141].

Notably, video can inherently contain audio and contextualizing the visual information by fusing the audio using a multiscale cross-modal attention mechanism can help effectively leverage the additional modality of information. Representative work has investigated specialized architectures for multimodal feature fusion and query generation [15, 256, 155, 171, 293, 151, 59]. However, such efforts have not led to models that can process both unimodal and multimodal data in a unified video transformer framework.

Recently, multiscale processing has been applied with transformers to assist in the understanding of single images (e.g., classification [247], detection [298, 31] and panoptic segmentation [37]) and videos (e.g., action recognition [54, 129]). More generally, multiscale processing is an established technique across computer vision, with a long standing history, e.g., [22, 106]. Some representative examples in the era of convolutional networks include edge detection [261], image segmentation [193], object detection [133] as well as Automatic Video Object Segmentation (AVOS) [91, 295, 188]. Multiscale processing also has the potential to enhance the learning abilities of transformers through cross-scale learning [31, 37, 54, 129, 247]. The resulting multiscale features can combine the semantic richness available through deep-layer abstractions with fine localization available at shallower layers. Additionally, cross-attention between multiple modalities can yield complementary feature fusion in the encoder.

Multiscale transformer models use spatiotemporal attention on the features across various levels of abstraction, which can facilitate mining long-range relationships at multiple spatial granularities and levels of semantic abstraction. This ability supports extraction of highly robust features that can improve the temporal consistency of segmentation. Also, this long-range contextualization across scale-space helps in mining information regarding complex motions. For example, a moving car has a translational motion while the wheels are rotating, and these motions may be best captured at different levels of representation. A gap exists, however, as prior research efforts focused on multiscale processing in either encoding or decoding in video transformers, leaving the potential integration throughout the model largely unaddressed. Recent work has focused on multiscale transformer encoding, e.g., [54, 129], yet does not incorporate multiscale processing in the transformer decoder. Other work has

proposed multiscale transformer decoding, e.g., [37, 35], yet did not fully exploit multiscale attention at encoding.

In this thesis, multiscale transformers are explored in the context of video segmentation. Segmentation algorithms assign each pixel a label from a finite set of classes to delineate the pixel’s membership. This label set depends on the qualitative criteria that define an output space based on the application of interest. Based on the qualitative criteria, there are various categories of video segmentation tasks, e.g., object segmentation, actor-action segmentation and semantic segmentation [294]. This inference from colour space to label space supports analysis and application for downstream objectives of an artificial intelligence processor or a human observer. Video segmentation is important for reasons both theoretical and practical. From a theoretical perspective, the association of pixels according to common attributes (e.g., colour, texture, dynamics) to form meaningful segments (e.g., objects, motion, actions) is a fundamental problem in visual information processing. Video segmentation also is useful for various applications including video compression [65], object tracking [101], activity recognition [79], video surveillance [222], robotics [265] and autonomous driving [30]. More generally, video segmentation allows for the manipulation of each segment independently, tracking objects of interest, recognizing human activities, analyzing scene content and retrieving specific segments or objects from a video database based on their semantic content or visual appearance. Despite being very challenging, the theoretical interest and application areas have driven a continuing increase in research efforts on video segmentation.

Notably, the majority of all the above approaches have given very limited attention to supporting their architectures with interpretability studies. For scientific [160], application [272] and societal [21, 70] reasons, such advances will be most impactful and widely deployed if their operations are interpretable. Indeed, legal precedents are being established that will preclude deployment of learning-based technologies in certain domains unless they are interpretable, e.g., [45, 117]. In recent years, scientific and social requirements have led to increased research on the interpretability of deep networks [227]; however, much remains to be done, especially with regard to how video transformers capture the temporal dimension throughout their multilayer architectures.

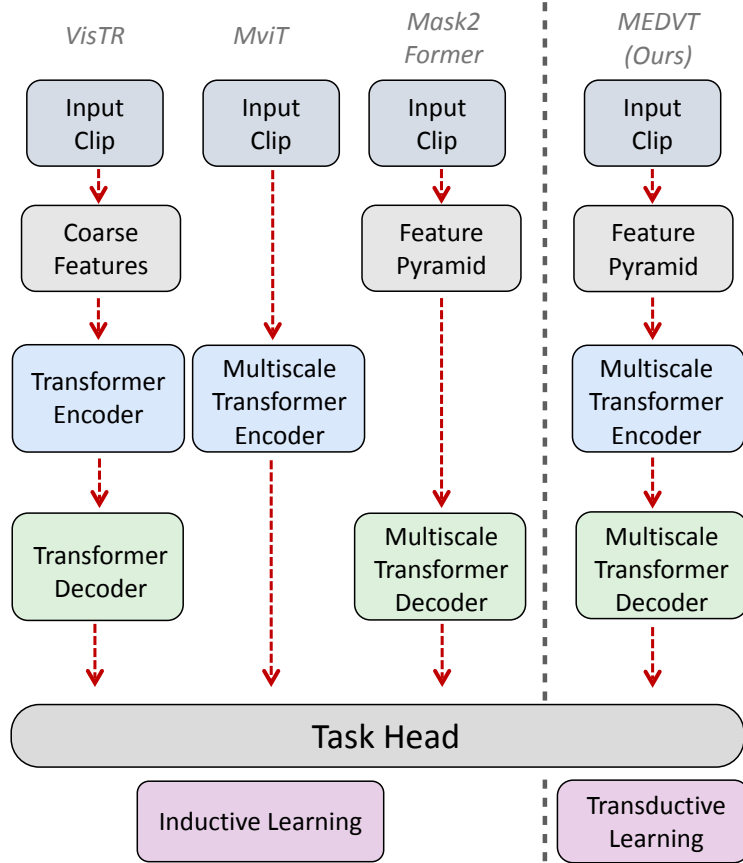


Figure 1.1: Comparison of State-of-the-art Multiscale Video Transformers and Our Approach. In particular, we show comparison with VisTR [251], MViT [54] and Mask2Former [35] by using schematic diagrams. Video transformers take an input clip and feed features to a transformer encoder-decoder, with alternative approaches using multiscale processing only in the encoder or decoder. We present a unified multiscale encoder-decoder video transformer (MED-VT), while predicting temporally consistent segmentations through transductive learning.

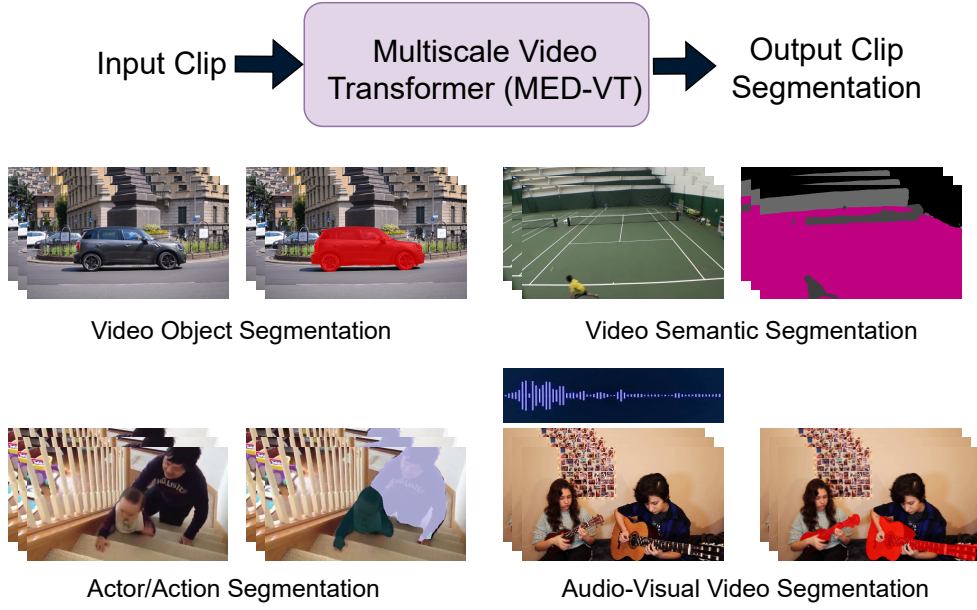


Figure 1.2: Multiscale Video Transformer (MED-VT, MED-VT++) on Different Unimodal and Multimodal Video Segmentation Tasks. For unimodal tasks, such as primary object segmentation, video semantic segmentation and actor-action segmentation, MED-VT takes an input clip and estimates target masks. For multimodal video segmentation, e.g., audio visual segmentation, MED-VT++ takes audio as additional input and uses it as a context cue in segmentation.

1.2 Contributions

The main objective of this dissertation is to advance the capability of video transformer models on dense video estimation tasks. For this purpose, we investigate three major sub-objectives: design and development of a unified multiscale encoder-decoder video transformer model for dense estimation, empirical evaluation of the design on several benchmark datasets across multiple video segmentation tasks and understanding of the model by interpretability analysis. The major contributions of this dissertation are enumerated below.

1. We present the first end-to-end Multiscale Encoder Decoder Video Transformer (MED-VT). Figure 1.1 overviews MED-VT compared to alternative multiscale video transformers. At encoding, its within and between-scale attention mechanisms allow it to capture both spatial, temporal and integrated spatiotemporal information. Notably, this information is extracted without reliance on optical flow, as only RGB frames are used for input. At decoding, it introduces learnable coarse-to-fine queries that allow for pre-

cise target delineation.

2. We present the first many-to-many label propagation scheme within a transductive learning paradigm to ensure temporally consistent estimates across an entire input clip. The label propagation provides a structured, temporally consistent prediction for a video clip using many-to-many propagation of the initial prediction from the task head. Overall, this extended operation enforces structured prediction across all frames in a clip by facilitating information sharing among different frames. This operation accommodates challenging scenarios where an object could be partially occluded, out of view, or some parts are not activated in the feature map due to other reasons in some frames, but they are activated well in other neighbouring frames.
3. We introduce an attention-based extension from unimodal to multimodal video segmentation (MED-VT++). The approach adds optional modules at the encoder and decoder to integrate multimodal features when available; no domain-specific modifications are made, which lends generality. In particular, we have implemented and evaluated the extended multimodal model by considering the case of audio; although, our approach to incorporating an auxiliary signal is not specific to that particular modality.
4. We document the wide applicability of the developed architecture via empirical evaluation on four video segmentation tasks: Automatic Video Object Segmentation (AVOS), actor-action segmentation, Video Semantic Segmentation (VSS) and Audio Visual Segmentation (AVS). Figure 1.2 overviews MED-VT/MED-VT++ as an architecture that applies to a wide range of video segmentation tasks. All of these tasks are important and challenging, as they are enablers of many subsequent visually-guided operations, e.g., autonomous driving and augmented reality. In all cases, we operate in the unsupervised inference setting, i.e., without using a mask in the first frame of a video to indicate the segmentation target, in contrast to the semi-supervised setting. These tasks share challenges common to any video segmentation task (e.g., target deformation and clutter). Notably, however, the requirement of complete automaticity imposes extra challenges, as there is no benefit of per video initialization. Lacking prior information, solutions must exploit appearance (e.g., colour, texture and shape) as well as motion to garner as much information as possible. MED-VT and MED-VT++ respond to these challenges. Their within and between-scale attention mechanisms capture both appearance and motion information as well as yield temporal consistency. Their learnable

coarse-to-fine queries allow semantically laden information at deeper layers to guide finer-scale features for precise object delineation. Their transductive learning through many-to-many label propagation ensures temporally consistent predictions.

5. We provide an extensive interpretability analysis, both quantitative and qualitative, highlighting the importance of multiscale integration in temporally consistent and robust video segmentation. To understand what has been learned by the multiscale encoder, decoder and label propagator, we apply both quantitative and qualitative analyses. The results verify that MED-VT learns to represent rich multiscale features with temporal consistency that underly its success across a range of segmentation tasks.

Code for the MED-VT/MED-VT++ model is available at rkyuca.github.io/medvt

1.3 Dissertation outline

This dissertation is organized in the following fashion. The first chapter has served to motivate and provide a broad overview of video transformers, multiscale processing and interpretability analysis in the context of video segmentation. Chapter 2 provides a detailed discussion of different categories of video segmentation, transformer-based architectures, interpretability of such models and state-of-the-art approaches for video segmentation. Chapter 3 provides a component-wise description of the developed MED-VT model for video segmentation. Chapter 4 provides results of empirically evaluating MED-VT on selected video segmentation tasks. Chapter 5 provides a detailed interpretability analysis of the presented model. Finally, Chapter 6 provides conclusions, including suggestions for future research directions.

Chapter 2

Related work

This chapter unfolds in five parts. The first provides an overview of video segmentation. The second presents the various video segmentation categories. Part three provides an overview of video transformer models, with an emphasis on their application to video segmentation. Part four provides a review of the state of the art in video segmentation. Finally, part five overviews interpretability studies of video transformers.

2.1 Video segmentation

2.1.1 Overview

Video segmentation is a fundamental computer vision task that divides each frame into regions based on specific qualitative objectives, e.g., delineation of objects, motion, actions and semantics [294]. Segmentation is of both theoretical and practical interest. Inferring coherent regions from temporal streams of images is a fundamental computer vision task. Moreover, segmentation supports a wide range of subsequent processing tasks, including surveillance [222], human-robot interaction [296] and autonomous driving [30].

Video segmentation shares many challenges with single image segmentation (e.g., photometric and geometric variation within a segment); however, additional challenges arise in the setting of temporal image streams, as segments need to be delineated and tracked or otherwise

associated across time. Attendant challenges include occlusion, motion blur, camera motion and object deformation. Moreover, operation with multiple video frames incurs additional computational expense compared to single frames.

Video segmentation also provides opportunities beyond those of single image segmentation. In particular, the availability of multiple images taken close together in time support the ability to recover image motion. Further, multiview image understanding is supported as a principled geometric approach to 3D inference. Moreover, the additional data provided by multiple images of (approximately) the same scene can be exploited to improve signal to noise ratio.

2.1.2 Early approaches to video segmentation

Given its theoretical and practical motivation, video segmentation has attracted much research effort for decades [294, 221, 287]. Although numerous categories of video segmentation have proliferated lately [294], early approaches primarily focused on the most basic video segmentation task of foreground-background separation. Early proposals for video segmentation are mainly analytic approaches that apply geometric constraints along with heuristics to design a solution. In addition to their historical importance, some analytic methods or components from those approaches are still relevant in the literature. The dominant analytical techniques of the early attempts fall into four categories: background subtraction, trajectory-based, object proposal-based and saliency-based. The remainder of this subsection overviews each of these approaches.

Background subtraction. Background subtraction approaches typically rely on heuristic assumptions about the background in the scene, e.g., pixels along frame boundaries are likely to be background. These approaches then attempt to detect moving objects by comparing pixel attributes to those of the hypothesized background. Connected component algorithms are used to estimate the connected region corresponding to an object or other distinctive region. Therefore, the above process is called background subtraction. Overall, video object segmentation is achieved by constructing a representation of the scene called the background model and then finding deviations from the model for each input frame.

There are various background subtraction methods that have been proposed in the lit-

erature. The variety of the proposed approaches primarily differ in terms of the heuristics and types of background model used e.g., stationary background [253, 208, 52, 68], background undergoing 2D parametric motion [189, 8] and background undergoing 3D motion [86, 224, 18]. They commonly use multi-class statistical models of colour pixels with a single Gaussian or a Gaussian Mixture Model (GMM). All of these methods rely on the restrictive assumption that the camera is stable and slowly moving; so, they are sensitive to model selection (2D or 3D) and cannot handle many challenging cases such as non-rigid objects and dynamic backgrounds. A detailed overview and comparison of background subtraction based video object segmentation is available elsewhere [18].

Point trajectories. Other early analytic approaches for video segmentation initially cluster point trajectories and use them as prior information to obtain the foreground object. The common motivation of these approaches is considering motion as a perceptual cue for segmenting a video into separate objects and point trajectories have potential to support analysis of motion information over long periods [202]. Interestingly, the notion of “common fate” for grouping points and objects can be traced to Gestalt psychology studies of perceptual organization [109]. Point trajectory-based methods generally use optical flow [17, 165, 32] or feature tracking [203, 164, 56] as a way of obtaining motion information. Optical flow allows the analysis of motion information between two consecutive frames using dense motion fields. Feature tracking allows object motion analysis over a longer range by tracking a sparse set of key feature points over a larger temporal extent of a video sequence. Methods with point trajectories benefit from explicitly exploiting motion information compared to background subtraction methods. Still, they face the challenge of grouping local motion estimates into coherent segments.

Object proposals. Object proposal based approaches generate likely regions (proposals) where objects appear based on colour, texture and motion. The proposals are then ranked based on confidence scores for subsequent aggregation across those that are highly ranked [120, 148, 284, 57, 107, 108]. These methods vary in the information used to generate object proposals, the object proposal generation methods and the method of ranking and aggregating those proposals. By definition, these approaches are limited by their particular definitions of objects. Indeed, they typically are not applicable to segmentation tasks where the regions of interest cannot be readily considered as objects, e.g., semantic segmentation where interest may include texture and colour defined regions.

Saliency based approaches. Other video segmentation approaches use saliency information as a prior or seed [184, 172, 53, 242, 226, 81]. Typically, these approaches compute local salience based on feature contrast, with features being measurements of colour, texture and motion. Subsequently, local salience is aggregated across space and/or time in various ways, e.g., using Conditional Random Fields (CRFs) [184], Non-Local Consensus voting (NLC) [53] or graph-based methods [81, 242]. Similar to object proposals, saliency approaches are limited by their definitions of salience and likely are inapplicable to cases where the regions of interest are not the most salient, e.g., small or low contrast regions.

2.2 Video segmentation tasks

2.2.1 Object segmentation

Object-based video segmentation aims to map groups of pixels in video frames to object masks based on qualitative criteria regarding object properties in the videos. Object-based segmentation approaches are mainly of two types: Video Object Segmentation (VOS), which only segments foreground objects and Video Instance Segmentation (VIS), which segments all object instances along with tracking across frames. Depending on the required level of prior information or human interaction, video object segmentation can be of three types: automatic VOS, semi-automatic VOS and interactive VOS. In the following, we discuss the problem formulations for each of these video object segmentation tasks.

Automatic video object segmentation. Automatic Video Object Segmentation (AVOS), also called primary object segmentation, is defined as grouping pixels into foreground object masks representing the primary object in a video sequence, where primary is distinguished as the object showing dominant motion or distinctive appearances across frames [277, 294]. For this reason, AVOS often has been termed foreground/background segmentation since an object with distinctive motion and/or appearance properties in a video sequence is generally the foreground object. Since no prior information is available, this kind of segmentation also is called unsupervised VOS or zero-shot VOS. In addition to the challenges common to most video segmentation task that were mentioned in Section 2.1.1, AVOS also faces the challenges of lack of prior information regarding the objects of interest.

Semi-automatic video object segmentation. Semi-automatic Video Object Segmentation (SVOS) is a mask-guided object segmentation where one or more annotated frames (commonly the first frame) are available as reference masks containing one or more objects as a target. The objective is to segment the target objects throughout the video. This task also is called semi-supervised VOS or one-shot VOS due to the availability of guidance or supervision. SVOS has its own challenges of mask propagation for target objects in video containing multiple objects, often from the same semantic category. Recently, a multimodal extension to this type of video object segmentation has become popular in the computer vision community where the reference is text or audio. We present a discussion of that variation in Section 2.2.4.

Interactive video object segmentation. Interactive Video Object Segmentation (IVOS) is semi-automatic object segmentation where there is an interaction between the system and a human providing hints to the object to be segmented (e.g., scribbles) [162, 6, 46, 290, 55]. This interaction happens in a loop with a series of interactions beginning with initial guidance and followed by feedback. Other forms of interaction include contours [146] and points [243]. IVOS allows for applications requiring high-quality segmentation at the trade-off of human supervision. This approach also is used to generate datasets for automatic or other semi-automatic video segmentation tasks. IVOS faces task specific challenges arise from the complexity of user interaction (e.g., target is given as a scribble, unlike a mask in semi-automatic VOS).

Video instance segmentation. Video instance segmentation (VIS) aims at the simultaneous detection, segmentation and tracking of instances in videos [269]. The problem also is referred to as Multi-Object Tracking and Segmentation (MOTS) [233]. The related task, video object detection and tracking, aims at detecting (bounding boxes) and tracking objects, while video instance segmentation additionally generates pixel-wise object segmentation masks. Again, in contrast to instance segmentation in images, which involves detection and segmentation [71], the requirement of tracking object instance masks across video frames makes it more challenging. In short, VIS simultaneously predicts object bounding boxes and object masks where the bounding boxes are labeled with semantic categories and segmentation masks are based on object identities across frames. The tracking of object masks makes it especially relevant for the application of video editing, autonomous driving and augmented reality [269].

2.2.2 Actor-action segmentation

Actor-action segmentation is defined as delineating regions from a given video with the corresponding label of an actor-action tuple [263]. Actors are objects that are participants in some actions. This task has been investigated far less intensively than object segmentation. Due to the exclusiveness of certain actor-action tuples, jointly segmenting actor-action has been found to result in better modeling than independently segmenting them in the context of action recognition. As examples, a car can not eat and a cat can not fly. The exclusiveness of these actor-action tuples makes it less likely for the model to learn to predict such tuples. Conversely, independent inference of actor and action can predict car as actor and climb as action simultaneously for a particular segment, even though the inference of actor-action tuple car-climb is unrealistic. This exclusiveness of actor-action tuples is evident from quantitative empirical results [263].

2.2.3 Scene segmentation

Scene segmentation aims to segment all the pixels in a scene with labels that assign them to a particular category or instance of a category. In this context, scene categories are defined either as “thing” or “stuff” [135]. In general, “things” are countable objects that typically have a well defined shape, whereas “stuff” is an uncountable entity with shape typically not being a defining attribute. For example: People and cars are countable objects, with shape being an important attribute; in contrast, sky and forest are not countable and shape is not important in their definition. Scene segmentation also is referred to as scene parsing in some contexts, since it aims to parse all the contents of a scene according to categories or instances.

Video semantic segmentation. Video Semantic Segmentation (VSS), also referred to as Video Scene Parsing (VSP) [267], aims to label all the pixels in a scene according to semantic categories of thing and stuff classes in an instance agnostic fashion [294]. Similar to VOS, VSS also can be automatic or semi-automatic. Automatic VSS segments all the frames without using any key-frame, while semi-automatic VSS uses annotated key-frames to segment other frames. Due to the ability of VSS to describe the high-level semantics of the physical environment, it can serve as upstream perception for many applications of autonomous systems, e.g., autonomous driving, robot navigation and human-machine interaction.

Video panoptic segmentation. Panoptic segmentation of images was first introduced as a holistic segmentation of all foreground instances and background regions in a scene [104], which was later extended to video [102]. Video Panoptic Segmentation (VPS) unifies video instance segmentation for thing classes and video semantic segmentation for stuff classes [102]. The difference between VPS with VSS is that in VSS things are segmented into semantic categories without tracking instances. Here, it also is worth noting that the difference between VPS and VIS is that the latter considers all stuff classes simply as background, while VPS distinguishes different classes of stuff. So, this is an instance-aware semantic segmentation with tracking of object IDs. Overall, VPS jointly provides estimates of object classes, bounding boxes, masks, instance ID tracking and semantic segmentation in video frames.

Depth-aware video panoptic segmentation. Depth-aware Video Panoptic Segmentation (DVPS) is defined as an inverse projection problem that restores the point clouds from a video and simultaneously provides each point with instance-level semantic interpretations [182]. This task also can be considered as an extension to VPS that jointly aims to provide estimates of panoptic segmentation and depth in a video.

Panoramic video panoptic segmentation. Panoramic Video Panoptic Segmentation (PVPS) extends VPS to panoramic videos generated simultaneously from multiple cameras. In addition to the multi-view setting, this task has another added challenge of tracking instance IDs across multiple cameras at each time. PVPS is an important requirement for autonomous driving, where there are multiple cameras on the vehicle to capture videos of the surrounding environment.

2.2.4 Multimodal segmentation

Multimodal Video Object Segmentation, also termed Referring Expression Object Segmentation, uses input from another modality (e.g., text or audio) to indicate the segmentation target. This formulation is similar to semi-supervised VOS, where reference is given as a mask annotation of the first frame. In contrast, in multimodal video segmentation, reference to the object is given by text or audio.

Text-guided video object segmentation. Text-Guided video object segmentation is defined as automatically separating foreground objects from the background in a video sequence

based on referring text expressions. Since text is the most common reference source for video object segmentation, it simply is called Referring Video Object Segmentation (RVOS). There are two types of annotations used for text-guided video object segmentation. The first is based on reference to an object description based on the first frame, which is very similar to semi-supervised VOS except the reference is text. This kind of reference text mostly describes the object appearance, e.g., ‘A man wearing a white cap’. The second type of referring text expression is based on the full video clip and also includes temporal dynamics of the object in the referring expression, e.g., ‘A person showing his skateboard skills on the road’.

Audio-guided video object segmentation. Audio-Guided Video Object Segmentation (AGVOS) is defined as automatically separating foreground objects from the background in a video sequence based on referring audio expressions comprised of spoken descriptions of target objects [171]. The authors of [171] refer to this task as AVOS, while we use AGVOS to avoid confusion with Automatic VOS. The motivation for audio-guided VOS is that audio often contains additional information compared to text, e.g., accent, emotion and speed.

In addition to segmentation based on speech-based referring audio expressions, an audio guided formulation known as Audio Visual Segmentation (AVS) [293] aims to segment objects as sound sources in the video. In contrast to the audio referring or audio guided video segmentation where the audio comes from a speaker asking or describing a target object, the audio of this setting comes from the scene captured in the video. The problem is to segment the objects that are producing the sounds.

2.2.5 Tasks summary

Table 2.1 provides a summary of the video segmentation tasks presented in this subsection, as well as a recapitulation of all their acronyms. Also indicated are current standard datasets used in evaluation.

Category	Task	Target	Instances	Tracking	Datasets
Objects	Automatic Video Object Segmentation (AVOS)	Primary moving object	-	-	DAVIS'2016 [178], MoCA [116], YouTube-VOS [266], YouTube-Objects [180]
	Semi-automatic VOS (SVOS)	Mask-guided object	-	-	DAVIS'2017 [179, 23]
	Interactive VOS (IVOS)	Scribble-guided object	-	-	DAVIS'2017 [179, 23]
	Video Instance Segmentation (VIS)	All Objects	✓	✓	YouTube-VIS [269], OVIS [181]
Actor-action	Actor-action segmentation	Primary Object related to actions	-	-	A2D [263]
Scene	Video Semantic Segmentation/ Video Scene Parsing (VSS/ VSP)	All thing and stuff classes	-	-	VIPER [190], VSPW [157]
	Video Panoptic Segmentation (VPS)	All thing and stuff classes	✓	✓	Cityscapes-VPS [102], VIPER [190], VIPSeg [156]
	Depth-aware Video Panoptic Segmentation (DVPS)	All thing and stuff classes and depth	✓	✓	Cityscapes-DVPS [182], SemanticKITTI-DVPS [182]
	Panoramic Video Panoptic Segmentation (PVPS)	All thing and stuff classes	✓	✓	WOD:PVPS [154]
Multimodal	Text guided VOS/Referring-VOS (RVOS)	Text reference guided object	-	-	A2D-Sentence [60], J-HMDB Sentence [60], Ref-DAVIS [100], RVOS [197]
	Audio Guided VOS (AGVOS)	Audio reference guided object	-	-	AVOS [171]
	Audio Visual Segmentation (AVS)	Sound producing objects	-	-	AVSBench [293]

Table 2.1: Summary of Video Segmentation Tasks.

2.3 Transformers for video segmentation

2.3.1 Encoder-decoder abstraction

The transformer model was originally proposed for machine translation as a fully attention-based deep neural network [231]. Here, attention refers to the model performing data associations between elements of its input, i.e., attending to those elements. Generally, two different types of attention are distinguished: self-attention and cross-attention. Self-attention establishes data relationships within a single input, while cross-attention works between two inputs. The model is a convolution free alternative to recurrent models for machine translation [147, 259]. The transformer model works on a set of tokens with the representation of each token as a feature vector ($N \times C$) where N is the length of the sets and C is the feature dimension. The proposed model processes the entire input set as a whole, in contrast to the sequential processing of recurrent models. Processing the entire input at once allows better long range context aggregation in transformers compared to ConvNets. Subsequently, the transformer model was adapted to computer vision in the Vision Transformer (ViT) model [50]. In this adaptation, ViT transformed an image into a set of tokens by dividing it into patches and representing each patch as a single token. Further adaptation of transformers to videos operates by making spatiotemporal patches [4]. Following the success of ViT, there is a surge of research in the computer vision community for various image and video under-

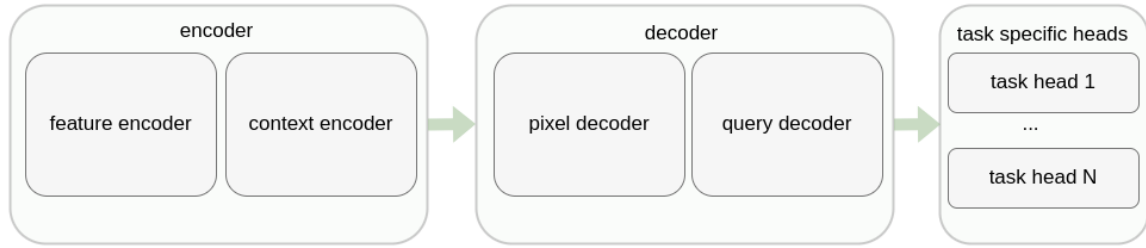


Figure 2.1: Overview of Video Segmentation Transformer Model Functional Components. The components can be broadly categorized into three functional types: encoding, decoding, and task specific specialization.

standing tasks, largely outperforming the previously dominant ConvNet-based approaches, e.g., [195, 228, 218].

2.3.1.1 Common components

The high level design of the functional components of a video segmentation transformer is presented in Figure 2.1. This diagram shows an abstraction of the encoder-decoder model found in recent transformer-based solutions, which also may be applied to tasks beyond segmentation. The model components are subdivided based on the functional purposes they serve. Since the models evolve over time through research efforts with incremental improvements, some are instantiated with a subset of these components. More recent models contain most of these components with diversity in design, typically driven by performance and/or efficiency concerns. Different segmentation tasks follow the general architecture, with task specialization embodied in a final task specific head. Typically, the task head itself is a learned network of variable design.

Despite the variety in segmentation tasks, video segmentation models typically use a common encoder-decoder approach. They begin by computing high resolution features with rich semantics from input image data and ultimately use a task-specific head to yield the desired segmentation. Rich semantics in this context means a representation containing high level information such as edges, contours, shapes and objects, in contrast to low level representation of pixel colour intensity. The encoder generates a semantically rich feature representation from the pixel space representation of the input video. The decoder generates high resolution semantic features and pixel-level masks. The final task head is a custom component for the

specific segmentation objective under consideration, e.g., semantic or instance segmentation.

The processing of an input video begins with a feature encoder, often referred as the feature backbone or extractor. The feature encoder takes input from a low dimensional colour space and maps it to a semantically richer high dimensional feature space. For this reason, the recent trend is to use models trained on coarse granularity tasks (e.g., classification) for the feature encoder and use transfer learning for application to other tasks of interest, e.g., [213, 74, 141]. The main reasons that models trained on classification tasks are good candidates for transfer learning with rich semantics are the availability of large-scale datasets for classification tasks and the manageable objective functions used in training that arise from the coarse granularity of classification. Recent video segmentation models generally use ConvNet models trained on ImageNet or transformer models trained on ImageNet [194] or Kinetics [25] as the backbone feature encoder, e.g., [213, 74, 141]. The context encoder, also known as the relation encoder, is designed to encode long range contextual relationships either spatially or spatiotemporally over the encoder features. This context encoder generally is equipped with a self-attention mechanism over a low resolution feature representation, e.g., [24, 251, 51]. Since attention is a permutation invariant operation, locality is induced on the features via the use of a positional embedding.

The spatial decoder (also called the pixel decoder) is developed for pixel level estimation tasks to generate high resolution features by propagating semantics from low resolution feature maps over a multiresolution feature space. Conventionally, this component is instantiated with the Feature Pyramid Network (FPN) [133] or its successors [104, 264, 215, 298]. The query decoder (also called the mask decoder) generates pixel-level mask proposals corresponding to salient features or object instances. The query decoder consists of self-attention, cross-attention and a feed-forward network.

The task specific head is the most heterogeneous component across different segmentation tasks. This component can be based on convolutional or linear layers. Depending on the task, it may also contain additional computation, e.g., Hungarian matching [251] for video instance segmentation.

2.3.1.2 Input preprocessing

Since the attention mechanism works on sets of tokens, the input video requires preprocessing steps to represent it as a set of tokens. A generalized discussion of preprocessing steps for video is presented elsewhere [195]. In this subsection, we provide details on the input preprocessing steps specific to video segmentation, some of which also may be applied to other tasks (e.g., tokenization essentially is needed for any application of transformers to raw pixel data to keep dimensionality manageable). In particular, for transformer models of video segmentation, the preprocessing steps involve tokenization, token-embedding and positional embedding.

Tokenization. In vision transformers, tokenization refers to dividing an input image or video into smaller units or patches called tokens [50]. All the tokens produced from an image or video are then serialized to produce a set that is transformed into token embeddings and then used as input to a transformer-based model. The patches are generally produced from non-overlapping spatial regions of various shapes for images, e.g., (8×8) or (16×16) . For videos, the patches can be spatiotemporal volumes for classification and spatial regions for pixel level estimation (due to the requirement of preserving the temporal dimension). Alternatively, in a hybrid model where transformers are used on the output of a ConvNet-based feature extractor, tokenization is done on the ConvNet features to produce a set of tokens from the feature maps. Since the ConvNet features are already in a lower spatial resolution, tokenization of these features may involve a small region, e.g., one pixel.

Token embedding. The token embedding transforms the spatial region or spatiotemporal volume of the token with multiple pixels to a single feature vector known as the token embedding. When using patch-embedding on high resolution data, such as RGB values or early layer features of a ConvNet, a spatial patch of size greater than a single pixel is used as the token. In that case, feature aggregation of all the elements under this window is used to transform it into a single feature vector. This operation is performed by a combination of simple averaging, concatenating the pixel features, convolution and linear layers. The convolutional or linear layer can additionally serve to transform the feature dimension to a specific dimension to be used in the subsequent transformer model. Alternatively, when using tokenization on the low resolution features of a ConvNet network, often a single pixel feature is transformed into a single token requiring no further processing for spatial dimension reduction. Even so, a

feature projection still can be used to transform the feature dimension of the token-embedding. The components involved here often are referred to as a token-embedding network.

Positional embedding. Positional embedding is used to augment the token embedding to capture the positional dependency of the tokens in the set. This addition is important because the attention mechanism is permutation invariant, while position can be important in many vision tasks, including segmentation. Different forms of position embedding are used in transformer-based models to encode the position of elements in an input set. The positional embedding can be fixed or learned depending on whether they are analytically defined or learned during the training step. Further, it can be absolute or relative, depending on how the position embedding is aggregated with the token-embedding. Details on the different forms of positional embeddings can be found elsewhere [50].

2.3.1.3 Output post processing

Some methods for video segmentation include post-processing steps, e.g., Conditional Random Fields (CRF) and object pruning. In particular, several methods for video object segmentation on Davis 2016 dataset used CRF [2, 295, 188, 92, 244, 145, 241, 289], while others used instance pruning [273, 289] or multiscale inference [289].

2.3.1.4 Transfer learning

Video segmentation aims to provide pixel-wise labels. Objective functions for learning at such fine granularity are difficult to train from scratch [229]. Moreover, generation of large-scale datasets that have been annotated with this level of granularity is not practical. For these reasons, video segmentation models use transfer learning from video-level or image-level granularity tasks. As one specific example, video object segmentation generally uses models trained on image or video classification tasks [194, 97] to initialize the feature extractor backbone. In particular, many video segmentation models use ConvNet models trained for ImageNet [194] classification as a backbone feature extractor, e.g., [213, 74]. These models use pre-trained weights from the classification tasks to initialize the segmentation model for transfer learning. VGG [206] and ResNet [74] models are the dominant backbones in those models. Recent video segmentation models with a transformer-based feature backbone use

transformer models trained on the Kinetics action classification dataset [97]. It also is found that first training the classification model for object detection (e.g., on COCO [135]) and subsequently using those trained weights to initialize the segmentation model leads to further improved learning [94, 95, 291, 240].

2.3.2 Encoder

Typically, the transformer encoder operates in two steps: feature encoding and context encoding. Feature encoding serves to extract an initial representation of the video from colour input that is suitable for delineating items of segmentation interest. The context encoding serves to capture relationships between the extracted features across both space and time and to do so over large extents to enrich the initial features. The remainder of this subsection describes these two steps in detail, with an emphasis video segmentation.

2.3.2.1 Feature encoder

In general, the first step of encoding in video segmentation transformers involves extracting high dimensional feature representations from the input data. A large number of transformer-based models for video segmentation (e.g., [176, 251, 36, 84, 270, 185, 236, 175, 80, 182, 35, 124, 182, 256, 171, 255, 103, 297, 280, 256, 171]) use a ConvNet (e.g., [206, 74, 42, 192, 282]) as feature extractor with a transformer in subsequent processing steps. Another trending approach (e.g., [271, 211, 103, 297, 280]) is to use an image transformer (e.g., [50, 260, 245, 140]) to extract features of individual frames and then aggregate to get video features. Most recently, models have appeared (e.g., [15, 256, 96]) that use spatiotemporal transformers (e.g., [141]) as their feature extractor. The input data to some video segmentation tasks includes information from other modalities in addition to images, e.g., text or audio for referring video object segmentation. In cases of multimodal segmentation, there is an additional feature extractor for the additional input modalities and the cross-modal features subsequently are aggregated. Recent models on multimodal video segmentation employ a transformer-based text encoder to extract text features [231, 48, 139] and Mel Frequency Cepstral Coefficients (MFCCs)-based models to extract audio features [16]. Yet another recent approach for audio visual sound source segmentation [293] used a ConvNet-based feature extractor for audio

input [78].

The ConvNet-based feature extractors follow a similar hierarchical architecture using convolution, rectification, normalization and pooling layers to generate a feature hierarchy of successively decreasing resolution and increasing semantics [206, 74, 42, 192, 282]. Different ConvNet models have their own characteristics in their internal architecture, e.g., VGG [206] presented an architecture using a moderately deep architecture with small filters, while ResNet [74] used residual connections to facilitate the training of a very deep network. These architectures have been widely in use as feature extractors for video segmentation due to their ability to generate hierarchical features from input images. A detailed discussion of these ConvNet models is available elsewhere [66].

Given the recent trend of using spatial transformers [50, 260, 245, 140] and spatiotemporal transformers [141] as backbone feature extractors in video segmentation, in the following we provide a discussion of transformer-based feature extractors. Early image transformers (e.g., ViT [50]) operate at a single scale, which limits applicability to segmentation where multiscale features are useful in distinguishing segmentation targets. Multiscale features are useful for segmentation as the distinguishing aspects of targets typically are manifest across a range of scales. In response, image transformers with multiscale features have been developed [44, 260, 245, 140], somewhat akin to multiscale ConvNet features. Notably, these features are captured on a framewise basis and thereby leave all temporal processing to subsequent stages.

Pyramid Vision Transformer (PVT) was the first model to include hierarchical multiresolution features in a transformer backbone [245]. PVT extracts features from images that are of successively decreased resolution. In particular, the PVT architecture is composed of four stages with progressive shrinking of output resolution by successively increasing the strides. Each stage is comprised of a patch embedding layer and several transformer encoders with multi-head self-attention. The resulting model benefits from both the hierarchical structure of ConvNets as well as the long range data association of transformers. Results show improvement over comparable ConvNets on object detection and semantic segmentation [245]. A subsequent extension also applied the features to video instance segmentation [271].

An efficient image transformer backbone used in video segmentation tasks is the Swin transformer [140]. The Swin transformer achieves improved efficiency by using a shifted

window mechanism in a hierarchical transformer backbone. The model is presented as a general purpose transformer backbone offering linear computational complexity in image size. The architecture involves four stages with patch merging in between stages. Each stage consists of alternating blocks with Windowed Multi-head Self-Attention (W-MSA) and Shifted-Windowed Multi-head Self-Attention (SW-MSA). The shifting window mechanism allows for cross-window attention. The model achieves linear complexity by limiting the self-attention locally within the non-overlapping windows of fixed size patches that partition an input spatially. The hierarchical feature representation is constructed by beginning with small-sized patches and gradually merging neighboring patches in deeper stages. The Swin transformer has been adopted in many recent video segmentation tasks [35, 255, 103, 297, 280, 256].

SegFormer introduced another notable hierarchical image transformer encoder named the Mix Transformer (MiT) encoder [260]. MiT is designed on top of PVT with a number of novel features, including overlapped patch merging and a positional-encoding-free design. For efficient self-attention, a sequence reduction process is adopted, following previous work [245]. The positional encoding free design allows MiT to not depend on interpolation of positional codes during testing with a different resolution input. Several variants also were presented using the same architecture with different sizes, named MiT-B0 to MiT-B5. The MiT encoder has been used as a feature extractor for transformer-based video semantic segmentation [211]. Recently, Hierarchical Feature Alignment Network (HFAN) [177] also used the MiT backbone for automatic video object segmentation.

Video transformers trained for video level granularity (e.g., action recognition [97]) have been used as the basis of feature extraction for segmentation. Early video transformer models trained on action classification used a transformer model on top of a 2D feature encoder backbone. Video Transformer Network (VTN) [163] used LongFormer [11] on top of ResNet [74] or DeiT [225], and augmented the features with an additional token for classification. Despite being efficient for long videos, VTN did not encode temporal information in the feature backbone. Video Vision Transformer (ViViT) [4] is a purely transformer-based model that encodes spatiotemporal tokens extracted from an input video using a transformer encoder consisting of a series of 3D spatiotemporal self-attention layers. To handle long input sets, a factorized spatiotemporal encoder was employed. Both VTN and ViViT models generate features at a single resolution making them less attractive for segmentation where fine localization from multiscale features is an important design principle.

The Multiscale Vision Transformer (MViT) is a transformer model that progressively grows the feature dimension with simultaneous reduction of the spatiotemporal resolution [54]. MViT achieves these hierarchical feature stages by using multi-head pooling attention blocks to reduce spacetime resolution at successive stages. In contrast to shrinking [245] or patch-merging [260], pooling operations on the (embedded) token sets are used to reduce the feature resolution (2D or 3D). Temporal dynamics modeling analysis by shuffling the frame orders show that this models presents strong temporal modeling capabilities [54]. MViT has been extended with improved embedding and pooling mechanisms [129] as well as via memory augmentation [254]; however, these versions have yet to be employed in video segmentation.

The Video Swin Transformer [141], an extension to the Swin Transformer [140], proposed a transformer model with shifted windows that combines the locality of ConvNets using windowing and the long range data association of transformers through attention mechanisms. The extension from the Swin transformer to the Video Swin Transformer is done by extending the scope of local attention computation from only the spatial domain to the spatiotemporal domain. The model is very fast due to the windowing mechanism. Further, the model allows bridging between the windows of two successive layers by means of the shifted windows. By associating information at window boundaries, the modeling power is enhanced, resulting in performance improvement on various tasks. Due to the benefits of both speed and performance in encoding video data, the Video Swin transformer is recently gaining popularity as a feature backbone in video segmentation tasks [15, 256].

Another method, the SIFA-Transformer [142], uses a form of deformable attention to rescale the offset estimates by the difference between two frames. Deformable attention uses dynamic sampling to adaptively generate a sparse attention matrix [298]. This method achieved superior results on video action recognition compared to MViT and Video Swin, suggesting a possible stronger feature encoding in the model [238]. While not yet applied to video segmentation tasks, the SIFA-Transformer has potential in this area.

2.3.2.2 Context encoder

The vast majority of transformer-based video segmentation approaches use ConvNets as their feature encoder and thereby can benefit from subsequent global context aggregation in the feature maps. Somewhat similarly, many transformer-based feature encoders use attention in a local window or pooling with large strides in attention. Moreover, in multimodal video

segmentation, there is a need for long range relationship aggregation on the combined features generated from multiple input modalities for multimodal video segmentation. To address these challenges, transformer-based models typically use a context encoder, also referred to as a relation encoder, equipped with global self-attention to capture long-range relationships between tokens. More generally, any feature encoder can achieve the added advantage of enriched features by global context encoding. The context encoder helps the model capture the spatial and temporal dependencies between elements in the video frames, allowing it to better understand the semantic content of the scene. This representation can lead to more accurate segmentation results, especially in complex scenes with multiple interacting objects.

The general architecture of the context encoder, as proposed in DETR for object detection [24], begins with feature dimension reduction and tokenization to become suitable for global self-attention processing. Next, the feature maps are tokenized to get a token embedding. Subsequently, a set of transformer encoder layers process the tokens. Each transformer encoder layer has a standard multi-head self-attention and a feed forward network (FFN). To incorporate position information in the permutation invariant transformer architecture, DETR added a fixed positional encoding [174, 9] with the tokens.

Approaches to video instance segmentation that have employed context encoding include the following. Video instance segmentation with Transformers (VisTR) [251] adapted the context encoder from DETR to model the similarities among all the pixel level features in a clip extracted by a ConvNet backbone. Inter-frame Communication transformers (IFC) [84] used additional tokens to store compact context information of each frame of a scene in memory. The inter-frame correlation is achieved by aggregating relationships between memory tokens. Specifically, two transformers are used to yield this encoder: one to encode the frame context summary and the other to exchange information between frames. Thus, the memory encoder works on each frame individually and the communication encoder works between the frames. The Sequential Transformer (SeqFormer) [255] used a transformer-based context encoder in each frame independently. SeqFormer argued for handling the 2D space domain and 1D time domain differently based on the view that the spatial and temporal domains are fundamentally distinct [288]. The framewise encoding employs deformable attention [298]. Subsequently, information is aggregated across all frames via region sampling. Ablation studies were presented to support the design of distinct spatial and temporal processing. A recent multiscale approach to video instance segmentation, MS-STC [220], uses multiscale spatiotemporal split

attention in its context encoder to capture spatiotemporal feature relationships across frames in a video. Specifically, it uses intra- and inter-scale temporal attention to generate enriched features during encoding. Additionally, it uses an adversarial loss to enhance foreground-background separability.

Various types of context encoder also are used in semi-supervised VOS. TransVOS [153] used a context encoder to model the spatiotemporal relationships among pixel-level features of the input token set. This approach used ConvNet features and adopted a context encoder architecture similar to DETR. Another approach to semi-supervised VOS, Sparse Spatiotemporal Transformers (SST) [51] used a context encoder with sparse-attention [40] for computational efficiency. The SST encoder used sparse multi-head attention, in contrast to global attention. In particular, a sparse connectivity between the key and value was used in the multi-head attention, based on the video spacetime coordinates.

Curiously, it appears that no approach to automatic VOS has employed a transformer-based context encoder. Probably the closest ones using the attention mechanism for AVOS are Hierarchical Feature Alignment Network (HFAN) [177] and Prototype Memory Network (PMN) [118]. HFAN used a mechanism similar to self-attention to fuse appearance and motion features at intermediate stages of encoding, while PMN used an attention mechanism to adaptively score and select the most useful superpixel-based component prototypes in each frame as candidates to a memory bank.

Finally, approaches have appeared that address multimodal segmentation using transformers. Multimodal Tracking Transformer (MTTR) for text Referring Video Object Segmentation (RVOS) presented an end-to-end transformer-based approach [15]. It employed a single multimodal transformer for context aggregation jointly across the two modalities. MTTR initially extracts features separately from the input text and video frames. Next, the features from the two modalities are flattened and concatenated to form a multimodal feature set. The context encoder then uses self-attention on the multimodal feature set to exchange information between the modalities. Another approach, ReferFormer, also used context encoding to fuse text and video features [256]. This approach adopted deformable transformer [298] as its transformer model for efficient global attention. The deformable transformer enhances the standard attention mechanism by allowing the model to selectively focus on different positions in the input with varying degrees of importance. By introducing learnable offsets, the deformable transformer adapts its attention window dynamically, capturing more fine-grained

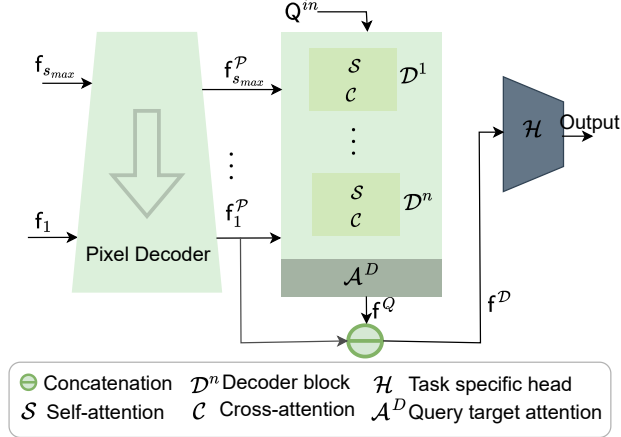
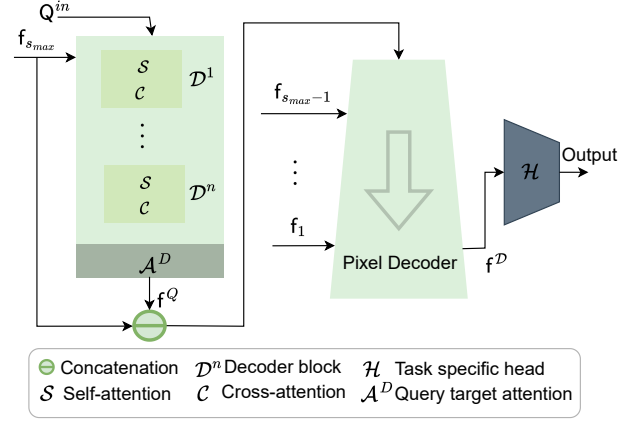
dependencies and improving performance on tasks requiring long-range context understanding. The approach projects multiscale visual features to lower dimension and multiplicatively combines the projection with text features. Finally, a context encoder operates on the fused features. Yet another multimodal video segmentation approach, Wnet, for audio-guided video object segmentation, used a transformer encoder to learn a cross-modal representation of the video and audio features [171]. In contrast to concatenating or multiplying, a wavelet-based cross-modal module is used to fuse the two modalities. In this context encoder, cross-attention is used with visual features as queries and audio features as keys and values. This cross-modal context encoder uses a wavelet based component in addition to the multi-head attention and feed forward network.

2.3.3 Decoder

The decoding phase of video segmentation tasks includes generating high resolution semantic features and mask proposals, followed by the final segmentation. Decoding typically operates in two steps, with the order of the steps depending on the design choices of the particular architecture considered. Some architectures use a query decoder initially to learn dynamic queries to represent a set of mask proposals, followed by a pixel decoder for generating high resolution semantic features. Other variants initially use a pixel decoder to produce multiresolution semantic features and then a query decoder to learn the dynamic queries from those features. In this section we will discuss various forms of spatial decoder and query decoder found in the literature. We present a brief visual overview of the general trends in the combination of pixel decoder and query decoder for video segmentation in Figure 2.2.

2.3.3.1 Pixel decoder

In the context of pixel level video understanding, we define a pixel decoder (also known as a spatial decoder) as a model component used to produce high resolution features with rich semantic abstraction. In general, a pixel decoder takes as input a set of multiscale features of different resolutions and different levels of abstraction. The multiscale features result from several intermediate stages of transformer encoding and may also be augmented with features from the backbone feature extractor. This multiscale feature set thereby contains features of successively lower spatial dimension, albeit higher in channel dimension and richer in semantics. In this regard, the pixel decoder is employed to propagate the rich semantics from deeper



(b) A pixel decoder is used before query decoding in multiscale decoding. Here, $f_1, \dots, f_{s_{max}}$ are the feature maps of successively deeper layers; $f_1^P, \dots, f_{s_{max}}^P$ are the multiscale feature maps from pixel decoder; $D^1, \dots, D^n$ are query decoder layers, each containing self-attention, $\mathcal{S}$, and cross-attention, $\mathcal{C}$; $\mathcal{A}^D$ is the query attention module that follows the query decoder; and f^Q is the output feature from the query attention module. In this type of architecture (e.g., Mask2Former [37, 35]) the output feature of the query attention module, f^q , and highest resolution feature from pixel decoder, f_1^P , are concatenated to produce decoder feature, f^D , which subsequently is used as input to a task specific head, $\mathcal{H}$.

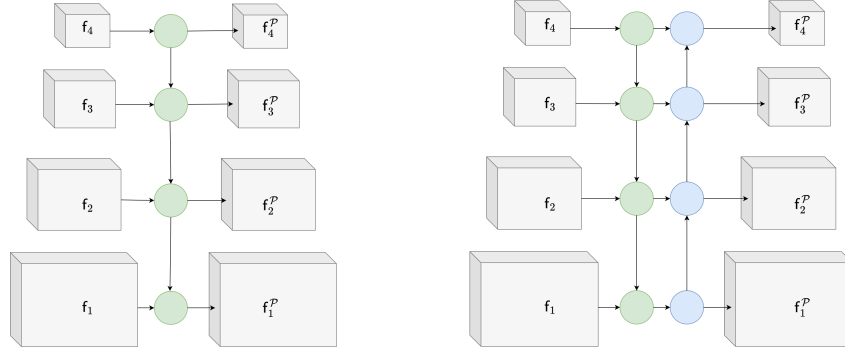
Figure 2.2: Pixel Decoder and Query Decoder in Pixel Level Estimation Model. In single scale designs, query decoding typically precedes pixel decoding (a); in multiscale designs, the opposite is true (b).

layer low resolution features to the early layer high resolution features. Thus, the multiscale feature output from the pixel decoder has features that all are rich in semantic information. Depending on the design choice, the subsequent modules of the pixel level estimation model can use any or all of the higher resolution features for task specific mask generation.

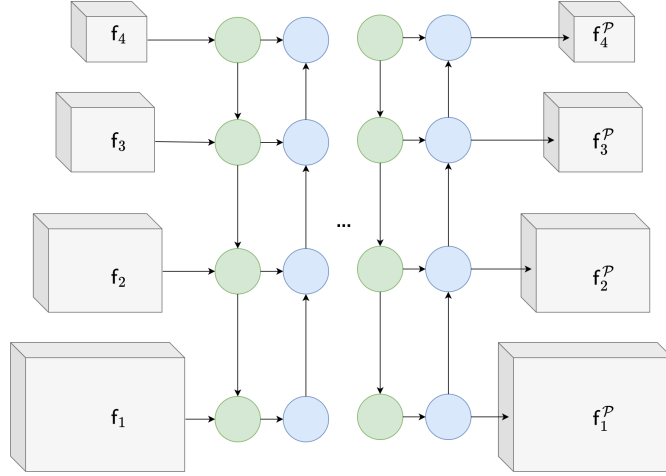
Pixel decoding addresses three important aspects of pixel level image/video understanding. First, small objects in a scene may have lost their detailed shape or appearance in the deep layer features due to successive striding operations. Moreover, in practical scenarios, objects can appear at various scales in an image or video frame. For example, the output features of a traditional ConvNet are generally 32 times smaller in spatial resolution compared to the input. A small object in a scene may have lost its original shape at this scale. Second, pixel level estimation requires precise localization at the granularity of a pixel. However, masks generated from low resolution features may lose precise localization when upsampled to the input size. Third, early layer features, while precise in localization, may lack in semantics (e.g., awareness of objects and context) to unambiguously detect or label on a local basis. The pixel decoder addresses these issues by producing high resolution features with semantics from deep layer features.

Recent developments in the design of pixel decoders are shown in Figure 2.3. A widely popular architecture for pixel decoding is the Feature Pyramid Network (FPN) [133], initially proposed for object detection. This architecture produces high-level semantic feature maps at all scales using lateral connections on the feature encoder in a top-down manner. The design operates by creating a feature pyramid based on hierarchical ConvNet features. Initially, each feature map of the pyramid is projected to a common embedding dimension (e.g., 256 channels). The feature embedding from the deepest layer is then up-sampled and added to the next higher resolution feature embedding. This feature embedding is then further up-sampled and added to the next highest resolution. This process is continued until the highest resolution feature embedding. Finally, all these new feature embeddings are input to common mask heads to generate estimates at multiple resolutions.

The Feature Pyramid Network was presented as a generic solution for building feature pyramids from the feature hierarchies of backbone feature extractors, e.g., ConvNets. The model was originally proposed for object detection and segmentation from images. For object detection, the model was adopted in the Region Proposal Network (RPN) [187] and Fast R-ConvNet [63]. Additionally, it also was demonstrated for segmentation proposal generation.



(a) FPN [133] with a single top-down pathway. (b) Panoptic-FPN [104] with a single top-down and an additional bottom-up pathway.



(c) BiFPN [215] with multiple alternative top-down and bottom-up pathways.

Figure 2.3: Recent Trends in Pixel Decoder Design. Shown are schematic diagrams of recent trends in the design of pixel decoder showing a) FPN [133], b) Panoptic-FPN [104] and c) BiFPN [215]. In all three diagrams, $f_1, \dots, f_4$ are four input features to the pixel decoder, $f_1^P, \dots, f_4^P$ are four output features from the pixel decoder, green circles are top-down pathways and blue circles are bottom-up pathways. Four features are shown for a concise depiction and comparison; the number of features can vary across different implementations. The circles of top-down and bottom-up pathways generally contain convolution, up-sample and arithmetic addition operations.

The original image based approach is now widely used for videos.

Panoptic-FPN [104] added a FPN to Mask R-CNN [73] for panoptic segmentation. Two task heads were employed, with one for instance segmentation to capture things and the other

for semantic segmentation to capture stuff. As part of the innovation, the architecture modified the original FPN by up-sampling all intermediate features to the highest resolution features. The resulting approach was found to perform better for semantic segmentation compared to Mask R-CNN. Several other approaches have followed this general path. PANet [137] adds an additional bottom-up pathway on top of a FPN for instance segmentation. PAN [123] added a global attention up-sample module to the FPN for the task of semantic segmentation. NAS-FPN [62] and Auto-FPN [264] presented an improved FPN architecture by using a Neural Architecture Search (NAS) approach. NAS is a method used to automatically search for optimal network architectures by exploring and evaluating a large space of possible architectures for a given task. EfficientDet [215] presented a weighted Bidirectional Feature Pyramid Network (BiFPN) for fast multiscale feature fusion. It extended the idea of PANet by adding repeated blocks of top-down and bottom-up feature aggregation. Additionally, it used a learnable weighting method for each feature during aggregation to emphasize their different contributions. The approach performed better on object detection and instance-level segmentation compared to using FPN. Another approach, FaPN, is designed to contextually align up-sampled higher-level features [83]. For this purpose, a feature selection module is used to emphasize deeper features with richer semantics. These features are upsampled and spatially registered to higher resolution features via a learnable alignment module. The approach is demonstrated across four pixel level tasks: object detection, semantic, instance and panoptic segmentation.

Recently, attention-based pixel decoding has been found to provide superior performance over ConvNet-based pixel decoding. Deformable-DETR [298] used an attention-based pixel decoder for object detection, which later was adopted for segmentation. This pixel decoder also is known as a deformable attention pixel decoder. The decoding combines the sparse spatial sampling of the deformable convolution operation and the relation modeling of the transformer to yield a multiscale deformable self-attention module. Subsequently, Mask2former [37] adopted this deformable attention pixel decoder for segmentation tasks. This multiscale attention module works on top of a multiscale feature pyramid and aggregates the transformer-based context encoder and FPN in a single module. Due to the quadratic complexity of self-attention, transformer-based context encoding is not practical for high resolution features from the feature encoder. The use of sparse sampling allowed the self-attention to be computationally feasible on high resolution feature maps. This pixel decoding approach was found to perform consistently better than other ConvNet-based pixel decoding approaches across var-

ious segmentation tasks.

2.3.3.2 Query decoder

The query decoder used in video segmentation takes as input the feature maps from a pixel decoder. This module also is known as mask decoder or bounding box decoder. The query decoder generates segmentation masks using successive iterations of self-attention and cross-attention mechanisms. The query decoder design is inspired by DETR [24], which used learned queries to formulate object detection as a set prediction problem. In particular, learned queries are used to capture the centre, height and width of bounding box estimates. Subsequently, the approach has been extended to pixel level segmentation for mask prediction [251, 37].

There are several benefits of using a query decoder for pixel level segmentation in images or videos. First, the query learned from a spacetime volume of video features can generate pixel level masks for the volume in an adaptive fashion. Second, the mask decoding approach can unify multiple segmentation tasks in a single framework [37]. Third, the resulting model is end-to-end trainable without requiring an additional proposal generation network for videos. Fourth, the model is able to capture long range relationships between video targets (e.g., objects).

For video, the query decoder takes two inputs: flattened spacetime feature volumes and a randomly initialized set of queries. The flattened spacetime feature volumes also are referred to as a key-value memory entries in this context. The query decoder processing steps include self-attention, cross-attention and a Multi-Layer Perceptron (MLP). Initially, self-attention is performed on the query input. Next, cross-attention is performed between query and spacetime features using the spacetime feature as key and value. Finally, the query is processed with a MLP. After multiple layers of such operations, the output queries operate on the memory features to generate attention maps. These attention maps can be used directly as object saliency or object binary masks for the segmentation. For video instance segmentation, an additional mask classification head is used along with the Hungarian algorithm [115] for bipartite matching between the sets of masks and classes [251].

Several approaches have emerged that follow the general paradigm that was sketched in

the previous paragraph. Video instance segmentation with Transformers (VisTR) presented a transformer-based query decoder aimed at decoding pixel features to represent object instances in each frame, which were termed instance level features [251]. Following a similar approach to DETR, it used instance queries learned with a transformer decoder operating on top layer features. These instance queries are then used with the top layer features to produce instance mask features at the pixel level. The instance mask features are concatenated with the top layer features and further processed with a pixel decoder to produce high resolution mask features by aggregating high resolution encoder features. Finally, the instance mask is generated from these instance features using a 3D ConvNet. Another approach, Inter-Frame Communication Transformers (IFC), used decoding instance masks with two modules: a spatial decoder and a transformer decoder [84]. The transformer decoder uses the encoder output to learn a fixed set of query embeddings corresponding to a set of objects. The number of queries is constant per clip, while in VisTR the number of queries grows linearly with the number of frames; this innovation enables more efficient decoding compared to VisTR. Two output heads work on top of the transformer decoder: a class head and a segmentation head. The segmentation head uses a FPN-like [133] pixel decoder to generate high resolution features before mask prediction. Another recent approach for video instance segmentation, SeqFormer, presented a query decompose transformer decoder that breaks the instance queries into frame-level box queries [255]. This query decomposition mechanism allows for restricting attention to focus on the same instance across frames. A fixed number of learnable instance queries for the same instance from each frame was employed, in contrast to the frame-level instances in VisTR. The decomposed transformer decoder learns one instance query for the whole clip and one box query for each frame. SeqFormer used deformable attention in its transformer decoder [298]. Since the deformable attention only attends to a small set of sampling points, this design results in efficient and less noisy attention. On top of the decoder, it used three task specific heads: mask head, box head and class head. Yet another transformer-based image and video segmentation approach, Mask2Former, has shown state-of-the art results on video instance segmentation by trivially generalizing the architecture for video segmentation [37, 35]. In contrast to other transformer decoding used for video instance segmentation, it learned instance queries from a set of multiscale features. Operations proceed through initial use of a FPN to enrich lower layer features with higher layer semantics, followed by a transformer decoder that learns instance queries from the enriched multiscale features. The decoder building blocks follow an architecture similar to VisTR, with the modification of alternating multiple

resolution feature sets in the decoder layers.

TransVOS [153] presented a transformer-based query decoder similar to VisTR [251] that also produces pixel mask features. It uses the query decoder to predict spatial positions of target objects in query frames for the task of semi-automatic (query frame mask guided) video object segmentation. A single object query is sufficient because semi-automatic video object segmentation aims to segment one given object in a query frame throughout the other frames in the video. Subsequently, a Target Attention Block (TAB) is used to get an attention map of the object at the query frame. This target object attention map is concatenated to the features of the frames to be segmented. A segmentation head is used to segment the target object in the current frame from these concatenated feature maps. Another approach for semi-automatic VOS, SST, used a generic convolutional segmentation network in the decoder that works on current frame features and a truncated history embedding instead of using a transformer decoder [51].

The multimodal transformer encoders that were introduced in the previous section [15, 171] also have corresponding decoders. Multimodal Tracking Transformer (MTTR) for text-referring video object segmentation used a transformer decoder on top of the multimodal encoder to learn object query embeddings [15]. Multiple object queries per input frame were used in this model with the decoder memory features from the encoder’s multimodal features. In this approach, corresponding queries of different frames share the same weights and each query represents an embedding specific to some instance. By using instance specific shared queries across frames, this decoder achieves natural tracking of object instances. For each of the instance query outputs from the transformer decoder, a feed forward network was used to generate conditional convolution kernels [223, 237]. Again, from the visual feature component of the multimodal output of the encoder, high resolution, semantically rich features were generated using a FPN-like [133] spatial decoder. Then, a sequence of segmentation masks is generated by convolving the generated convolution kernels with the corresponding frame features. Finally, a text association scoring is used to determine which of the object query sets has the strongest association with the object described in the given text. Another approach for the same problem, ReferFormer, uses a transformer decoder with object queries for the referring objects conditioned on the language expression [256]. This approach of using conditional object queries simplifies the segmentation by directly decoding only the referred objects. Hence, this approach does not require object and text association on all of the objects

in the scene. The transformer decoder architecture is similar to VisTR [251] with the difference of shared queries across frames. The use of shared queries allows support for videos of variable lengths. In addition, the text features are jointly fed to the decoder as input to make the query learning conditioned on the text features. Finally, three task heads were used to work on the instance embeddings and cross-modal segmentation features: class head, box head and mask head. A Cross-Modal Feature Pyramid Network (CM-FPN), adapted from the FPN [133], was used to generate high resolution cross-modal features. For this purpose, they added multi-head self-attention to the FPN. The mask head used the instance embeddings to generate dynamic convolution kernels [223] and convolve with the encoder features.

Multimodal video segmentation approach, Wnet [171] for audio-guided video object segmentation, uses a self-attention free transformer decoder to learn instance queries, following Fnet [121]. This paper argues that the joint representation of audio and video is more suitable for processing in the frequency domain. For this reason, it replaced the self-attention in the transformer decoder layer with a Fourier sublayer: A 2D Discrete Fourier Transform (DFT) is applied to the embedded input, with one dimension of the transform operating along the token dimension and the other operating along the hidden dimension. Subsequently, only the real part of the transform is used as instance queries to generate object segmentation masks.

2.3.4 Universal video segmentation transformers

Recently, video segmentation research has focused on designing universal segmentation models capable of handling multiple segmentation tasks in a single model. This trend is motivated by the evolution of various segmentation models toward having similar components with a difference in customized task heads and training objectives. These models typically use a Multi-Task Learning (MTL) approach [26, 27, 286, 230]. MTL in the context of deep learning aims to design a universal network architecture capable of learning generalization across multiple tasks. MTL has been gaining interest in the computer vision community to improve generalization, reduce latency and allow multiple tasks to benefit from sharing complementary information. The recent advancement in artificial intelligence using transformer-based models has led to increased interest in unified modeling approaches [88, 87, 110, 186]. In particular, there has been an increased interest in unified modeling of multiple video segmentation tasks.

The recently presented TubeFormer-DeepLab [103] is a pioneering effort aimed at providing a unified model for multiple video segmentation tasks. Specifically, a model is presented to solve video semantic segmentation, instance segmentation and panoptic segmentation with a possibility for extension to Depth-aware Video Panoptic Segmentation (DVPS). This approach formulated video segmentation as partitioning video frames into tubes with various semantic labels. In this formulation, multiple video segmentation tasks are generalized to tube labeling. The model architecture is developed by extending mask transformer [237]. The processing begins by generating a set of pairs of tube embeddings and class predictions. Next, the tube prediction is obtained from the tube embedding and pixel level video features. The generalization comes from the new formulation for directly predicting class labeled tubes. This universal architecture achieved better or competitive results across several video segmentation tasks on different benchmarks compared to specialized single or ensemble models.

Another recent approach, TarViS, presented a unified model for object detection and segmentation [5]. This model is trained jointly on multiple video segmentation tasks. The model is able to ‘hot-swap’ between tasks during inference by specifying a segmentation target without requiring task-specific retraining. The model is demonstrated on four different tasks: Video Instance Segmentation (VIS), Video Panoptic Segmentation (VPS), Video Object Segmentation (VOS) and Point Exemplar-guided Tracking (PET). In contrast to TubeFormer-DeepLab, TarViS used multiple sets of output queries instead of multiple task heads, following PerceiverIO [88, 87]. The main idea of TarViS is to encode a set of task specific targets as queries and then produce pixel level masks for each target using a transformer-based model. Swapping between the tasks is achieved by providing task specific targets during the inference. The segmentation masks are generated by the inner product of learned output queries and video features. In comparison to TubeFormer-DeepLab, TarViS reported competitive results on CityscapesVPS [102] and better results on VIPSeg dataset [156].

2.3.5 Miscellaneous video transformers

In addition to the above mentioned common model components and transformer-based models, we include some miscellaneous approaches in the following.

There have been several research efforts on improving the efficiency of video transformers. One notable approach, TimeSFormer, explored various methods of spacetime separable

attention [12]. In particular, TimeSFormer compares spacetime attention, divided spacetime attention and axial attention. In the context of video action recognition, the divided spacetime attention, where temporal attention and spatial attention are applied separately, was found to perform better compared to others. Another approach to computational overhead, Spacetime Mixing Attention, uses the full spatial region of video frames, but a temporal window that grows with layer depth, somewhat reminiscent of ConvNet design [20]. This approach allows for the computational complexity to be independent of the temporal length of the video, because temporal window size remains constant. Yet another approach, Temporally Efficient Vision Transformer (TeViT) [271] for video instance segmentation, uses a messenger token shift and spatiotemporal query interaction for efficient computation. The messenger token shift is a nearly parameter-free mechanism used for early temporal context fusion. TeViT obtains state-of-the-art results on VIS with an efficient computation model.

Other research on transformer-based video understanding models explored various methods for dynamically sampling attention maps, rather than using global attention. A prominent method in this direction is the deformable attention mechanism [298], which was introduced in Section 2.3.2.2. Another transformer-based approach to video captioning, SwinBERT [132], also uses adaptive sampling to learn a sparse attention mask. It used this learnable sparse attention mask as a regularizer to improve long-range video sequence modeling with less noisy attention. To achieve this goal, the model imposes a sparsity constraint overlaid on the attention masks of the visual features. The model yielded state-of-the-art results on multiple datasets for video captioning. Although not yet applied to video segmentation, this approach of sparse attention has the potential for adaptation to address computational issues for segmenting long duration videos.

2.4 Currently dominant video segmentation approaches

In this section we revisit the various video segmentation tasks defined in Section 2.2, now with an emphasis on the currently most popular and best performing transformer-based approaches for each.

2.4.1 Objects

Automatic video object segmentation. Dominant approaches to Automatic Video Object Segmentation (AVOS) rely on optical flow as an additional input along with colour images [295, 188, 205, 91]. These models generally use a two-stream architecture for extracting appearance features from colour images and motion features from optical flow. There are different feature-fusion approaches available in the literature to aggregate these features, e.g., the commonly used late fusion [91] vs. recently presented multistage fusion [295, 177]. In contrast, Treating Motion as Option TMO [41] used optical flow but reduced the dependency on flow features by treating it as optional information and fusing it to appearance features only during decoding, if it is available. Another family of approaches considers attention to capture recurring objects without requiring optical flow features in a video via simple mechanisms, e.g., co-attention [241, 145]. Among other recent approaches, Iterative Mask Prediction (IMP) applied a Semi-automatic Video Object Segmentation (SVOS) model to AVOS [119]. This approach used a Salient Object Detection (SOD) model to generate an initial pixel-wise estimate for all frames and then sampled the easy frames using a separately trained model. Finally, it propagated these sampled frame estimates to all other frames using an SVOS model. Memory based approaches also have been in use for AVOS. One such recent approach is Prototype Memory Network (PMN) that scores and samples superpixel based aggregated features as memories to be used in subsequent frames [118]. Yet another recent approach, Isomorous Transformer (Isomer) [281], used two variants of transformer: Context-Sharing Transformer (CST) for global-shared contextual information and Semantic Gathering-Scattering Transformer (SGST) for semantic correlation modeling for automatic video object segmentation.

Semi-automatic video object segmentation. In general, Semi-automatic Video Object Segmentation (SVOS) approaches employ feature alignment or attention-guided mechanisms to propagate the mask of target objects (also called query objects in some contexts) from a given annotated reference frame to the entire video sequence. Matching-based approaches learn to match features of the target frames with the reference frame [82]. In some approaches, the matching also uses estimates from several previous frames to segment the current frame [166, 232, 274, 276]. Another research direction leverages a memory network to remember an embedding of previous predictions and uses a memory-guided attention mechanism to segment the current frame [144, 168, 198]. One such notable approach, Hierarchical Memory Matching Network (HMMN) uses a memory-based method with two

advanced memory read modules to perform memory reading at multiple scales and exploit temporal smoothness [199]. Conversely, Space-Time Correspondence Network (STCN) uses direct image-to-image correspondence for efficient and robust matching purposes [39]. Alternatively, Per-Clip Video Object Segmentation (PCVOS) presented a clip-wise mask propagation approach using an intra-clip refinement and a progressive memory matching module [173]. Recent approaches also use attention-based mechanisms for SVOS to propagate masks [51, 130, 275]. To improve further the performance of challenging multi-object scenarios, an attention-guided decoder also has been used [130]. Another line of research uses a long short-term transformer for constructing hierarchical matching and propagation to associate multiple targets into a high-dimensional embedding space for simultaneous multi-object segmentation [275].

Interactive video object segmentation. Recent Interactive Video Object Segmentation (IVOS) approaches typically work by following an interaction-propagation approach [76, 158, 77, 38]. Interaction-and-Propagation Networks (IPNet) introduced two internally connected networks jointly trained for user interaction processing and transferring the annotations to neighboring frames [167]. Another approach, the Memory Aggregation Networks (MANet), used integrated interaction and propagation in a single network to improve efficiency [158]. An effective memory aggregation mechanism was used in this method to record the informative knowledge from the previous interaction rounds. Later, Annotation Transfer Network (AT-Net), used an Annotation Network (A-Net) and a Transfer Network (T-Net) that alternate to generate the segmentation results [76]. The A-Net first generates a segmentation result from user scribbles on a frame and then the T-Net propagates the segmentation result bidirectionally to the other frames by employing global and local transfer modules. Yet another approach, Guided Interactive Segmentation (GIS), presented a reliability-based attention module and intersection-aware propagation module [77]. Conversely, Modular interactive Video Object Segmentation (MiVOS), decouples the interaction-propagation as interaction-to-mask and mask propagation, where an interaction module converts user interactions to an object mask and then the mask is temporally propagated [38]. For efficient temporal mask propagation, a spacetime memory model is employed in this method with a novel top-k filtering strategy to read the memory.

Video instance segmentation. Video Instance Segmentation (VIS) recently has been the subject of especially intensive investigation. Recent dominant approaches for VIS follow a

simultaneous segmentation and tracking approach using transformer-based attention mechanisms [251, 84, 255]. Crossover learning for Video Instance Segmentation (CrossVIS), used a crossover learning scheme for online VIS that localizes an instance in other frames from the instance feature of the current frame [270]. This approach also used an additional branch, called global balanced instance embedding, for a stable online instance association. Another approach, Video instance segmentation with TRansformers (VisTR), used a transformer-based encoder-decoder approach followed by multiple task-heads for detection, segmentation and matching-based tracking to work on an entire video sequence in a single inference [251]. This formulation allows for simple yet efficient video instance segmentation. Yet another approach, Inter-Frame Communication Transformers (IFC), used a concise memory token representation as an efficient encoding of the clip context [84]. The inter-frame tracking is achieved via an exchange of information between the memory tokens. Conversely, Sequence Mask R-CNN (Seq Mask R-CNN), presented a propose-reduce paradigm to generate prediction for a whole clip in a single step [131].

Recently, Efficient Video Instance Segmentation (EfficientVIS), presented a fully end-to-end solution that achieves whole video instance segmentation in a single pass without requiring additional data association and post-processing [258]. To address computational complexity, Temporally efficient Vision Transformer (TeViT), uses message passing based on frame-level temporal context and query-based instance-level temporal context to achieve strong temporal consistency on the VIS task [271]. Another approach addressing computational complexity, Sequential transFormer (SeqFormer) [255], used deformable attention [298] in the decoder to first locate the object bounding box queries in each frame and then aggregated to the instance queries to predict dynamic mask head parameters. The tracking is achieved by convolving the encoded feature maps using a mask head. Another recent approach, Tubelet Transformer DeepLab (TubeFormer-DeepLab) demonstrated a transformer-based unified model for multiple video segmentation tasks including video semantic, instance and panoptic segmentation [103]. Similarly, video Kernel Network (K-Net), also used a video panoptic segmentation model equipped with a dynamic kernel for video instance segmentation as a subtask [128].

Another research direction formulated the problem not to use annotation during training; in particular, FreeSOLO [249] presented a simple yet effective solution to learn instance segmentation following a contrastive learning approach [250]. Alternatively, motivated by online problem settings, In Defense of OnLine models (IDOL) is a transformer-based ap-

proach to VIS that used contrastive learning to learn more discriminative and robust instance embeddings for frame-to-frame association [257]. Another approach, Video Mask Transfomer (VMT), introduced a method to leverage high resolution temporal features for mask refinement across frames using an efficient video transformer [98]. Following similar motivation, Multi-Scale SpatioTemporal Split attention transformer (MS-STs) introduced a multiscale spatiotemporal split attention module to capture spatiotemporal feature relationships across multiple scales for video instance segmentation [220].

Very recently, Instance Motion for object-centric video segmentation (InstMove) introduced a method to leverage instance-level motion information by using a memory network that can be integrated into existing models to boost the performance of various video segmentation tasks, including video instance segmentation [136]. Alternatively, Generalized Video Instance Segmentation (GenVIS) introduced a method for unified video label assignment using a framework to sequentially associate clip-wise predictions [75]. Another approach, Context-Aware Relative Object Queries (CAROQ) introduced an approach to continuously propagate object queries frame-by-frame to track objects without post-processing [43]. Yet another approach in the open-world setting for segmenting unfamiliar objects, Cut-and-LEaRn (CutLER), used a cut-and-learn pipeline to discover objects [248]. Another unified video segmentation model, TarViS [5], employed transformer based queries for video instance segmentation. Conversely, Decoupled Video Instance Segmentation Framework (DVIS) [285] proposed a decoupling strategy for VIS by dividing it into three independent sub-tasks of segmentation, tracking and refinement. Another recent approach, Temporally Consistent Online Video Instance Segmentation (TCOVIS) [125], presented a novel online strategy for segmenting video instances, emphasizing the optimal use of temporal data within video clips. The approach prioritizes a global strategy for assigning instances and a module for enhancing spatiotemporal properties to improve the coherence of temporal features. Yet another recent approach, Consistent Training for Online Video Instance Segmentation (CTVIS) [279], presented a simple yet effective training strategy termed consistent training, which is based on aligning the training and inference pipelines to construct contrastive items. Another universal video segmentation approach, Tube-Link [127], also presented video instance segmentation along with several other video segmentation tasks following a temporal contrastive learning approach to build instance-wise discriminative features for tube level association.

2.4.2 Actors and Actions

Trending research on actor-action segmentation typically include 2D or 3D ConvNets as feature backbones and use optical flow as an extra input along with colour (RGB) frames to localize object motion as indicative of a probable actor [93, 47, 185]. Some approaches formulate the model as simultaneous multi-task learning by adding multiple branches of task heads on top of a feature backbone [93]. Specifically, separate prediction branches are used for mask segmentation, actor classification and action classification on top of fused features from RGB frames and optical flow inputs. Another notable approach uses a region of interest-based approach with region masks from an instance segmentation algorithm followed by non-maximum suppression and a classification head [47]. Recently, Single Shot Actor-Action Detection (SSA2D) introduced a simple yet effective proposal-free model using 3D convolutions to perform pixel-level joint actor-action detection in a single shot [185].

2.4.3 Scenes

Video semantic segmentation. Most common approaches in Video Semantic Segmentation (VSS), also referred to as Video Scene Parsing (VSP), follow propagation based segmentation from a key-frame (semi-automatic) or previous frames (automatic) to other frames. Typically, these approaches rely on optical flow based warping to propagate their labels [58, 138, 49, 89]. However, trending research instead uses inter-frame attention [176, 236, 124] to capture the temporal relations between consecutive frames without depending on optical flow. For example, Temporal Memory Attention Network (TMANet) adaptively integrates the long-range temporal relations over a video sequence with a temporal memory attention module to capture the relation between the current frame and a memory generated from several past frames [236].

To further address efficiency, Sparse Temporal Transformer (STT) used sparse inter-frame cross-attention on a query and key selection strategy to reduce the time complexity of attentional processing [124]. Another approach, Temporal Context Blending (TCB), introduced a model to effectively harness long-range contextual information across video frames using a spatiotemporal pyramid pooling mechanism [157]. Alternatively, SegFormer, presented a hierarchically structured transformer encoder, named Mix Transformer (MiT), that operates

without using positional embedding coupled with a simple decoder [260]. Another approach, video Kernel Network (K-Net), used a video panoptic segmentation model equipped with a dynamic kernel for video semantic segmentation as a subtask [128]. As yet another approach, Coarse-to-Fine Feature Mining (CFFM) used inter-frame cross-attention with coarse features on past frames for efficient cross-attention [211]. An interesting unified segmentation model, Tubelet Transformer DeepLab (TubeFormer-DeepLab), also demonstrated its performance on video semantic segmentation [103]. Recently, another universal video segmentation approach, Tube-Link [127], also presented video semantic segmentation following a temporal cross-tube matching approach.

Video panoptic segmentation. Recent efforts on Video Panoptic Segmentation (VPS) use multiple task head branches for detection, segmentation and tracking. The Video Panoptic Segmentation Network (VPSNet) initially aggregates warped features of past frames to the current frame, performs spatiotemporal attention followed by multi-task branches and then explicitly models cross-frame instance association specifically for tracking [102]. Another approach, Siamese model with Tracking head (SiamTrack), used a similar architecture and jointly learned coarse segment-level matching and fine pixel-level matching for temporal associations using a supervised contrastive learning approach [252]. Yet another approach, video Kernel Network (K-Net), introduced a content-aware dynamic kernel with a bipartite matching strategy [128]. Video inverse Projection DeepLab (ViP-DeepLab) for depth-aware video panoptic segmentation jointly performs video panoptic segmentation and monocular depth estimation as two subtasks of the main problem using a unified model [182]. This model achieved state-of-the-art results on the task of video panoptic segmentation as the first subtask of depth-aware video panoptic segmentation. Another unified multitask model, Tubelet Transformer DeepLab (TubeFormer-DeepLab), presented a mask transformer-based approach to partition a video clip into class-labeled tubes that yielded a general solution to Video Semantic Segmentation (VSS), Video Instance Segmentation (VIS) and Video Panoptic Segmentation (VPS) [103].

Yet another approach, panoptic Slots for VPS (Slot-VPS), presented an end-to-end framework with an object-centric representation learning approach for video panoptic segmentation [297]. It encoded all panoptic entities in a video with a unified representation, called panoptic slots, that capture their visual appearance and spatial information across time. It used a video panoptic retriever to retrieve and encode coherent spatiotemporal object information

into the panoptic slots. The retriever is a three-stage pipeline that includes slot assignment, slot prediction and slot refinement. The approach used three prediction heads to produce classification, mask and object ID from the panoptic slots. A recent video instance segmentation approach, Context-Aware Relative Object Queries (CAROQ), also demonstrated the effectiveness of object query propagation for panoptic video segmentation [43].

Depth-aware video panoptic segmentation. Recent efforts on Depth-aware Video Panoptic Segmentation (DVPS) mostly extended non-depth aware panoptic segmentation approaches with an extra depth estimation head. One notable approach, ViP-DeepLab was the first presented model for DVPS that used an additional prediction head for monocular depth estimation [182]. Subsequently, TubeFormer-DeepLab presented a unified model to combine multiple video segmentation tasks in a single model, which thereby went beyond just DVPS [103]. The approach begins by partitioning a video clip into tubes with class prediction and tube embedding vector pairs that later are used to generate a tube prediction. For this purpose, hierarchical structure is employed, with two types of dual-path transformer blocks using message passing between video features and a latent memory or a global memory. Yet another recent approach, PolyphonicFormer presented a transformer-based query learning approach for DVPS that unifies both panoptic segmentation and depth estimation [280]. This unified modeling enables interaction between panoptic context and depth information allowing them each to benefit from the other. Overall, the model has three components. First, a feature extractor generates features for panoptic segmentation and depth estimation. Second, a polyphonic head learns the queries for these tasks. Third, a tracking head is used to learn the feature embedding among the frames. Another recent approach, Bi-directional Guidance Learning (BGL) [72], presented a framework that integrates depth information into panoptic segmentation by simultaneously conducting segmentation and depth estimation for each segment using shared object queries. It employs a geometric query enhancement method and bi-directional guidance learning to enhance the relationship between the tasks.

Panoramic video panoptic segmentation. Currently, a single approach to Panoramic Video Panoptic Segmentation (PVPS) has appeared [154]. It operates by augmenting the ViP-DeepLab video panoptic segmentation algorithm to accept multi-camera input [182]. For this purpose, it trains separate models for each camera view and an additional model on a panorama generated from all views. To track instance IDs across camera views, a similar method to that used in ViP-DeepLab to track instance IDs across time is followed. This instance ID

tracking method is based on comparing IoU between region pairs, e.g., two masks with high IoU overlap are re-assigned with the same instance ID.

2.4.4 Multimodal video segmentation

Text-guided video object segmentation. Early efforts on Text Guided VOS (TGVOS), also known as Referring-VOS (RVOS), generally follow a two stage pipeline: First, a referring bounding box is tracked across a video; second, the box is segmented to indicate the object of interest on a pixel-wise basis [99]. A notable recent approach, Unified Referring Video Object Segmentation (URVOS), presented an efficient and unified model to initially segment some masks guided by the referring expressions, followed by a propagation based semi-supervised VOS to produce the final results [197]. Another approach, You Only inFer Once (YOFO), presented a multiscale cross-modal feature mining structure and an augmented memory based meta-transfer mechanism that learns target specific features [122]. Recently, transformer-based models have been introduced for RVOS that use an encoder-decoder architecture. One such approach, Multimodal Tracking Transformer (MTTR), initially extracts features from the text and video for subsequent concatenated multimodal feature sequences [15]. Then, a multimodal transformer encoder is used as feature relation encoder followed by a transformer decoder to decode instance-level queries. Finally, corresponding mask and reference labels are generated. Another transformer-based approach, ReferFormer, presented a simple and unified end-to-end transformer-based framework that uses object queries conditioned on language expressions [256]. The tracking of objects is achieved by linking the object queries across frames. Another recent work, Hybrid Temporal-scale Multimodal Learning (HTML) [69] employs a multiscale strategy to synchronize language and visual features for recognizing crucial object meanings in videos. Conversely, Temporal Collection and Distribution (TempCD) [216], presented a referent-guided dynamic temporal decoding for text-referring video object segmentation. For this purpose, a global referent token and a sequence of object queries are used to capture language guided video-level referent and high quality object localization, respectively. Yet another approach, Spectrum-guided Multigranularity (SgMg) [155], proposed segmenting encoded features and leveraging visual details to enhance mask optimization. Additionally, this approach used a Spectrum-guided Cross-modal Fusion (SCF) to foster effective multimodal representation by enabling global interactions within frames in the spectral domain. This approach was extended to multi-object R-VOS, facilitating simultaneous

segmentation of multiple referred objects in a video.

Audio-guided video object segmentation. Currently, it appears that there is only a single effort that has considered Audio Guided VOS (AGVOS), where the audio is of a spoken description of the segmentation target. In particular, Wavelet-based cross-modal denoising network (Wnet), used a cross-modal denoising approach to clean the audio signal and a wavelet-based encoder network to learn the cross-modal representations of the target video and query audio [171]. Overall, the model is composed of five modules: visual encoder, audio encoder, transformer encoder, transformer decoder and segmentation module. The visual encoder is a ConvNet (ResNet-50 [74]) that is used to extract features from the input video clip. The audio encoder uses a pretrained acoustic feature extractor network [16], followed by convolution. Following feature extraction, a wavelet-based transformer encoder is used to encode the cross-modal features. Next, a transformer decoder is used to learn object instance queries. Finally, the queries are used together with the cross-modal features to segment instances with a segmentation head.

Audio visual sound source segmentation. In addition to the above audio-guided video object segmentation based on spoken descriptions, there is another formulation of Audio Visual Segmentation (AVS) aiming to segment sound sources in video frames from audio. Similar to the audio guided video segmentation, AVS is also critically under explored. AVS-Bench [293] presented a baseline approach that employs a module facilitating pixel-level audio visual interaction over time, integrating audio cues to guide the visual segmentation. Another recent approach, Explicit Conditional Multimodal Variational Auto-Encoder (ECM-VAE) [152], addressed AVS by explicitly modeling the contribution of each modality while emphasizing the importance of cross-modal shared representation learning.

2.4.5 Training and testing efficiency

Currently, computational efficiency is not consistently reported for video segmentation models. Indeed, it is unusual for anything other than inference speed to be discussed and even that is rare. Furthermore, the computational platforms on which efficiency is reported varies; although, in all cases contemporary middle to high end GPU based processors are employed. In the following, a few examples are given to provide an indication of the current state of the art regarding inference efficiency in video segmentation. For video object segmentation, amor-

tized inference time per frame is 0.9s for MATNet [295], 1.3s for COSNet [145] and 1.9s for RTNet [188]. Among VIS methods, CrossVIS [270] reported processing more than 35 Frames Per Second (FPS) in YouTube-VIS-2019 [269] while, also introducing a faster variant of the model able to process over 48 FPS with a slightly reduced accuracy. EfficientVIS [258] reported achieving inference efficiency of 32-36 FPS. In contrast, TeViT [271] reported an inference efficiency of over 68 FPS on the same dataset. Among the most recent approaches to VIS, CAROQ [43] reported an inference speed of 20 FPS for their best model using the Swin-L [140] backbone and 40 FPS when using the ResNet-50 [74] backbone. For the VSS task, SegFormer [260] reported over 2 FPS inference speed on the Cityscapes [102] dataset for their highest accuracy variant, while the efficient variant can achieve over 47 FPS speed with slightly lower accuracy. For the VPS task, VPSNet [102] has roughly 1 FPS using a ResNet-50 [74] backbone, while Slot-VPS [297] reported an inference speed of 4.2 and 2.0 FPS when using ResNet-50 [74] and Swin-L [140] backbones, respectively. Alternatively, SiamTrack [252] reported an inference speed of 5.1 FPS using ResNet-50 [74], but the accuracy is lower than Slot-VPS [297]. Among the models for the RVOS task, YOFO [122] reported an inference speed of 10 FPS per object.

2.5 Video transformer interpretability

In this section we present a summary of notable results on transformer interpretability. Included are both model agnostic methods and attention specific methods.

2.5.1 Internal representations in transformers

Filter visualization. Visualization of learned filter kernels has been used for analyzing deep networks. Visualization of early filters in ConvNets as grey-scale or colour images has shown that they largely resemble oriented bandpass filters, e.g., [283, 149, 114, 67]. Theoretical reasons for the emergence of such filters also has been provided in terms of the correspondence between convolution and linear shift invariant systems [67]. In contrast, visualization of the linear embedding layer of transformers [50] suggests that while oriented structure is observed, it has more global support compared to ConvNets, which is consistent with the relative ability of transformers to perform data association across their entire input domain. Also, beyond

unimodal orientation, transformers appear to implement more complicated data associations.

Attention distance. ViT used attention distance between tokens on various heads of the multi-head attention layer as an analogue to the receptive field in ConvNets [50]. In particular, attention weights were used to compute the average distance in image space across which information is integrated. A comparison was presented showing attention distance of ViT and a hybrid model with ConvNet+ViT. It was evident in their findings that the attention distance is large for both the pure transformer model and the hybrid model. This evidence shows that in both of the cases deeper layer attention can integrate information across a long distance in image space. In contrast, the early attention layers for both of the models has attention spreading from short to long distance. Again, the early attention heads for the pure transformer model has highest spread of attention distance, encompassing short to long. This result suggests that some attention heads in the early layers learn localized attention similar to the smaller receptive fields of ConvNets. The overall results motivate the use of pure transformer models, without relying on ConvNets to provide localized support.

Optimization-based feature visualization. Similar to its use with ConvNet models, gradient descent based optimization has been used to explore the inner workings of transformer models. One particular example is an optimization-based feature visualization that analyses several transformer models [61]. The approach initially generated synthetic images that maximize the activation of a particular model component and subsequently identifies natural images that have the same behaviour. It was found that the feed forward layer features with high-dimensionality produce more interpretable visualizations than the low-dimensional features from self-attention. This work conducted visualizations for a variety of vision transformer models. The results show that early layers of vision transformer models also learn to recognize local features, e.g., colour, edges and texture, while deeper layers learn semantic abstraction similar to ConvNets. However, different from ConvNets, vision transformers are found to use background information in addition to object features and show resilience to dependence on texture.

Canonical correlation analysis. Canonical Correlation Analysis (CCA) is a statistical method to analyze the relationship between two sets of variables that can enable comparison of learned representations of deep neural networks, including model layers at different stages of training epochs or intermediate layer features for pairs of inputs. CCA based methods have been employed for quantitative comparisons of representations within and across three

networks [183, 161, 111]. The experiments reveal several key differences between the representation learning mechanism of the two currently dominant classes of models, ConvNets and transformers. One of the key findings is that vision transformers are able to learn global features at early layers in contrast to ConvNets, which is to be expected based on the relative early layer support of transformers vs. ConvNets. Another finding indicates that vision transformers produce comparatively stronger intermediate features compared to ConvNets, making the vision transformer a better backbone for dense estimation tasks.

2.5.2 Attention interpretability

Attention rollout. ViT used attention maps from the output token to the input space to demonstrate that the model attends to image regions that are semantically relevant for classification [50]. In particular, attention rollout was used to project the attention map to the input space [1]. The approach involves averaging attention weights across all heads and then recursively multiplying the weight matrices of all layers for the mixing of attention across tokens through all layers. The attention maps generated by this method are shown to largely correspond to the semantically important input image regions for making the desired classifications. However, this approach does not provide insights on the information captured in different heads at different attention layers.

Relevance propagation. Total Relevance Propagation (TRP) is a method for computing and propagating relevancy scores through all layers in transformer networks while maintaining the total relevancy across layers [29]. This method has been demonstrated on the vision transformer ViT [50]. The experiments include comparing classification outputs and interpretability saliency using segmentation metrics. The results suggest that the network focuses on the most salient components of the objects for classification. For instance, the model tends to focus on the abdomen to capture the stripes for the class zebra. In comparison, in classifying elephants, the focus tends to be on the head region. Evaluation with segmentation metrics considered the input relevance visualization as soft-segmentation of the image and compared it to the segmentation ground truth using standard evaluation metrics for segmentation. According to these metrics, TRP yielded saliency maps that are in better accord with the ground truth compared to rollout [1], GradCAM [196], Layerwise Relevance Propagation (LRP) [13] and partial LRP [234].

Generic attention-model explainability. The generic transformer interpretability method extends interpretation to attention layers beyond self-attention, encompassing bi-modal co-attention and cross-attention in encoder-decoder architectures, enabling explanation of predictions across various transformer-based models [28]. The approach was demonstrated on transformer models containing various categories of attention, e.g., self-attention, multimodal co-attention and cross-attention decoder. In particular, the self-attention vision-language model VisualBERT [126] and vision model ViT [50] were used to demonstrate the relevance score for self-attention. For VisualBERT, perturbing the tokens with high relevance shows dramatic drop in model accuracy indicating the practical significance of the generated relevance. The presented results using ViT showed superior performance in producing class-specific visualizations. In an experiment with multimodal attention, positive and negative perturbation tests were performed on each modality separately using the multimodal transformer LXMERT [214] for visual question answering [3]. Again, perturbing the tokens with high relevance shows a dramatic drop in model accuracy, indicating the practical significance of the generated relevance. This interpretability approach was demonstrated for encoder-decoder transformer models using an object detection model, DETR [24], trained on the MS COCO dataset [135]. In particular, Otsu’s threshold method [169] was used to produce segmentation masks from the saliency; subsequently, standard segmentation evaluation protocols were followed. The segmentation masks produced from input relevancy shows that the generated masks are consistent with the bounding boxes.

2.5.3 Understanding temporal modeling in video transformers

Several methodologies have been developed specifically for probing the ability of deep networks to encode the temporal dimension. Three of these are relevant here. The static vs. dynamic bias metric was proposed to measure how much static information (available in a single frame, e.g., colour, texture, etc.) vs. dynamic information (requiring temporal ordering of multiple frames, e.g., motion) is capitalized on by a model [112]. The method was demonstrated with experiments on several video models, revealing that state-of-the-art video models are mostly static-biased [54, 12]. Another approach presented the Appearance Free Dataset (AFD) to analyse how state-of-the-art video models perform when given a perturbed dataset containing no static information, which forces them to work solely with dynamics [85]. Their experiments reveal the limitation of many video action recognition models in capturing

temporal dynamics in the absence of appearance, including the transformer MViT [54]. Yet another recent approach, ATemporal Probe (ATP), introduced a frame selection method to enable comparison between the performance of video models on video input with the performance of image models on the selected frame as an image input [19]. The experiments show that many top video-language benchmarks are heavily biased towards static features, lacking temporal correspondence, resulting in image models using single-frame information performing similarly to video models.

2.5.4 Inherently interpretable vision transformers

The ongoing debate between post-hoc explainability vs. inherently interpretable models has resulted in recent proposals of inherently interpretable transformers. A notable work, ConceptTransformer (CT) presented a transformer-based module designed to be used as a classifier head in deep learning models [191]. The module generates classification outputs using cross-attention between input features and a set of embeddings representing high level interpretable concepts. This CT classifier is capable of enhancing the model with domain knowledge in the form of plausible cross-attention weights. The CT classifier module only adds an overhead of a modification to the loss function in the host model to enforce plausibility of the explanation during the training phase. Another work, introduced B-cos transformers as a holistically explainable model capable of inherently providing explanations for its decisions [14]. This formulation allows for summarizing the entire model with a single linear transformation. The generated summaries from the transformer model with dynamic linear layers are highly interpretable as the parameters are implicitly optimised to align with relevant input patterns. Experimental result of applying this approach to the ViT model shows that Bcos-ViTs can perform competitively to the baseline with the additional advantage of inherent explainability. The inherent linear summaries also are found to outperform state-of-the-art post-hoc explanation methods. However, this direction still requires further research efforts as there is no known inherently interpretable video model to date.

2.5.5 Interpretability for efficient transformer design

IA-RED² is an interpretability-aware redundancy reduction method for vision transformers to improve computational efficiency [170]. The paper argues that some input patches are redundant and such redundancy is input-dependant. On that basis, an interpretable module is introduced to dynamically drop several input patches with minimal correlation to reduce the redundant computation. A policy network is adopted by leveraging the idea of dynamic inference to identify and discard uninformative patches. This way of dynamically dropping some less informative patches in the original input reduces the length of the input sequence to yield computational efficiency without suffering accuracy. Further, by using the policy network to discriminate regional importance, this approach is inherently interpretable. Empirically, on both images and videos a notable speed up on state-of-the-art models was achieved with minimal drop in accuracy. In addition, qualitative examples were provided to verify the ability of the multi-head interpreter compared to raw-attention. Also, quantitative results were presented using weakly-supervised image segmentation on ImageNet [64] to quantify the superiority of the multi-head interpreter in identifying semantically important image regions. Although this method was not applied to video segmentation, it paves the way for similar work to follow in the video domain.

2.6 Summary

This chapter has provided an extensive survey of work related to the MED-VT architecture. This survey has included a taxonomy of video segmentation tasks, a presentation of the basic video transformer model, as well as a review of transformers for video segmentation and relevant interpretability methodologies.

Key aspects of the current state of the art in multiscale processing for video transformers that motivate our research are summarized in Table 2.2. In recent years, video understanding methods, from classification to segmentation, have widely adopted transformer-based encoder-decoder architectures, which have excelled in establishing data relationships across space and time. For instance, the VisTR [251] model for video instance segmentation exemplifies this success. Architectures like MViT [54] for action classification leverage multiscale transformers at the encoding stage to mine global or long-range contexts, while models such

as Mask2Former [35] employ multiscale decoding for powerful semantic disentangling. Typically, these methods use a shallow CNN model as a task head for per-pixel classification, lacking provisions for temporal consistency in video understanding. However, we observed a gap in recent literature regarding the full integration of multiscale architectures from encoding to decoding, despite the promise shown by individual multiscale approaches. Furthermore, temporal consistency and robustness in video understanding remain underexplored, and multimodal audio-video understanding has focused on specialized rather than generalized architectures [293]. Additionally, interpretability, crucial for model diagnostics and explainability, is underdeveloped in video segmentation tasks. We emphasize the need for integrated multiscale architectures, as multiscale representation enhances model learning and yields robust latent representations by taking advantage of both rich semantics in coarse scale and precise localization in fine scale, thereby improving the effectiveness of video understanding models.

Model	Publication	Tasks	Core architecture
DETR [24]	ECCV'2020	Detection and segmentation	A transformer-based end-to-end encoder-decoder architecture for object detection and panoptic segmentation.
VisTR [251]	CVPR'2021	Segmentation	A transformer-based encoder-decoder for video instance segmentation with potential extension to other video understanding tasks.
MViT [54]	ICCV'2021	Classification	A multiscale vision transformers using the fundamental concept of multiscale feature hierarchies with the transformer model during feature encoding.
CrossViT [31]	ICCV'2021	Classification	Application of cross-scale attention between two stream of different scale of input.
Mask2Former [35]	CVPR'2022	Segmentation	Multiscale decoding approach using masked attention for universal image segmentation with extension to video segmentation.
CrossFormer [247]	ICLR'2022	Classification, detection, segmentation	Application of cross-scale attention in feature encoding to blend multi-scale features and to split self-attention to reduce computation.
ABSBench [293]	ECCV 2022	Audio-visual segmentation	A specialized architecture providing a strong baseline for multimodal audio visual segmentation task using a module for encoding pixel-wise audiovisual interactions within temporal video sequences.

Table 2.2: Related approaches motivating MED-VT design choices.

Chapter 3

MED-VT Architecture

In this chapter we detail our Multiscale Encoder-Decoder Video Transformer (MED-VT). We begin with preliminaries that serve to formalize the transformer model that was introduced in Section 2.3.1. Next, we define the core MED-VT model, followed by its extension to the multimodal case. Finally, we provide implementation details.

3.1 Preliminaries

3.1.1 Overview

Transformer models are constructed on self-attention and cross-attention mechanisms. Intuitively, self-attention is designed to capture the relationship among the different elements of the same token set, while cross-attention is designed to capture the relationship among the elements of two different sets of tokens. The transformer model follows a general encoder-decoder architecture. A transformer encoder uses self-attention and linear layers as building blocks. While, transformer decoder building blocks consist of self-attention, cross-attention and linear layers. Both the self-attention and cross-attention mechanisms use a multi-head attention mechanism. In the following, we provide a brief preliminary discussion on multi-head attention, self-attention and cross-attention.

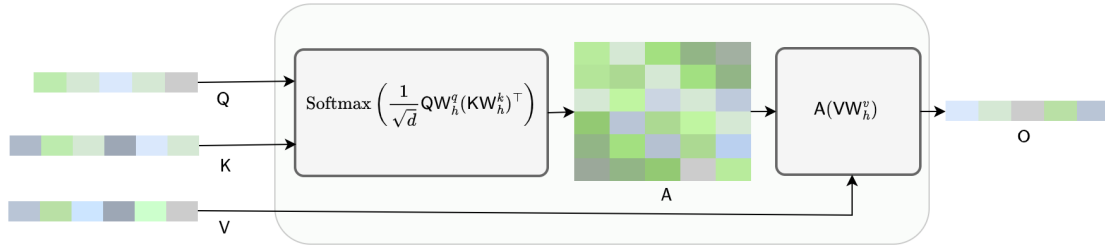


Figure 3.1: Single Attention Head Mechanism. Depicted is single attention head with query, Q, key, K, and value, V. The Q and K (with their associated weights, W_h^q , W_h^k) are first used to produce attention affinity matrix, A, and then the output, O, is computed from the attention affinity matrix and value set. The colours in the set distinguish different tokens and the colours in the attention affinity matrix distinguish the different attention affinity scores between different pairs of tokens and do not correspond to any particular numerical values.

3.1.2 Input format

Inputs to the attention function are represented as sets of tokens, also referred as vectors or sequences in some literature, especially in regards to natural language processing. A token is defined as a discrete unit of input with a numerical representation treated as a whole during processing. For example, a token can be a word, a sub-word or a character in language models and it can be a small spatial region or spatiotemporal volume in a vision transformer. As explained in Section 2.3.2.1, image patch tokens are embedded via linear layers to produce features for transformer input. For a single input, X , the set of N tokens with C features has the form $X^{N \times C}$. For batched data with B elements in the batch, the token set has the form $X^{B \times N \times C}$.

3.1.3 Single-head attention

An attention function takes three sets of tokens as input and produces a set of tokens as its output, O. The three inputs are called query, key and value. The terms query, key and value are used in analogy to their use in an information retrieval system, where data is retrieved based on an input “query”, e.g., a text description, which is compared to a set of “keys”, e.g., a set of video titles to produce “values”, e.g., similarity scores, which are used to rank the videos with respect to the text. For transformers, the query, Q, key, K, and value, V, are matrices whose

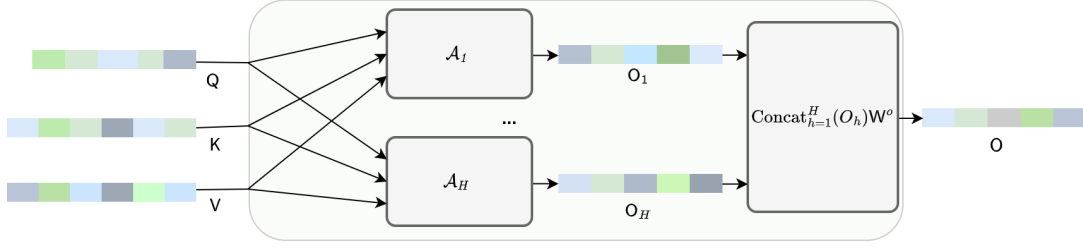


Figure 3.2: Multi-head Attention Mechanism. Multi-head attention using H attention heads is shown. The query, key and value are projected to each head, $\mathcal{A}_1, \dots, \mathcal{A}_H$, and then outputs, $O_1, \dots, O_H$, from these heads are concatenated and projected using output weights, W^o , to produce final output, O . The colours in the sets distinguish different tokens and do not correspond to any particular numerical values.

rows are tokens. A single attention head, $\mathcal{A}_h$, is defined in terms of query, key and value as

$$\mathcal{A}_h(Q, K, V) = \text{Softmax} \left(\frac{1}{\sqrt{d}} QW_h^q (KW_h^k)^\top \right) VW_h^v, \quad (3.1)$$

where d is the feature dimension of Q and K , while W_h^q , W_h^k and W_h^v are the corresponding learned weight matrices for head h . Thus, $QW_h^q (KW_h^k)^\top$ forms a matrix whose ij^{th} element results from the inner product of row i of the (weighted) queries, Q , and row j of the (weighted) keys, K . Since by construction the rows of Q and K each correspond to a token, these operations systematically capture the relationships between the tokens, which subsequently are scaled and normalized by the token dimension, d , and the Softmax to provide what is termed *attention affinity*. In turn, the attention affinity is applied to the (weighted) values, V , to capture its relationship and yield the final score. In short, an attention function first computes a global relationship score among all pairs of tokens in the input query and key sets in the form of attention affinity and then incorporates the tokens from the value set to emphasize this global relationship. These operations result in a representation that captures relevance or relationship among the tokens in the input sets. A visual depiction is shown in Figure 3.1.

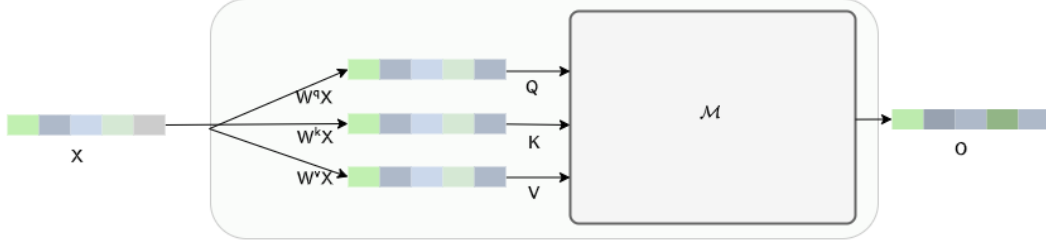


Figure 3.3: Self-attention Mechanism. Self-attention mechanism explained in terms of multi-head attention, $\mathcal{M}$. The single input X is projected to produce query, Q , key, K , and value, V , sets for the multi-head attention to produce final output, O . Position embeddings are not shown in favor of simplicity of visualization. The colours in the sets distinguish different tokens and do not correspond to any numerical values.

3.1.4 Multi-head attention

Multi-head attention uses multiple single-head attention mechanisms working in parallel and concatenates the outputs of all the heads. Thus, multi-head attention, $\mathcal{M}$, is defined from equation 3.1 as

$$\mathcal{M}(Q, K, V) = \text{Concat}_{h=1}^H \left(\mathcal{A}_h(Q, K, V) \right) W^o, \quad (3.2)$$

where H is the number of attention heads in the multi-head attention and W^o is the weight matrix for the final multi-head output. By aggregating outputs from multiple attention heads that operate on different projections of the inputs, the transformer model achieves increased representational capacity and enhanced generalization ability. As a result, the models can provide robust feature generation by capturing the diverse relationships among the tokens of input sets. A pictorial depiction of the operations involved is shown in Figure 3.2.

3.1.5 Self-attention

Self-attention, $\mathcal{S}$, is defined as an instance of attention with a single input, X , and an optional positional embedding, p . Commonly, it is instantiated in terms of multi-head attention, equation 3.2, as

$$\mathcal{S}(X, p) = \mathcal{M}(X + p, X + p, X). \quad (3.3)$$

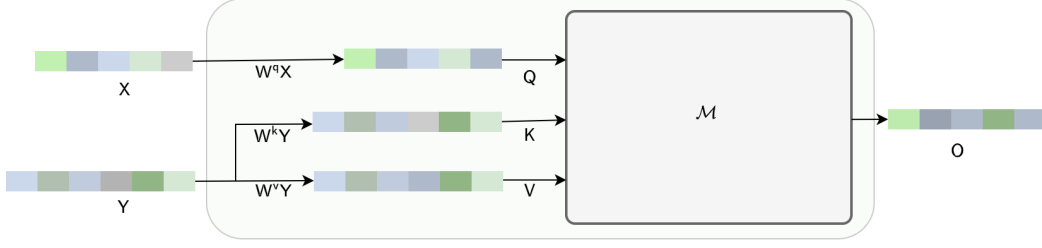


Figure 3.4: Cross-attention Mechanism. Cross-attention mechanism explained in terms of multi-head attention, $\mathcal{M}$. One input, X , is projected to produce query, Q , while the other input, Y , is projected to produce key, K , and value, V , sets for the multi-head attention to produce final output, O . Position embeddings are not shown in favor of simplicity of visualization. The colours in the sets distinguish different tokens and do not correspond to any particular numerical values.

The positional embedding captures the position of tokens within a set to emulate that property of a sequence [50]. It is used as an additional input to the attention mechanism to introduce locality, since attention itself is a permutation invariant operation. Self-attention relates different positions of a single set of tokens to compute a representation that captures global or long range context within the tokens of the input set. A visual depiction is shown in Figure 3.3.

3.1.6 Cross-attention

Cross-attention, $\mathcal{C}$, is defined as an instance of attention with two inputs, X, Y , and their corresponding positional embeddings, p^X, p^Y . Commonly, it is instantiated in terms of multi-head attention, equation 3.2, as

$$\mathcal{C}(X, Y, p^X, p^Y) = \mathcal{M}(X + p^X, Y + p^Y, Y). \quad (3.4)$$

Hence, cross-attention is a four input operation used to mix or combine two different input sets and their positional embeddings into a single output. Here, one input serves as query and another input is used as both key and value. In this way, cross-attention allows a combination of information from two different modalities. A visual depiction is shown in Figure 3.4.

3.2 MED-VT architecture components

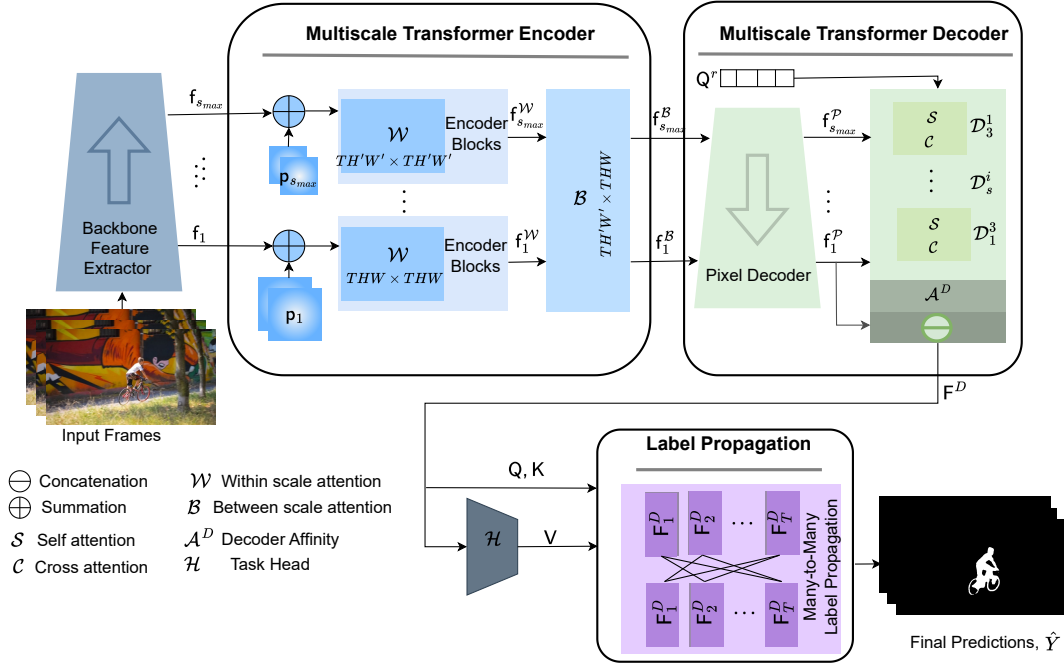


Figure 3.5: Detailed (MED-VT) Architecture with Unified Multiscale Encoder-Decoder Transformer, Illustrated with Application to Automatic Video Object Segmentation (AVOS). The model has five functionally distinct components. (i) Backbone feature extractor to extract per frame features, f_s , at multiple scales, s . (ii) Multiscale transformer encoder consisting of spatiotemporal within and between scale attention with resulting features, $f_s^{\mathcal{W}}$ and $f_s^{\mathcal{B}}$, resp; the multi-head attention transformation, equation 3.2, is used for both. (iii) Multiscale transformer decoder consisting of pixel decoding, which produces decoded features, $f_s^{\mathcal{P}}$, and a series of multiscale query learning decoder blocks, $\mathcal{D}_s^i$, that iterate i times across s scales, each of which entail self and cross attention, again using the multi-head attention transformation, equation 3.2. The input to the blocks are the decoded features $f_s^{\mathcal{P}}$ and the query resulting from the previous block, with a randomized query, Q^r , initialization; the output is a final object query, Q^o . The decoder applies an affinity, equation 3.9, between Q^o and the finest scale decoded features, $f_1^{\mathcal{P}}$, to yield an object attention map, which is concatenated with the finest scale decoded features for final decoder output, F^D . (iv) A task specific head (a shallow 3D ConvNet) produces initial predictions from F^D . (v) A many-to-many label propagator, equation 3.12, that inputs initial predictions and F^D to produce final temporally consistent segmentation masks, $\hat{Y}$.

3.2.1 Overview

MED-VT is an end-to-end video transformer that inputs a clip and provides dense segmentation predictions without the need of explicit optical flow. Processing unfolds in five main stages; see Figure 3.5. (i) A feature pyramid is extracted using a backbone network. (ii) The extracted stacks of feature pyramids are processed by a transformer encoder and (iii) decoder. (iv) A task specific head produces initial predictions. (v) A many-to-many temporal label propagator refines the initial predictions by enforcing temporal consistency. Our architecture is unique in its unified approach to encoding and decoding at multiple scales, as well as its use of many-to-many label propagation.

Feature extraction is standard. Given a video clip, we first extract a set of multiscale features, $F = \{f_s : s \in S\}$, where $f_s \in \mathbb{R}^{T \times H_s \times W_s \times C_s}$ represent features extracted at scale s , $S = \{1, \dots, s_{max}\}$ indexes the scales from coarse to fine and $\{T, H_s, W_s, C_s\}$ are the clip length, height, width and channel dimension at scale s , resp. Prior to subsequent processing, backbone features are down projected to d dimensions, $\bar{f}_s = \phi(f_s) \in \mathbb{R}^{T H_s W_s \times d}$, where ϕ is a simple 1×1 convolutional layer, followed by flattening. Our model is not specific to a particular backbone feature extractor; indeed, we illustrate with multiple in Section 3.5. Moreover, our model is not specific to a particular task head and we also illustrate with multiple in Chapter 4. The rest of this section, details our novel encoder-decoder and label propagator.

3.2.2 Multiscale transformer encoder

Transformer encoders built on spatiotemporal self-attention mechanisms can capture long range object representation relationships across both space and time for video recognition tasks [12, 20, 54]. They thereby naturally support learning of both spatial and temporal features as well as integrated spatiotemporal features. Notably, however, standard encoders that operate over only coarse scale feature maps limit the ability to capture fine grained pattern structure as well as fail to support precise localization [251]. To overcome these limitations, our encoder encompasses two main operations of within and between-scale spatiotemporal attention on multiple feature abstraction levels with different resolutions that are extracted from a backbone convolutional network, as discussed in Section 3.2.1.

We formulate the operations of within and between scale attention via standard multi-head

self- and cross-attention mechanisms, following equation equation 3.3 and equation equation 3.4, respectively.

Within scale attention. We formulate multi-head within scale attention, $\mathcal{W}$, by instantiating multi-head self-attention, equation 3.3, as

$$\mathcal{W}(\bar{\mathbf{f}}_s, \mathbf{p}_s) = \mathcal{S}(\bar{\mathbf{f}}_s, \mathbf{p}_s) = \mathcal{M}(\bar{\mathbf{f}}_s + \mathbf{p}_s, \bar{\mathbf{f}}_s + \mathbf{p}_s, \bar{\mathbf{f}}_s), \quad (3.5)$$

with $\mathbf{p}_s \in \mathbb{R}^{TH_sW_s \times d}$ per scale learned positional encodings to preserve location information, cf. [251]. $\mathcal{W}$ is applied successively across multiple encoder layers. We compute the final encoded feature maps, $F^{\mathcal{W}} = \{\bar{\mathbf{f}}_s^{\mathcal{W}} : s \in S\}$, for each corresponding scale, to capture globally consistent representation of objects of interest. Successive application of spatiotemporal within scale attention yields globally coherent representation.

Between scale attention. For between-scale attention, $\mathcal{B}$, we apply attention on the encoded features, $F^{\mathcal{W}}$. Coarse scale feature maps capture rich semantics by virtue of having gone through multiple abstraction layers. Correspondingly, the feature map from a coarser scale, s , i.e., $\bar{\mathbf{f}}_s^{\mathcal{W}}$, is used to affect the immediately finer scale feature map, $\bar{\mathbf{f}}_{s-1}^{\mathcal{W}}$, based on between-scale attention affinity. To achieve this goal, we use multi-head cross-attention, equation 3.4, now as

$$\mathcal{B}(\bar{\mathbf{f}}_{s-1}^{\mathcal{W}}, \mathbf{p}_{s-1}, \bar{\mathbf{f}}_s^{\mathcal{W}}, \mathbf{p}_s) = \mathcal{C}(\bar{\mathbf{f}}_{s-1}^{\mathcal{W}}, \bar{\mathbf{f}}_s^{\mathcal{W}}, \mathbf{p}_{s-1}, \mathbf{p}_s) = \mathcal{M}(\bar{\mathbf{f}}_{s-1}^{\mathcal{W}} + \mathbf{p}_{s-1}, \bar{\mathbf{f}}_s^{\mathcal{W}} + \mathbf{p}_s, \bar{\mathbf{f}}_{s-1}^{\mathcal{W}}), \quad (3.6)$$

where $\mathbf{p}_s$ and $\mathbf{p}_{s-1}$ are learned positional embeddings. This operation enhances between-scale communication to promote globally consistent, semantically rich feature maps and is conducted between each pair of adjacent scales. We denote the output features from between-scale attention, $\mathcal{B}$, as $F^{\mathcal{B}} = \{\bar{\mathbf{f}}_s^{\mathcal{B}} : s \in S\}$.

3.2.3 Multiscale transformer decoder

Our multiscale encoder’s between-scale attention promotes spatiotemporal consistency across scales, while the decoder promotes multiscale query learning to localize object-level properties. Our decoder works in two steps: (i) pixel decoding, which propagates coarse scale semantics to fine scale localization and (ii) transformer decoding, which generates adaptive

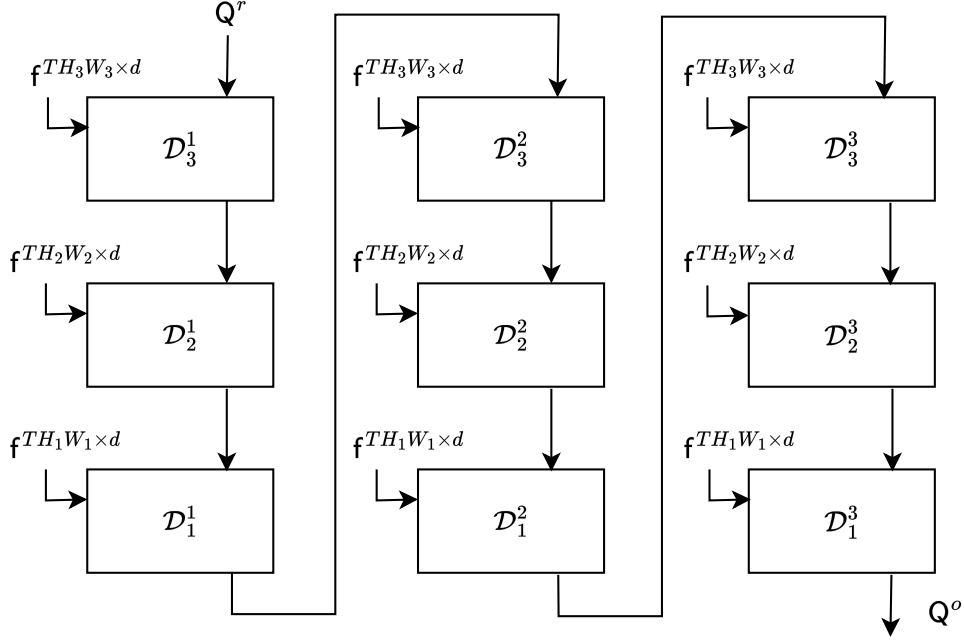


Figure 3.6: Decoder Stacked Coarse-to-fine Processing. Our multiscale decoder inputs a multiscale feature pyramid, $F^{\mathcal{P}}$, and randomly initialized queries, Q^r , and outputs final object queries, Q^o . The input is processed coarse-to-fine and iteratively through multiple decoder blocks, $\mathcal{D}_s^i$, with s indicating input feature scale and i indicating iteration. For simplicity, we show $s = 3$ scales and $i = 3$ iterations, with f - denoting features from each level of the pyramid, $F^{\mathcal{P}}$, where corresponding dimensions of the three levels are, $TH_3W_3 \times d, TH_2W_2 \times d, TH_1W_1 \times d$, resp.

queries.

Pixel decoding. In pixel decoding we seek to propagate semantics of coarse scale features to finer scales. For this purpose, we use a Feature Pyramid Network (FPN) [133]. The FPN works top down from coarse features with highest abstraction to fine features by injecting coarser scale information into each finer scale. It thereby allows for better communication from high level to low level semantics with finer details preserved before queries are generated in the actual transformer decoder. The FPN inputs the between-scale attention features, $F^{\mathcal{B}}$, and outputs a feature pyramid $F^{\mathcal{P}} = \{\bar{f}_s^{\mathcal{P}} : s \in S\}$.

In the FPN we combine feature outputs from our multiscale encoder along with the two finest resolution outputs directly from the backbone. The FPN operates on these coarse-to-fine scheme by (i) performing a 1×1 convolutional operation that maps the features to channel dimension 384, (ii) performing ReLU and bilinear upsampling to match the next scale and

(iii) pointwise adding the upsampled feature map to the next scale. The levels from the feature pyramid network output are used as input to our multiscale query learning mechanism. Further details on how FPNs operate are available in the original paper [133].

Decoding and adaptive queries. Improved object queries, Q^o , are learned via multiscale coarse-to-fine processing, where the queries adapt to the input features for object localization. This multiscale processing enriches the learnable queries so they work better on the finest resolution. These adaptive queries, $Q^o \in \mathbb{R}^{N_q \times d}$, with N_q the number of queries, are learned jointly from the multiscale feature maps, F^P , using a series of transformer decoder blocks, $\mathcal{D}_s^i$ operating coarse-to-fine across scales, s , and with multiple iterations, i ; see Figure 3.6. Each transformer decoder block, $\mathcal{D}_s^i$, inputs pixel decoded features, $\bar{f}_s^P$, at a particular scale, s . At each iteration, i , the blocks operate, coarse-to-fine, with queries output from the previous serving as input (along with decoded features from F^P) to the next. The process iterates N_d times, as notated by superscripts i on the blocks, $\mathcal{D}_s^i$, i.e., $i \in \{1 \dots N_d\}$. The entire process starts with a randomly initialized query, Q^r , and culminates in the final adaptive object queries, Q^o . The adaptive queries compactly represent the foreground object with its different appearance changes and deformation within the input clip.

In each decoder block both self- and cross-attention operates, as in encoding, Section 3.2.2. To define self attention, $\mathcal{S}$, we instantiate multi-head self-attention, equation 3.3, as

$$\mathcal{S}(Q_s^i, p_s^Q) = \mathcal{M}(Q_s^i + p_s^Q, Q_s^i + p_s^Q, Q_s^i), \quad (3.7)$$

with Q_s^i the input query to block $\mathcal{D}_s^i$ and $p_s^Q \in \mathbb{R}^{N_q \times d}$ learnable *query positional embeddings*. To define Cross attention, we instead instantiate equation 3.4 as

$$\mathcal{C}(Q_s^i, p_s^Q, \bar{f}_s^P, p_s, \hat{p}_s^\sigma) = \mathcal{M}(Q_s^i + p_s^Q, \bar{f}_s^P + p_s + \hat{p}_s^\sigma, \bar{f}_s^P), \quad (3.8)$$

with $p_s \in \mathbb{R}^{TH_sW_s \times d}$ *fixed sinusoidal feature positional embeddings* and $\hat{p}_s^\sigma \in \mathbb{R}^{TH_sW_s \times d}$ derived from learnable scale embeddings, $p_s^\sigma \in \mathbb{R}^{1 \times d}$, after being repeated across T, H_s, W_s ; this operation allows cross attention to be scale sensitive, cf. [37].

After all decoder blocks have produced the query, Q^o , a final cross attention block is used to establish affinity between the query and finest scale features, $\bar{f}_1^P$, and thereby generate an object attention map, F^A . Since only the query and features are considered, a two argument

affinity is used,

$$\mathbf{F}^A = \mathcal{A}^D(\mathbf{Q}, \mathbf{K}) = \text{Concat}_{h=1}^{N_h} \left[\text{Softmax} \left(\frac{1}{\sqrt{d}} \mathbf{Q} \mathbf{W}_h^q (\mathbf{K} \mathbf{W}_h^k)^\top \right) \right] \quad (3.9)$$

with $\mathbf{Q} = \mathbf{Q}^o$, $\mathbf{K} = \bar{\mathbf{f}}_1^{\mathcal{P}}$, N_h the number of heads and $\mathbf{W}_h^q$, $\mathbf{W}_h^k$ being learnable parameters for head h . The final decoder output, $\mathbf{F}^D$, is formed as the concatenation of the finest scale features, $\mathbf{f}_1^{\mathcal{P}}$, with the attention maps, $\mathbf{F}^A$, i.e., $\mathbf{F}^D = \mathbf{F}^A \ominus \mathbf{f}_1^{\mathcal{P}}$, with $\ominus$ channel-wise concatenation. This augmentation further enhances localization precision.

3.2.4 Many-to-many temporal label propagation

Label propagation is a standard technique that can be used in transductive reasoning [292]. In semi-supervised VOS, it was proposed to train an end-to-end model to propagate the labels from many/all previous frames, $\{\dots, t-1\}$, to a single current frame, t , hence causal many-to-one propagation within a transductive setting [150]. This operation provides structured prediction across frames instead of independent predictions per frame. We extend this idea by allowing label propagation from all other frames, $\{\dots, t-1, t+1, \dots\}$, in a clip to each frame, t , hence many-to-many label propagation. This extended operation enforces structured prediction across all frames in a clip.

In our many-to-many label propagator, three operators are applied sequentially: (i) a label encoder, $\mathcal{E}_L$, (ii) a spatiotemporal affinity based label propagator using masked attention, $\mathcal{M}^m$, and (iii) a label decoder, $\mathcal{D}_L$. The input to the encoder is an initial prediction, $\mathbf{Y}' = \mathcal{H}(\mathbf{F}^D)$, generated by a task head, $\mathcal{H}$, from the output of the previous decoding, $\mathbf{F}^D$. The label encoder then generates an encoding of dimension D from the initial predictions, $\mathbf{Y}'$, and flattens it, $\bar{\mathbf{Y}} = \mathcal{E}_L(\mathbf{Y}') \in \mathbb{R}^{TH_1W_1 \times D}$, cf. [150]. The label propagator extends the encoded labels temporally in a many-to-many fashion. The label decoder, $\mathcal{D}_L$, takes these propagated encoded labels and generates the final class-wise predictions. The label encoder, $\mathcal{E}_L$, is a similar ConvNet to that used elsewhere [150]. In particular, it generates richer target information from the single-channel segmentation mask by using successive convolution layers to increase the feature dimension and pooling operations to reduce the spatial dimension. The label decoder, $\mathcal{D}_L$, is a three-layer ConvNet.

We devise the label propagator as a masked attention module [231, 210] to capture the

long-distance dependencies between labels while preserving efficiency. The mask, $M \in \mathbb{R}^{TH_1W_1 \times TH_1W_1}$, restricts attention to regions centered around the predicted data point, akin to the notion of *clique* in conditional random fields [113] or, more generally, graph theory. The mask can be defined to promote communication between data points in a wide variety of fashions (e.g., within frame, between frames, many-to-one, many-to-many) [113]. We use this mechanism for temporal many-to-many propagation to encourage information sharing among different frames. The mask of two arbitrary positions is set to, $M_{ij} = -\infty$, if they are in the same frame, otherwise as zero. Formally, the mask is defined as

$$M_{ij} = \begin{cases} -\infty, & \text{if } \tau(i) = \tau(j) \\ 0, & \text{otherwise.} \end{cases} \quad (3.10)$$

where $\tau(\cdot)$ returns the frame index of given data points. Masked attention is then defined by augmenting the standard single, equation 3.1, and multi-head, equation 3.2, attention operations, to become

$$\mathcal{A}_h^m(Q, K, V, M) = \text{Softmax}\left(\frac{1}{\sqrt{d}}QW_h^q(KW_h^k)^\top + M\right)VW_h^v, \quad (3.11a)$$

and

$$\mathcal{M}^m(Q, K, V, M) = \text{Concat}_{h=1}^{N_h}(\mathcal{A}_h^m(Q, K, V, M))W^o, \quad (3.11b)$$

where Q, K, V, M are the queries, keys, values and mask resp., while W_h^q, W_h^k, W_h^v are learnable parameters for head h , and W^o is the final weighting for combining heads. Our label propagator instantiates masked attention, equation 3.11b, as

$$\tilde{Y} = \mathcal{M}^m(\bar{F}^D, \bar{F}^D, \bar{Y}, M), \quad (3.12)$$

where $\bar{F}^D \in \mathbb{R}^{TH_1W_1 \times (d+N_h)}$ is the flattened decoded features. As defined, this operation propagates labels across all data points in the entire clip, both spatial position and frames, unlike previous efforts that were limited to propagating to the current frame only [150].

Finally, the overall class-wise predictions, $\hat{Y}$, are produced by combining the propagated label decodings, $\mathcal{D}_L(\tilde{Y})$, and the initial predictions from the segmentation head, Y' according to

$$\hat{Y} = \frac{1}{2} \left(\mathcal{D}_L(\tilde{Y}) + Y' \right). \quad (3.13)$$

We combine the initial predictions per frame and the propagated predictions from all other frames because while label propagation enforces temporal consistency, it also can sacrifice boundary precision due to the smoothing it incurs. The final combination, equation 3.13, provides both temporal consistency and precise boundaries.

Label propagation and spectral clustering. There is a theoretical connection between spectral clustering and our label propagator, which we include to offer insights on the latter operation. We begin with formal definitions of spectral clustering [235]. A similarity graph is defined in terms of a set of vertices, v_i , and edges with weights, w_{ij} , on edges connecting vertices, v_i , and, v_j . The weight, w_{ij} , denotes the similarity between the two vertices; the similarity matrix, W , is comprised of the w_{ij} . For a vertex, v_i , we define the degree, $\delta_i = \sum_j^n w_{ij}$. The degree matrix, D , is defined as a diagonal matrix with degree $\delta_1, \dots, \delta_n$, where each δ_i sums the degree of connectivity for each vertex and its neighbours.

The normalized graph Laplacian matrix, which is a corner stone of the label propagation [292], represents the smoothness of the graph function. It has two main forms, the symmetric and random walk, where the former was used in label propagation prior to deep learning [292]. Here, we work with the random walk graph Laplacian, as it aligns with transformer operations; it is defined as $L_{rw} = I - D^{-1}W$, with I the identity matrix. Since the identity matrix is constant, we can see the main component of the normalized graph Laplacian, $D^{-1}W$, corresponds to the calculated per head, h , attention map in our label propagation scheme, equation 3.11a, $\text{Softmax} \left(\frac{1}{\sqrt{d}} QW_h^q (KW_h^k)^\top + M \right)$.

The part of the attention operation that computes the relevance between the query and key tokens with masking that controls predefined neighbourhood, $\frac{1}{\sqrt{d}} QW_h^q (KW_h^k)^\top$, corresponds to building the similarity matrix, W . We compute the similarity between two nodes in the similarity matrix, W , in terms of a scaled inner product, as standard in multi-head attention. The masking operation, M , serves to further enforce cliques within the similarity graph based on the structure of the data (e.g., the temporal structure). On the other hand, the normalization of the graph Laplacian using the degree matrix, D^{-1} , corresponds to the Softmax normalization in the attention operation. Overall, given that the attention map in equation 3.11a when applied to our values, V , controls the label propagation, we see that it acts analogously to the way the graph Laplacian enforces smoothness.

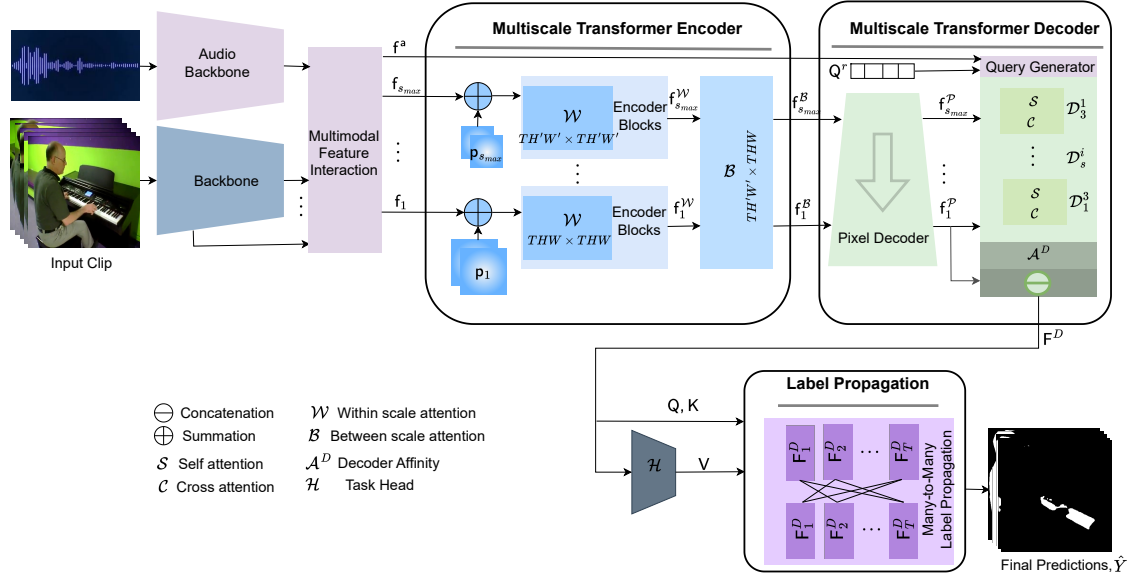


Figure 3.7: Detailed MED-VT++ Architecture for Multimodal Processing Illustrated with Application to Audio Visual Segmentation (AVS). It shares the five core modules of MED-VT (backbone, encoder, decoder, task head and label propagator, cf. Figure 3.5) and adds three additional modules: (i) Audio backbone to extract audio features, (ii) Multimodal feature interaction that uses bidirectional cross-attention, equation 3.16b, and (iii) Multimodal query generator that initializes the query, Q^r , from additional modality feature output from the feature fusion module (e.g., audio feature, f^a) instead of random initialization used for AVOS. Our key innovations lie in the multimodal feature interaction and query generator.

3.3 Multimodal extension (MED-VT++)

We have extended MED-VT to multimodal video segmentation tasks, i.e., to encompass an additional input signal beyond video, as illustrated in Figure 3.7. The specific additional modality that we have considered is audio; although, our approach to incorporating that signal is not specific to that particular modality. To extend MED-VT for this setting, we added three additional components: (i) a new modality feature extractor, (ii) an attention based multimodal fusion mechanism and (iii) a context-based query generator. For the sake of generality, we make no assumptions about the nature of the auxiliary modality, except that it can be represented as backbone features, f^a .

We developed an attention based context module immediately after the feature extractors for information exchange between the features of two modalities. In particular, we use

a bidirectional cross-attention mechanism [239] for information exchange between the audio features and each of the four stages of the visual features. The bidirectional cross attention mechanism is a simplification of cross-attention from one modality, X , to the other modality, Y , and vice versa, yet performed in one shared attention matrix to reduce the compute and memory footprints. To provide a concrete explanation of this bidirectional multi-headed cross-attention mechanism, we show the computation for a single attention head since the multi-head mechanism remains the same. In conventional cross attention, there are two inputs, say X and Y , one of which is used as query, Q , and the other as key and value, K, V . Rewriting equation 3.1 using two inputs for cross-attention with $Q = X, K = Y, V = Y$ and renaming W_h^q, W_h^k as W_h^x, W_h^y , we can formulate computation of a single head for cross-attention as

$$X' = \mathcal{A}_h(X, Y, Y) = \text{Softmax} \left(\frac{1}{\sqrt{d}} X W_h^x (Y W_h^y)^\top \right) Y W_h^v. \quad (3.14)$$

Similarly, computation of a single head of another cross-attention with $Q = Y, K = X, V = X$ can be expressed as

$$Y' = \mathcal{A}_h(Y, X, X) = \text{Softmax} \left(\frac{1}{\sqrt{d}} Y W_h^y (X W_h^x)^\top \right) X W_h^v. \quad (3.15)$$

Inspection of equation 3.14 and equation 3.15 shows that in realizing the two cross-attention operations, we can compute the attention matrix for one and reuse it for the other by taking the transpose. So, we can formulate a single head of the bidirectional cross-attention as

$$A = \frac{1}{\sqrt{d}} X W_h^x (Y W_h^y)^\top, \quad (3.16a)$$

and

$$(X', Y') = \left(\text{Softmax}(A) Y W_h^v, \text{Softmax}(A^\top) X W_h^v \right). \quad (3.16b)$$

Finally, we include a residual connection from the same modality feature to compute, $X'' = X' + X$, and $Y'' = Y' + Y$. As standard [74], the residual connection is used to avoid diminished gradients.

MED-VT++ uses a bidirectional encoder for each scale, s , of the visual features. The same auxiliary features, $f^a \in \mathbb{R}^{T \times c}$, are paired with all the scales of the visual features, where T is the number of frames and c is the dimension. As a preliminary, the backbone visual, f_s ,

and auxiliary features, f^a , are projected to a common dimensionality, d , via separate linear layers and flattening to yield $\bar{f}_s$ and $\bar{f}^a$, resp. Then, as one input to the bidirectional encoder we use the projected visual features from each scale, i.e., $X_s = \bar{f}_s$ for the s^{th} scale. For the other input, we always use the same projected auxiliary features, $\bar{f}^a$, i.e., $Y_s = \bar{f}^a$ for the s^{th} scale. Using these definitions of X_s and Y_s in equation 3.16a and equation 3.16b followed by residual connections yields a pair of visual, $\bar{f}_s''$, and auxiliary, $\bar{f}_s^{a''}$, features for each scale, s . The visual features, $\bar{f}_s''$, are used as input to next stage of the transformer encoder, exactly as in the original MED-VT. The auxiliary features, $\bar{f}_s^{a''}$, are used to provide context to the MED-VT decoder, as described in the remainder of this section.

In MED-VT, for unimodal video segmentation, the initial queries of the transformer decoder, Q^r , are randomly initialized and subsequently updated through self- and cross-attention mechanisms to get object queries, Q^o . When an additional modality is available, e.g., audio, it can be used for context-based initialization. In particular, we input randomly initialized queries, Q^r , and averaged bidirectionally encoded audio features,

$$\tilde{f}^a = \frac{1}{S_{max}} \sum_{s=1}^{S_{max}} \bar{f}_s^{a''}, \quad (3.17)$$

to an additional transformer decoder, which operates prior to MED-VT's original decoder, to yield context-based queries, Q^a . Here, the feature averaging is motivated by the fact that the additional decoder only inputs a single feature, not a set of (multiscale) features. The queries from this additional transformer decoder are used as input to the multiscale transformer decoder instead of Q^r (Section 3.2.3). Next, we detail how to obtain, Q^a .

Self-attention in the query generator is formulated analogously to equation 3.7 in the MED-VT decoder. Omitting the iterations, i , for simplicity, we have

$$\mathcal{S}(Q^r, p^a) = \mathcal{M}(Q^r + p^a, Q^r + p^a, Q^r), \quad (3.18)$$

with Q^r the input to the query generator, $p^a \in \mathbb{R}^{N_q \times d}$ learnable *query positional embeddings* and the output taken as $Q^a = \mathcal{S}(Q^r, p^a)$. Analogously, cross-attention for the query generator is similar to equation 3.8 and yields

$$\mathcal{C}(Q^a, p^a, \tilde{f}^a, \tilde{p}^a) = \mathcal{M}(Q^a + p^a, \tilde{f}^a + \tilde{p}^a, \tilde{f}^a), \quad (3.19)$$

with $\tilde{\mathbf{p}}^a \in \mathbb{R}^{T \times d}$ learnable *audio feature positional embeddings* and $\mathbf{Q}^a = \mathcal{C}(\mathbf{Q}^a, \mathbf{p}^a, \tilde{\mathbf{f}}^a, \tilde{\mathbf{p}}^a)$ taken as the final output queries. The self- and cross-attention operations, equation 3.18 and equation 3.19, are applied consecutively for m decoding layers to yield the final query, $\mathbf{Q}^a$. These output queries serve as input to the MED-VT transformer decoder, i.e., $\mathbf{Q}_s^0$ in equation 3.7 and equation 3.8.

The rest of the MED-VT architecture remains the same, except that we do not employ label propagation. Label propagation is not used because of the task nature: The visual segmentation targets can change abruptly depending on the audio signal (e.g., a person is segmented only in frames where they are talking, but not in other frames).

Notably, MED-VT++ does not entail any modality specific changes to MED-VT. Features from a source other than audio could serve as complements to visual features in the bidirectional encoder.

3.4 Training and inference

MED-VT/MED-VT++ employs an end-to-end training mechanism. Task specific losses are employed comprised of main and auxiliary components. The main loss, $\mathcal{L}_m$, is with respect to the final model prediction. The auxiliary loss, $\mathcal{L}_a$, is with respect to the initial prediction directly following the task head, i.e., before label propagation. The same task specific losses are used for both. The final loss computation is a weighted sum of the main and auxiliary losses, i.e.,

$$\mathcal{L} = w_m \mathcal{L}_m + w_a \mathcal{L}_a, \quad (3.20)$$

with w_m and w_a weights. This auxiliary loss allows us to train the model end-to-end in a simpler and faster way compared to relying solely on the main loss.

For inference, the same clip length is used as in training for each segmentation task. Output masks are upsampled to match input size.

3.5 Implementation details

For unimodal segmentation, we have implemented MED-VT with respect to two backbones that are used commonly on the considered tasks, ResNet-101 [74] and Video-Swin [141], to extract multiscale feature pyramid, F , from a clip of $T = 6$ frames by taking features, f_s , as activations at the last block of successive stages. For the multimodal segmentation task, we instead use pretrained PVTv2 [246] and VGGish [78] as visual and audio backbone, respectively, because those are used by our comparison approaches; see Section 4.4. For this task, we use $T = 5$ as clip size because the employed dataset only provides five annotated frames per video.

For the multiscale transformer encoder, we use only the two coarsest scale features for memory efficiency reasons; we use six and one encoder blocks for the two smallest scale features. In the FPN we combine the two coarsest resolution feature outputs from the encoder with the two finest resolution outputs directly from the backbone. Again, we only use the two coarsest resolutions for memory efficiency. The last three (coarsest) levels from the FPN are input to our multiscale query learning mechanism. For query learning, we use three feature scales and three iterations, $N_d = 3$, resulting in nine decoding layers. All multi-head attention operations and the attention block, $\mathcal{A}^D$, have, $N_h = 8$, attention heads and use a channel of dimension $d = 384$. All input backbone features, i.e., visual features for all cases and audio features for the multimodal case are projected to dimension $d = 384$ using a linear projection layer before subsequent processing. The weights for the loss function, $\mathcal{L}$, are $w_m = 1.0$ and $w_a = 0.5$.

The segmentation head, $\mathcal{H}$, is a three-layer convolutional module. The final decoding layer of $\mathcal{H}$ is defined according to the segmentation task to match the number of categories. Additional information on task specific implementation, training and inference details is provided in the corresponding evaluation sections of Chapter 4.

3.6 Summary

This chapter has provided detailed technical specification of the core MED-VT architecture, as well as its MED-VT++ extension. The basic mechanisms of attention were reviewed and

served as fundamental building blocks for all MED-VT components. MED-VT was seen to provide an end-to-end integrated multiscale approach to transformer encoding and decoding for rich feature extraction. Addition of an attention-based label propagator serves to enhance temporal coherence in estimation. MED-VT++ augments the core model by allowing for fusion of an auxiliary input information source, e.g., audio.

Chapter 4

Empirical evaluation

This chapter documents the performance of MED-VT and MED-VT++ on a diverse subset of the video segmentation tasks that were introduced in Chapter 2. In particular, MED-VT is evaluated on Automatic Video Object Segmentation (AVOS), actor-action segmentation and Video Semantic Segmentation (VSS), while MED-VT++ is evaluated on Audio Visual Segmentation (AVS). For each task, we present the considered benchmark datasets, task specific implementation details and provide quantitative as well as qualitative results.

In addition to comparing our results to extant state-of-the-art approaches, to highlight the particular benefits of multiscale processing, we also present qualitative and ablation results with a single scale baseline architecture. This baseline operates with only single scale transformer encoder and decoder using just the top layer feature map of the backbone feature extractor, i.e., the coarsest scale.

Following these performance comparisons, we provide a detailed ablation study and also report runtimes.

4.1 Automatic video object segmentation

4.1.1 Datasets and protocols

For evaluation on the AVOS task, we consider three datasets containing natural images that present diverse challenges. For instance, we include Moving Camouflaged Animals (MoCA), a natural camouflage video dataset; DAVIS 2016, widely used as a benchmark for video segmentation in recent works; and YouTube Objects, which includes additional challenges such as multi-object scenarios and diverse motion characteristics.

MoCA. Moving Camouflaged Animals (MoCA) is an especially challenging video object segmentation dataset containing camouflaged animals [116]. This is the most challenging motion segmentation dataset currently available, as in the absence of motion the camouflaged animals are almost indistinguishable from the background by appearance alone (i.e., colour and texture). Notably, in many cases the camouflage is so strong that the target animal is indistinguishable from the background in a single frame; however, it is revealed in the motion exhibited across frames. The dataset contains 141 video sequences of resolution 720p collected from YouTube depicting various camouflaged animals of 67 categories moving in natural scenes. Models trained on another dataset are tested on this dataset to validate generalization ability, as MoCA does not provide a training subset. As standard [295, 188], we train on DAVIS 2016.

The ground-truth is provided as bounding boxes, and we follow previous work [268] in removing videos that contain no predominant target locomotion, to evaluate on a subset of 88 videos constituting approximately 2803 frames. We follow the standard evaluation protocol provided for this dataset. The evaluation protocol for segmentation on this dataset computes the Intersection over Union (IoU) between the ground-truth bounding box and the minimum box that includes the estimated segmentation mask. In addition, the evaluation protocol on MoCA includes success rate at various thresholds. We evaluate using mIoU with the largest bounding box on the predicted segmentation and also report the success rate with varying IoU thresholds, $\tau \in \{0.5, \dots, 0.9\}$.

DAVIS 2016. Densely Annotated Video Segmentation (DAVIS) 2016 [178] is an automatic foreground-background segmentation dataset containing 50 high-quality video se-

quences with a total of 3,455 annotated frames of 480p and 720p resolution. There are 30 videos for training and 20 for testing. The videos cover multiple occurrences of common video object segmentation challenges, such as occlusions, motion blur and appearance changes. The video sequences contain a single object or two spatially connected objects annotated in each frame as the primary foreground object with a distinctive motion. There also are video-level attributes describing challenges present in the sequence, such as occlusions, fast-motion, non-linear deformation and motion-blur to enable a deeper understanding of the results of algorithms on this dataset.

Pixel-precision ground-truth is provided for the DAVIS 2016 dataset. We follow the standard evaluation metrics reported in recent approaches in this dataset. Three complementary metrics are used to evaluate the performance of video object segmentation models on this dataset. These metrics measure the spatial accuracy of the segmentation, the quality of the silhouette and the temporal coherence. Spatial accuracy is measured using region similarity, Intersection-over-Union (IoU), also known as the Jaccard index [204]. Silhouette quality is measured in terms of contour accuracy. To compute contour accuracy, the segmentation mask is considered as a set of closed contours, then the F-measure [34] is calculated from the contour based precision and recall between the segmentation mask and the ground-truth. Temporal stability is measured in terms of the segmentation contours between frames. For this purpose, polygonal representations of the contours of adjacent frames are matched using dynamic time warping to minimize the shape context descriptor (SCD) distance [10]. The temporal stability is calculated as the mean cost per matched point. Following recent work [145, 295, 188], we only report results as mean Intersection over Union (mIoU), $\mathcal{J}$, and boundary accuracy, $\mathcal{F}$, using 480p resolution frames.

YouTube-Objects. The YouTube-Objects [180] dataset contains a total of 155 web videos collected from YouTube. There are objects of 10 semantic classes and each class has nine to 24 videos of duration between 30 seconds and three minutes at resolution 720p. These videos are split into 1,407 clips containing a total of approximately 570,000 frames.

For video object segmentation, a subset of this dataset is used which has 126 shots with 20,647 frames covering all of the 10 semantic classes. A shot in this context is a short clip from a video containing the same objects. Generally, a shot is a continuous sequence from a video that was produced from a single capture of the camera and hence contains mostly the same objects with uniform camera settings. This subset provides pixel-level foreground-background

annotations of every 10^{th} frame [90]. This subset is used to validate the generalization ability of automatic video object segmentation models and has no training set of its own. For this purpose, models trained on other datasets are validated on the pixel-level annotated subset of the YouTube-Objects dataset. We follow standard protocol [295, 188] and train on DAVIS 2016. We follow the standard evaluation protocol used for this dataset. In particular, we used both per class mean intersection over union and the average of the mean intersection over union for all the classes across whole dataset. Spatial accuracy is measured using region similarity, Intersection-over-Union (IoU), also known as the Jaccard index [204]. More specifically, accuracy is reported as mean IoU (mIoU) per category as well as the mean over all categories.

4.1.2 Task specific details

The only necessary specialization of the MED-VT architecture is to set the number of outputs in the last layer of the segmentation task head, $\mathcal{H}$, to the number of segment categories. For AVOS, the only category is foreground object; correspondingly, we set a single output.

For object segmentation, we train the model using a combination of distribution-based and region-based losses. In particular, we combine the distribution-based focal [134] and the region-based Dice [159] losses for computing both the main, $\mathcal{L}_m$, and auxiliary, $\mathcal{L}_a$, losses in equation 3.20 (See Appendix A for definition of loss functions.) Notably, this combination ameliorates the challenge of class imbalance [134, 159], as in the evaluated datasets background pixels are more frequent than other classes. To support many-to-many label propagation in AVOS we use the entire clip ground-truth to compute the loss. We use a clip length, $T = 6$. We train the model once to evaluate on all three AVOS datasets. During the training we use the 30 videos from train split of DAVIS 2016 and augment them using additional videos from the training set of Youtube-VOS [266] following recent works [295]. We employ data augmentation strategies, including vertical and horizontal flipping as well as multiscale training [188] and the input images are downsampled so that the smallest side length becomes 300 pixels. We use AdamW [143] as optimizer, with initial learning rate 10^{-6} for the backbone and 10^{-4} for the rest of the model. We use weight decay of 1×10^{-4} and train for 15 epochs on the augmented training set using polynomial learning rate decay with power 0.9.

For inference we use the same clip length as in training, $T = 6$, in a sliding window with the predicted logits upsampled to the original image size. Each temporal window of frames

Measures	Uses RGB+Flow			Uses RGB only		
	COD (two-stream) [116]	MATNet [295]	RTNet [188]	COSNet [145]	Ours	Ours [†]
$\mathcal{J}$ Mean $\uparrow$	55.3	64.2	60.7	50.7	69.6	78.5
Success Rate $\uparrow$	$\tau = 0.5$	0.602	0.712	0.679	0.588	0.877
	$\tau = 0.6$	0.523	0.670	0.624	0.534	0.851
	$\tau = 0.7$	0.413	0.599	0.536	0.457	0.798
	$\tau = 0.8$	0.267	0.492	0.434	0.337	0.706
	$\tau = 0.9$	0.088	0.246	0.239	0.167	0.468
	SR_{mean}	0.379	0.544	0.502	0.417	0.740

Table 4.1: Results of Moving Camouflaged Object Segmentation on MoCA Dataset. The best overall results are shown in **bold**. Results shown as mean Intersection over Union (mIoU) and localization success rate for various thresholds, τ . Our results are reported with ResNet-101 backbone, as used in the state of the art [145, 295, 188], as well as Video-Swin backbone, labelled with [†].

serves as input to our model to predict the segmentation of its centre frame. During inference on DAVIS 2016 only, we use multiscale inference for postprocessing with predictions averaged over different scales using scale multipliers, (0.7, 0.8, $\dots$, 1.2), as it is standard practice to show results on DAVIS with and without postprocessing [33]. We scale the smallest side of the input images to 473 pixels for DAVIS 2016, 473 pixels for MoCA and 360 pixels for YouTube Objects respectively while maintaining the original aspect ratio of the test video frames.

4.1.3 Results on MoCA

Quantitative evaluation. Table 4.1 shows MoCA results, with comparison to the previous state of the art. It is evident that MED-VT outperforms all others by a notable margin when using the same backbone (i.e., ResNet-101 [74]) as the previous state of the art [295, 188]. Moreover, MED-VT performance improves even further when using the recent attention-based Video-Swin backbone [141]. Interestingly, even though our model does not use optical flow, it succeeds on this dataset where motion is the primary cue to segmentation due to the camouflaged nature of the animals. This fact supports the claim that our encoder is able to learn rich spatiotemporal features, even without optical flow input.

Qualitative evaluation. Figure 4.1 presents qualitative results on the MoCA dataset using the segmentation masks overlaid with input images in red. Note that the ground-truth is only specified as bounding boxes, whereas MED-VT performs pixel-precision segmentation to delineate actual object shape. The examples show the segmentation performance of MED-VT

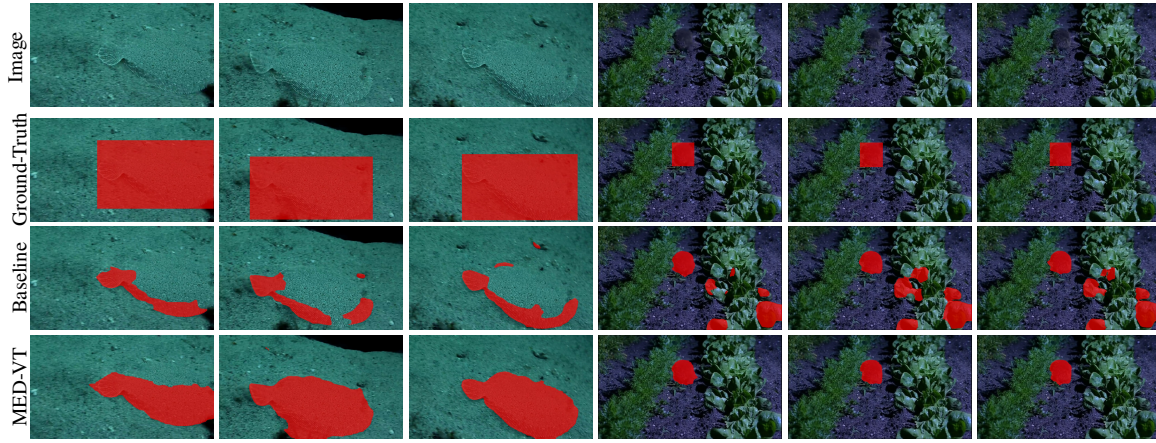


Figure 4.1: Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on MoCA Dataset. **Left:** Three frames of *flounder-6*. **Right:** Three frames of *hedgehog-1*. MED-VT segments with fine precision and temporal consistency, even in the presence of severe camouflage. MoCA ground-truth only specified as bounding boxes; although, MED-VT goes further to delineate actual object shape.

in the presence of strong camouflage. In particular, MED-VT successfully handles the false negatives of the single scale baseline to produce better segmentation masks for the case of *flounder-6*. For the case of *hedgehog-1*, it is seen that MED-VT successfully eliminates many false positives of the single scale baseline and delineates the object well.

4.1.4 Results on DAVIS 2016

Measures	Uses RGB + Optical Flow							Uses RGB only				
	MATNet [295]	RTNet [188]	FSNet [92]	HFAN [177]	TMO [41]	PMN [118]	Isomer [281]	COSNet [145]	DFNet [289]	IMP [119]	Ours	Ours†
Mean ↑	-82.4	84.3/85.6	82.1/83.4	86.8/88.0	85.6/-	85.4/85.6	88.8/-	-80.5	-83.4	84.5/-	83.3/83.7	85.0/86.7
$\mathcal{J}$ Recall ↑	-94.5	-96.1	-	96.1/96.2	-	-	-	-94.0	-	92.7/-	93.6/93.3	95.6/ 96.6
Decay ↓	-5.5	-	-	4.3/4.5	-	-	-	-0.0	-	2.8/-	0.05/0.05	0.06/0.07
Mean ↑	-80.7	83.0/84.7	83.0/83.1	88.2/89.3	86.6/-	86.4/86.2	91.1/-	-79.4	-81.8	86.7/-	84.3/83.7	87.6/86.9
$\mathcal{F}$ Recall ↑	-90.2	-93.8	-	95.3/ 95.4	-	-	-	-90.4	-	93.3/-	93.6/93.7	95.1/94.5
Decay ↓	-4.5	-	-	1.1/2/0	-	-	-	-0.0	-	0.8/-	0.03/0.03	0.04/0.04

Table 4.2: Results on DAVIS 2016 Validation Set. For those using post processing (e.g., conditional random fields [2, 295, 188, 92, 244, 145, 241, 289], instance pruning [273, 289], multiscale inference [289]), results shown as x/y , with x and y results without and with post processing, resp. Boundary F-measure, $\mathcal{F}$, and mean Intersection over Union (mIoU), $\mathcal{J}$ are shown. We show our results with the ResNet-101 backbone and the Video-Swin backbone, indicated by †. Best results in **bold**.

Quantitative evaluation. Table 4.2 shows DAVIS 2016 results, with and without postprocessing. With the Video-Swin backbone, MED-VT outperforms alternatives that, like MED-VT, do not employ optical flow information in mean/recall F-measure, $\mathcal{F}$, and mean/recall IoU,

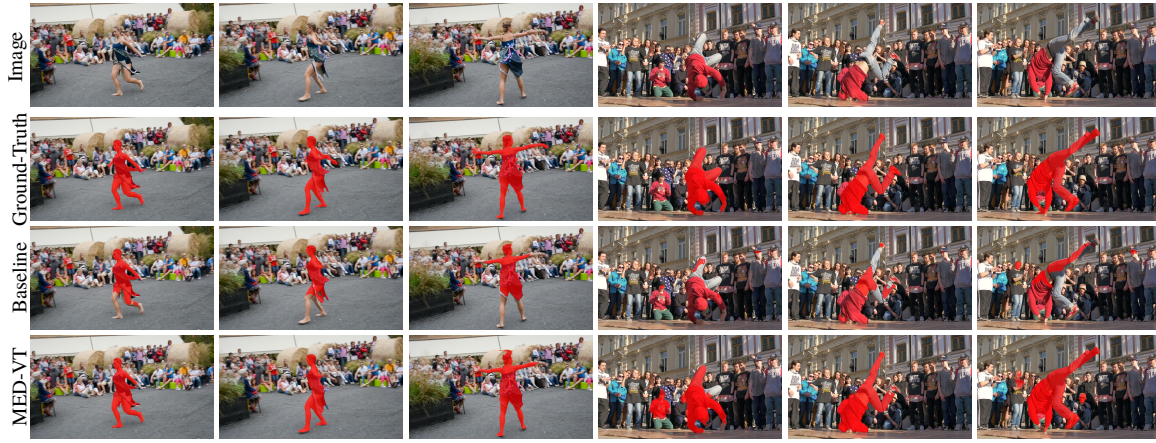


Figure 4.2: Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on DAVIS 2016. **Left:** Three frames of *dance_twirl*. **Right:** Three frames of *breakdance*. MED-VT segments with fine precision and temporal consistency, even in the challenging situation of thin object parts and articulated motion.

$\mathcal{J}$. When comparing to alternatives that use optical flow as additional information, MED-VT still outperforms the majority of approaches, except for HFAN [177] and Isomer [281], where MED-VT is still competitive. Again when reverting to the ResNet-101 backbone for MED-VT, MED-VT outperforms all similar alternatives that work directly on video frames (i.e., RGB without optical flow) on mean F-measure and mIoU while being competitive with the approaches relying on additional optical flow input. Notably, MED-VT relies only on RGB frames while other state-of-the-art approaches use optical flow as an additional input to exploit object motion. Moreover, most of the DAVIS 2016 results include CRF postprocessing [113]. In contrast, we do not employ such complex postprocessing; rather, we follow a simpler multiscale inference strategy similar to that used by another approach [289].

There is evidence that success on DAVIS 2016 is driven largely by the ability to capitalize on single frame/static appearance information (e.g., colour, texture), rather than dynamic (e.g., motion) information [112]. Unlike MED-VT, RTNet [188], uses extra pre-training on a saliency segmentation dataset, which aligns with success on DAVIS 2016 being tied to single frame information. Nevertheless, not only can MED-VT be competitive with RTNet (e.g., on mean F-measure without postprocessing) when using the same ResNet-101 backbone, but by switching MED-VT to the Video-Swin backbone it yields better performance on boundary accuracy and also slightly better performance on mIoU, without an extra dataset or optical flow.

Qualitative evaluation. Figure 4.2 presents qualitative results on the DAVIS 2016 dataset using the segmentation masks overlaid with input images in red. The examples show the segmentation performance of MED-VT in the presence of complex articulated motions and objects with thin body parts. MED-VT deals with all of these challenges in a temporally consistent fashion, with consistency from the encoder as well as label propagator and localization from the decoder. In particular, we observe that MED-VT can segment the thin parts of the dancer’s body on *dance_twirl*, e.g., hands and legs, consistently across frames. Similarly, MED-VT precisely segments the dancer in *breakdance* consistently across frames even in the presence of complex articulated motion and other stationary persons in the background.

4.1.5 Results on YouTube-Objects

Category	Uses RGB+Flow					Uses RGB only					Ours†
	MATNet [295]	RTNet [188]	HFAN [177]	TMO [41]	PMN [118]	PDB [207]	AGS [244]	COSNet [145]	AGNN [241]	Ours	
Airplane(6)	72.9	84.1	84.7	85.7	-	78.0	87.7	81.1	81.1	88.9	88.9
Bird(6)	77.5	80.2	80.0	80.0	-	80.0	76.7	75.7	75.9	73.6	85.4
Boat(15)	66.9	70.1	72.0	70.1	-	58.9	72.2	71.3	70.7	77.4	77.9
Car(7)	79.0	79.5	76.1	78.0	-	76.5	78.6	77.6	78.1	77.7	86.6
Cat(16)	73.7	71.8	76.0	73.6	-	63.0	69.2	66.5	67.9	79.5	83.4
Cow(20)	67.4	70.1	71.2	70.3	-	64.1	64.6	69.8	69.7	76.1	75.3
Dog(27)	75.9	71.3	76.9	76.8	-	70.1	73.3	76.8	77.4	75.9	81.1
Horse(14)	63.2	65.1	71.0	66.2	-	67.6	64.4	67.4	67.3	68.6	67.7
Motorbike(10)	62.6	64.6	64.3	58.6	-	58.3	62.1	67.7	68.3	65.9	69.2
Train(5)	51.0	53.3	61.4	47.0	-	35.2	48.2	46.8	47.8	70.1	77.2
Mean	69.0	71.0	73.4	71.5	71.8	65.4	69.7	70.5	70.8	75.4	79.3

Table 4.3: Results on YouTube-Objects. Results shown as mean Intersection over Union (mIoU) per category as well as overall average across all categories. Numbers in parentheses following category names indicate the number of instances per category. MED-VT results reported with the standard ResNet-101 backbone as well as the Video-Swin backbone (†). Best results are highlighted in **bold**.

Quantitative evaluation. Table 4.3 shows YouTube-Objects results. For overall results averaged across categories, it is seen that our approach once again outperforms all others by a considerable margin when using the standard ResNet-101 backbone without optical flow and further improves using Video-Swin, again without optical flow. Also, it is observed that although the performance of most of the alternatives varies widely across different categories (e.g., *Airplane* vs. *Train*), our approach is consistent across categories, which demonstrates its robustness. This observation is especially evident in the most challenging category, *Train*, where our approach outperforms the previous state-of-the-art approach, HFAN [177], that uses optical flow input by 15.8%.

Qualitative evaluation. Figure 4.3 presents qualitative results on the YouTube-Objects

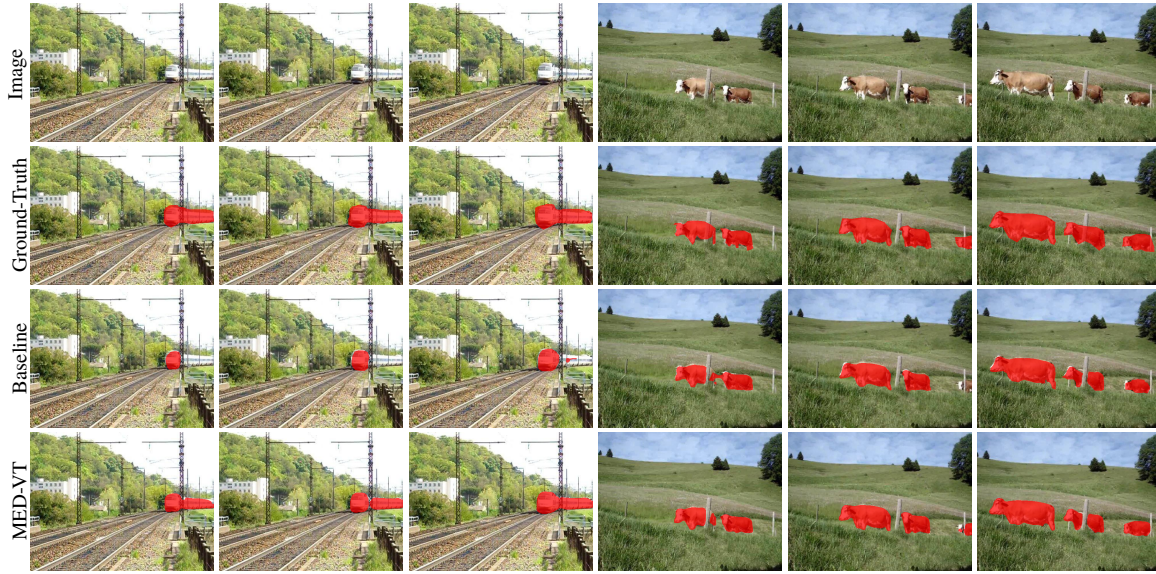


Figure 4.3: Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on the YouTube Objects Dataset. **Left:** Three frames of clip 0025 from *train* class. **Right:** Three frames of clip 008 of *cow* class. MED-VT segments with fine precision and temporal consistency, both in scenarios of multiple foreground objects and objects with connected components undergoing articulated motion.

dataset using the segmentation masks overlaid with input images in red. It is evident from the visual examples that MED-VT segments with fine precision and temporal consistency, both in scenarios of multiple foreground objects and objects with connected components undergoing articulated motion. In the example of *train*, the primary foreground object (i.e., the train) is undergoing articulated motion with multiple connected coaches. As shown in the figure, MED-VT successfully delineates the whole train across different frames, whereas the baseline delineates only the front compartment. On *cow* there are multiple cows moving in the field with the additional challenges of occlusion and deocclusion as well as new objects entering the scene. The ground-truth considers all of these cows as foreground objects. MED-VT is able to delineated the objects in such a scenario, whereas the baseline shows limited ability to delineate the suddenly appearing additional foreground object in the multi-object scenario (see the fifth column).

4.2 Actor-action segmentation

4.2.1 Datasets and protocols

For actor-action segmentation, we use the A2D dataset [263] as it is the largest dataset of natural images available til date containing actor-action tuples as annotation for joint segmentation of multiple actor and actions. Since our task of interest is segmentation, we do not consider datasets providing bounding box annotations (e.g., VidOR [201]).

A2D. The A2D [263] dataset provides actor-action segmentation with annotation for seven actor classes (adult, baby, ball, bird, car, cat and dog), eight action classes (climb, crawl, eat, fly, jump, roll, run and walk) and the no-action class. This dataset facilitates three specific problems including actor-action recognition with single-label, multiple-label and actor-action segmentation. The A2D dataset has 3,782 unconstrained ‘in-the-wild’ videos with varying characteristics collected from YouTube and provides pixel-level annotations. There are 43 valid actor-action tuples with at least 99 instances per tuple for joint actor-action segmentation tasks. The training split has 3,036 videos and test split has 746 videos divided evenly over all actor-action tuples. Both pixel level and bounding box annotations are provided per actor-action pair.

The evaluation of actor-action segmentation on this dataset includes global pixel accuracy, mean per-class accuracy and mean pixel Intersection-over-Union (mIoU) for both separate actor and action segmentation as well as joint actor-action segmentation [262]. The global pixel accuracy measures the overall percent of pixels with correct label. The mean per-class accuracy measures the percent of pixel level accuracy initially per class and then averaged across classes. The mean pixel Intersection-over-Union (mIoU) evaluates the region similarity using Intersection over Union (IoU), aggregated similarly to per-class accuracy. We follow the standard evaluation metrics reported in recent methods for the sake of comparability. Recent work considers mIoU as the most representative metric for correct pixel prediction over all classes [185] and that is what we report in our evaluation.

Method	Input	Backbone	mIoU
Ji et al. [93]	RGB + Optical Flow	ResNet-101	36.9
Dang et al. [47]		ResNet-101	38.6
SSA2D [185]		I3D	39.5
MED-VT (Ours)	RGB	ResNet-101 Video-Swin	45.7 57.1

Table 4.4: State-of-the-art Comparisons for Actor-action Segmentation on A2D Dataset. The best results are shown in **bold**. Results given as mean Intersection over Union (mIoU).

4.2.2 Task specific details

The only necessary specialization of the MED-VT architecture is to set the number of outputs in the last layer of the segmentation task head, $\mathcal{H}$, to the number of segment categories. For actor-actions, the number of categories is the number of actor-action tuples. For the A2D dataset that number is 43 and we set the number of outputs correspondingly.

For actor/action segmentation, we train the model using only a distribution-based loss. In particular, we use the distribution-based multiclass focal cross entropy loss [134] (See Appendix A for definition of loss functions.) Thus, it is used for both the main and auxiliary loss terms in equation 3.20. For the A2D dataset, we only have a ground-truth label for the centre frame. So, we segment all the frames in a clip together but minimize the loss for the center frame only. Since the whole clip is processed altogether, the model learns to segment all the frames in a clip with supervision from only the center clip ground-truth. This strategy allows us to apply label propagation in a similar way as done for the AVOS task. We train the model on the train set of A2D without any additional data. We employ data augmentation strategies, including vertical and horizontal flipping, use AdamW [143] as optimizer, with initial learning rate 10^{-6} for the backbone and 10^{-4} for the rest of the model. We use weight decay of 1×10^{-4} and train for 30 epochs using polynomial learning rate decay with power 0.9. Unlike AVOS, we do not use any sliding window during inference on A2D since the whole clip fits to our memory at once. Clip inference length is six and the predicted logits are upsampled to the original image size. During both training and inference, we scale the smallest side of the input images to 320 pixels for A2D dataset.

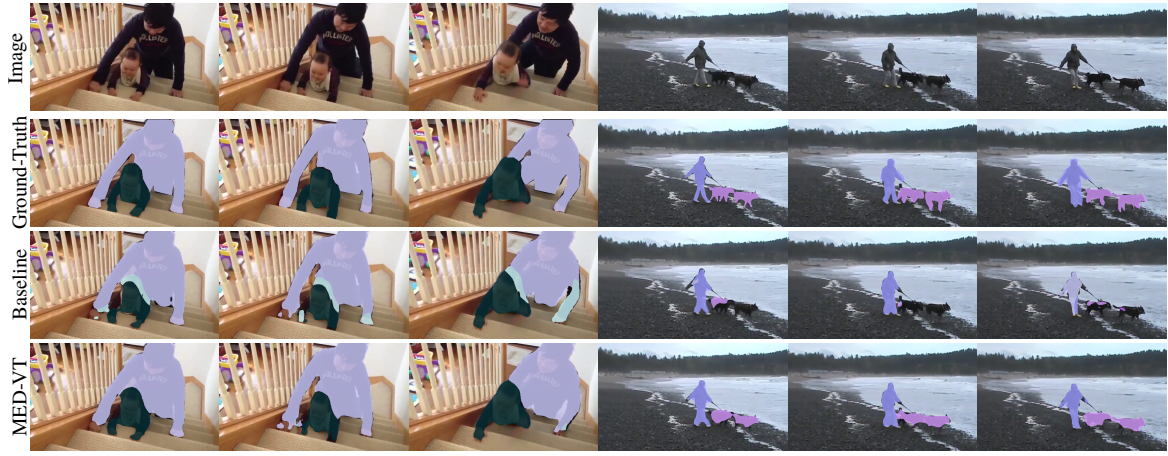


Figure 4.4: Qualitative Segmentation Results Comparing MED-VT to Baseline Algorithm on A2D Dataset. **Left:** Three frames of clip *2yu9Qkdo4HY* with adult-none and baby-climbing actor-action tuples. **Right:** Three frames of clip *CmOTlqhp_4s* with adult-walking and dog-walking actor-action tuples. MED-VT segments with fine precision and classifies correct actor-action tuples, both in scenarios of multiple objects performing different actions and involving complex object motions.

4.2.3 Results on A2D

Quantitative evaluation. Table 4.4 compares our approach on actor-action segmentation to a number of alternatives, all of which use optical flow as an extra input, across different feature backbones. Our results are consistently better than the alternatives, even when we operate under the simplest setting, i.e., one input modality (RGB images) and weaker features, i.e., ResNet-101 vs. I3D used by the previous state-of-the-art (SSA2D). When trained and tested with the stronger Video-Swin backbone (yet maintaining only RGB input), the improvements are even more notable, e.g., we outperform the previous best (SSA2D[185]) by more than 10% with Video-Swin features [141].

Qualitative evaluation. Figure 4.4 presents qualitative results on the A2D dataset using the segmentation masks overlaid on input images in red. The examples show that MED-VT can delineate segmentation regions precisely with correct actor-action classes compared to the baseline where both segmentation and classification errors are present. Notice, for example, the superior ability of MED-VT to delineate the boundary between the non-moving adult and the climbing baby in the left example and precise segmentation and correct classification of adult walking and dog walking in the right example. Overall, MED-VT segments with fine

precision and classifies correct actor-action tuples in scenarios of multiple objects performing different actions and involving complex object motions.

4.3 Video semantic segmentation

4.3.1 Datasets and protocols

We compare our proposed method with state of the art VSS methods on large scale benchmark of VSPW dataset. We consider evaluating our model on the VSPW dataset due to its comprehensive coverage of real-world scenarios and categories in video scene parsing featuring well-trimmed long-temporal clips, making it the largest benchmark for video semantic segmentation using real-world videos to date. Again, we focus on evaluating our model across multiple task categories to emphasize generality, rather than more datasets per task for a smaller number of tasks.

VSPW. Video Scene Parsing in the Wild (VSPW) [157] is a large-scale video semantic segmentation dataset containing 3,536 videos, each approximately five seconds long. The dataset provides pixel-level annotations for 124 categories. There are 2,806 videos in the training split, 343 videos in the validation split and 387 videos in the test split. The dataset covers a wide range of real-world scenarios, including both indoor and outdoor environments. Annotations are provided for both thing and stuff classes.

The evaluation metrics for this dataset includes mean IoU, weighted IoU and Temporal Consistency (TC). The mean IoU is the region similarity measure using intersection-over-union between the estimated and ground-truth pixels averaged over all the classes. The weighted IoU also calculates the region similarity but is weighted by the total pixel ratio of each class. The Temporal Consistency (TC) is computed as the intersection-over-union between the estimated mask of frame t and the warped estimated mask from frame $t - 1$ to frame t . Optical flow-based warping is used for this purpose. We follow the standard evaluation metrics reported in recent approaches for the sake of comparability. Following the disparity of reported metrics in various recent methods, we report on the superset of the reported metrics found in recent works.

4.3.2 Task specific details

The only necessary specialization of the MED-VT architecture is to set the number of outputs in the last layer of the segmentation task head, $\mathcal{H}$, to the number of segment categories. For semantic segmentation, the number of categories is the number of thing and stuff classes. For the VSPW dataset that number is 124 and we set the number of outputs correspondingly.

For semantic segmentation, we train the model using a combination of distribution-based and region-based losses. In particular, we combine the distribution-based focal [134] and the region-based Dice [159] losses (See Appendix A for definition of loss functions.) Thus, this combination is used for both the main and auxiliary loss terms in equation 3.20. Notably, this combination ameliorates the challenge of class imbalance [134, 159]. We train the model on the train set of VSPW without any additional data. We employ standard data augmentation strategies, including vertical and horizontal flipping and random scaling. We downsample the input images during training so that the smallest side becomes 360 pixels. We use AdamW [143] as optimizer, with initial learning rate 10^{-6} for the backbone and 10^{-4} for the rest of the model. We use weight decay of 1×10^{-4} and train for 10 epochs using polynomial learning rate decay with power 0.9. Both training and inference is done with a clip size, $T = 6$. Similar to AVOS, we use sliding window during inference and the predicted logits are upsampled to the original image size. During inference, we scale the smallest side of the input images to 473 pixels for the VSPW dataset.

4.3.3 Results on VSPW

Quantitative evaluation. Table 4.5 compares our approach on video semantic segmentation to a number of alternatives on the VSPW validation set. Overall, MED-VT without major task specific modification of the model achieved competitive results on VSPW (on-par with the second best results, Tube-Link [127]). Notably, the best reported results, TubeFormer-DeepLab [103], used additional data augmentation that has not been released, clip-level copy-paste.

Qualitative evaluation. Figure 4.5 shows qualitative segmentation results comparing MED-VT to the baseline algorithm on the VSPW dataset. These visual examples show that MED-VT can better delineate semantic categories compared to the baseline. For instance,

Method	Backbone	mIoU	fwIoU	mVC_8	mVC_{16}
TCB [157]	ResNet-101	37.8	59.5	86.9	82.1
Video K-Net [128]	ResNet-101	38.0	-	87.2	82.3
CFFM [211]	MiT-B5	49.3	65.8	90.8	87.1
MRCFA [212]	MiT-B5	49.9	66.0	90.9	87.4
TubeFormer-DeepLab [103]	Axial-ResNet-50x64	63.2	-	92.1	88.0
Tube-Link [127]	Swin-L	59.7	-	90.3	88.4
MED-VT (Ours)	Swin-B	58.1	72.9	89.7	83.3
MED-VT (Ours)	Swin-L	59.6	73.1	90.5	86.7

Table 4.5: State-of-the-art Comparisons for Video Semantic Segmentation on VSPW Dataset. The results report mean intersection over union (mIoU), frequency-weighted intersection over union (fwIoU) and video consistency (mVC_8 , mVC_{16}), with 8 and 16 the number of frames over which consistency is calculated. Best results in **bold**.

the left three columns show that MED-VT delineates the tennis net consistently across the video frames, whereas the baseline model fails. The example on the right three columns show MED-VT segmenting the cross walk stripes better than the baseline, as the stripes appear with various thicknesses that are best captured through multiscale representation.

4.4 Audio visual segmentation

4.4.1 Dataset and protocol

For multimodal audio visual segmentation, we use the AVSBench dataset suite as it is the only dataset containing sound source segmentation annotation. Again this dataset suite offers two datasets containing two different types of challenge discussed below. We do not use datasets containing bounding box annotations as our objective is segmentation.

AVSBench. The AVSBench [293] dataset is a benchmark for audio visual segmentation, aiming to segment sound sources in video frames according to the audio signal. This formulation of multimodal video segmentation is different from the speech referring video object segmentation as the audio is a sound generated from some objects in the video rather than a target reference given by a speaker. The dataset provides pixel-level annotations for the sound-ing objects in audible videos. The videos are split into train, validation and test sets. There

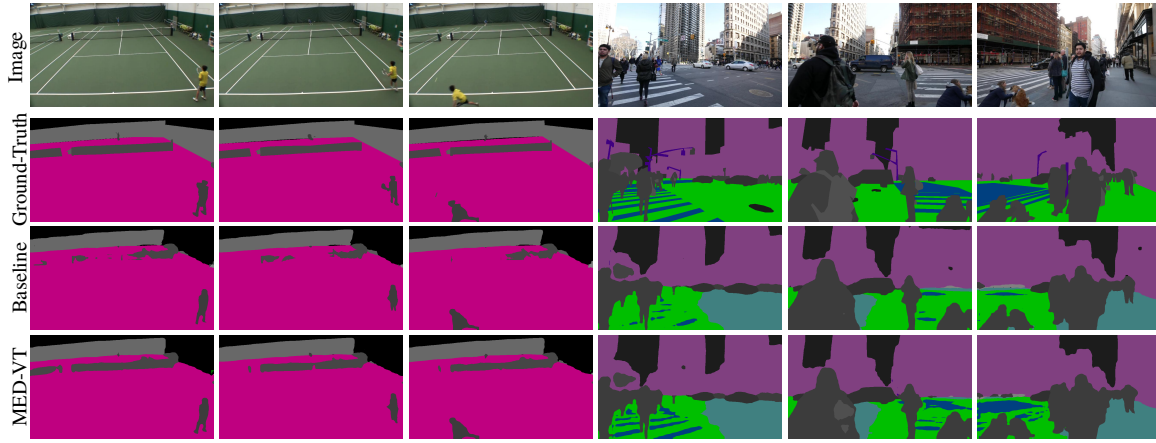


Figure 4.5: Qualitative Segmentation Results (red masks) Comparing MED-VT to Ground-truth and Baseline Algorithm on VSPW Dataset. **Left:** Three frames of *1017_8aIZCJKQLls*. **Right:** Three frames of *2312_R3mhAhexxyY*. MED-VT segments with fine precision and temporal consistency, even in the challenging situation of dense segmentation in the presence of small moving objects and clutter.

also are two subsets of data, single-source (S4) and multi-source (MS3) audio. The subset focused on a single sound source comprises 4,932 videos in total with train, valid and test splits of 3,452, 740 and 740 videos, respectively. The multi-source subset has 424 videos in total with train, valid and test splits of 296, 64 and 64 videos, respectively. Both of the subsets are comprised of videos spanning 23 categories, encompassing various sounds produced by humans, animals, vehicles and musical instruments, with the sound producing object considered as foreground. Similar to other object segmentation tasks, the evaluation metrics used are mean IoU and F-measure. We follow the standard evaluation metrics reported in recent approaches in this dataset for the sake of comparability which includes both mean IoU and F-measure.

4.4.2 Task specific details

The necessary specialization of the MED-VT architecture for AVS leads to the extension of MED-VT++ described in Section 3.3. We instantiate MED-VT++ using pretrained PVTv2 [246] and VGGish [78] as visual and audio backbone, respectively, because those are used by our comparison approaches. The number of output classes in the task head is set to 1, as the objective is to segment the sound producing object (single-source) or objects (multi-source)

as foreground. We use $T = 5$ as clip size because the employed AVS dataset only provides five annotated frames per video. We infer the whole sequence at once without using a sliding window. We also do not use label propagation as the the sound source segmentation requires the segmentation to follow the audio signal, not temporal context from visuals (e.g., a person can be present in all frames but segmented only in the frames where talking or singing.)

For training, we follow the same objectives used in earlier AVS work [293]. In particular, we use cross-entropy loss and Kullback-Leibler (KL) divergence for the multiple sound source setting (MS3) and only cross-entropy for the single sound source setting (S4). (See Appendix A for definition of loss functions.) Since label propagation is not used for this task, we do not need to use the auxiliary loss in the objective equation 3.20. We train and evaluate on the two subsets, S4 and MS3 separately. In both of the training, we use smallest side of the input images as 384 pixels. We use AdamW [143] as optimizer, with initial learning rate 10^{-5} for the backbone and 10^{-4} for the rest of the model. We use weight decay of 1×10^{-4} and train for 50 epochs using polynomial learning rate decay with power 0.9. Additionally, when training on the MS3 subset, we initialize the model with the learned weights from the S4 subset. During both training and inference, we used clip length, $T = 5$ and infer the whole clip at once without using sliding window. During inference, we scale the smallest side of the input images to 384 pixels for both S4 and MS3 subset of AVSbench dataset.

Quantitative evaluation. Table 4.6 compares our approach on audio visual segmentation to the most recent alternatives on the AVSBench dataset [293]. MED-VT++ outperforms the prior state of the art on both the single (S4) and multiple (MS3) sound source settings under both mean intersection over union and F1 scores. These results indicate that the multi-modal extension of MED-VT, with a simple attention based feature interaction across different modalities, can produce superior results compared to modality specific specialized feature interaction mechanisms [293].

4.4.3 Results on AVSBench

Qualitative evaluation. Qualitative visualizations for this task are shown in Figure 4.6. We observe that MED-VT++ accurately segments the sound source object corresponding to ‘man talking’ and ‘man talking and playing piano’ in the video frames. In the baseline model, person is not properly segmented and false positive segmentation of the piano occurs, even

Method	S4		MS3	
	M_J	M_F	M_J	M_F
AVS [293]	78.74	87.9	54.00	64.5
DiffusionAVS [151]	81.38	90.2	58.18	70.9
AVSegFormer [59]	83.06	90.5	61.33	73.0
MED-VT++ (Ours)	83.22	94.4	66.61	75.5

Table 4.6: State-of-the-art Comparison of Audio Visual Segmentation on AVSBench. The results report mean intersection over union (M_J) and F1-score (M_F). S4 and MS3 are two different problem formulations representing single sound source segmentation and multiple sound source segmentation. Best results highlighted in **bold**.

when the piano is not playing. In contrast, MED-VT++ successfully follows the sound sources to segment the desired objects.

4.5 Ablation study

The previous section documented the overall strength of MED-VT, with an integral part of that presentation being a confirmation of one of our key contributions: state-of-the-art performance on multiple video segmentation tasks without extra optical flow input. In this section we conduct ablation experiments on automatic video object segmentation (DAVIS 2016, MoCA, YouTube-Objects), actor-action segmentation (A2D) and video semantic segmentation (VSPW) to investigate two more of our contributions: the multiscale encoder-decoder video transformer and many-to-many label propagation.

Table 4.7 shows that the multiscale decoder immediately improves over the single scale baseline for all of the tasks (AVOS, actor-action segmentation and VSS). Addition of the multiscale encoder further improves the results to demonstrate their complementarity and the importance of unified multiscale encoding-decoding. Moreover, addition of many-to-many label propagation to the unified encoder-decoder consistently leads to the best overall performance. This enhancement can be traced to the label propagation yielding more temporally consistent predictions.

We also do a direct analysis of the benefits of our many-to-many label propagation vs. an alternative many-to-one label propagation technique, where many-to-one only propagates

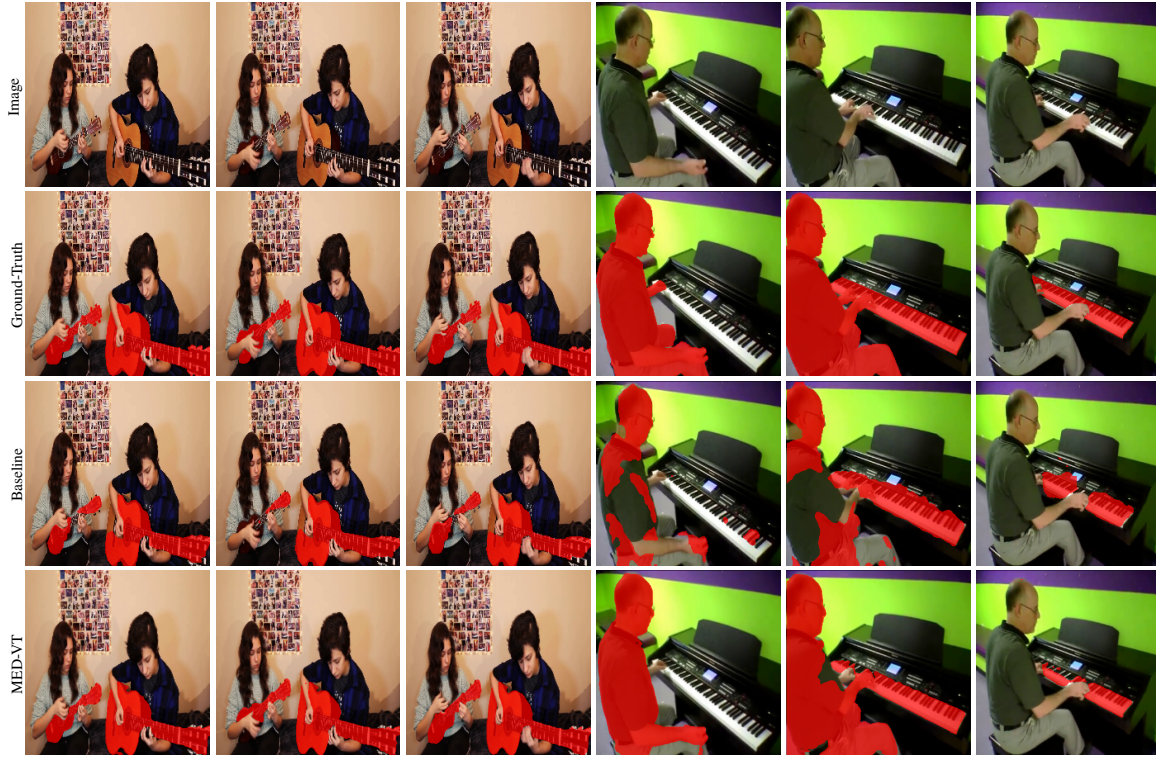


Figure 4.6: Qualitative Segmentation Results (red masks) Comparing MED-VT++ to Ground-truth and Baseline Algorithm on AVSBench Dataset. Shown are frames 1, 3 and 5 from two sequences. The audio of first sequence on the left refers to the sound from two musical instruments that are delineated by the segmentation model. The audio of the second sequence on the right has sounds for ‘man talking’, ‘man talking and playing piano’ and ‘piano playing’ corresponding to the shown frames. MEDVT++ can successfully follow the audio signal to only segment the sound producing objects and only in the frames where the object is producing sound even in challenging scenarios where the sound source changes over time.

Decoder Multiscale	Encoder Multiscale	Label Propagation	DAVIS 2016	YouTube Objects	MoCA	A2D	VSPW
-	-	-	79.5	73.9	67.5	52.3	56.3
✓	-	-	81.5	74.2	67.7	54.6	56.7
✓	✓	-	82.2	74.4	69.1	55.7	57.6
✓	✓	✓	83.3	75.4	69.6	57.1	58.1

Table 4.7: Multiscale Encoder, Decoder and Label Propagation Ablations. The results report mIoU. Best results highlighted in **bold**.

from previous frames to the current. Table 4.8 shows consistent improvement with many-to-many label propagation over the many-to-one alternative.

Method	DAVIS 2016	YouTube Objects	MoCA
-	82.2	74.4	69.1
Mto1	81.7	74.5	68.6
MtoM	83.3	75.4	69.6

Table 4.8: Comparison of Many-to-One (Mto1) vs. Our Many-to-Many (MtoM) Label Propagation. Best results in **bold**.

4.6 Temporal properties

To document the overall ability of MED-VT to exploit the temporal dimension, without computing optical flow, we consider the impact of corrupting the temporal information in a video. For this experiment we use the MoCA dataset due to its camouflage nature highlighting temporal information. We use two variants of temporal perturbation: shuffle the frame order and replicating a single frame T times (T is the clip length) to constitute a clip with no temporal dynamics. We perform these experiments with variable temporal sampling rates. Results are shown in Table 4.9. It is seen that shuffling reduces performance for all sampling rates, which demonstrates the importance of frame order. Notably, a far greater performance drop is observed when the temporal information is fully diminished by repeating the same frame. This evidence documents that MED-VT can exploit both correlated information from multiple frames and temporal order in segmenting objects. Conventional video segmentation approaches that simply extend single image segmentation models to process individual frames fail to capture this temporal information, while approaches relying on optical flow limit the ability for considering temporal information to only consecutive frames. Conversely, MED-VT exploits temporal information via clip-based inference with multiscale processing throughout.

As a demonstration of temporal consistency, we provide a quantitative result showing IoU over time (i.e., across frames) for a sequence from DAVIS 2016 and a sequence from MoCA comparing our single scale baseline and MED-VT using a ResNet-101 backbone. The result is plotted in Figure 4.7. It is seen that our results are comparably smoother than the baseline, e.g., showing less zig-zag pattern, and thereby documenting better temporal consistency in the segmentation.

Method	Sampling rate		
	1	5	10
-	78.5	78.0	77.6
Shuffle frame order in clip	77.1	77.4	77.3
Repeat a single frame in the clip	70.4	72.1	72.1

Table 4.9: Analysis of Temporal Dynamics by Means of Dynamic Perturbations on Clips Using MoCA Dataset. Best results highlighted in **bold**.

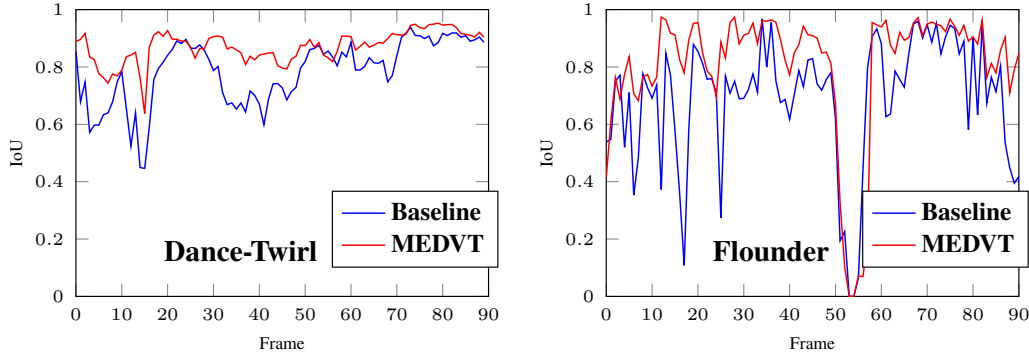


Figure 4.7: Temporal Consistency Analysis Using IoU over Time on Dance-twirl Video of DAVIS’16 and Flounder Video of MoCA Dataset. Flounder IoU drops around frame 53 due to sudden large camera motion.

4.7 Training and inference efficiency

We perform all benchmarking experiments on a Linux Server equipped with Intel(R) Xeon(R) W-2155 3.30GHz CPU and an NVIDIA Quadro P6000 24 GB GPU. We run all experiments on a single GPU. The training time is approximately two days. We run other models using their publicly available code and trained models. Table 4.10 shows a comparison of inference time, memory use and mIoU score on the MoCA dataset. It is seen that our approach is indeed efficient as it achieves the best performance in accuracy, while being faster in run time. This efficiency is a result of our approach not depending on time consuming optical flow estimation and the parallelization enabled by our ability to process a video clip as a whole rather than sequentially frame-by-frame. Due to the use of a multiscale transformer encoder, our approach requires more GPU memory than other approaches. An interesting future direction is to investigate memory efficient video transformers, e.g., [105].

Model	Time (s)	mIoU	Memory (MB)
MATNet [54]	0.9	64.2	2577
RTNet [32]	1.9	60.7	3615
COSNet [23]	1.3	50.7	9255
MED-VT (ours)	0.6	69.6	9509

Table 4.10: Comparison on Run Time and Memory Consumption. Best results highlighted in **bold**.

4.8 Summary

This chapter has documented empirical evaluation of MED-VT on three unimodal video segmentation tasks, Automatic Video Object Segmentation (AVOS), actor-action segmentation and Video Semantic Segmentation (VSS) as well as MED-VT++ on the multimodal Audio Visual Segmentation (AVS) task. For each task, evaluation datasets and protocols have been documented as well as task specific implementation details. In all cases, MED-VT/MED-VT++ has shown to be competitive with or exceeded the performance of current state-of-the-art approaches to these tasks.

Chapter 5

Interpreting MED-VT

To gain insight into the learned representations of MED-VT, we provide systematic probes to the architecture at key processing stages: backbone feature extractor, multiscale encoder, multiscale decoder and label propagator. At each stage, we provide both quantitative and qualitative analyses. For these investigations, we focus on MED-VT trained on Automatic Video Object Segmentation (AVOS) using the Video-Swin backbone [141]. For the sake of comparison, we also present results on the single scale baseline version of MED-VT, which was used for comparison in Chapter 4.

5.1 Methodology

5.1.1 Quantitative analysis.

To study the latent representations maintained by the backbone, transformer encoder and transformer decoder, we extend a recent method for analysing the extent to which deep models capture static *vs.* dynamic properties in video [112]. Here, static is defined as the information that can be captured from a single frame (e.g., colour and texture), while the dynamic factor refers to the information conveyed from multiple frames (e.g., motion). Together, the static and dynamic factors provide insight into fundamental aspects of video representation.

The original methodology calculated the extent that static *vs.* dynamic properties are rep-

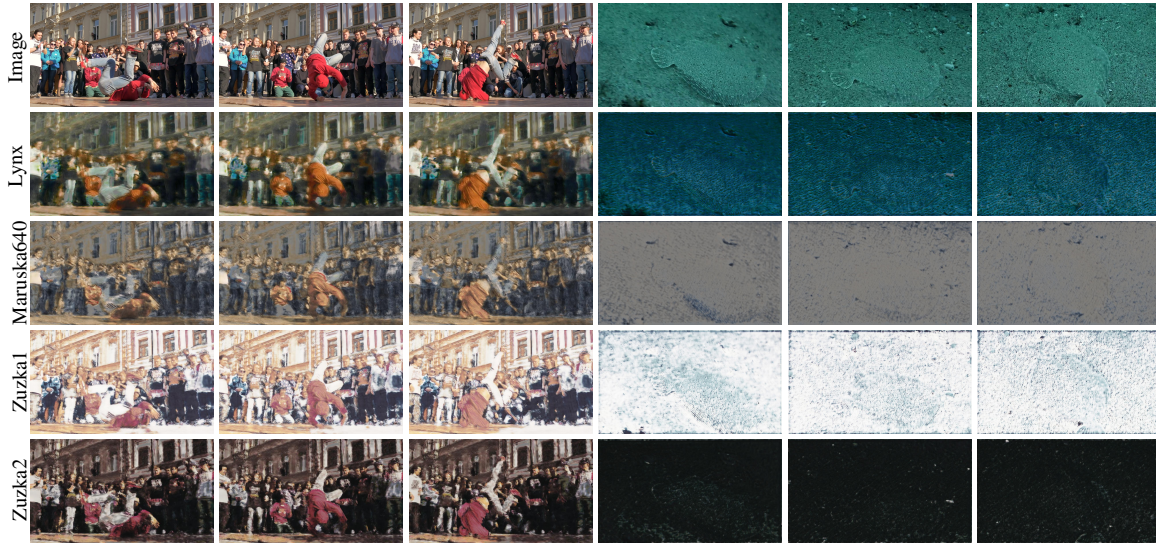


Figure 5.1: Example of Stylization Used for Interpretability Showing Stylized Frames from Both Stylized DAVIS 2016 and Stylized MoCA. **Left:** Three frames of the *breakdance* video from the DAVIS 2016 dataset. **Right:** Three frames of the *flounder-6* video from MoCA. Columns show original images and four different styles. Stylization perturbs the appearance while keeping the temporal dynamics.

resented at layer-wise and unit-wise granularities. This property was calculated in terms of mutual information between pairs of inputs generated from the same video clip, by perturbing either the static or dynamic information. If the mutual information was decreased more during perturbation of one type of information (static or dynamic) compared to other, then that information was deemed to be more characteristic of what is captured at that layer or unit. Static information was perturbed through appearance stylization, e.g., [219] (see Figure 5.1); dynamic information was perturbed through frame shuffling or optical flow jittering. Details are available in the original presentation [112].

In comparison to the original methodology, we employ two necessary modifications for our video object segmentation setup: (i) We modify the dataset used in measuring the static/dynamic bias to be less static biased; (ii) we use a different scheme for perturbing the dynamic information. First, in addition to adopting use of the Stylized DAVIS 2016 dataset, we extend to generate a stylized version of the MoCA dataset using a similar approach [219]. The reason for using stylized MoCA is inspired by the findings of the original work that DAVIS 2016 itself is static biased [112]. Thus, we chose another dataset that we believe to reduce the static bias by its design that focuses on camouflaged objects, where motion is a critical factor. Second, we

repeat the same frame throughout the clip to perturb dynamics in contrast to shuffling frames used in the original methodology. This choice is rooted in our ablation experiments with MED-VT, revealing that in video object segmentation shuffling the frame order is not sufficient to perturb dynamics (Table 4.9); although, it may be the case for action recognition [112]. Note, that the previous interpretability study focused on video object segmentation techniques that use appearance (RGB frames) and motion (optical flow). However, our model, MED-VT, relies on only appearance information without optical flow preprocessing; so, optical flow jittering is not applicable. Finally, we use both layer-wise and element-wise analysis using both Stylized MoCA and Stylized DAVIS 2016 using the static vs dynamic bias measure.

For quantitative analysis of label propagation, we employ the Receiver Operating Characteristic (ROC), which supports evaluation of a classification model across different thresholds to show the trade-off between sensitivity and specificity. We use ROC analysis for label propagation because it employs a compact label encoding of the initial predictions in the label space. This encoding intrinsically contains a very different representation compared to the latent features in the intermediate layers. In particular, a quantitative method to interpret classifier characteristics is suitable for investigating its properties. We show the ROC curves for the original videos vs. static or dynamic perturbed videos. For each variant, we extracted the first principal component of the label encoding, computed true positive and false positive rates and plotted the ROC.

5.1.2 Qualitative analysis.

For qualitative analysis, we visualize MED-VT’s internal representations using heat-maps of the intermediate layer features after performing principal component analysis [7, 217]. Since our interest is pixel-wise semantically discriminatory feature generation throughout model stages, we apply principal component analysis to the high-dimensional features and analyze the region-wise distinguishable activation of the main principal component. A rich feature representation for pixel-wise estimation tasks is expected to have features that show activation regionally distinguishable across semantics of the scene. As a representative input for demonstration, we use the *Flounder_6* video from the MoCA dataset for the following reasons. First, this sequence represents a very challenging scenario where the object (fish) in the scene is heavily camouflaged and very difficult to distinguish in a single frame. Second,

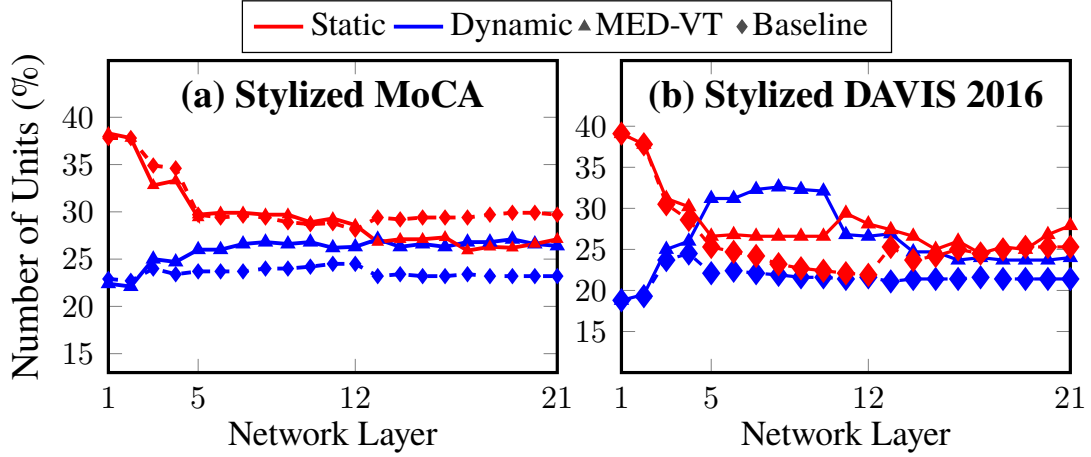


Figure 5.2: Layerwise Static vs. Dynamic Bias Analysis of MED-VT Compared to Our Single Scale Baseline on MoCA and DAVIS 2016 Datasets. Left: MoCA layer-wise analysis. Right: DAVIS layer-wise analysis. Overall, the analysis indicates that multiscale encoding-decoding enriches dynamic information in encoding that stays consistent during decoding.

this sequence poses additional challenges, where the fish changes motion directions and encompasses various speeds. Third, the object also presents challenges of partially going out of view and re-appearing in the scene.

5.2 Component-wise results

5.2.1 Backbone

Quantitative analysis. In Figure 5.2 the first four data points in the layer-wise analysis correspond to the four backbone stages. It is seen that the backbone learns to encode more dynamics and less static information in the MED-VT multiscale model compared to the single scale baseline, with most pronounced differences in stages three and four. In Figure 5.3, the unit-wise analysis on MoCA also shows that the backbone encodes dynamic information in more units for MED-VT compared to the baseline; although, the effect is lessened for DAVIS 2016. Furthermore, the joint encoding units increase in our model with respect to the baseline on DAVIS 2016, with the increase largely coming at the expense of decreased residual units. (Residual units are those that show little bias to either static or dynamic information, while joint encoding units show approximately equal bias.) Notably, DAVIS 2016 itself is

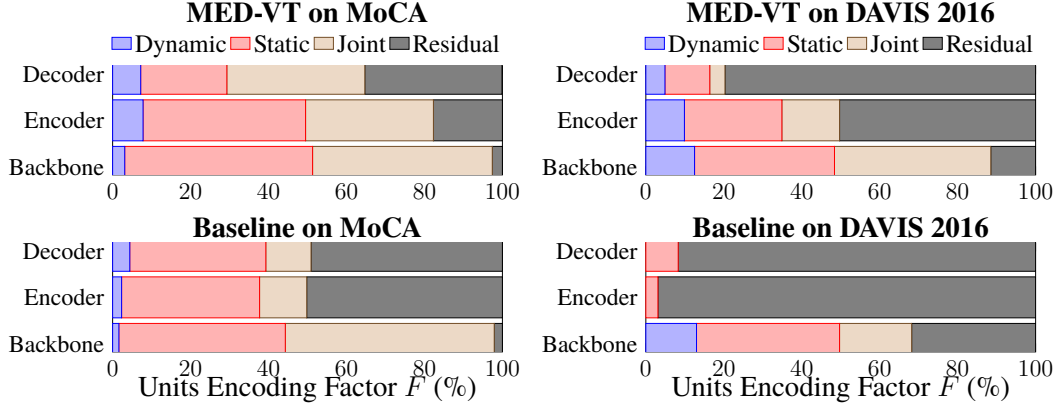


Figure 5.3: Unitwise Static vs. Dynamic Bias Analysis of MED-VT Compared to Our Single Scale Baseline on the MoCA and DAVIS 2016 datasets. Left: MoCA unit-wise analysis. Right: DAVIS unit-wise analysis. Dynamic and Static indicate units that are biased toward the corresponding factor, while Joint indicates units that are equally biased to both and Residual indicates units that have little response to either.

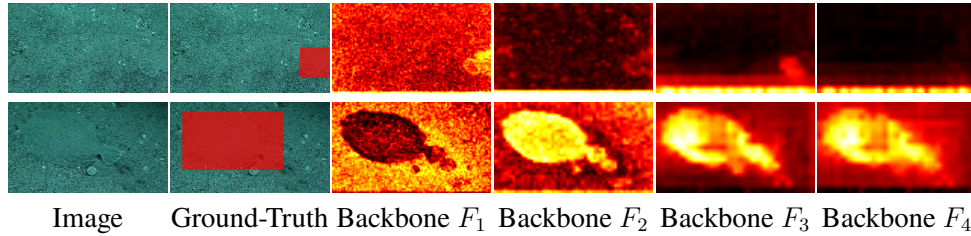


Figure 5.4: Activations (first principle component) at Different Backbone Layers as Heatmaps. Top and bottom rows show two frames of a camouflaged fish. It is seen that distinct information is captured at different scales (e.g., different object parts and textures). Keeping information across all scales yields richest representation.

static biased, but still requires some consideration of dynamics [112]; so, this dataset encourages conversion of residual to joint encodings. These results support that using multiscale processing in the downstream transformer compels even the earliest feature extraction, *i.e.* the backbone, to acquire a nuanced understanding of task demands, e.g., importance of dynamics in video segmentation, especially where motion is critical, e.g., in camouflaged MoCA.

Qualitative analysis. The visualizations in Figure 5.4 provides further insights. The backbone features (columns three - six) show that the maximum activation is not always at the last layer. As examples: The first row shows a scenario of severely diminished activation of the object region in the deepest layer feature of the backbone compared to other intermediate

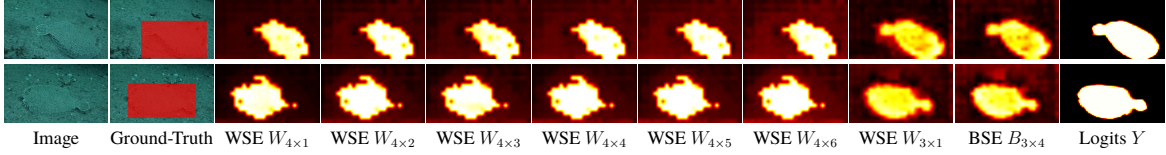


Figure 5.5: Activations (first principle component) at Different Encoder Layers as Heatmaps. Top and bottom rows show two frames of a camouflaged fish. Left-to-right: Input and bounding box ground-truth, six encoder layers operating on the deepest backbone layer (columns 3-8), between and within scale encoder operating on penultimate backbone layer (columns 9, 10), logit predictions (column 11). $W_{s \times k}$ denotes k^{th} layer of within scale encoder on stage s . $B_{i \times j}$ denotes between scale encoder between stage i and j .

layers; the second row reveals a different scenario where the second layer has activation from the object’s main body, while other parts (e.g., tail) are best captured by deeper layers. These results show that when operating as part of MED-VT the backbone is able to capture important object characterizations across a range of scales and pass all of them forward for subsequent processing, rather than just the deepest layer, which could be lacking in some details.

5.2.2 Encoder

Quantitative analysis. In Figure 5.2 data points 5-12 correspond to transformer encoder layers. The layer-wise analysis shows that MED-VT consistently increases in dynamic information with increased depth compared to the baseline. The unit-wise analysis in Figure 5.3 also shows an upswing of the dynamic factor during MED-VT encoding compared to the single scale baseline for both MoCA and DAVIS 2016. Also seen is an increase in joint encodings compared to the baseline. Interestingly, while the MED-VT encoder increases dynamics over the MED-VT backbone on MoCA, it slightly decreases in dynamic encodings (as well as static and joint encodings) on DAVIS 2016, with a corresponding increase in residual units. Even so, however, the MED-VT encoder has far fewer residual units than the baseline, which indicates that our multiscale processing encourages units to focus on static, dynamic and joint encodings. Overall, these results reveal that MED-VT’s multiscale encoder captures more dynamic information than its single scale baseline counterpart and it does so progressively with layer depth.

Qualitative analysis. Figure 5.5 visualizes different encoder layer activations. It is seen

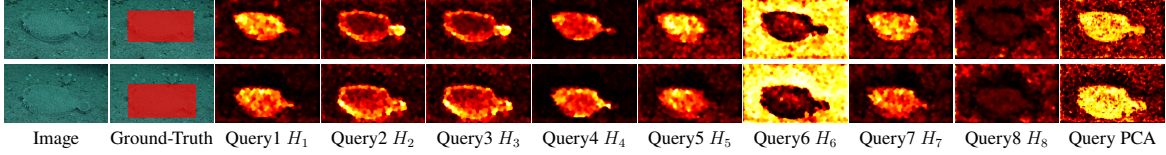


Figure 5.6: Activations (first principle component) Across Decoder Attention Heads as Heatmaps. Top and bottom rows show two frames of a camouflaged fish. Left-to-right: Input, bounding box ground-truth, decoder attention heads 1-8 ($H_1 \dots H_8$) and all attention heads.

that going deeper in encoder layers provides an increase of distinguishable activation in the object region, with the effect most pronounced in the top row. For both cases, the target object is greatly enhanced compared to the backbone alone, cf. Figure 5.4. Comparing encoding on the deepest layer backbone features, $W_{4 \times 1}$ to encoding on the penultimate layer backbone features, $W_{3 \times 1}$ reveals complementarity provided by multiscale processing: The penultimate scale provides more precise localization of the object boundaries, but with background noise; the coarser scale ameliorates noise, but compromises on the boundaries. Our multiscale approach can reap the benefits of both scales. Finally, comparison of between scale, $B_{3 \times 4}$, and within scale, $W_{3 \times 1}$, encoding shows the ability of between scale to enhance object delineation beyond what is provided by within scale alone, as here is where the scales combine to exploit complementarity. For example, notice how the body of the object better matches the final shape and attains more precise, less blurry boundaries, especially in the upper row.

5.2.3 Decoder

Quantitative analysis. In Figure 5.2 data points 12-21 correspond to decoder layers. The layer-wise results on MoCA show that the decoder maintains, but does not increase dynamic encoding. Ideally, the decoder’s role is generation of adaptive queries without altering the latent representation learned by the encoder; the observed behaviour is consistent with that goal. Conversely, the dynamic factor decreases in the decoding stages on DAVIS 2016, which apparently seems opposite to the expected properties of MED-VT. This outcome could be attributed to the static bias that is inherent in the DAVIS 2016 dataset [112]. The decoder generates dynamic queries based on encoded features to address the segmentation objective. On the DAVIS 2016 dataset, biased towards static samples, decoder queries prioritize static information over dynamics in encoder features. Conversely, the MoCA dataset, which is biased

towards dynamic samples, enables the decoder to extract queries emphasizing dynamic factors from encoded features. On the other hand, the unit-wise results in Figure 5.3 show an increase in dynamic units in the decoder for the multiscale MED-VT architecture compared to the single-scale baseline for both datasets, with a more pronounced impact on the MoCA dataset. These findings also support extending the experiment to use the stylized MoCA dataset for comparing static vs. dynamic bias instead of just the stylized DAVIS 2016 dataset: It is preferable to interpret a model property using a dataset that inherently contains the specific factor of interest. Overall, these results support that the decoder can capitalize on the information provided by the encoder in a dataset dependent fashion.

Qualitative analysis. Figure 5.6 shows the object attention generated by attention blocks, $\mathcal{A}^D$, per head. We observe that some attention heads learn to focus on various object aspects, including the object extremities. However, most attention heads generate well-localized saliency maps for the primary objects in the video. Indeed, the majority of attention heads in the examples yield activation for the primary object despite camouflage (e.g., H_1 , H_2 , H_3 , H_4 , H_5 , H_7). In complement, however, H_8 and especially H_6 capture the background. These results offer insight into the nature of attention generated through decoder queries, a level of detail not previously presented in work on transformer decoders.

5.2.4 Label propagation

Quantitative analysis. Figure 5.7 shows label propagation ROC analysis comparing original input to three alternative scenarios: appearance perturbed, motion perturbed and random guess. On the DAVIS 2016 dataset, there is a noticeable performance drop from original input to both appearance and motion perturbed inputs. The performance drop shows a similar pattern on the MoCA dataset; however, the change is less notable. The explanation for this difference might be due to the fact that success on camouflaged MoCA depends on motion information, which inherently involves temporal coherence; so, there is less room for label propagation to contribute. In contrast, for DAVIS 2016, where motion is a less important cue, as that dataset is static biased [112], label propagation can play a greater role in enforcing temporal coherence. Overall, both perturbations decrease performance in both datasets, with the effect most pronounced for dynamic perturbation. These results support that MED-VT can exploit dynamics for temporally consistent predictions through label propagation.

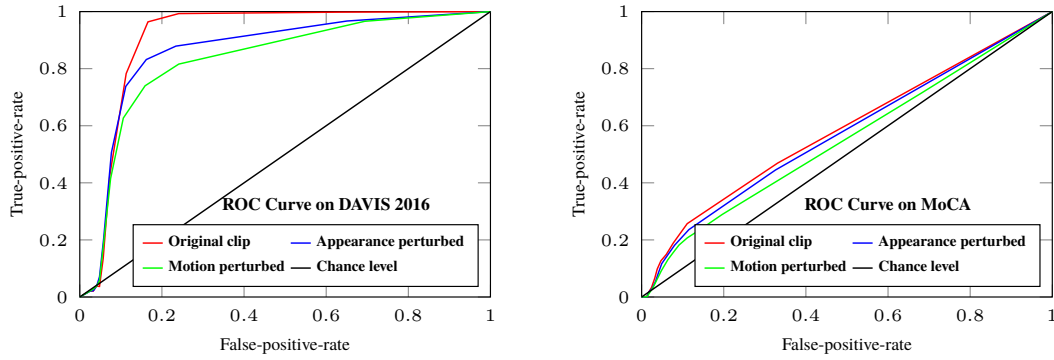


Figure 5.7: Receiver Operating Characteristic (ROC) Curve for the Label Propagator Encoder Comparing Three Scenarios. These are: i) original input images, ii) appearance perturbed input clip and iii) motion perturbed input clip. Left: ROC curve on DAVIS 2016; Right: ROC curve on MoCA. The results indicate performance drop for both appearance and motion perturbation, with higher impact of motion perturbation. The result is more noticeable in DAVIS 2016 dataset.

Qualitative analysis. Figure 5.8 visualizes label propagation. Both rows (frames) provide examples where label propagation recovers missing parts of the target object, with a more pronounced improvement in the second row. The example in the second row shows a large part of the fish with false negative labels in the task head logits; however, the label propagator associates information from surrounding frames to fill in the missing parts. The initial prediction from the task head considers the latent representation of each pixel independently, thereby limiting the ability to associate the object with the pixels of neighbouring frames. Label propagation associates the labels estimated in neighbouring frames to correct initially false negative labels.

5.3 Summary

Due to its reliance on deep-learning, the technical specification of MED-VT, given in Chapter 3 yields little insight on the internal representations that the model maintains in operation. To address that limitation, this chapter has provided an extensive interpretability analysis using both quantitative and qualitative methods. These investigations have shown that MED-VT’s multiscale representations capture dynamic information without use of optical flow, even while also maintaining useful static information. The model also has been shown to represent

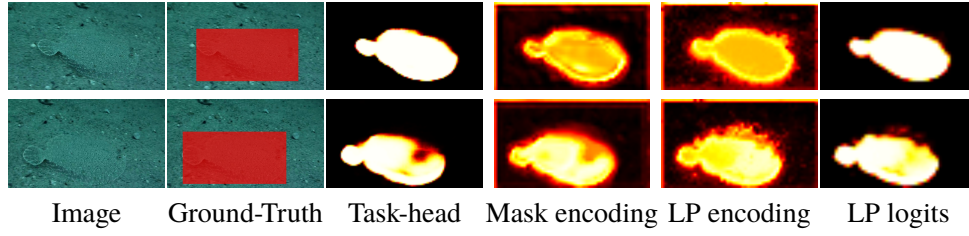


Figure 5.8: Activations (first principle component) at Different Stages of Label Propagation as Heatmaps. LP stands for Label Propagator. Top and bottom rows show two frames of a camouflaged fish. Left-to-right: Image, bounding box ground-truth, task-head logits, mask-encoding, propagator mask encoding and label propagator logits. Both rows show label propagation correcting errors in a given frame (e.g., missing parts), with second row providing stronger illustration.

and integrate object and scene information across scales. Moreover, label propagation has been shown to capitalize on across frame temporal structure to yield temporally consistent segmentations.

Chapter 6

Conclusions

6.1 Summary and overall discussion

This dissertation has presented an end-to-end trainable and unified multiscale encoder-decoder transformer. The model is targeted as a general multiscale architecture for advancing the ability of deep learning models on dense video estimation. Toward this goal, it is designed and developed for video segmentation in both unimodal and multimodal settings. Dense video estimation tasks aim at pixel-level inference from input video clips. Video segmentation, e.g., delineation of objects, motions, actions and other scene components, is one of the fundamental dense video estimation tasks. These particular tasks continue to be a subject of great interest due to both theoretical importance (e.g., delineation of visually distinct components) as well as practical importance (e.g., applications to autonomous driving and video editing). We have investigated this area by exploring unified multiscale processing throughout the processing pipeline of feature encoding, context encoding and object decoding in an encoder-decoder model. In particular, we presented a Multiscale Encoder-Decoder Video Transformer (MED-VT) that uses multiscale representation throughout and employs an optional input beyond video (e.g., audio), when available, for multimodal processing (MED-VT++).

A broad overview of video transformers, multiscale processing and interpretability analysis in the context of video segmentation were introduced in Chapter 1. Next, a detailed discussion of different categories of video segmentation, transformer-based-architectures, interpretability of such models and state-of-the-art approaches for video segmentation were dis-

cussed in Chapter 2. A component-wise description of the developed unified model, MED-VT/MED-VT++, for tackling these video segmentation tasks was provided in Chapter 3. Empirical evaluation with quantitative and qualitative results of MED-VT/MED-VT++ on three unimodal video segmentation tasks and a multimodal segmentation task (Automatic Video Object Segmentation (AVOS), actor-action segmentation, Video Semantic Segmentation (VSS) and Audio Visual Segmentation (AVS)) was presented in Chapter 4. A detailed interpretability study, encompassing both quantitative and qualitative analyses to document details of the model’s internal learned representations was presented in Chapter 5. Overall, emphasizing multiscale processing, interpretability and application across multiple video segmentation tasks, this research aimed to provide insights not only into these particular tasks and models, but also more broadly into the field of computer vision and Artificial Intelligence (AI).

In more detail, the key contributions of this dissertation are as follows.

1. We presented a novel Multiscale Encoder-Decoder Video Transformer (MED-VT). MED-VT is the first video transformer to unify multiscale encoding and decoding. This design is motivated by the fact that object and other scene components appear in images across a range of scales. Multiscale encoding yields rich and temporally consistent spatiotemporal features derived from only RGB input (i.e., without optical flow or object proposals). Multiscale decoding yields semantically informed, precise localization.
2. We explored temporal consistency across time and presented a transductive learning scheme through many-to-many label propagation to provide temporally consistent video estimates. The label propagation provides structured estimation beyond the task head’s pixel-level classification by enforcing consistency across all frames in a clip through information sharing among different frames. This approach has been shown to handle complex situations in unimodal video segmentation tasks where an object may be partially obscured, out of sight, or where certain parts are inactive in the feature map due to various factors in some frames, yet are well activated in neighboring frames.
3. Furthermore, we used a seamless attention based integration of the audio modality, when available, to apply MED-VT to audio visual segmentation without requiring a specialized multimodal architecture. This extension, MED-VT++, uses optional modules at the encoder and decoder to integrate multimodal features without relying on any domain-specific hand-crafted engineering. While we have focused on integration of au-

dio with video, it is important to note that our approach to integrating this signal is not limited solely to the realm of audio and others, such as text, could be considered.

4. We demonstrated the wide applicability of the developed architecture through empirical evaluation on four video segmentation tasks: Automatic Video Object Segmentation (AVOS), actor-action segmentation, Video Semantic Segmentation (VSS), and Audio Visual Segmentation (AVS). All these tasks are significant from an applications point of view, serving as enablers for many subsequent visually-guided operations, such as autonomous driving and augmented reality. MED-VT and MED-VT++ addressed numerous challenges of these tasks and were shown to yield excellent performance compared to previous state-of-the-art approaches in Chapter 4. The multiscale mechanism and label propagation is evidently found to be successful in addressing common challenges of video segmentation as well as task specific challenges.
5. We presented an extensive interpretability analysis, both quantitative and qualitative, highlighting the importance of multiscale integration in temporally consistent and robust video segmentation. To understand what has been learned by the multiscale encoder, decoder and label propagator, our interpretability study began with a quantitative analysis of the model’s ability to represent static vs. dynamic information across layers as well as individual units. To gain further insights, we followed with qualitative visualizations of the model’s generated feature maps at different stages of the processing pipeline. The resulting solid understanding of the representations learned by the model help decode the black-box of our model and contribute to advancing the understanding of deep learning models of computer vision.

6.2 Societal impact

Unified architectures for video segmentation tasks, e.g., automatic video object segmentation, actor-action segmentation, semantic segmentation and audio visual segmentation, can have multiple positive societal impacts as they can be used for a variety of useful applications.

- Automatic video object segmentation, actor-action segmentation and semantic segmentation can be used for a variety of **autonomous systems**, e.g., autonomous driving

and robot navigation. The class agnostic segmentation of moving objects and actors in autonomous driving can encourage safety critical decision making and reduce accidents from unknown objects outside the closed set of classes predefined in large-scale datasets. Their use in robot navigation also can serve a wide variety of impactful applications, e.g., human-robot interaction.

- By integrating various segmentation tasks into a unified architecture, general surveillance systems can be built for enhanced **safety and security**. In particular, these systems can be used to better detect and track objects in environments with stringent safety requirements, e.g., automatically inspecting Personal Protective Equipment (PPE) use by workers and detecting environmental safety hazards at construction sites. In complement, actor-action segmentation can be used to identify malicious actions and corresponding actors in environments where security is important, e.g., airports. These applications enhance social safety and security by enabling automated and timely identification of potential emergencies, unsafe actions, suspicious actions or intruders.
- Video segmentaion can help improving **health and elder care**. Remote monitoring of patients and elderly people is possible by using video actor-action segmentation to detect alarming incidents, e.g., fall detection. Also, using semantic masks for storage and transmission will provide more privacy to individuals. In complement, video segmentation applied to medical imaging can assist healthcare professionals in diagnosing diseases, monitoring patient conditions and providing personalized treatment plans, thereby improving healthcare outcomes and reducing medical errors.
- Unified video segmentation architectures support applications in **environmental monitoring** and conservation efforts. For instance, by segmenting video data captured by drones or surveillance cameras in natural habitats, researchers can track and analyze wildlife behavior as well as monitor changes in ecosystems. One particular example is the performance of the proposed MED-VT model in segmenting naturally camouflaged animals. The empirical evalutions in this dissertation suggests that these models can be used in very challenging natural scenarios, even those containing camouflaged animals. This information helps inform conservation strategies and contributes to the preservation of biodiversity and natural resources.
- Unified video segmentation architectures can empower **content creators and media professionals** to produce more engaging and immersive experiences across various plat-

forms, including film, television, gaming and virtual reality. By accurately segmenting video content into objects, actions and semantic categories, these systems, enable the creation of interactive narratives, personalized recommendations and augmented reality experiences, thereby enriching entertainment offerings and enhancing user engagement.

Overall, the integration of multiple video segmentation tasks within a unified architecture not only improves the efficiency and effectiveness of video analysis systems but also fosters innovation and positive societal impacts across diverse domains. However, as with many AI abilities, video segmentation also can have negative societal impacts as follows.

- Video segmentation serves as a pivotal tool for automated target detection within **military systems**, enabling various technologies such as object-based, actor-action-based, or semantic-based target detection. These techniques allow for precise identification and tracking of potential threats or objectives in video footage, enabling swift and informed decision-making processes. Moreover, the integration of audio visual sound source segmentation proves invaluable in military applications, particularly in detecting sound-emitting targets, regardless of their semantic categorization. This approach empowers military personnel to pinpoint and assess potential threats solely based on auditory cues, enhancing situational awareness and bolstering overall operational effectiveness. To some extent, movements are emerging to limit such applications, e.g., pledges on the part of researchers to ban use of artificial intelligence in weaponry systems.
- Video segmentation, whether unimodal or multimodal, presents a double-edged sword as it not only enables legitimate applications, e.g., object removal for photo personalization, but also facilitates the **falsification of video documents**, particularly in the era of social media. Advances in unified architectures for precise video segmentation amplify the risk of misinformation dissemination by individuals seeking to deceive, undermining trust in video surveillance systems and potentially falsifying evidence. The technology's misuse extends beyond social media, impacting legal proceedings and investigations. Consequently, urgent measures are needed to develop authentication mechanisms and raise awareness to mitigate the proliferation of manipulated visual content and preserve its integrity.

- The interpretability analysis of MED-VT presented herein employs a post-hoc approach suitable for enhancing model understanding and diagnostic capabilities. However, the inherent **explainability** of video segmentation models in general remains elusive. This absence of an inherent explanatory component, supplementary to primary outputs, obstructs users’ ability to comprehend the circumstances and methodologies underlying potential errors in model predictions. Consequently, deploying such models in safety-critical systems, such as autonomous driving, poses a significant challenge. Without clear insights into the model’s decision-making processes, stakeholders face heightened risks and uncertainties regarding system reliability and safety. Therefore, urgent research efforts should focus not only on interpretability but also on enhancing the explainability of such models. Furthermore, there is a pressing need for models that are equipped with explainable inferences for real world safety-critical applications.

6.3 Future work

In this final section, several directions for future work are suggested.

Universal video segmentation model. There have been recent efforts focused on designing universal segmentation models capable of handling multiple segmentation tasks within a single model using a Multi-Task Learning (MTL) approach [26, 27, 286, 230]. In deep learning, MTL aims to create a single network for multiple tasks, improving generalization, reducing latency and sharing information. The recent advances in artificial intelligence using transformer-based models have led to increased interest in unified modeling approaches [88, 87, 110, 186]. In particular, some video segmentation models have been proposed to combine multiple segmentation tasks, such as semantic segmentation, instance segmentation and panoptic segmentation [103, 5]. MED-VT/MED-VT++ presents a unified general encoder-decoder architecture for unimodal video object segmentation, action-action segmentation, semantic segmentation and multimodal audio visual segmentation. Future work can focus on furthering the generality of MED-VT/ MED-VT++, e.g., via extensions to instance and text referring segmentation, as well as the other tasks overviewed in Chapter 2.

Temporal modeling in videos. Despite the performance gains of recent approaches on current benchmarks in various video understanding tasks, e.g., action recognition [254, 129,

140] and video segmentation [103, 255, 171, 96], interpretability experiments reveal their limited ability in capturing temporal dynamics [200, 112, 85]. Conversely, the interpretability analysis of MED-VT reveal the improvement in modeling temporal dynamics compared to the baseline; however, there is still much room for further improvement. The path toward advancing the abilities of deep learning models may include curating better video datasets with more inclusive motion characteristics, designing training strategies emphasizing motion (e.g., training clip sampling with varying temporal strides and temporal consistency loss) as well as improved model design. Such advances have potential to achieve superior performance in video segmentation in challenging dynamic scenarios. From this point of view, future research on harnessing temporal information in video transformers is of paramount importance.

Better multiscale modeling for segmentation. The requirement of fine localization in video segmentation has motivated approaches for aggregating features from multiple resolutions during encoding [96] and learning object attributes from multiscale features during decoding [37, 35]. However, these methods show limitation in their multiscale modeling in both the encoder and decoder. First, the majority of the methods use attention guided multiscale feature encoding separately at individual scales [54, 129, 37, 35], with only a few efforts using attention across scales [31, 96]. Second, multiscale feature generation with top down semantic propagation is limited to different variants of FPNs [133, 104, 137, 62, 215]. A few recent efforts explored using attention-based approaches for top down semantic propagation [298] and multiscale decoding [96, 37, 35]. These efforts on specific categories of video segmentation (including those of MED-VT) indicate potential of attention-based approaches for multiscale feature aggregation. Further research is required to consider their generalization across a broader range of segmentation tasks.

Interpretability of spatiotemporal modeling. Current analytic methods to evaluate the temporal modeling ability of video transformers are mainly limited to evaluating models on curated or synthetic datasets [200, 85, 112]. Again, the majority of interpretability methods for transformer models are focused on image classification plus modest work on video action recognition, which leaves video segmentation significantly under explored. Visualization based interpretability of video segmentation transformers is very subjective and does not provide quantitative understanding. Lack of research for transformer interpretability with a focus on video segmentation is a notable constraint on the development of video segmentation models capable of generalisation over a wide range of scenarios. Improved interpretability of

algorithms and representations is necessary to truly understand the role of attention on dense video understanding tasks and their enabling mechanisms. From this perspective, there is a demand for further research on video transformer interpretability methods equipped with statistical tools focused on segmentation tasks. Such developments should shed further light into MED-VT/MED-VT++, as well as other contemporary models. Even as it stands, the interpretability tools employed in Chapter 5 to the core MED-VT model also could be applied to MED-VT++ to shed light on the nature of the multimodal representations that it maintains.

Scalable and efficient video transformers. Previous research toward scalable and efficient video transformers primarily uses various heuristics, e.g., attention locality and separability. Such methods proposed to improve transformer efficiency using sparse attention [51], axially separable attention [12] or dynamic sampled attention [298]. Currently, interpretability analysis in natural language processing has improved efficiency by pruning some attention heads. Interpretability-aware redundancy reduction of vision transformers also has been proposed that dynamically drops selected input tokens for computational efficiency [170]. Such interpretability driven scalable and efficiency improvement is an under explored area and can contribute to the computer vision community. With respect to MED-VT, heavy memory consumption (Section 4.7) motivates future work aimed to ameliorate this shortcoming.

Self-explainable video transformers. The growing interest in explainable AI due to social and regulatory reasons has lead to research interest in inherently interpretable vision transformer models [45, 117]. These concerns only add to the fundamental scientific motivation of understanding the details of model operations. This dissertation presented an extensive interpretability analysis, both quantitative and qualitative, for a componentwise understanding of the presented video segmentation model. This analysis is quite useful for diagnostic purposes. The potential next step could be designing a self-explainable video segmentation model that additionally outputs an explanation of the primary output. Currently, there is very limited research on self-explainable models, with a narrow focus on image classification, e.g., concept transformers [191] and B-cos transformers [14]. Hence, self-explainable models for video segmentation are worth further investigation.

Bibliography

- [1] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, 2020. 49
- [2] I. Akhter, M. Ali, M. Faisal, and R. Hartley. EpO-Net: Exploiting geometric constraints on dense trajectories for motion saliency. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1273–1283, 2019. 20, 78
- [3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2425–2433, 2015. 50
- [4] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 2, 16, 23
- [5] A. Athar, A. Hermans, J. Luiten, D. Ramanan, and B. Leibe. TarViS: A unified approach for target-based video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18738–18748, 2023. 36, 41, 110
- [6] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: Robust video object cutout using localized classifiers. *ACM Transactions on Graphics*, 28(3):1–11, 2009. 12
- [7] M. Bakken, J. Kvam, A. A. Stepanov, and A. Berge. Principal feature visualisation in convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, pages 18–31, 2020. 97

- [8] O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, 20(6):1709–1724, 2010. 10
- [9] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3286–3295, 2019. 25
- [10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 75
- [11] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 23
- [12] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning*, pages 813–824, 2021. 37, 50, 60, 112
- [13] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 63–71, 2016. 49
- [14] M. Böhle, M. Fritz, and B. Schiele. Holistically explainable vision transformers. *arXiv preprint arXiv:2301.08669*, 2023. 51, 112
- [15] A. Botach, E. Zheltonozhskii, and C. Baskin. End-to-end referring video object segmentation with multimodal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4985–4995, 2022. 2, 21, 24, 26, 34, 45
- [16] L. Bouchakour and M. Debyeche. MFCCs and Gabor features for improving continuous Arabic speech recognition in mobile communication modified. In *Proceedings of the International Conference on Advanced Aspects of Software Engineering*, pages 115–121, 2018. 21, 46

- [17] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the European Conference on Computer Vision*, pages 282–295, 2010. 10
- [18] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1937–1944, 2011. 10
- [19] S. Buch, C. Eyzaguirre, A. Gaidon, J. Wu, L. Fei-Fei, and J. C. Niebles. Revisiting the “video” in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2927, 2022. 51
- [20] A. Bulat, J. M. Perez Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos. Space-time mixing attention for video transformer. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 19594–19607, 2021. 37, 60
- [21] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018. 3
- [22] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. 2
- [23] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 2018. 16
- [24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 18, 25, 32, 50, 53
- [25] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 18
- [26] R. A. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the International Conference on Machine Learning*, pages 41–48, 1993. 35, 110

- [27] R. A. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997. 35, 110
- [28] H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021. 50
- [29] H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021. 49
- [30] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2722–2730, 2015. 3, 8
- [31] C.-F. R. Chen, Q. Fan, and R. Panda. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 357–366, 2021. 2, 53, 111
- [32] L. Chen, J. Shen, W. Wang, and B. Ni. Video object segmentation via dense trajectories. *IEEE Transactions on Multimedia*, 17(12):2225–2234, 2015. 10
- [33] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 77
- [34] T. Y. Chen, F.-C. Kuo, and R. Merkel. On the statistical properties of the F-measure. In *Proceedings of the IEEE International Conference on Quality Software*, pages 146–153, 2004. 75
- [35] B. Cheng, A. Choudhuri, I. Misra, A. Kirillov, R. Girdhar, and A. G. Schwing. Mask2Former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 2, 3, 4, 21, 23, 28, 33, 53, 111
- [36] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. 21

- [37] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 2, 3, 28, 31, 32, 33, 63, 111
- [38] H. K. Cheng, Y.-W. Tai, and C.-K. Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021. 39
- [39] H. K. Cheng, Y.-W. Tai, and C.-K. Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 11781–11794, 2021. 39
- [40] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 26
- [41] S. Cho, M. Lee, S. Lee, C. Park, D. Kim, and S. Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5140–5149, 2023. 38, 78, 80
- [42] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 21, 22
- [43] A. Choudhuri, G. Chowdhary, and A. G. Schwing. Context-aware relative object queries to unify video instance and panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6377–6386, 2023. 41, 44, 47
- [44] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 9355–9366, 2021. 22

- [45] E. Commission. “Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts”. *Office for Official Publications of the European Communities Luxembourg*, 2021. 3, 112
- [46] A. Criminisi, T. Sharp, C. Rother, and P. Pérez. Geodesic image and video editing. *ACM Transactions on Graphics*, 29(5):134–1, 2010. 12
- [47] K. Dang, C. Zhou, Z. Tu, M. Hoy, J. Dauwels, and J. Yuan. Actor-action semantic segmentation with region masks. In *Proceedings of the British Machine Vision Conference*, 2018. 42, 83
- [48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Association for Computational Linguistics*, pages 4171–4186, 2019. 21
- [49] M. Ding, Z. Wang, B. Zhou, J. Shi, Z. Lu, and P. Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10713–10720, 2020. 42
- [50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 16, 19, 20, 21, 22, 47, 48, 49, 50, 58
- [51] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor. SSTVOS: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5912–5921, 2021. 2, 18, 26, 34, 39, 112
- [52] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002. 10
- [53] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *Proceedings of the British Machine Vision Conference*, 2014. 11

- [54] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multi-scale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2, 4, 24, 50, 51, 52, 53, 60, 111
- [55] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. JumpCut: Non-successive mask transfer and interpolation for video cutout. *ACM Transactions on Graphics*, 34(6):195–1, 2015. 12
- [56] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1846–1853, 2012. 10
- [57] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2014. 10
- [58] R. Gadde, V. Jampani, and P. V. Gehler. Semantic video CNNs through representation warping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4453–4462, 2017. 42
- [59] S. Gao, Z. Chen, G. Chen, W. Wang, and T. Lu. AVSegFormer: Audio-visual segmentation with transformer. *arXiv preprint arXiv:2307.01146*, 2023. 2, 90
- [60] K. Gavriluk, A. Ghodrati, Z. Li, and C. G. Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5966, 2018. 16
- [61] A. Ghiasi, H. Kazemi, E. Borgnia, S. Reich, M. Shu, M. Goldblum, A. G. Wilson, and T. Goldstein. What do vision transformers learn? A visual exploration. *arXiv preprint arXiv:2212.06727*, 2022. 48
- [62] G. Ghiasi, T.-Y. Lin, and Q. V. Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2019. 31, 111
- [63] R. Girshick. Fast R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1440–1448, 2015. 29

- [64] M. Guillaumin, D. Küttel, and V. Ferrari. ImageNet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110:328–348, 2014. 52
- [65] H. Hadizadeh and I. V. Bajić. Saliency-aware video compression. *IEEE Transactions on Image Processing*, 23(1):19–33, 2013. 3
- [66] I. Hadji and R. P. Wildes. What do we understand about convolutional networks? *arXiv preprint arXiv:1803.08834*, 2018. 22
- [67] I. Hadji and R. P. Wildes. Why convolutional networks learn oriented bandpass filters: Theory and empirical support. *arXiv preprint arXiv:2011.14665*, 2020. 47
- [68] B. Han and L. S. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1017–1023, 2011. 10
- [69] M. Han, Y. Wang, Z. Li, L. Yao, X. Chang, and Y. Qiao. HTML: Hybrid Temporal-scale Multimodal Learning framework for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13414–13423, 2023. 45
- [70] S. O. Hansson, M.-Å. Belin, and B. Lundgren. Self-driving vehicles – an ethical overview. *Philosophy & Technology*, 34(4):1383–1408, 2021. 3
- [71] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 297–312, 2014. 12
- [72] J. He, Y. Wang, L. Wang, H. Lu, B. Luo, J.-Y. He, J.-P. Lan, Y. Geng, and X. Xie. Towards deeply unified depth-aware panoptic segmentation with bi-directional guidance learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4111–4121, 2023. 44
- [73] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2961–2969, 2017. 30
- [74] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 18, 20, 21, 22, 23, 46, 47, 68, 71, 77

- [75] M. Heo, S. Hwang, J. Hyun, H. Kim, S. W. Oh, J.-Y. Lee, and S. J. Kim. A generalized framework for video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14623–14632, 2023. 41
- [76] Y. Heo, Y. Jun Koh, and C.-S. Kim. Interactive video object segmentation using global and local transfer modules. In *Proceedings of the European Conference on Computer Vision*, pages 297–313, 2020. 39
- [77] Y. Heo, Y. J. Koh, and C.-S. Kim. Guided interactive video object segmentation using reliability-based attention maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7322–7330, 2021. 39
- [78] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. CNN architectures for large-scale audio classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 131–135, 2017. 22, 71, 88
- [79] H. Holloway, E. K. Jones, A. Kaluzniacki, E. Blasch, and J. Tierno. Activity recognition using video event segmentation with text (VEST). In *Proceedings of the SPIE Conference on Signal Processing, Sensor/Information Fusion, and Target Recognition*, pages 225–234, 2014. 3
- [80] P. Hu, F. Perazzi, F. C. Heilbron, O. Wang, Z. Lin, K. Saenko, and S. Sclaroff. Real-time semantic segmentation with fast attention. *IEEE Robotics and Automation Letters*, 6(1):263–270, 2020. 2, 21
- [81] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *Proceedings of the European Conference on Computer Vision*, pages 786–802, 2018. 11
- [82] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. VideoMatch: Matching based video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 54–70, 2018. 38

- [83] S. Huang, Z. Lu, R. Cheng, and C. He. FaPN: Feature-aligned pyramid network for dense image prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 864–873, 2021. 31
- [84] S. Hwang, M. Heo, S. W. Oh, and S. J. Kim. Video instance segmentation using inter-frame communication transformers. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 13352–13363, 2021. 21, 25, 33, 40
- [85] F. Ilic, T. Pock, and R. P. Wildes. Is appearance free action recognition possible? In *Proceedings of the European Conference on Computer Vision*, pages 156–173, 2022. 50, 111
- [86] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, 1998. 10
- [87] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. J. Henaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. 35, 36, 110
- [88] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the International Conference on Machine Learning*, pages 4651–4664, 2021. 35, 36, 110
- [89] S. Jain, X. Wang, and J. E. Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8866–8875, 2019. 42
- [90] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *Proceedings of the European Conference on Computer Vision*, pages 656–671, 2014. 76
- [91] S. D. Jain, B. Xiong, and K. Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2126, 2017. 2, 38

- [92] G.-P. Ji, K. Fu, Z. Wu, D.-P. Fan, J. Shen, and L. Shao. Full-duplex strategy for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4922–4933, 2021. 20, 78
- [93] J. Ji, S. Buch, A. Soto, and J. C. Niebles. End-to-end joint semantic segmentation of actors and actions in video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 702–717, 2018. 42, 83
- [94] R. Karim, M. A. Islam, and N. D. Bruce. Recurrent iterative gating networks for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1070–1079, 2019. 21
- [95] R. Karim, M. A. Islam, and N. D. Bruce. Distributed iterative gating networks for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2844–2853, 2020. 21
- [96] R. Karim, H. Zhao, R. P. Wildes, and M. Siam. MED-VT: Multiscale encoder-decoder video transformer with application to object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6323–6333, 2023. 2, 21, 111
- [97] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 20, 21, 23
- [98] L. Ke, H. Ding, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu. Video mask transfiner for high-quality video instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 731–747, 2022. 41
- [99] A. Khoreva, A. Rohrbach, and B. Schiele. Video object segmentation with referring expressions. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 7–12, 2018. 45
- [100] A. Khoreva, A. Rohrbach, and B. Schiele. Video object segmentation with language referring expressions. In *Proceedings of the Asian Conference on Computer Vision*, pages 123–141, 2019. 16

- [101] C. Kim and J.-N. Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):122–129, 2002. 3
- [102] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9859–9868, 2020. 14, 16, 36, 43, 47
- [103] D. Kim, J. Xie, H. Wang, S. Qiao, Q. Yu, H.-S. Kim, H. Adam, I. S. Kweon, and L.-C. Chen. TubeFormer-DeepLab: Video mask transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13914–13924, 2022. 2, 21, 23, 36, 40, 43, 44, 86, 87, 110, 111
- [104] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 14, 18, 30, 111
- [105] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. 93
- [106] J. J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 2
- [107] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7417–7425, 2017. 10
- [108] Y. J. Koh, Y.-Y. Lee, and C.-S. Kim. Sequential clique optimization for video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 537–556, 2018. 10
- [109] W. Köhler. *Gestalt Psychology: An Introduction to New Concepts in Modern Psychology*. WW Norton & Company, 1970. 10
- [110] A. Kolesnikov, A. Susano Pinto, L. Beyer, X. Zhai, J. Harmsen, and N. Houlsby. UViM: A unified modeling approach for vision with learned guiding codes. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 26295–26308, 2022. 35, 110

- [111] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *Proceedings of the International Conference on Machine Learning*, pages 3519–3529, 2019. 49
- [112] M. Kowal, M. Siam, M. A. Islam, N. D. Bruce, R. P. Wildes, and K. G. Derpanis. A deeper dive into what deep spatiotemporal networks encode: Quantifying static vs. dynamic information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13999–14009, 2022. 50, 79, 95, 96, 97, 99, 101, 102, 111
- [113] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 109–117, 2011. 65, 79
- [114] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 47
- [115] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 32
- [116] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Proceedings of the Asian Conference on Computer Vision*, pages 488–503, 2020. 16, 74, 77
- [117] Law Commission of Ontario. “Regulating AI: Critical issues and choices”. *LCO Issue Paper*, 2021. 3, 112
- [118] M. Lee, S. Cho, S. Lee, C. Park, and S. Lee. Unsupervised video object segmentation via prototype memory network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5924–5934, 2023. 26, 38, 78, 80
- [119] Y. Lee, H. Seong, and E. Kim. Iteratively selecting an easy reference frame makes unsupervised video object segmentation easier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1245–1253, 2022. 38, 78
- [120] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1995–2002, 2011. 10

- [121] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, 2022. 35
- [122] D. Li, R. Li, L. Wang, Y. Wang, J. Qi, L. Zhang, T. Liu, Q. Xu, and H. Lu. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1297–1305, 2022. 45, 47
- [123] H. Li, P. Xiong, J. An, and L. Wang. Pyramid attention network for semantic segmentation. In *Proceedings of the British Machine Vision Conference*, 2018. 31
- [124] J. Li, W. Wang, J. Chen, L. Niu, J. Si, C. Qian, and L. Zhang. Video semantic segmentation via sparse temporal transformer. In *Proceedings of the ACM International Conference on Multimedia*, pages 59–68, 2021. 2, 21, 42
- [125] J. Li, B. Yu, Y. Rao, J. Zhou, and J. Lu. TCOVIS: Temporally consistent online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1097–1107, 2023. 41
- [126] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. VISUALBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 50
- [127] X. Li, H. Yuan, W. Zhang, G. Cheng, J. Pang, and C. C. Loy. Tube-Link: A flexible cross tube framework for universal video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13923–13933, 2023. 41, 43, 86, 87
- [128] X. Li, W. Zhang, J. Pang, K. Chen, G. Cheng, Y. Tong, and C. C. Loy. Video K-Net: A simple, strong, and unified baseline for video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18847–18857, 2022. 40, 43, 87
- [129] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. MViTv2: Improved multiscale vision transformers for classification and detection. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4804–4814, 2022. 2, 24, 111

- [130] H. Lin, X. Qi, and J. Jia. AGSS-VOS: Attention guided single-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3949–3957, 2019. 39
- [131] H. Lin, R. Wu, S. Liu, J. Lu, and J. Jia. Video instance segmentation with a propose-reduce paradigm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1739–1748, 2021. 40
- [132] K. Lin, L. Li, C.-C. Lin, F. Ahmed, Z. Gan, Z. Liu, Y. Lu, and L. Wang. SWINBERT: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958, 2022. 37
- [133] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 2, 18, 29, 30, 33, 34, 35, 62, 63, 111
- [134] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2980–2988, 2017. 76, 83, 86, 147
- [135] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 13, 21, 50
- [136] Q. Liu, J. Wu, Y. Jiang, X. Bai, A. L. Yuille, and S. Bai. InstMove: Instance motion for object-centric video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6344–6354, 2023. 41
- [137] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 31, 111

- [138] S. Liu, C. Wang, R. Qian, H. Yu, R. Bao, and Y. Sun. Surveillance video parsing with single frame supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 413–421, 2017. 42
- [139] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 21
- [140] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 21, 22, 24, 47, 111
- [141] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 2, 18, 21, 22, 24, 71, 77, 84, 95
- [142] F. Long, Z. Qiu, Y. Pan, T. Yao, J. Luo, and T. Mei. Stand-alone inter-frame attention in video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3192–3201, 2022. 24
- [143] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 76, 83, 86, 89
- [144] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool. Video object segmentation with episodic graph memory networks. In *Proceedings of the European Conference on Computer Vision*, pages 661–679, 2020. 38
- [145] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli. See more, know more: Unsupervised video object segmentation with co-attention Siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3623–3632, 2019. 20, 38, 47, 75, 77, 78, 80
- [146] Y. Lu, X. Bai, L. Shapiro, and J. Wang. Coherent parametric contours for interactive video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 642–650, 2016. 12

- [147] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015. 16
- [148] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–677, 2012. 10
- [149] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5188–5196, 2015. 47
- [150] Y. Mao, N. Wang, W. Zhou, and H. Li. Joint inductive and transductive learning for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9670–9679, 2021. 64, 65
- [151] Y. Mao, J. Zhang, M. Xiang, Y. Lv, Y. Zhong, and Y. Dai. Contrastive conditional latent diffusion for audio-visual segmentation. *arXiv preprint arXiv:2307.16579*, 2023. 2, 90
- [152] Y. Mao, J. Zhang, M. Xiang, Y. Zhong, and Y. Dai. Multimodal variational auto-encoder based audio-visual segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 954–965, 2023. 46
- [153] J. Mei, M. Wang, Y. Lin, Y. Yuan, and Y. Liu. TransVOS: Video object segmentation with transformers. *arXiv preprint arXiv:2106.00588*, 2021. 2, 26, 34
- [154] J. Mei, A. Z. Zhu, X. Yan, H. Yan, S. Qiao, L.-C. Chen, and H. Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 53–72, 2022. 16, 44
- [155] B. Miao, M. Bennamoun, Y. Gao, and A. Mian. Spectrum-guided multi-granularity referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 920–930, 2023. 2, 45
- [156] J. Miao, X. Wang, Y. Wu, W. Li, X. Zhang, Y. Wei, and Y. Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21033–21043, 2022. 16, 36

- [157] J. Miao, Y. Wei, Y. Wu, C. Liang, G. Li, and Y. Yang. VSPW: A large-scale dataset for video scene parsing in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4133–4143, 2021. 16, 42, 85, 87
- [158] J. Miao, Y. Wei, and Y. Yang. Memory aggregation networks for efficient interactive video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10366–10375, 2020. 39
- [159] F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the International Conference on 3D Vision*, pages 565–571, 2016. 76, 86, 147
- [160] C. Molnar. Interpretable machine learning: A guide for making black box models explainable. *Independently Published*, 2022. 3
- [161] A. Morcos, M. Raghu, and S. Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 5732–5741, 2018. 49
- [162] N. S. Nagaraja, F. R. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3235–3243, 2015. 12
- [163] D. Neimark, O. Bar, M. Zohar, and D. Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021. 23
- [164] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 614–621, 2012. 10
- [165] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2013. 10
- [166] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018. 38

- [167] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5247–5256, 2019. 39
- [168] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 38
- [169] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 50
- [170] B. Pan, R. Panda, Y. Jiang, Z. Wang, R. Feris, and A. Oliva. IA-RED²: Interpretability-aware redundancy reduction for vision transformers. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 24898–24911, 2021. 52, 112
- [171] W. Pan, H. Shi, Z. Zhao, J. Zhu, X. He, Z. Pan, L. Gao, J. Yu, F. Wu, and Q. Tian. Wnet: Audio-guided video object segmentation via wavelet-based cross-modal denoising networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1320–1331, 2022. 2, 15, 16, 21, 27, 34, 35, 46, 111
- [172] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1777–1784, 2013. 11
- [173] K. Park, S. Woo, S. W. Oh, I. S. Kweon, and J.-Y. Lee. Per-clip video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1352–1361, 2022. 39
- [174] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In *Proceedings of the International Conference on Machine Learning*, pages 4055–4064, 2018. 25
- [175] M. Paul, M. Danelljan, L. Van Gool, and R. Timofte. Local memory attention for fast video semantic segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1102–1109, 2021. 2, 21

- [176] M. Paul, C. Mayer, L. V. Gool, and R. Timofte. Efficient video semantic segmentation with labels propagation and refinement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2873–2882, 2020. 21, 42
- [177] G. Pei, F. Shen, Y. Yao, G.-S. Xie, Z. Tang, and J. Tang. Hierarchical feature alignment network for unsupervised video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 596–613, 2022. 23, 26, 38, 78, 79, 80
- [178] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 16, 74
- [179] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 16
- [180] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3289, 2012. 16, 75
- [181] J. Qi, Y. Gao, Y. Hu, X. Wang, X. Liu, X. Bai, S. Belongie, A. Yuille, P. H. Torr, and S. Bai. Occluded video instance segmentation: A benchmark. *International Journal of Computer Vision*, 130(8):2022–2039, 2022. 16
- [182] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen. ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3997–4008, 2021. 14, 16, 21, 43, 44
- [183] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 6078–6087, 2017. 49
- [184] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä. Segmenting salient objects from images and videos. In *Proceedings of the European Conference on Computer Vision*, pages 366–379, 2010. 11

- [185] A. J. Rana and Y. S. Rawat. We don't need thousand proposals: Single shot actor-action detection in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2960–2969, 2021. 21, 42, 82, 83, 84
- [186] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. d. Freitas. A generalist agent. *Transactions on Machine Learning Research*, 11:1–42, 2022. 35, 110
- [187] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016. 29
- [188] S. Ren, W. Liu, Y. Liu, H. Chen, G. Han, and S. He. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15455–15464, 2021. 2, 20, 38, 47, 74, 75, 76, 77, 78, 79, 80
- [189] Y. Ren, C.-S. Chua, and Y.-K. Ho. Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24(1-3):183–196, 2003. 10
- [190] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2213–2222, 2017. 16
- [191] M. Rigotti, C. Mikovic, I. Giurgiu, T. Gschwind, and P. Scotton. Attention-based interpretability with concept transformers. In *International Conference on Learning Representations*, 2021. 51, 112
- [192] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2017. 21, 22
- [193] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 2
- [194] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual

- recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 18, 20
- [195] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés. Video transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12922–12943, 2023. 17, 19
- [196] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 618–626, 2017. 49
- [197] S. Seo, J.-Y. Lee, and B. Han. URVOS: Unified referring video object segmentation network with a large-scale benchmark. In *Proceedings of the European Conference on Computer Vision*, pages 208–223, 2020. 16, 45
- [198] H. Seong, J. Hyun, and E. Kim. Kernelized memory network for video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 629–645, 2020. 38
- [199] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim. Hierarchical memory matching network for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12889–12898, 2021. 39
- [200] L. Sevilla-Lara, S. Zha, Z. Yan, V. Goswami, M. Feiszli, and L. Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 535–544, 2021. 111
- [201] X. Shang, D. Di, J. Xiao, Y. Cao, X. Yang, and T.-S. Chua. Annotating objects and relations in user-generated videos. In *Proceedings of the International Conference on Multimedia Retrieval*, pages 279–287, 2019. 82
- [202] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1160, 1998. 10
- [203] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 10

- [204] R. Shi, K. N. Ngan, and S. Li. Jaccard index compensation for object segmentation evaluation. In *IEEE International Conference on Image Processing*, pages 4457–4461, 2014. 75, 76
- [205] M. Siam, C. Jiang, S. Lu, L. Petrich, M. Gamal, M. Elhoseiny, and M. Jagersand. Video object segmentation using teacher-student adaptation in a human robot interaction (HRI) setting. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 50–56, 2019. 38
- [206] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 20, 21, 22
- [207] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European Conference on Computer Vision*, pages 715–731, 2018. 80
- [208] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. 10
- [209] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. 2017. 147
- [210] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. VideoBERT: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 64
- [211] G. Sun, Y. Liu, H. Ding, T. Probst, and L. Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3126–3137, 2022. 2, 21, 23, 43, 87
- [212] G. Sun, Y. Liu, H. Tang, A. Chhatkuli, L. Zhang, and L. Van Gool. Mining relations among cross-frame affinities for video semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 522–539, 2022. 87

- [213] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 18, 20
- [214] H. Tan and M. Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 5100–5111, 2019. 50
- [215] M. Tan, R. Pang, and Q. V. Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. 18, 30, 31, 111
- [216] J. Tang, G. Zheng, and S. Yang. Temporal collection and distribution for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15466–15476, 2023. 45
- [217] Y. Tang, Z. Jiang, Z. Xie, Y. Cao, Z. Zhang, P. H. Torr, and H. Hu. Breaking shortcut: Exploring fully convolutional cycle-consistency for video correspondence learning. *arXiv preprint arXiv:2105.05838*, 2021. 97
- [218] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022. 17
- [219] O. Texler, D. Futschik, M. Kučera, O. Jamriška, Š. Sochorová, M. Chai, S. Tulyakov, and D. Šỳkora. Interactive video stylization using few-shot patch-based training. *ACM Transactions on Graphics*, 39(4):73–1, 2020. 96
- [220] O. Thawakar, S. Narayan, J. Cao, H. Cholakkal, R. M. Anwer, M. H. Khan, S. Khan, M. Felsberg, and F. S. Khan. Video instance segmentation via multi-scale spatio-temporal split attention transformer. In *Proceedings of the European Conference on Computer Vision*, pages 666–681, 2022. 25, 41
- [221] D. M. Thounaojam, A. Trivedi, K. Manglem Singh, and S. Roy. A survey on video segmentation. In *Proceedings of the International Conference on Advanced Computing, Networking, and Informatics*, pages 903–912, 2014. 9

- [222] Y.-L. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1182–1187, 2005. 3, 8
- [223] Z. Tian, C. Shen, and H. Chen. Conditional convolutions for instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 282–298, 2020. 34, 35
- [224] P. H. Torr and A. Zisserman. Concerning Bayesian motion segmentation, model averaging, matching and the trifocal tensor. In *Proceedings of the European Conference on Computer Vision*, pages 511–527, 1998. 10
- [225] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning*, pages 10347–10357, 2021. 23
- [226] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *Proceedings of the European Conference on Computer Vision*, pages 760–775, 2016. 11
- [227] A. Tursunaliyeva, D. L. Alexander, R. Dunne, J. Li, L. Riera, and Y. Zhao. Making sense of machine learning: A review of interpretation techniques and their applications. *Applied Sciences*, 14(2):496, 2024. 3
- [228] A. Ulhaq, N. Akhtar, G. Pogrebna, and A. Mian. Vision transformers for action recognition: A survey. *arXiv preprint arXiv:2209.05700*, 2022. 17
- [229] A. Van Opbroek, M. A. Ikram, M. W. Vernooij, and M. De Bruijne. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE Transactions on Medical Imaging*, 34(5):1018–1030, 2014. 20
- [230] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3614–3633, 2021. 35, 110
- [231] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 6000–6010, 2017. 16, 21, 64

- [232] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen. FEELVOS: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019. 38
- [233] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. MOTs: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019. 12
- [234] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, 2019. 49
- [235] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 66
- [236] H. Wang, W. Wang, and J. Liu. Temporal memory attention for video semantic segmentation. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2254–2258, 2021. 21, 42
- [237] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021. 34, 36
- [238] J. Wang and L. Torresani. Deformable video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14053–14062, 2022. 24
- [239] P. Wang. Bidirectional Cross Attention. <https://github.com/lucidrains/bidirectional-cross-attention>. GitHub repository, Accessed on April 14, 2024. 68
- [240] W. Wang, M. Feiszli, H. Wang, and D. Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10776–10785, 2021. 21

- [241] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao. Zero-shot video object segmentation via attentive graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9236–9245, 2019. 20, 38, 78, 80
- [242] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3395–3402, 2015. 11
- [243] W. Wang, J. Shen, and F. Porikli. Selective video object cutout. *IEEE Transactions on Image Processing*, 26(12):5645–5655, 2017. 12
- [244] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. Hoi, and H. Ling. Learning unsupervised video object segmentation through visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3064–3074, 2019. 20, 78, 80
- [245] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 21, 22, 23, 24
- [246] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 71, 88
- [247] W. Wang, L. Yao, L. Chen, B. Lin, D. Cai, X. He, and W. Liu. CrossFormer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations*, 2022. 2, 53
- [248] X. Wang, R. Girdhar, S. X. Yu, and I. Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 41
- [249] X. Wang, Z. Yu, S. De Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez. FreeSOLO: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022. 40

- [250] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 40
- [251] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021. 4, 18, 21, 25, 28, 32, 33, 34, 35, 40, 52, 53, 60, 61
- [252] S. Woo, D. Kim, J.-Y. Lee, and I. S. Kweon. Learning to associate every segment for video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2705–2714, 2021. 43, 47
- [253] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfnder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997. 10
- [254] C.-Y. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, and C. Feichtenhofer. MeMViT: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 24, 111
- [255] J. Wu, Y. Jiang, S. Bai, W. Zhang, and X. Bai. SeqFormer: Sequential transformer for video instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 553–569, 2022. 21, 23, 25, 33, 40, 111
- [256] J. Wu, Y. Jiang, P. Sun, Z. Yuan, and P. Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4984, 2022. 2, 21, 23, 24, 26, 34, 45
- [257] J. Wu, Q. Liu, Y. Jiang, S. Bai, A. Yuille, and X. Bai. In defense of online models for video instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 588–605, 2022. 41
- [258] J. Wu, S. Yarram, H. Liang, T. Lan, J. Yuan, J. Eledath, and G. Medioni. Efficient video instance segmentation via tracklet query and proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 959–968, 2022. 40, 47

- [259] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Å. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 16
- [260] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 12077–12090, 2021. 21, 22, 23, 24, 43, 47
- [261] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015. 2
- [262] C. Xu and J. J. Corso. Actor-action semantic segmentation with grouping process models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3083–3092, 2016. 82
- [263] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso. Can humans fly? Action understanding with multiple classes of actors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2264–2273, 2015. 13, 16, 82
- [264] H. Xu, L. Yao, W. Zhang, X. Liang, and Z. Li. Auto-FPN: Automatic network architecture adaptation for object detection beyond classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6649–6658, 2019. 18, 31
- [265] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. 3
- [266] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 585–601, 2018. 16, 76

- [267] X. Yan, H. Gong, Y. Jiang, S.-T. Xia, F. Zheng, X. You, and L. Shao. Video scene parsing: An overview of deep learning methods and datasets. *Computer Vision and Image Understanding*, 201, 2020. 13
- [268] C. Yang, H. Lamdouar, E. Lu, A. Zisserman, and W. Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021. 74
- [269] L. Yang, Y. Fan, and N. Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. 12, 16, 47
- [270] S. Yang, Y. Fang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu. Crossover learning for fast online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8043–8052, 2021. 21, 40, 47
- [271] S. Yang, X. Wang, Y. Li, Y. Fang, J. Fang, W. Liu, X. Zhao, and Y. Shan. Temporally efficient vision transformer for video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2885–2895, 2022. 21, 22, 37, 40, 47
- [272] W. Yang, Y. Wei, H. Wei, Y. Chen, G. Huang, X. Li, R. Li, N. Yao, X. Wang, X. Gu, M. B. Amin, and B. Kang. Survey on explainable AI: From approaches, limitations and applications aspects. *Human-Centric Intelligent Systems*, 3(3):161–188, 2023. 3
- [273] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. H. Torr. Anchor diffusion for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 931–940, 2019. 20, 78
- [274] Z. Yang, Y. Wei, and Y. Yang. Collaborative video object segmentation by foreground-background integration. In *Proceedings of the European Conference on Computer Vision*, pages 332–348, 2020. 38
- [275] Z. Yang, Y. Wei, and Y. Yang. Associating objects with transformers for video object segmentation. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 2491–2502, 2021. 39

- [276] Z. Yang, Y. Wei, and Y. Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4701–4712, 2021. 38
- [277] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou. Video object segmentation and tracking: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(4):1–47, 2020. 11
- [278] M. Yi-de, L. Qing, and Q. Zhi-Bai. Automated image segmentation using improved PCNN model based on cross-entropy. In *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 743–746, 2004. 146
- [279] K. Ying, Q. Zhong, W. Mao, Z. Wang, H. Chen, L. Y. Wu, Y. Liu, C. Fan, Y. Zhuge, and C. Shen. CTVIS: Consistent training for online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 899–908, 2023. 41
- [280] H. Yuan, X. Li, Y. Yang, G. Cheng, J. Zhang, Y. Tong, L. Zhang, and D. Tao. PolyphonicFormer: Unified query learning for depth-aware video panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 582–599, 2022. 21, 23, 44
- [281] Y. Yuan, Y. Wang, L. Wang, X. Zhao, H. Lu, Y. Wang, W. Su, and L. Zhang. Isomer: Isomeric transformer for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 966–976, 2023. 38, 78, 79
- [282] S. Zagoruyko and N. Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016. 21, 22
- [283] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 818–833, 2014. 47
- [284] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 628–635, 2013. 10

- [285] T. Zhang, X. Tian, Y. Wu, S. Ji, X. Wang, Y. Zhang, and P. Wan. DVIS: Decoupled video instance segmentation framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1282–1291, 2023. 41
- [286] Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021. 35, 110
- [287] Y.-J. Zhang. *Advances in Image and Video Segmentation*, chapter 1: An Overview of Image and Video Segmentation in the Last 40 Years, pages 1–16. IGI Global, Hershey, Pennsylvania, 2006. 9
- [288] Y. Zhao, Y. Xiong, and D. Lin. Trajectory convolution for action recognition. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 2204–2215, 2018. 25
- [289] M. Zhen, S. Li, L. Zhou, J. Shang, H. Feng, T. Fang, and L. Quan. Learning discriminative feature with CRF for unsupervised video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 445–462, 2020. 20, 78, 79
- [290] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *ACM Transactions on Graphics*, 31(6):1–10, 2012. 12
- [291] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong. Squeeze-and-attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13065–13074, 2020. 21
- [292] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 321–328, 2003. 64, 66
- [293] J. Zhou, J. Wang, J. Zhang, W. Sun, J. Zhang, S. Birchfield, D. Guo, L. Kong, M. Wang, and Y. Zhong. Audio-visual segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 386–403, 2022. 2, 15, 16, 21, 46, 53, 87, 89, 90, 147
- [294] T. Zhou, F. Porikli, D. J. Crandall, L. Van Gool, and W. Wang. A survey on deep learning technique for video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7099–7122, 2022. 3, 8, 9, 11, 13

- [295] T. Zhou, S. Wang, Y. Zhou, Y. Yao, J. Li, and L. Shao. Motion-attentive transition for zero-shot video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13066–13073, 2020. 2, 20, 38, 47, 74, 75, 76, 77, 78, 80
- [296] T. Zhou, W. Wang, S. Qi, H. Ling, and J. Shen. Cascaded human-object interaction recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4263–4272, 2020. 8
- [297] Y. Zhou, H. Zhang, H. Lee, S. Sun, P. Li, Y. Zhu, B. Yoo, X. Qi, and J.-J. Han. Slot-VPS: Object-centric representation learning for video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3093–3103, 2022. 2, 21, 23, 43, 47
- [298] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2, 18, 24, 25, 26, 31, 33, 37, 40, 111, 112

Appendix A

Loss functions

In Chapter 4 several loss functions were mentioned in conjunction with training details for the various tasks to which MED-VT/MED-VT++ was applied. To focus on our original contributions, these standard functions were not defined formally in that chapter. Here, for the sake of completeness we provide the formal definitions.

To present formal definition of the loss functions, let $\pi = (x, y, t)$ index spatiotemporal position, while $Y(\pi)$ and $\hat{Y}(\pi)$ are binary groundtruth and our predicted foreground probability masks, resp. The definition of the individual loss functions are as follows.

Cross-entropy loss. The pixelwise binary cross-entropy loss [278] is defined as

$$\mathcal{L}_{\text{bce}}(Y, \hat{Y}) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} -\log \left(\delta(Y(\pi), \hat{Y}(\pi)) \right), \quad (\text{A.1})$$

with

$$\delta \left(Y(\pi), \hat{Y}(\pi) \right) = \begin{cases} \hat{Y}(\pi) & \text{if } Y(\pi) = 1 \\ 1 - \hat{Y}(\pi) & \text{otherwise,} \end{cases}$$

where Π is the spatiotemporal support of Y and $\hat{Y}$, i.e., $\pi \in \Pi$.

Focal loss. The distribution-based focal loss [134] is defined as

$$\mathcal{L}_{\text{focal}}(Y, \hat{Y}) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} -\alpha \left(1 - \delta(Y(\pi), \hat{Y}(\pi))\right)^\gamma \log \left(\delta(Y(\pi), \hat{Y}(\pi))\right), \quad (\text{A.2})$$

with

$$\delta(Y(\pi), \hat{Y}(\pi)) = \begin{cases} \hat{Y}(\pi) & \text{if } Y(\pi) = 1 \\ 1 - \hat{Y}(\pi) & \text{otherwise,} \end{cases}$$

where Π is the spatiotemporal support of Y and $\hat{Y}$, i.e., $\pi \in \Pi$. The focal loss, equation A.2, is a modified cross-entropy that places more emphasis on hard examples by means of a modulating factor, $1 - \delta(Y, \hat{Y})^\gamma$, and a balancing factor α . Following [134], we use $\alpha = 0.25$ and $\gamma = 2$.

Dice loss. The region-based Dice loss [159] is defined as

$$\mathcal{L}_{\text{dice}}(Y, \hat{Y}) = 1 - \frac{\sum_{\pi \in \Pi} 2 \times Y(\pi) \times \hat{Y}(\pi) + 1}{\sum_{\pi \in \Pi} Y(\pi) + \sum_{\pi \in \Pi} \hat{Y}(\pi) + 1}. \quad (\text{A.3})$$

The Dice loss, equation A.3, is adapted from the Dice coefficient and used for tasks where region similarity serves as the performance metric [209], specialized here for the two class case.

Kullback-Leibler (KL) divergence loss. The Kullback-Leibler (KL) divergence loss between two modalities of features [293], e.g., visual features, $Z(x, y, t)$, and audio features, $A(t)$, is defined as

$$\mathcal{L}_{\text{kld}}\left(\hat{Y}(x, y, t), Z(x, y, t), A(t)\right) = \sum_{t \in T} KL\left(\left(\frac{1}{|X \cdot Y|} \sum_{x \in X, y \in Y} Z(x, y, t) \cdot \hat{Y}(x, y, t)\right), A(t)\right), \quad (\text{A.4})$$

where X and Y are spatial width and height, respectively, and KL is the Kullback-Leibler divergence. The average over the spatial area ensures that the masked visual features are mapped to the shape of the audio features which has only temporal dimension.