Advancing Video Segmentation by Integrating Advanced Learning Paradigms

by

Islam Abdelfattah

M.Eng., University of Ottawa, 2022
B.Sc., Hons., Arab Academy for Science, Technology & Maritime Transport , 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The College of Graduate Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

October 2024

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis entitled:

Advancing Video Segmentation by Integrating Advanced Learning Paradigms

submitted by Islam Abdelfattah in partial fulfillment of the requirements

of the degree of Master of Science

Examining Committee:

Mohamed S. Shehata, The University of British Columbia, Okanagan Campus
Supervisor

Fatemeh Fard, The University of British Columbia, Okanagan Campus
Supervisory Committee Member

Gema Rodríguez-Pérez, The University of British Columbia, Okanagan Campus
Supervisory Committee Member

Shahria Alam, The University of British Columbia, Okanagan Campus
University Examiner

Abstract

Video segmentation involves isolating specific objects or the foreground from the background in a video, for example, separating people or vehicles from the rest of the scene. This task is crucial in many applications such as video editing, surveillance, autonomous vehicles, and augmented reality. Advanced methods often utilize Deep Learning (DL) to manage the complexities of video sequences, including motion and occlusion. Despite recent progress, current state-of-the-art methods face challenges such as catastrophic forgetting, poor domain generalization, and the loss of object details when objects are occluded by other objects.

This thesis addresses these critical challenges in video segmentation by proposing innovative methodologies to enhance segmentation accuracy, robustness, and domain generalization. We introduce a Knowledge Distillation Network (KDNet) and a new training technique to combat catastrophic forgetting and improve domain generalization by leveraging advanced learning paradigms such as federated learning, continual learning, knowledge distillation, and few-shot learning. This approach yields significant improvements across diverse datasets while ensuring data privacy. Additionally, we propose a Mask Enhancement Model (MEM) to mitigate the loss of object details by integrating a foundational model output and employing mask fusion techniques, resulting in better segmentation masks.

Experimental results demonstrate KDNet's ability to achieve state-of-the-art domain generalization performance on multiple datasets in foreground segmentation and adapt to new unseen domains using as few as one labelled image (one-shot learning). MEM also proves effective, versatile, and robust performance, achieving state-of-the-art results in Semi-Supervised Video Object Segmentation (S-SVOS).

Lay Summary

Video segmentation separates important objects, such as people or cars, from the background in videos, which is essential for tasks such as video editing, security monitoring, and self-driving cars. Current methods that are using Deep Learning face challenges like forgetting learned information, poor performance on new videos, and losing object details when occluded.

This thesis introduces KDNet and a new training technique to prevent forgetting and enhance domain generalization performance using federated learning, knowledge distillation, continual learning, and few-shot learning, ensuring better results and data privacy. It also presents MEM, which preserves object details by combining outputs from different state-of-the-art models for superior segmentation masks.

Experiments show that KDNet achieves state-of-the-art performance in domain generalization on multiple datasets in foreground segmentation and excels in adapting to new unseen video types with minimal labelled data, and MEM achieves state-of-the-art results in semi-supervised video object segmenting.

Preface

Chapters 3 and 4 of this thesis contain content from work submitted to conferences in which Islam Abdelfattah was both primary and co-author. Other co-authors for published or submitted work include Islam Osman and Mohamed S. Shehata.

Chapter 3 contains a version of content from "Domain Generalization for Foreground Segmentation Using Federated Learning" which was published in "Bebis, G., et al. Advances in Visual Computing. ISVC 2023. Lecture Notes in Computer Science, vol 14361. Springer, Cham". This work has been a collaborative work between Islam Osman and Islam Abdelfattah. Mohamed S. Shehata supervised all research and technical contributions and reviewed the final manuscript before submission. The initial concept for this research was proposed by Islam Osman, and the entirety of the work—including experiments, code development, and paper writing—was carried out through equal collaboration.

Chapter 4 contains a version of content from "MEM: Mask Enhancement Model for Video Object Segmentation" which was submitted and accepted by the International Symposium on Visual Computing (ISVC). Islam Abdelfattah led all research for this work. Mohamed S. Shehata oversaw the supervision of the research and reviewed the final manuscript before submission.

Grammarly and ChatGPT were occasionally used to improve writing during the preparation of this thesis (e.g., for grammar and typo checks). No other generative AI tools or technologies were employed for data analysis, writing, editing, or any other part of this document's development.

Table of Contents

List of Tables

List of Figures

Glossary of Notation

KDNet	Knowledge Distillation Network
MEM	Mask Enhancement Model
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
GMM	Gaussian Mixture Model
MRFs	Markov Random Fields
CRFs	Conditional Random Fields
SVMs	Support Vector Machines
CNNs	Convolutional Neural Networks
FCNs	Fully Convolutional Networks
STCN	Space-Time Correspondence Network
DAVIS16	Densely Annotated VIdeo Segmentation 2016
DAVIS17	Densely Annotated VIdeo Segmentation 2017
CDNet	ChangeDetection.Net
VOS	Video Object Segmentation
IOU	Intersection Over Union
SVOS	Supervised Video Object Segmentation
RNNs	Recurrent Neural Networks
S-SVOS	Semi-Supervised Video Object Segmentation

UVOS	Unsupervised Video Object Segmentation
GANs	Generative Adversarial Networks
MAE	Maske Auto Encoders
SAM	Segment Anything Model

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Mohamed Shehata, for his support, encouragement, and patience throughout my studies. I would also like to extend my sincerest thanks to Islam Osman, my lab mate and dear friend. Your valuable feedback, support and guidance made it possible for this work to fruit.

Dedication

I dedicate this work to my loving family and dear friends. To my mom, dad, and my two sisters, your love and support have given me a calm heart and unwavering determination. To my close and dear friends Moataz, Mokhtar, Omar, Ali, and Leandra, your constant support helped me through the most challenging moments of my studies and life when things seemed bleak. I will always be grateful for having you as part of my life.

Chapter 1

Introduction

Over the past few decades, video segmentation tasks have advanced significantly due to progress in computer vision and machine learning. These techniques are crucial for various applications, including video surveillance, autonomous driving, human-computer interaction, and video editing. Video segmentation encompasses multiple tasks, such as foreground segmentation, video object segmentation, video instance segmentation and more. The common goal among these tasks is to analyze a video sequence and extract valuable information from it. This could involve segmenting the entire foreground as a single entity from the background, e.g., foreground segmentation, isolating each object within the foreground, e.g., video object segmentation, or isolating each object based on its class, e.g., video instance segmentation.

Initially, video segmentation techniques depended on statistical methods in the early 2000s. However, a revolution occurred in the late 2000s with the emergence of Artificial Intelligence (AI) and Machine Learning (ML). Although these advancements addressed numerous problems, they also introduced new challenges.

Current state-of-the-art methods face challenges like catastrophic forgetting, poor domain generalization, and the loss of object details when objects are close to or occluded by other objects. Catastrophic forgetting is a phenomenon in machine learning, particularly in neural networks, where a model abruptly and significantly loses performance on previously learned tasks when trained on new tasks. This occurs because the learning process for the new task interferes with the neural pathways established for the old tasks, leading the model to "forget" what it had previously learned. This is a significant issue in fields like continual learning, where models need to learn new information over time without losing the ability to perform previously learned tasks. Poor domain generalization refers to the inability of a machine learning model to perform well on data that is significantly different from the data it was trained on. In other words, a model with poor domain generalization performs well on the training data (or data similar to it) but fails to generalize its knowledge to new, unseen data from different domains or distributions. This is a critical issue because real-world applications of-

ten involve deploying models in varied environments, and a lack of domain generalization limits the model's usefulness and robustness in such scenarios. In video segmentation, occlusion refers to the situation where one object in a video frame overlaps or hides another object, making it challenging to accurately segment and track the obscured object. Occlusion can significantly complicate the task of identifying and following objects over time because the object may be partially or entirely covered by other objects or scenes, leading to gaps or ambiguities in the segmentation process. Effective video segmentation algorithms need to handle occlusion by predicting or inferring the presence of the occluded parts of objects and maintaining consistent tracking despite the partial visibility. Techniques to manage occlusion often involve leveraging temporal information from previous frames, using context from surrounding pixels, or employing sophisticated models that can infer the obscured regions based on learned patterns.

This thesis addresses these critical challenges in video segmentation by proposing innovative methodologies to enhance segmentation accuracy, robustness, and domain generalization in video segmentation tasks. We introduce KDNet and a new training technique to combat catastrophic forgetting and improve domain generalization by leveraging advanced learning paradigms such as federated learning, continual learning, knowledge distillation, and few-shot learning. These approaches yield significant improvements across diverse datasets while ensuring data privacy. Additionally, we propose a novel model, MEM, to mitigate the loss of object details by integrating foundational model outputs and employing mask fusion techniques, resulting in superior segmentation masks.

1.1 Research Motivation

The motivation for this research stems from the ongoing challenges in deep learning, particularly in video segmentation tasks. Existing video segmentation methods often struggle when trained on diverse datasets from varying distributions/domains or when subject to unseen data. Moreover, they face difficulties in accurately segmenting objects in the presence of occlusions or sudden changes in appearance within video sequences. This research is fueled by a strong interest in investigating innovative and unconventional learning paradigms and combining them with current techniques to tackle these challenges. Our aim is to contribute to this field of study and pave the way for new and exciting research directions.

1.2 Research Objective

To build models that achieve state-of-the-art performances while overcoming the existing problems in video segmentation tasks.

1. Propose a novel deep-learning model and training technique to learn from single or multiple datasets effectively. Hence, achieves state-of-the-art domain generalization performance on multiple datasets.

2. Propose a training technique that allows the model to adapt to new domains using as few as one labelled image (One-shot).

3. Develop an approach that adapts the foundational model SAM for video Object Segmentation tasks to make it a zero-shot VOS model, This approach produces state-of-the-art performance in Unsupervised Video Object Segmentation (UVOS)

4. Propose a model that combines the strengths of both SAM (Foundational model) and Cutie (current state-of-the-art in Video Object Segmentation (VOS)), producing state-of-the-art results in VOS.

1.3 Publications

This thesis is based on the following published/submitted work:

1. Osman, I., Abdelfattah, I., Shehata, M.S. (2023). Domain Generalization for Foreground Segmentation Using Federated Learning. In: Bebis, G., et al. Advances in Visual Computing. ISVC 2023. Lecture Notes in Computer Science, vol 14361. Springer, Cham.

2. MEM: Mask enhancement model for Video Object Segmentation. Abdelfattah, I., Shehata, M.S., International Symposium on Visual Computing (ISVC), 2024. (Accepted)

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 provides a comprehensive background and literature review of video segmentation tasks, including foreground segmentation and video object segmentation. It explores mainstream approaches such as supervised, semi-supervised, and unsupervised methods and lays a foundation for the following technical chapters.

Chapter 3 introduces a novel deep-learning model and training technique that effectively learns from single or multiple datasets, achieving state-of-the-art out-of-domain performance across multiple datasets. This chapter is based on the published paper "Domain Generalization for Foreground Segmentation Using Federated Learning.".

Chapter 4 proposes a model that integrates the strengths of SAM (a foundational model) and Cutie (the current state-of-the-art in video object segmentation), resulting in state-of-the-art VOS performance. This chapter is based on the submitted paper "MEM: Mask Enhancement Model for Video Object Segmentation.".

Finally, Chapter 5 concludes the thesis and suggests avenues for future research in this area.

Chapter 2

Background and Related Work

2.1 Overview

Video segmentation began with basic image processing techniques. In the early days, background subtraction [2] was the most common approach. This method is a gaussian mixture-based background/ foreground segmentation algorithm that involves comparing each frame of a video with a background model to identify moving objects. Techniques such as frame differencing and median filtering were employed to create a static background model. However, these methods struggled with dynamic backgrounds, lighting changes, and camera movements. To address the limitations of simple background subtraction, more sophisticated statistical models were introduced. One notable advancement was the Gaussian Mixture Model (GMM) [3]. GMM models the background as a mixture of multiple gaussian distributions, allowing it to adapt to gradual changes in the scene. This approach significantly improved the robustness of video segmentation tasks in real-world scenarios.

In the early 2000s, the use of Markov Random Fields (MRFs) [4] and Conditional Random Fields (CRFs) [5] became popular. These probabilistic models consider spatial and temporal dependencies between pixels, leading to more accurate segmentation results. By modelling the relationships between neighbouring pixels, MRFs and CRFs can better handle noise and preserve object boundaries.

The advancements in machine learning in the late 2000s brought new possibilities for video segmentation tasks like foreground segmentation, where each frame of the video is separated into a background and a foreground that is seen as a single object. Support Vector Machines (SVMs) [6] and other classifiers were used to distinguish foreground objects from the background based on features such as colour, texture, and motion. These methods leveraged training data to learn discriminative models, improving the accuracy and generalization of segmentation algorithms.

The advent of deep learning in the 2010s revolutionized the field of video segmentation. Convolutional Neural Networks (CNNs) [7] and Fully Convolutional Networks (FCNs) [8] demonstrated unprecedented performance in

various computer vision tasks, including video segmentation tasks. Methods such as DeepLab [9], Mask R-CNN [10], and U-Net [11] set new benchmarks by leveraging deep hierarchical features and end-to-end learning frameworks.

2.2 Video Segmentation Tasks

With the continuous advancements in video segmentation, various specialized tasks have emerged, each addressing unique challenges and applications. Foreground segmentation focuses on distinguishing foreground objects as a single object from the background. Video object segmentation aims to segment and track multi-objects of interest throughout the video. Video instance segmentation extends this by differentiating between multiple instances of the same object class, providing detailed and instance-specific object tracking necessary for complex scene understanding. Moving object segmentation specifically targets the detection and segmentation of objects in motion, essential for dynamic environments and activity recognition. In this thesis, the main focus relies on foreground and video object segmentation. These tasks saw significant advancements with deep learning.

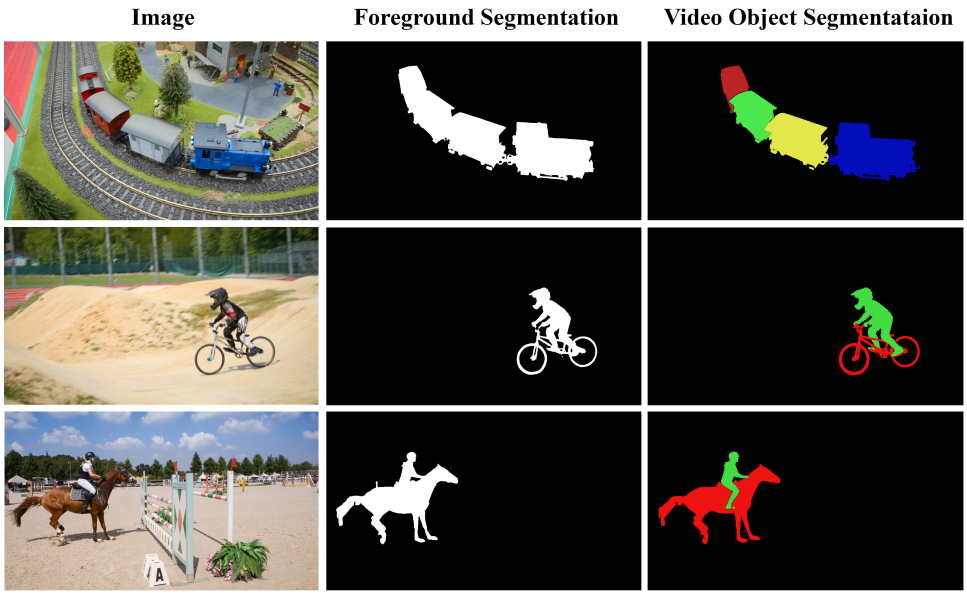

Figure 2.1: Foreground segmentation vs video object segmentation. Images are from DAVIS 17 Dataset [1]

2.2.1 Foreground Segmentation

2.2.1.1 Problem definition

Foreground segmentation is a method in computer vision that distinguishes objects from the background in a sequence of frames. This segmentation results in a binary image in which each pixel is assigned a value of 0 or 1, indicating whether it is part of the background (0) or the foreground (1) as seen in Figure 2.1. The most used model architecture for this task is a CNN-based Encoder-Decoder architecture.

2.2.1.2 Foreground Segmentation Related Work

Foreground segmentation models are showing outstanding performance nowadays for the segmentation task. Foreground segmentation models could be categorized into statistical-based models and deep-learning-based models. Statistical-based foreground segmentation models depend mainly on the change in the pixel intensity of the background of any image and also depend on the reasons behind this change to be able to separate the objects from the background. Furthermore, statistical methods also depend on motion vectors to separate the object pixels from the foreground. Optical flow is a widely used statistical method that depends on motion vectors. Optical flow depends on processing a sequence of images to approximate the 3-D velocities of surface points. Then, these vectors could be used for various tasks, including object segmentation. Optical flow has been used for foreground segmentation as seen in [12] and [13]. Statistical models struggle to isolate the background when environmental challenges exist in the image, like illumination change, moving camera, and dynamic background.

Recently, deep neural networks have been used for the task of foreground segmentation, and they have been showing a significant improvement in the task of foreground segmentation. Deep-learning networks are divided into CNNs and transformers. One of the earlier CNN models that used an encoder-decoder type neural network is FgSegNet [14]. Several models use neural network-based architectures like CascadeCNN [15], which is based on a multi-resolution CNNs with a cascaded architecture. RANet+[16], which is Ranking Attention Network, uses an encoder-decoder framework to learn pixel-level similarity and segmentation in an end-to-end manner. The most recent model, Space-Time Correspondence Network (STCN) [17], uses a Space-Time Correspondence Network and negative squared Euclidean distance to compute affinities. On the other hand, transformer-based models like TransBlast [18] are also being used and showing promising results. Trans-

Blast is trained using self-supervised learning to leverage the limited data and learn a strong object representation while using the augmented subspace loss function. However, all these models suffer from catastrophic forgetting when subject to domain shifts. This thesis proposes a federated learning technique to mitigate the effect of catastrophic forgetting and achieve domain generalization, even though federated learning is being used mainly for domain generalization on multiple mobile devices.

2.2.1.3 Foreground Segmentation Datasets

Densely Annotated VIdeo Segmentation 2016 (DAVIS16) [1] is a benchmark for evaluating video object segmentation methods, featuring 50 high-resolution Full HD videos with pixel-level ground truth annotations for each frame. The primary task is to segment and track foreground objects across complex scenes, which include challenges like dynamic backgrounds, occlusions, and varying object motions. It provides dense, consistent annotations and is commonly used to assess performance through metrics like the Jaccard Index (IoU) and F-measure, making it a key dataset for video object segmentation, tracking, and temporal consistency evaluation.

SegTrackV2 [19] is a benchmark used for video object segmentation, consisting of 14 video sequences with pixel-level ground truth annotations, and the number of frames ranged from 21 to 279 in each video. These sequences include complex challenges like occlusions, overlapping objects, and changes in object appearance. Some videos contain multiple objects, making the task more difficult by requiring accurate multi-object segmentation. The dataset is widely used to evaluate segmentation methods based on both spatial accuracy and temporal consistency, making it suitable for benchmarking algorithms in video object segmentation and tracking.

ChangeDetection.Net (CDNet) 2014 [20] is a benchmark dataset that detects changes between video frames. These changes are the moving objects. The dataset consists of 11 different challenges. Each challenge has a number of videos ranging from 4 to 6, and the number of frames in each video ranges from 1000 to 8000. The total number of videos in the dataset is 53 videos. It covers diverse challenges like dynamic backgrounds, camera jitter, illumination changes, and various object motions. The dataset includes ground truth masks for foreground objects and provides different categories, such as shadows, night videos, and weather changes, to assess robustness

across scenarios. CDNet is widely used to benchmark methods for change detection, foreground-background segmentation, and object tracking.

2.2.1.4 Evaluation Metrics

The evaluation metrics used in foreground segmentation:

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.3}$$

$$FNR = \frac{FN}{TP + FN} \tag{2.4}$$

$$FM = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2.5}$$

TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. FM is the F1-Measure. FPR and FNR are the false-positive rate and false-negative rate. Precision indicates the percentage of pixels correctly classified as moving objects, while Recall represents the percentage of object pixels correctly identified. The false-positive rate shows the percentage of background pixels incorrectly classified as foreground, and the false-negative rate indicates the percentage of foreground pixels mistakenly identified as background. The F1-measure, a key metric in foreground segmentation, is the harmonic mean of precision and recall.

2.2.2 Video Object Segmentation

2.2.2.1 Problem definition

Video Object segmentation is a method in computer vision that isolates objects from the background and each other in a sequence of frames. This segmentation results in a multi-class image in which each pixel is assigned a value of 0 to n, indicating whether it is part of the background (0) or the objects (n) as seen in Figure 2.1. The most used model architecture for this task is an Encoder-Decoder architecture.

2.2.2.2 Video Object segmentation Related Work

VOS has seen significant advancements over recent years, driven by innovations in deep learning and computer vision. This section reviews the three main types of VOS and the latest and most impactful models in each type.

Supervised Video Object Segmentation (SVOS) involves training models with a large dataset of annotated videos where the objects of interest are labelled frame by frame. Traditional techniques in SVOS relied on methods like optical flow [21] and graph-based segmentation [22], which often struggled with issues of scalability and generalization. The rise of deep learning brought about CNNs and Recurrent Neural Networks (RNNs), which improved the accuracy and robustness of SVOS. However, these methods require extensive labelled data and are prone to catastrophic forgetting, where the model forgets previously learned information when trained on new data. Recently, models such as Mask R-CNN [10] and the development of temporal convolutional networks have further enhanced SVOS, yet the need for large, annotated datasets remains a significant bottleneck.

Semi-supervised Video Object Segmentation (S-SVOS) aims to reduce the reliance on extensive annotated data by leveraging a few annotated frames and propagating this information across the video. Early techniques employed keyframe selection and interpolation methods, which often resulted in suboptimal segmentations due to error accumulation over frames. With the advent of AI, S-SVOS approaches began to utilize deep learning models that combine both spatial and temporal information to improve segmentation accuracy. Current techniques could be categorized into memory-based methods, transformer-based approaches, and object-level reasoning techniques.

Memory-based S-SVOS techniques utilize past frames to inform the segmentation of current frames, enhancing temporal coherence and robustness.

FEELVOS [23] achieves real-time performance by utilizing pixel-wise embeddings from previous frames. FEELVOS employs a distance-based matching mechanism between the embeddings of the current frame and a reference frame, allowing it to segment objects efficiently without extensive computational resources. STM[24] introduced a novel memory read-write mechanism, where previous frames are stored in a memory bank and matched with the current frame to propagate the segmentation mask. This approach significantly improves the accuracy of VOS by maintaining detailed temporal information through frames. MiVOS [25] extends the STM framework by incorporating multi-level memory features. This model not only stores high-level semantic information but also retains low-level pixel details, enabling it to handle complex scenes with intricate object interactions. XMem [26] introduces multi-scale memory representations to better handle occlusions and rapid object movements. By leveraging both local and global memory features, XMem can robustly track and segment objects even in challenging scenarios

Transformers have brought significant improvements to S-SVOS by employing attention mechanisms to model long-range dependencies between pixels across frames. AOT [27] employs an associative attention mechanism to dynamically link object instances across frames. This approach ensures that the model maintains consistent object identities throughout a video sequence, providing robust performance in complex environments. An adaptation of DETR for VOS [28], this model extends object detection capabilities to handle the temporal dimension. SST [29] focuses on reducing the computational complexity of transformers by applying sparse attention to regions of interest. Although this approach can sometimes compromise accuracy, it significantly improves processing speed, making it suitable for real-time applications.

Object-level reasoning approaches focus on modelling and tracking individual objects, providing a more holistic understanding of video content. ISVOS [30] injects instance-level features from a pre-trained instance segmentation network into a memory-based VOS framework. This integration enhances the performance of VOS by leveraging detailed instance-level information, although it requires separate instance segmentation steps. Cutie [31] introduces object-level memory reading and query-based transformers, achieving state-of-the-art performance while maintaining computational efficiency. This model can robustly segment objects even in complex scenarios, avoiding the need for computationally expensive operations between high-resolution feature maps.

The robustness and efficiency of modern VOS methods facilitate their integration into broader video segmentation pipelines, addressing tasks such as

open-vocabulary segmentation and unsupervised segmentation. Track Anything [32] combines robust object tracking with segmentation capabilities, enabling seamless tracking and segmentation across diverse video sequences. Track Anything leverages advancements in VOS to handle complex video scenes in real-time. DEVA [33] utilizes VOS techniques to automatically annotate video events, demonstrating the applicability of VOS in real-world scenarios. This model integrates VOS into comprehensive video understanding tasks, enhancing the automation of video analysis.

The field of VOS continues to evolve with innovative approaches that push the boundaries of accuracy and efficiency. Memory-based methods, transformer architectures, and object-level reasoning techniques are pivotal in addressing the inherent challenges of S-SVOS. Future research will likely build on these foundations, exploring new ways to integrate and optimize these techniques for even more robust and versatile video segmentation solutions.

Unsupervised Video Object Segmentation (UVOS) is the most challenging task, which involves segmenting objects without any labelled data. Traditional methods relied on motion segmentation and clustering techniques, which were limited by their inability to differentiate between objects with similar motion patterns effectively. The introduction of deep learning has led to the development of more sophisticated models that utilize Generative Adversarial Networks (GANs) [34] and self-supervised learning techniques. These models can learn robust features from vast amounts of unlabeled data, yet they often suffer from generalization issues and the problem of distinguishing between objects in complex scenes. Recent approaches have focused on leveraging spatiotemporal coherence and attention mechanisms to improve the accuracy and reliability of UVOS.

2.2.2.3 Video Object Segmentation Datasets

Densely Annotated VIdeo Segmentation 2017 (DAVIS17)[1] is a benchmark for video object segmentation, offering high-resolution videos with pixel-level annotations for diverse and challenging sequences. It includes 60 videos for training, 30 for validation, and 60 for testing (30 in the test-dev set and 30 in the test-challenge set). It supports semi-supervised and unsupervised segmentation tasks, providing detailed evaluation metrics like the Jaccard index.

YouTube-VOS 2019 dataset [35] is a large-scale benchmark for video object segmentation, offering high-resolution videos with detailed annotations. It includes 3,471 videos for training, 507 for validation, and 541 for testing. Supporting both semi-supervised and unsupervised tasks, this dataset provides extensive data for training and benchmarking segmentation models.

2.2.2.4 Evaluation Metrics

Video object segmentation evaluation metrics:

$$\mathcal{J}(U, V) = \frac{|U \cap V|}{|U \cup V|} \tag{2.6}$$

$$\mathcal{J}\&\mathcal{F} = \frac{\mathcal{J} + \mathcal{F}}{2} \tag{2.7}$$

$\mathcal{J}$ is Jaccard index, $\mathcal{F}$ is the F measure. The Jaccard index is calculated by calculating the intersection over union between the detected object pixels and the object ground truth pixels similar to Intersection Over Union (IOU). $\mathcal{J}\&\mathcal{F}$ is the average of both the Jaccard index and F measure and is considered the main metric in Video object segmentation.

In this thesis, we employ and integrate advanced learning paradigms with the proposed models to create models that are robust, generalized, and capable of achieving state-of-the-art performance in video segmentation tasks as demonstrated in the technical chapters 3 and 4.

Chapter 3

KDNet: Domain Generalization for Foreground Segmentation Using Federated Learning

3.1 Overview and Motivation

In this chapter, we propose a deep learning model and a novel training procedure to learn from multiple domains while achieving domain generalization and overcoming catastrophic forgetting by employing continual learning techniques. The proposed model is an encoder with a multi-head decoder that combines shared and isolated parameters for different domains. The proposed training procedure is inspired by federated learning, where client models are created such that each client learns from a specific domain. After that, the main model learns the knowledge gained by each client using knowledge distillation with regularization to prevent the main network from forgetting the previously gained knowledge. Experiments demonstrate the effectiveness of our proposed work in comparison with state-of-the-art models as well as the model's ability to adapt to new domains with few-shot learning.

3.1.1 Domain Generalization

Domain generalization seeks to improve a model's ability to perform well on unseen domains, addressing the problem of domain shift where the training and testing data come from different distributions. Traditional models often struggle with generalization beyond their training domain, leading to poor performance in real-world applications. Domain generalization techniques, such as invariant feature learning, data augmentation, and adversarial training, aim to create robust models that can generalize across various domains without requiring access to target domain data during training.

3.1.2 Continual Learning

Continual learning, also known as lifelong learning, aims to develop models capable of learning from a continuous stream of data without forgetting previously acquired knowledge. Traditional machine learning models suffer from catastrophic forgetting, where new information overwrites existing knowledge, a problem known as catastrophic forgetting. Continual learning addresses this by employing strategies such as rehearsal, regularization, and architectural approaches to retain past knowledge while integrating new information, making it crucial for real-world applications where data evolves over time. The approaches that address continual learning could be divided into three main categories, replay-based approaches, regularization-based approaches, and parameter isolation approaches. Replay-based approaches rely on storing some examples from previous training sessions to reuse them during training on new tasks [36, 37]. In regularization-based techniques, the main focus is on changing the loss function by adding regularization terms to restrain overfitting to new tasks [38, 39]. Furthermore, in parameter isolation techniques, the model parameters are separated and isolated for each task and training session. Freezing the desired weights allows the model to train on multiple tasks without forgetting the knowledge learned from previous tasks [40], Another approach is the add more weights and increase the model size with each new task [41].

3.1.3 Federated Learning

Federated learning is a decentralized approach to training machine learning models across multiple devices or organizations without centralizing the data. This method enhances privacy and security by keeping the data localized and only sharing model updates. Federated learning leverages techniques like secure aggregation, differential privacy, and homomorphic encryption to ensure data confidentiality while collaboratively training models. It is particularly valuable in privacy-sensitive applications like healthcare and finance, where data centralization is not feasible.

In federated learning, the model is trained locally on each device using the data on that device, and the updated weights are shared with a central server. The central server then updates its weights using the aggregated updates from all the devices and releases the updated model to the devices. This process is repeated to keep the model up to date with new data. As different settings for federated learning have been used in networking and IoT like Cross-silo [42] and Cross-device [43]. Cross-silo federated learning

refers to the collaborative training of machine learning models across multiple organizational silos, where each silo represents a distinct entity with its private dataset. This approach enables the pooling of data while ensuring data privacy and security. In contrast, cross-device federated learning involves training models directly on decentralized devices, such as smartphones or IoT devices, without the need for data transfer to a central server. The focus is on leveraging the computational power of diverse devices while preserving user privacy. Both approaches contribute to privacy-preserving machine learning, but their distinction lies in the data distribution and collaboration mechanisms employed. It has been found that Cross-silo is the best setting for the proposed solution in this paper, as the main focus is not on data privacy but on domain generalization. Federated learning has also been used for some computer vision tasks in medical imaging like [44], and [45] for decentralized learning. To the best of our knowledge, federated learning has yet to be used to overcome the problem of catastrophic forgetting in the context of foreground segmentation.

3.1.4 Knowledge Distillation

Knowledge distillation is a technique used to transfer knowledge from a large, complex model (the teacher) to a smaller, more efficient model (the student). This process involves training the student model to mimic the behaviour of the teacher model, often using the teacher's soft predictions as a guide. Knowledge distillation helps in deploying models on resource-constrained devices by reducing the model size and computational requirements without significantly sacrificing performance, making it a key approach in optimizing deep learning models for practical use. Knowledge distillation is particularly promising as it achieves performances close to those of larger models, despite having significantly fewer parameters and lower computational complexity. For instance, Mobile-SAM [46] exemplifies the power of knowledge distillation, being 100 times smaller and 50 times faster than SAM [47], with only a 20% performance drop.

3.1.5 Few-Shot Learning

Few-shot learning focuses on enabling models to generalize from a limited number of training examples. Unlike traditional learning paradigms that require large datasets, few-shot learning is designed to work effectively with minimal data, which is particularly useful in scenarios where data collection is expensive or time-consuming. Techniques like metric learning, self-supervised learning-based methods, and data augmentation are commonly used to enhance the model's ability to recognize new classes or tasks from just a few examples, thereby improving efficiency and reducing the need for extensive labelled data. Metric learning is a learning technique that focuses on clustering and identifying similar features in the embedding space by learning an embedding function that identifies similarities and separates similar features in the embedding space [48–50]. Self-supervised learning methods primarily aim to extract knowledge from unlabeled data by generating their own labels through techniques like random masking or applying random transformations to images. This process enables the model to learn abstract and high-level features from the data [51–53]. Subsequently, the model undergoes fine-tuning on a smaller, labelled dataset using transfer learning, enhancing its performance on specific tasks. Implementing this approach enhances learning efficiency and improves model generalization, advancing AI and ML capabilities. One of the commonly used self-supervised models for few-shot and domain generalization is Maske Auto Encoders (MAE) [51]. Applying augmentation techniques on limited training examples results in increasing the number of the labelled data points which helps the model to learn more features from the provided data. Some of the commonly used augmentation techniques are rotation, scaling, translation, flipping (Horizontal and Vertical), cropping, colour jittering (brightness, contrast, saturation), Gaussian noise, Cutout [54], Mixup [55], CutMix [56].

3.2 Proposed Work

The proposed work comprises two main components: 1) model architecture consisting of an encoder with a multi-head decoder. 2) training technique that combines federated learning with continual learning.

3.2.1 Model Architecture

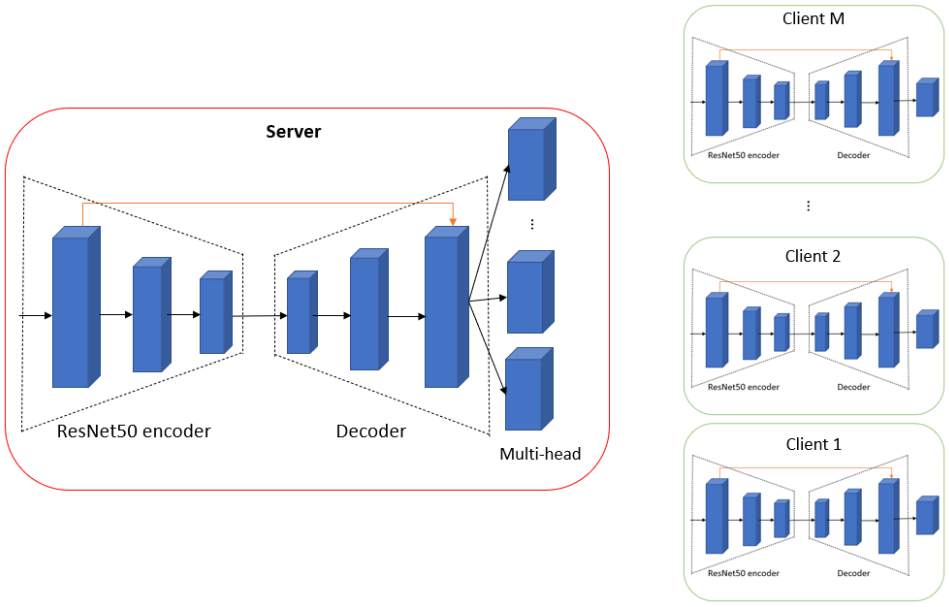

Figure 3.1: The architecture of KDNet. On the left is the server network with multi-head outputs, and on the right are the client networks.

The proposed model is an encoder-decoder architecture. The encoder is ResNet50, and the decoder is three blocks. Each decoder block consists of three convolutional layers. Each convolutional layer is followed by a normalization layer, and an up-sampling layer follows each decoder block. The number of filters of all convolutional layers in each block is the same but differs from one block to another. Such that convolution layers in the first block have 256 filters, in the second block have 128 filters, and in the last block have 64 filters. This size of all filters in all decoder blocks is 3×3. After the three blocks, a multi-head is added to produce the outputs for different domains. The head is defined as two convolutional layers. The first one has $3 \times 3 \times 64$ filters followed by a batch normalization layer. The second

convolutional layer has a $1 \times 1 \times 1$ filter with a sigmoid activation function. This architecture is for the main model (server), while the client's model is the same but has a single head. Figure 3.1 shows the server and client architecture.

3.2.2 Training Technique

As shown in Figure 3.2, the training procedure is a repeated cycle of four main steps: 1) instantiating the clients, 2) training the clients, 3) gathering the clients, and 4) training the main model (server).

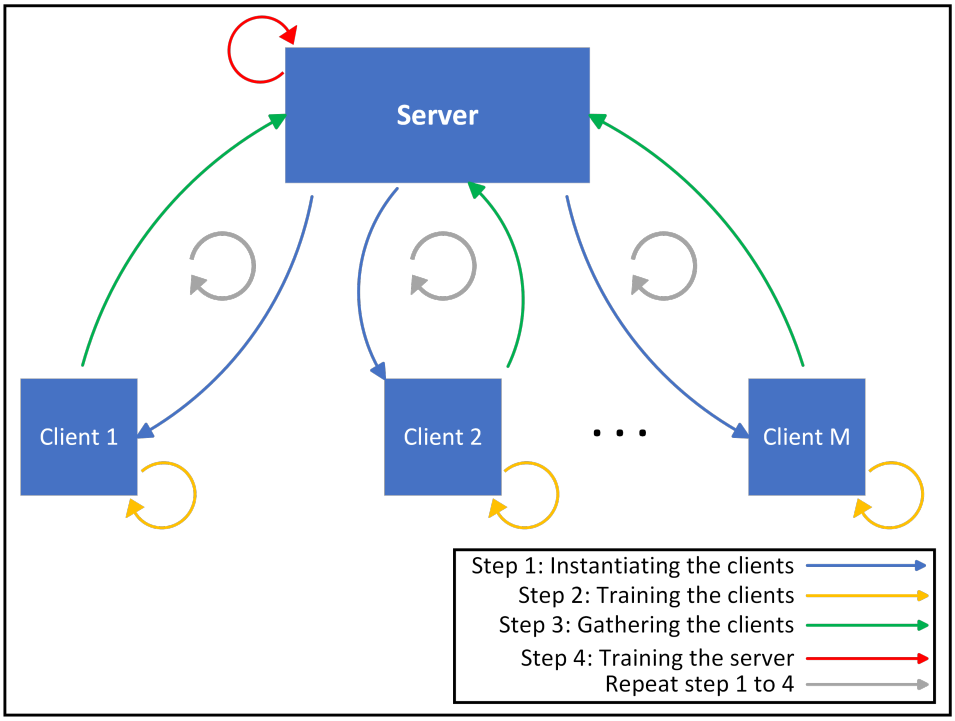

Figure 3.2: Training procedure of KDNet.

Instantiating the client: In this step, all clients are instantiated from the server (i.e., a clone of the shared weights of the server is sent to the client). The clients initialize their network parameters with the cloned weights of the server. It is important to mention that only the encoder and decoder parameters are copied to the client (i.e., the heads' parameters are not copied). After that, a head is initialized with random weights added at the end of the

client network.

Training the clients: The clients' training is straightforward. Each client is trained on its local dataset using supervised learning. The goal is that each client successfully learns to segment the foreground objects in the local dataset.

Gathering the clients: The server gathers only the trained client networks (i.e., the local dataset of each client is not included)

Training the server: This is the most significant step where the server network learns the knowledge gained by all client networks. The server uses a large unlabeled dataset. Then, for each batch in this dataset, only one client will process this batch to produce the output mask. The clients are selected in a round-robin order (i.e., client^1, client^2, ..., client^M, client^1, ..., etc.). After that, the server applies the knowledge distillation loss function to learn to produce an output similar to the client's output. However, the server should learn from each client without affecting the knowledge gained by other clients. Hence, each client updates only one head of the server. Also, a term in the loss function is added to regularize the updates of shared parameters between clients (i.e., the encoder and decoder parameters). This term minimizes the distance between the output of other clients after the update and before the current client's update. The loss function used is defined as follows:

$$\mathcal{L}^i = \ell(y^i, y^i_n) + \frac{1}{M}\sum_{j\neq i}^{M}\ell(y^j_o, y^j_n) \tag{3.1}$$

$$\mathcal{L}^i = KDL(y^i, y^i_n) + \frac{1}{M}\sum_{j\neq i}^{M}MSE(y^j_o, y^j_n) \tag{3.2}$$

the $\mathcal{L}^i$ is the loss function for updating the server with the i^{th} client. The first loss term is reducing the distance between the server's output of i^{th} after the update y^i_n and the output of the i^{th} client y^i. The second term is reducing the distance between the output of all other heads of the server before y^j_o, and after y^j_n, the update and M represents the number of clients. The second term acts as a regularization term to force the network to update the parameters in a way that does not cause a huge change in the output of other heads. As shown in the loss function, the ground truth (label) of the batches in the dataset is not used.

3.3 Experiments

To evaluate KDNet, Three experiments are conducted. The first experiment is traditional foreground segmentation, where the model is trained using datasets individually (separated training). This experiment compares the results of KDNet with other foreground segmentation models. The second experiment tests the model generalization by training using multiple datasets (multiple domains). In this experiment, there are two training methods, namely joint and fine-tuning (sequential). Joint training means multiple datasets are combined to form one large dataset. Then, the model is trained using this large dataset. In sequential training, the model is trained using the dataset one after the other. After the model is trained using the last dataset, the models are evaluated using all datasets. The last experiment is few-shot learning. In this experiment, we test the ability of the model to adapt to new domains using a few labeled images.

3.3.1 Datasets

Densely Annotated VIdeo Segmentation 2016 (DAVIS16) [1] is a video object segmentation dataset. DAVIS consists of 50 videos. Each video has several frames ranging from 50 to 104. In each video, a single object is annotated, which is the object of interest in this video.

SegTrackV2 [19] is a video multiple objects segmentation dataset. The number of videos in the dataset is 14, and the number of frames ranged from 21 to 279 in each video. In this dataset, each frame has multiple ground-truth corresponding to each object in the video.

ChangeDetection.Net (CDNet) 2014 [20] is a benchmark dataset that detects changes between video frames. These changes are the moving objects. The dataset consists of 11 different challenges. Each challenge has a number of videos ranging from 4 to 6, and the number of frames in each video ranges from 1000 to 8000. The total number of videos in the dataset is 53 videos.

ImageNet [57] is a large-scale dataset used as a benchmark for image classification tasks. In this paper, we use the images in the dataset without labels to train the main model to learn the knowledge gained by clients using knowledge distillation.

3.3.2 Implementation Details

In all experiments, we report the results of five different versions of KDNet. These versions 1) KDNet-base is the proposed model with a single head

decoder, 2) KDNet-FL is the proposed model trained with the federated learning settings. In this version, the dataset is into sub-datasets. Then, a client is instantiated to learn from each sub-dataset. After that, the multi-heads of the main model learn from all clients using knowledge distillation loss function without the regularization term in equation (3.2). 3) KDNet-CL is the same as KDNet-FL, but the main model learns directly from the sub-datasets instead of learning from the trained clients. The loss function used is a mean squared error (MSE) with the regularization term in equation (3.2). 4) KDNet-FL-CL is the combination of both federated learning and continual learning. The loss function used in this version is the proposed function in equation (3.2). 5) KDNet-FL-CL-U is the same as KDNet-FL-CL, but the knowledge distillation between the main model and the clients is applied using a completely different unseen dataset (ImageNet) from the one used to train the clients. The point of adding the last version is to show that the proposed work does not need to access the datasets used in training the clients (data privacy).

During the training of KDNet, if the training data is a single dataset, the dataset is randomly split into five groups of videos to simulate multiple dataset training. In this case, the main model has five heads, each one responsible for learning from one of the five groups of videos. On the other hand, if the training data is multiple datasets, then the number of heads of the main model will be the same as the number of different datasets.

In the testing phase, for all versions of KDNet with multi-head, we use one labelled frame for each video to find which head is suitable for segmenting this video. We do this by feeding the frame as an input to the model. Then, the f-measure of each head of KDNet is calculated. The head with the highest f-measure is used to segment foreground objects from all frames of this video.

3.3.3 Traditional Foreground Segmentation Results

This experiment evaluates KDNet using two datasets, DAVIS16 and SegTrackV2. The model is trained and tested using each dataset individually. For this experiment, we show the results of the five versions of KDNet mentioned in the previous subsection. The reported metric for all experiments is the f-measure ($\mathcal{F}$).

Table 3.1 shows the results of KDNet against state-of-the-art foreground segmentation models. The results show that KDNet is in second place by a small gap of 0.06% in f-measure less than the top-performing model.

For the SegTrackV2 dataset, Table 3.2 shows the results of KDNet against other foreground segmentation models using the SegTrackV2 dataset. As

shown in Table 3.2, the performance of three versions of KDNet is in the top places. KDNet outperforms the second-best model (REFNet) in literature by 1.8%. While the second version of KDNet outperforms REFNet by 1.2%. These tables show the effectiveness of the proposed work in learning from individual datasets, even though the proposed work is mainly designed to learn effectively from multiple datasets.

Table 3.1: Traditional foreground segmentation results of KDNet compared with state-of-the-art models on the **DAVIS16** dataset. The number inside the circle defines the order of the model performance concerning each evaluation metric.

Model	$\mathcal{F}$
STM [24]	0.901
MHP-VOS[58]	0.895
CFBI [59]	0.905
CFBI+ [59]	0.911
FgSegNet [60]	0.847
CascadeCNN [15]	0.814
STCN [17]	**0.930**①
REFNet [61]	0.883
TransBlast [18]	0.863
KDNet-base	0.891
KDNet-FL	0.909
KDNet-CL	0.897
KDNet-FL-CL	0.924②
KDNet-FL-CL-U	0.921③

Table 3.2: Traditional foreground segmentation results of KDNet compared with state-of-the-art models on the **SegTrackV2** dataset. The number inside the circle defines the order of the model performance concerning the evaluation metric.

Model	$\mathcal{F}$
DSRFCN3D [62]	0.878
GDHF [63]	0.868
UFO [64]	0.863
PDB [65]	0.864
STRCF [66]	0.899
FgSegNet [60]	0.880
CascadeCNN [15]	0.867
REFNet [61]	0.904
TransBlast [18]	0.902
KDNet-base	0.905
KDNet-FL	0.911③
KDNet-CL	0.909
KDNet-FL-CL	**0.920**①
KDNet-FL-CL-U	0.914②

3.3.4 Domain Generalization Results

To highlight the effectiveness of the proposed work, we conducted multi-domain experiments. The model is trained in these experiments using three benchmark datasets: CDNet, DAVIS16, and SegTrackV2. We show the results of two training procedures. The first training procedure is Joint training, where the models are trained using the three datasets simultaneously as if they are one large dataset. The second training procedure is Fine-tuning,

where the models are fine-tuned using one dataset after the other (sequentially). After training is done in both scenarios, the model is evaluated using the testing set of all three datasets.

The results are compared with state-of-the-art foreground segmentation models. Four models with their source code publicly available from the previous experiment are selected to compare KDNet's generalization against them. These models are REFNet, sEnDec, FgSegNetV2, and CascadeCNN.

Tables 3.3 and 3.4 show the results of the models when trained using multiple datasets either simultaneously or sequentially. In these tables, it is clear that all models degraded in performance tremendously compared to their performance in the separated training. For example, REFNet has an average of 26.3% drop in performance when trained using multiple datasets. In Table 3.3, most of the models have higher performance in the first dataset (CDNet) as it is the largest dataset, which made the models biased toward it. However, our model solves this problem by using isolated parameters for each dataset (multi-head decoder). The performance gain in our model is 14.1% compared to REFNet, which is a massive gain in performance. In Table 3.4, the performance of all models in the last dataset is the highest. This is because the models tend to forget the knowledge gained by previous datasets due to sequential training (catastrophic forgetting). However, our proposed model overcomes this problem by the isolated parameters and the continual learning regularization term in the loss function. Hence, the performance of our proposed model is 13% higher than the second-best model.

Table 3.3: Comparison of KDNet against state-of-the-art methods when trained using the three datasets **CDNet, DAVIS16, SegTrackV2 jointly** (i.e., all three datasets are combined and considered as one large dataset). The values in the table represent the F-measure.

Model	CDNet	DAVIS16	SegTrackV2	Average
sEnDec [67]	0.657	0.571	0.526	0.584
FgSegNetV2 [60]	0.639	0.567	0.495	0.567
CascadeCNN [15]	0.582	0.535	0.460	0.525
REFNet [61]	**0.737**①	0.630	0.505	0.624
KDNet-base	0.627	0.706	0.580	0.637
KDNet-FL	0.649	0.735③	0.721③	0.701③
KDNet-CL	0.633	0.719	0.658	0.670
KDNet-FL-CL	0.662②	**0.756**①	**0.867**①	**0.765**①
KDNet-FL-CL-U	0.659③	0.751②	0.816②	0.742②

Table 3.4: Comparison of KDNet against state-of-the-art methods when trained using the three datasets **CDNet, DAVIS16, SegTrackV2 sequentially** (i.e., the models are fine-tuned on datasets sequentially. After the model is trained using the last dataset, it is evaluated using the last dataset and all previous ones). The values in the table represent the F-measure.

Model	CDNet	DAVIS16	SegTrackV2	Average
sEnDec [67]	0.451	0.562	0.829	0.614
FgSegNetV2 [60]	0.419	0.518	0.755	0.564
CascadeCNN [15]	0.398	0.503	0.746	0.549
REFNet [61]	0.422	0.529	**0.917**①	0.622
KDNet-base	0.421	0.590	0.858	0.620
KDNet-FL	0.476	0.643	0.872	0.664
KDNet-CL	0.638③	0.716③	0.874	0.743③
KDNet-FL-CL	**0.655**①	**0.721**①	0.880②	**0.752**①
KDNet-FL-CL-U	0.647②	0.716②	0.876③	0.746②

To show the output of the trained model using some example videos from the three datasets, some visual results from this experiment are shown in Fig. 3.4.

The effectiveness of the proposed federated learning technique is shown in Fig. 3.3. In this figure, the average loss of all clients gets lower and converges faster after each federated cycle. Hence, the trained server has good knowledge of different domains and can adapt agilely to new domains.

3.3.5 Few-shot Results

We conducted a few-shot learning experiment to prove the trained KDNet's versatility and agility to learn new domains with limited data. In this experiment, the KDNet trained using CDNet, SegTrackV2, and DAVIS16 is fine-tuned using a few labelled frames from the DAVIS17 dataset. The number of labelled frames per video is either 1 or 5 frames, which is the default configuration for few-shot learning. Other models are trained jointly using the same three datasets and fine-tuned using the labelled frames from DAVIS17. In Table 3.5, KDNet (Server) is fine-tuning the jointly trained model, while KDNet (Server-Client) is fine-tuning the model trained using federated learning. The results show the superiority of the KDNet trained with federated learning in the few-shot learning performance. In 1-shot (i.e., one labelled frame per video), the performance of the proposed model is

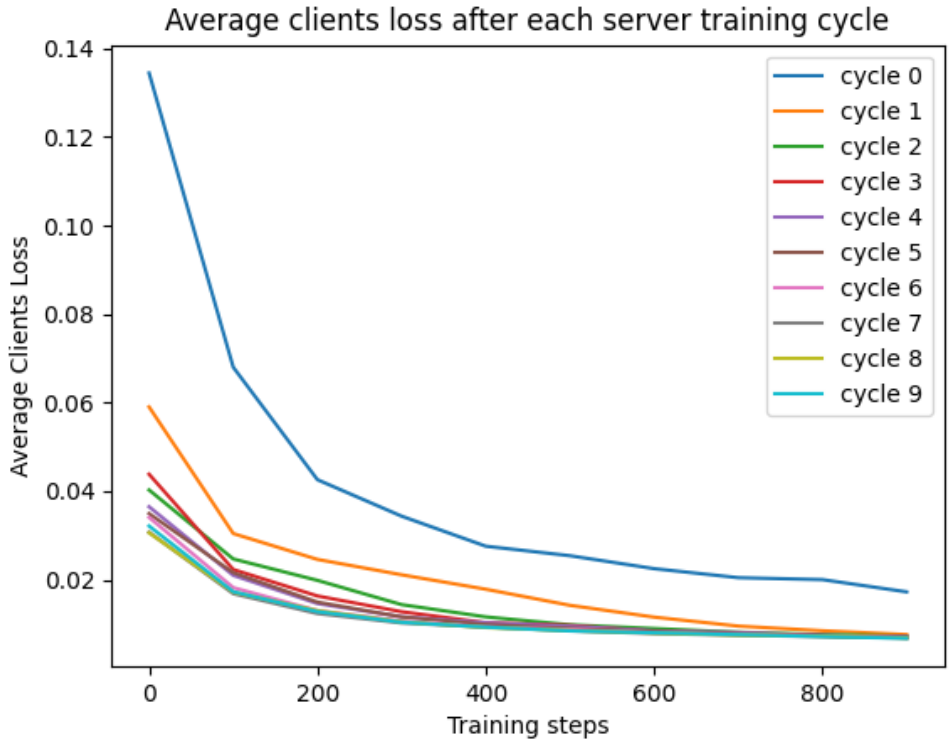

Figure 3.3: The average loss of clients of all three datasets after each federated learning cycle.

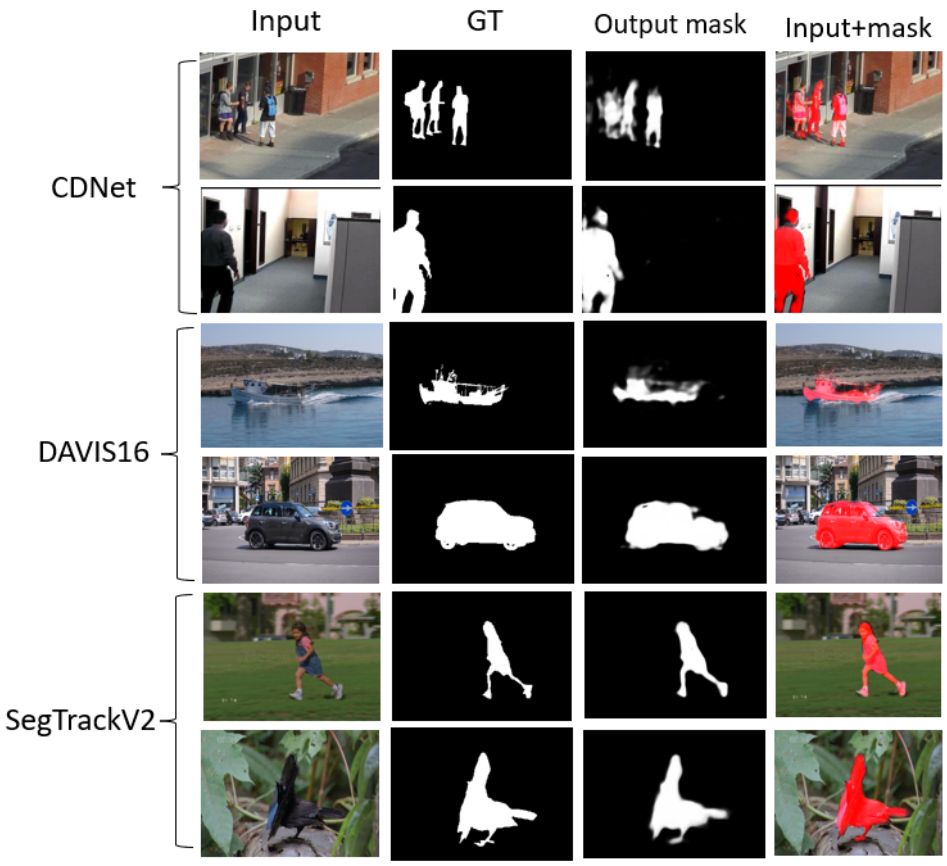

Figure 3.4: Sample results of KDNet from domain generalization experiment.

around 6.8% above the second model REFNet, which is expected as shown in Fig. 3.3 the trained KDNet makes the convergence towards a new task faster and easier. Hence, only a few fine-tuning iterations are required for the trained KDNet to converge.

Table 3.5: Few-shot foreground segmentation results of KDNet compared to state-of-the-art methods on DAVIS17.

Model	1-shot	5-shot
CascadeCNN [15]	0.624	0.661
FgSegNetV2 [60]	0.676	0.742
sEnDec [67]	0.681	0.759
REFNet [61]	0.692	0.767
KDNet-base	0.714	0.785
KDNet-FL-CL	**0.760**	**0.814**

3.4 Limitations and possible improvements

Foreground segmentation focuses on distinguishing the whole foreground as a single object from a static or minimally changing background, often leading to simpler binary masks and struggles with complex or dynamic backgrounds and facing major challenges for multi-object segmentation. Therefore, the natural progression in this case is to adapt VOS methods as demonstrated in the next technical chapter 4.

3.5 Conclusion

In this Chapter, we proposed a novel federated learning technique and an encoder-decoder network with multi-heads using multi-domain foreground segmentation datasets. The proposed model is called KDNet. KDNet instantiates a number of client networks corresponding to the number of different datasets. Each client is responsible for training using one of the datasets. Then, KDNet learns the knowledge gained by each client using knowledge distillation while minimizing the catastrophic forgetting of knowledge gained by previous clients. The results show that the proposed KDNet outperforms state-of-the-art foreground segmentation models by 1.8% when trained/tested using the SegTrackV2 dataset and in the second place with a small gap of 0.06% using DAVIS16. On the other hand, When the models are trained using multiple datasets to achieve domain generalization, KDNet outperforms other models by a noticeable amount of 14.1% and 13% in Joint training and fine-tuning, respectively.

However, This work faces challenges when it comes to multi-object segmentation in the foreground since the focus of this work is to isolate the whole foreground as a single object. In this case, the future work focus shifts to a multi-object segmentation capable video segmentation task such as VOS as demonstrated in the next technical chapter 4.

Chapter 4

MEM: Mask Enhancement Model

4.1 Overview and Motivation

In this chapter, we propose a deep-learning model that achieves state-of-the-art metrics in video object segmentation task by leveraging two main techniques such as foundational model integration and mask fusion. We employ two different models with two different functions. The first model is the Cutie model [31], which represents the state-of-the-art in VOS and is utilized in this chapter. Memory-based models such as Cutie operate by maintaining a record of previously segmented frames within the video sequence, enabling the retrieval of features to enhance the segmentation of future frames. On the other hand, Cutie often fails when highly similar objects move in close proximity or occlude each other. In these cases, neither the pixel memory nor the object memory can pick up sufficiently discriminative features for the object transformer to operate on. The second model is SAM [47], which is a large foundational model that is trained on the SA-1B dataset, which is one of the largest segmentation datasets, containing over 1 billion multi-object masks across 11 million images. This diverse dataset covers a wide range of objects, lighting conditions, and environments, allowing SAM to generalize across various segmentation tasks. SAM works by combining a Vision Transformer (ViT) backbone for feature extraction with a prompt-based approach. Users can provide different types of prompts, such as points, bounding boxes, or free-form text, to guide the model in identifying objects of interest. The model processes the prompt and image, generating a high-quality mask for the object specified. This flexibility allows SAM to perform zero-shot segmentation without fine-tuning on new datasets, making it highly adaptable across domains. However, SAM is not designed for video sequences or video object segmentation tasks. To use SAM for segmenting an object in a video, each frame and the object's location in each frame must be provided individually and iteratively along the video sequence. The strength that SAM provides in this work is that it doesn't need knowledge from previous frames

to segment an object like Cutie.

Given the mentioned strengths and weaknesses of both models, fusing the output masks of these two models will produce a model with better generalization and segmentation accuracy. Additionally, we investigated visually the output masks of both models. As shown in Figure 4.1, there are instances where SAM performs better than Cutie, while Cutie outperforms SAM in other instances. Hence, fusing these models will produce overall better performance.

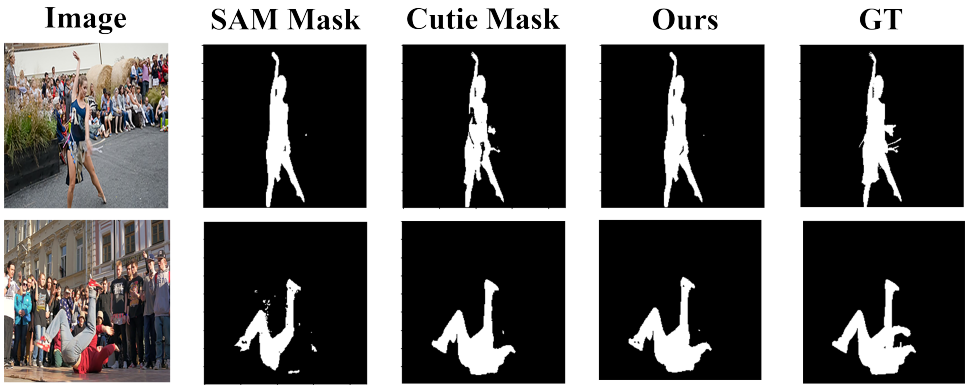

Figure 4.1: Two different frames are selected from two different video sequences that show our model's mask in comparison to SAM and Cutie

4.2 Methodology

We propose MEM that leverages the strengths of both existing models. MEM combines the current frame, SAM mask, and Cutie mask as inputs to generate an improved output mask, ultimately surpassing the individual performance of each original model. First, we illustrate how SAM is used to segment the target object in each video. Then, we explain the proposed MEM architecture.

4.2.1 VOS with SAM

It was observed that VOS models naturally follow objects and typically create masks that cover most, if not all, of the object. The approach involved finding the bounding box coordinates of all non-zero elements in the VOS model output masks, as illustrated in Figure 4.2. These coordinates were then used to generate the SAM response.

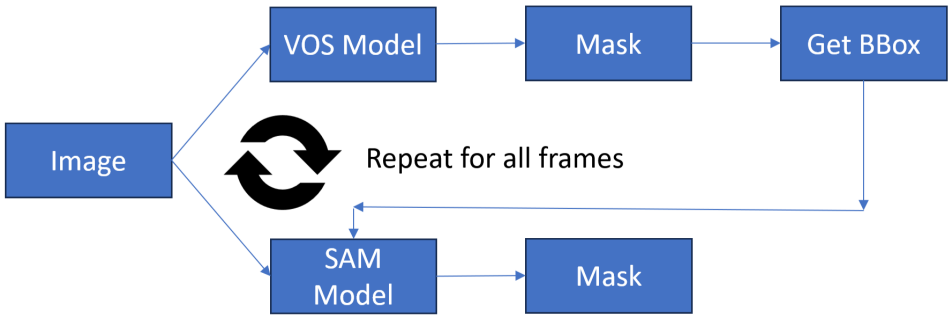

Figure 4.2: Getting SAM response after extracting bounding box from VOS model. The process is performed in an iterative way overall video sequence frames.

4.2.2 Architecture

4.2.2.1 MEM

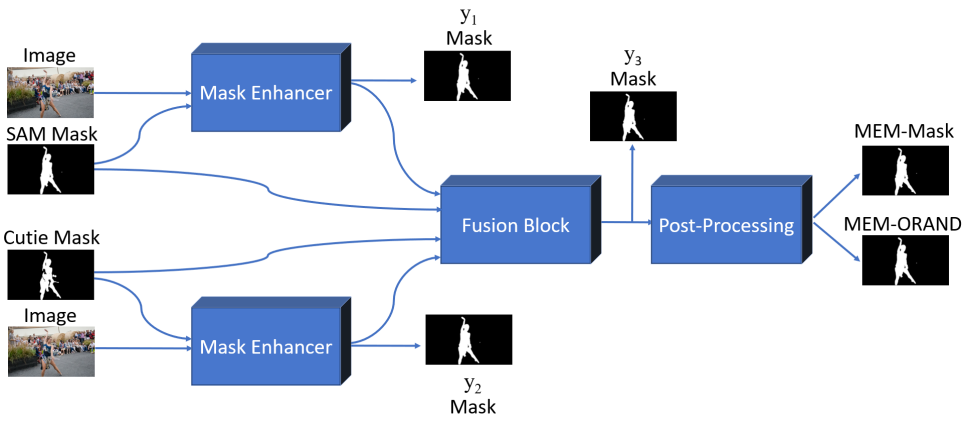

Figure 4.3: MEM model architecture. Each Mask Enhancer block takes the current frame and model mask as input. The Fusion Block takes both Mask Enhancer features and both models' masks as inputs. y1, y2, and y3 are used only during training time while MEM-Mask and MEM-ORAND are the model output during inference time.

MEM consists of two main components. The first component is the Mask Enhancer, which learns to produce an output better than that of one of the two primary models. The second component is the Fusion Block, which combines

the embeddings from both Mask Enhancer blocks to generate an improved mask as seen in figure 4.3. The main building block of all components is a modified Res-block.

4.2.2.2 Modified Res-Block

The Modified Res-Block comprises three convolution layers with a kernel size of 3×3 and 'same' padding. A batch normalization layer and a ReLU activation function follow each convolution layer.

4.2.2.3 Stem Stage

The Stem Stage extracts a feature map from the image to prepare it for the Mask Enhancer component. A single Stem Stage is employed to extract image features for both Mask Enhancer components. This stage consists of one convolution layer, followed by a batch normalization layer and a ReLU activation function. The resulting feature map size is $480 \times 480 \times 32$.

4.2.2.4 Mask Enhancer

The Mask Enhancer is utilized twice in this model, each with distinct sets of weights. Each Mask Enhancer stage is constructed using three consecutive Modified Res-Blocks. The Mask Enhancer takes two inputs: the feature map generated from the Stem Stage and the corresponding model mask scaled by 32. These inputs are concatenated along the channel dimension, creating an input size of $480 \times 480 \times 64$. Due to the 'same' padding style, the output dimensions remain unchanged, maintaining the same size as the input.

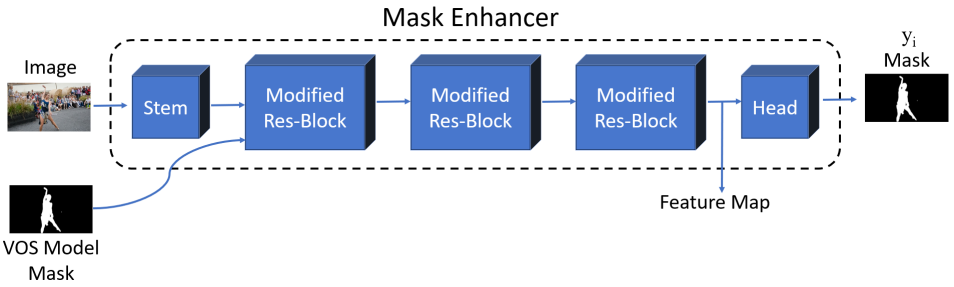

Figure 4.4: Detailed view of Mask Enhancer components. The Stem stage has shared weights between both Mask Enhancers.

4.2.2.5 Fusion Block

The Fusion Block architecture mirrors that of the Mask Enhancer, comprising three consecutive Modified Res-Blocks. The key difference lies in its four inputs, which increase the number of input channels. These inputs include the final feature maps from both Mask Enhancer stages, the Cutie mask repeated 32 times along the channel dimension, and the SAM mask similarly repeated 32 times along the channel dimension. This results in an input size of $480 \times 480 \times 128$

4.2.2.6 Prediction Heads

The Prediction Heads are at the end of each component, heads are used to generate a mask from the feature maps produced by each component. The heads used for the Mask Enhancer stages consist of a single convolution layer with a kernel size of 3×3 and 'same' padding, followed by a sigmoid activation layer. The prediction head for the fusion block is composed of two convolution layers with the same kernel size and padding as the other heads, followed by a sigmoid activation layer.

4.2.2.7 Post Processing Block

The Post Processing Block refines the model output by removing noise and applying a threshold to the output from the sigmoid function, resulting in two masks: MEM-Mask and MEM-ORAND. MEM-Mask is generated by applying a threshold of 0.5 to the sigmoid values, converting any pixel value above 0.5 to 1 and any value below 0.5 to 0. This threshold has been selected after inspecting different threshold values. However, inspection of the MEM-Mask output revealed occasional false positives from the background part of the image as noise. To address this issue, the output should be restricted to the scope of the object in the mask. Given that both Cutie and SAM masks isolate the object with minimal noise, the solution involves performing a logical OR operation on both masks, followed by a logical AND operation between the OR-mask and MEM-Mask. This process significantly reduces background noise, resulting in the MEM-ORAND Mask.

4.3 Experiments

Several experiments have been conducted to validate the proposed method to use SAM for VOS as well as evaluate the proposed MEM model performance on standard benchmark datasets.

4.3.1 Implementation Details

4.3.1.1 Optimizeation

The Adam optimizer is used with a learning rate of 1e-5 and a batch size of 1.

4.3.1.2 Loss

The loss is calculated on each of the three masks (y_1,y_2,y_3). The final loss is the summation of all three losses, as shown in Equation 4.1. ℓ is Mean Squared Error Loss (MSE), y_i is the mask, y is the Ground Truth, and $\mathcal{L}$ is the sum of all losses over the three masks

$$\mathcal{L} = \sum_{i}^{M} \ell(y_i, y) \tag{4.1}$$

4.3.1.3 Training

The model is trained on the training sets of both DAVIS17 and youTube-VOS individually. We train the model on each object separately. After isolating each object in a separate mask, both forward and backpropagation are performed in an n-stream manner, where n is the number of objects in the mask.

4.3.1.4 Testing

The model has been evaluated locally and on datasets' competition servers. For DAVIS-17 val, the model has been evaluated locally and both the Jaccard index and F-score have been calculated locally. For both the DAVIS-17 test and YouTube VOS-2019 val, the model output has been evaluated on the competition servers. The main evaluation metric is $\mathcal{J}\&\mathcal{F}$ [68], which is the average of both the Jaccard index and F-score.

4.3.2 Datasets

Densely Annotated VIdeo Segmentation 2016 (DAVIS16) [68] is a foreground segmentation dataset. DAVIS consists of 50 videos. Each video has several frames ranging from 50 to 104. In each video, a single object is annotated, which is the object of interest in this video.

Densely Annotated VIdeo Segmentation 2017 (DAVIS17)[1] is a benchmark for video object segmentation, offering high-resolution videos with pixel-level annotations for diverse and challenging sequences. It includes 60 videos for training, 30 for validation, and 60 for testing (30 in the test-dev set and 30 in the test-challenge set). It supports semi-supervised and unsupervised segmentation tasks, providing detailed evaluation metrics like the Jaccard index.

YouTube-VOS 2019 dataset [35] is a large-scale benchmark for video object segmentation, offering high-resolution videos with detailed annotations. It includes 3,471 videos for training, 507 for validation, and 541 for testing. Supporting both semi-supervised and unsupervised tasks, this dataset provides extensive data for training and benchmarking segmentation models.

4.3.3 Unsupervised VOS SAM Results

Table 4.1: Comparison between unsupervised VOS $\mathcal{J}\&\mathcal{F}$ scores for the state-of-the-art models and SAM for VOS.

	DAVIS-16 val	DAVIS-17 val	DAVIS-17 test
Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
RTNet [69]	85.2	-	-
PMN [70]	85.9	-	-
UnOVOST [71]	-	67.9	58.0
Propose-Reduce [72]	-	70.4	-
DEVA [33]	**88.9**①	73.4②	62.1②
SAM [47](Modified for VOS)	87.1②	**86.6**①	**78.4**①

As seen in Table 4.1, The SAM model for video object segmentation (VOS) achieved top $\mathcal{J}\&\mathcal{F}$ scores of 86.6% on the DAVIS-17 validation set and 78.4% on the DAVIS-17 test set. However, on the DAVIS-16 validation set, SAM scored 87.1%, which is the second-best score after DEVA's 88.9%. Unlike

VOS datasets like DAVIS-17 and YouTube-VOS, DAVIS-16 focuses on segmenting all foreground objects as a single entity. This difference explains the drop in SAM's performance on DAVIS-16, as it was trained on datasets that involve multiple objects and parts segmentation.

4.3.4 MEM Results

MEM is compared to state-of-the-art models on DAVIS 2017 validation/testdev and YouTubeVOS validation standard benchmarks. Both Jaccard index $\mathcal{J}$, F1-score $\mathcal{F}$, and their average $\mathcal{J}\&\mathcal{F}$ are reported for all benchmark datasets.

As observed in Table 4.2, MEM model is achieving the state-of-the-art $\mathcal{J}\&\mathcal{F}$ scores on both DAVIS-17 val and DAVIS-17 test with 89.0% and 86.0% respectively. However, observing the results on Youtube-VOS, MEM model still achieves a state-of-the-art overall $\mathcal{G}$ score of 86.6% despite achieving lower $\mathcal{F}_u$ and $\mathcal{F}_s$ scores than Cutie.

Table 4.2: Comparison of VOS state-of-the-art models on two different datasets DAVIS-17 and YouTube VOS

	DAVIS-17 val			DAVIS-17 test				YouTube VOS-2019 val			
Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$
STCN [73]	85.4	82.2	88.6	76.1	72.7	79.6	82.7	81.1	85.4	78.2	85.9
AOT-R50 [27]	84.9	82.3	87.5	79.6	75.9	83.3	85.3	83.9	88.8	79.9	88.5
RDE [74]	84.2	80.8	87.5	77.4	73.6	81.2	81.9	81.1	85.5	76.2	84.8
JointFormer [75]	-	-	-	65.6	61.7	69.4	73.3	75.2	78.5	65.8	73.6
XMem [26]	86.0	82.8	89.2	79.6	76.1	83.0	85.6	84.1	88.5	81.0	88.9②
DeAOT-R50 [76]	86.0	83.1	88.9	82.8	79.1	86.5	85.3	84.2	89.0	79.9	88.2
DEVA [33]	87.0	83.8	90.2	82.6	78.9	86.4	85.4	84.9	89.4③	79.6	87.8
Cutie-base [31]	88.8③	85.6③	**91.9**①	85.3②	81.4③	89.3②	86.5②	85.4③	**90.0**①	81.3②	**89.3**①
SAM [47](Modified for VOS)	86.6	83.4	89.7	78.4	75.2	81.7	78.0	80.0	82.5	71.1	78.3
MEM (Ours)	88.9②	86.7②	91.1③	85.2③	81.6②	88.7③	86.3③	85.7②	89.4③	81.2③	88.7③
MEM-ORAND (Ours)	**89.0**①	**86.8**①	91.2②	**86.0**①	**82.6**①	**89.5**①	**86.6**①	**86.1**①	89.7②	**81.6**①	88.9②

4.3.5 Ablation Study

In this section, we show other methods that have been tested to employ SAM for VOS tasks.

4.3.5.1 Bounding Box Regression Head and Object Tracking Model

As illustrated in Figure 4.5, a bounding box regression head was implemented. This head took SAM embeddings for two consecutive frames as input and output the bounding box coordinates in the form of (X1, Y1, X2, Y2).

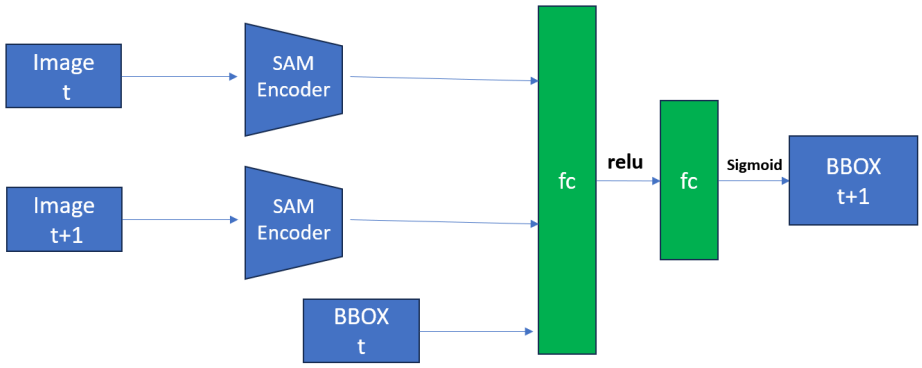

Figure 4.5: Bounding Box Regression Head architecture. Both SAM Encoders have shared weights. Both Encoder embeddings are fed to a fully connected layer followers by ReLU then another fully connected layer followed by a Sigmoid to calculate a normalized bounding box corner coordinates values.

The OStrack [77] model was evaluated and showed acceptable results and IOU scores but to use SAM for the VOS task, higher IOU scores were needed.

4.3.5.2 Get Bounding Box From VOS Model Output

As seen in figure 4.2, the bounding box generation algorithm is applied to the output masks from the Cutie model and then provided to SAM along with the corresponding image. This method shows the best results among other methods as it leverages the tracking nature of VOS models.

4.3.5.3 Modified SAM Results

Table 4.3: The highest IOU results are reported in this table to show the difference between all three tested methods. Several experiments have been conducted to develop a better bounding box regression model and the reported result was the best achieved results for this method. For the OSTrack model, it has been used without any training from our side. These results are reported after using the training set of DAVIS-17

Method	Bounding Box Regression	OS Track [77]	IOU from VOS
IOU Score	56.1	78.4	**97.5**

For evaluation, a standard metric Intersection Over Union (IOU) is used to measure how accurate is the predicted bounding box in comparison to the ground truth bounding box. Ground truth bounding boxes have been generated by applying non-zero value analysis on ground truth masks similar to what is shown in Figure 4.2.

As seen in Table 4.3, getting the bounding box from the VOS model's output masks resulted in the highest IOU of 97.5% while the bounding box regression model resulted in an IOU of 56.1% which is not suitable for our application as the bounding boxes need to be as accurate as possible to avoid missed or partial segmentation with SAM.

4.4 Limitations and Possible Improvements

Upon reviewing the results, we notice that MEM achieves less-than-ideal F1 scores on certain benchmarks. This shortcoming arises from the model's difficulty in accurately capturing the fine details of the outer silhouettes of some objects. This issue could be tackled in future work, considering recent advancements in fine edge detection.

4.5 Conclusion

This chapter introduces MEM as a novel approach to video object segmentation by combining the strengths of the Segment Anything Model (SAM) and the Cutie model. Our results show that MEM enhances segmentation accuracy, achieving top $\mathcal{J}\&\mathcal{F}$ scores on standard benchmark datasets. The integration of SAM's zero-shot segmentation capabilities with Cutie's memory-based approach allows MEM to effectively handle complex video sequences with occlusions and highly similar objects. Despite MEM's limitations, the experiments conducted validate the robustness and MEM's ability to achieve higher $\mathcal{J}$ and $\mathcal{J}\&\mathcal{F}$ scores than the state-of-the-art model due to the effectiveness of foundational model integration techniques proposed in this chapter, making it a valuable contribution to the field of video object segmentation. Future work will explore further optimization and enhancement of MEM.

Chapter 5

Conclusions and Future Work

This thesis begins by presenting the motivation and objectives of the research, aiming to enhance video object segmentation through innovative techniques. The primary contributions of the introduction chapter 1, include defining the problem space and setting the stage for the approaches explored in technical chapters. The introduction highlights the problems in today's video segmentation as well as the importance of accurate video segmentation in various applications, such as autonomous driving and video surveillance, and outlines the potential impact of the proposed solutions on these fields.

In chapter 2, an overview of existing video segmentation tasks and techniques is provided with the main focus on Foreground segmentation and video object segmentation. The key contribution here is the presentation of current methods and state-of-the-art methods, identifying their implemented methods. This chapter serves as the foundation for understanding the advancements proposed later in the thesis. The discussion on various segmentation approaches, including supervised, semi-supervised, and unsupervised methods, sets the context for the need for improved models like KDNet and MEM.

Chapter 3 introduces a novel approach to foreground segmentation through domain generalization using federated learning. The main contribution is the development and evaluation of the KDNet architecture, which enables effective learning across different domains without sharing raw data. The results demonstrate significant improvements in segmentation accuracy and robustness across multiple datasets, showcasing the potential of federated learning in enhancing generalization capabilities when paired with knowledge distillation and continual learning.

Chapter 4 presents the Mask Enhancement Model (MEM) which leverages the strengths of existing state-of-the-art models (SAM and Cutie) by fusing their outputs to generate superior segmentation masks. The key contributions include the design of the MEM architecture and its components, such as the Mask Enhancer and Fusion Block, which collectively improve segmentation accuracy. Experimental shows the model's ability to achieve state-of-the-art performance.

In conclusion, this thesis has explored innovative approaches and learning paradigms to advancing video segmentation, addressing critical challenges through the development of KDNet and MEM. By leveraging federated learning, continual learning, knowledge distillation, and few-shot learning, we demonstrated significant improvements in foreground segmentation accuracy and robustness across diverse datasets without compromising data privacy as a bonus. The MEM model, which integrates the strengths of existing segmentation techniques, showed remarkable effectiveness in enhancing mask quality, particularly in out-of-domain scenarios. These contributions underscore the potential of the proposed methodologies to impact real-world applications, such as autonomous driving and video surveillance. The findings of this research help advance video segmentation and lay new paths for future explorations aimed at further enhancing segmentation accuracy and generalization capabilities through the integration of advanced learning paradigms.

The possible future work proposed in this thesis is, 1) improving MEM's ability to capture fine details of object silhouettes. This involves leveraging advancements in fine edge detection techniques to enhance segmentation accuracy. 2) further optimizing and enhancing the MEM architecture to improve performance and efficiency. This could include refining the Mask Enhancer and Fusion Block components to better handle complex video sequences with occlusions and highly similar objects as well as adapting some techniques from KDNet to enhance MEM's generalization.

Bibliography

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

[2] Pakorn KaewTraKulPong and Richard Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," *Video-based surveillance systems: Computer vision and distributed processing*, pp. 135–144, 2002.

[3] Chris Stauffer and W Eric L Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*. IEEE, 1999, vol. 2, pp. 246–252.

[4] S. Z. Li, "Markov random field models in computer vision," in *Computer Vision — ECCV '94*, Jan-Olof Eklundh, Ed., Berlin, Heidelberg, 1994, pp. 361–370, Springer Berlin Heidelberg.

[5] John Lafferty, Andrew McCallum, Fernando Pereira, et al., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Icml*. Williamstown, MA, 2001, vol. 1, p. 3.

[6] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[12] Suyog Dutt Jain and Kristen Grauman, "Supervoxel-consistent foreground propagation in video," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*. Springer, 2014, pp. 656–671.

[13] Hajer Fradi and Jean-Luc Dugelay, "Robust foreground segmentation using improved gaussian mixture model and optical flow," in *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*. IEEE, 2012, pp. 248–253.

[14] Long Ang Lim and Hacer Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, 2018.

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

[16] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao, "Ranet: Ranking attention network for fast video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3978–3987.

[17] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11781–11794, 2021.

[18] Islam Osman, Mohamed Abdelpakey, and Mohamed S Shehata, "Transblast: Self-supervised learning using augmented subspace with transformer for background/foreground separation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 215–224.

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

[20] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 387–394.

[21] James J Gibson, "The perception of the visual world.," 1950.

[22] Pedro F Felzenszwalb and Daniel P Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, pp. 167–181, 2004.

[23] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh 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*, 2019, pp. 9481–9490.

[24] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9226–9235.

[25] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7376–7385.

[26] Ho Kei Cheng and Alexander G Schwing, "Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *European Conference on Computer Vision*. Springer, 2022, pp. 640–658.

[27] Zongxin Yang, Yunchao Wei, and Yi Yang, "Associating objects with transformers for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2491–2502, 2021.

[28] Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu, "Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer," 2024.

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

[30] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Chuanxin Tang, Xiyang Dai, Yucheng Zhao, Yujia Xie, Lu Yuan, and Yu-Gang Jiang, "Look before you match: Instance understanding matters in video object segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 2268–2278.

[31] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing, "Putting the object back into video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 3151–3161.

[32] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng, "Track anything: Segment anything meets videos," *arXiv preprint arXiv:2304.11968*, 2023.

[33] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee, "Tracking anything with decoupled video segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1316–1326.

[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[35] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv preprint arXiv:1809.03327*, 2018.

[36] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[37] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[38] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[39] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 139–154.

[40] Arun Mallya, Dillon Davis, and Svetlana Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 67–82.

[41] Amir Rosenfeld and John K Tsotsos, "Incremental learning through deep adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 3, pp. 651–663, 2018.

[42] Yansheng Wang, Yongxin Tong, Dingyuan Shi, and Ke Xu, "An efficient approach for cross-silo federated learning to rank," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 1128–1139.

[43] Muhammad Habib ur Rehman, Ahmed Mukhtar Dirir, Khaled Salah, Ernesto Damiani, and Davor Svetinovic, "Trustfed: A framework for fair and trustworthy cross-device federated learning in iiot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8485–8494, 2021.

[44] Dong Yang, Ziyue Xu, Wenqi Li, Andriy Myronenko, Holger R Roth, Stephanie Harmon, Sheng Xu, Baris Turkbey, Evrim Turkbey, Xiaosong

Wang, et al., "Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan," *Medical image analysis*, vol. 70, pp. 101992, 2021.

[45] Bernardo Camajori Tedeschini, Stefano Savazzi, Roman Stoklasa, Luca Barbieri, Ioannis Stathopoulos, Monica Nicoli, and Luigi Serio, "Decentralized federated learning for healthcare networks: A case study on tumor segmentation," *IEEE Access*, vol. 10, pp. 8693–8708, 2022.

[46] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong, "Faster segment anything: Towards lightweight sam for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.

[47] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 4015–4026.

[48] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[49] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7260–7268.

[50] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc, "Dense classification and implanting for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9258–9267.

[51] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16000–16009.

[52] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue, "Self-supervised learning for few-shot image classification," in

ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 1745–1749.

[53] Yang Li, Feiyun Xu, and Chi-Guhn Lee, "Self-supervised metalearning generative adversarial network for few-shot fault diagnosis of hoisting system with limited data," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2474–2484, 2022.

[54] Terrance DeVries and Graham W Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[55] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[56] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[58] Shuangjie Xu, Daizong Liu, Linchao Bao, Wei Liu, and Pan Zhou, "Mhp-vos: Multiple hypotheses propagation for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 314–323.

[59] Zongxin Yang, Yunchao Wei, and Yi Yang, "Collaborative video object segmentation by foreground-background integration," in *European Conference on Computer Vision*. Springer, 2020, pp. 332–348.

[60] Long Ang Lim and Hacer Yalim Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, vol. 23, no. 3, pp. 1369–1380, 2020.

[61] Islam Osman, Agwad Eltantawy, and Mohamed S Shehata, "Task-based parameter isolation for foreground segmentation without catastrophic

forgetting using multi-scale region and edges fusion network," *Image and Vision Computing*, vol. 113, pp. 104248, 2021.

[62] Trung-Nghia Le and Akihiro Sugimoto, "Deeply supervised 3d recurrent fcn for salient object detection in videos.," in *BMVC*, 2017, vol. 1, p. 3.

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

[64] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, and Qingyao Wu, "A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection," *arXiv preprint arXiv:2203.04708*, 2022.

[65] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 715–731.

[66] Trung-Nghia Le and Akihiro Sugimoto, "Video salient object detection using spatiotemporal deep features," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5002–5015, 2018.

[67] Thangarajah Akilan and QM Jonathan Wu, "sendec: An improved image to image cnn for foreground localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4435–4443, 2019.

[68] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 724–732.

[69] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He, "Reciprocal transformations for unsupervised video object segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15455–15464.

[70] Minhyeok Lee, Suhwan Cho, Seunghoon Lee, Chaewon Park, and Sangyoun Lee, "Unsupervised video object segmentation via prototype memory network," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 5924–5934.

[71] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe, "Unovost: Unsupervised offline video object segmentation and tracking," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 2000–2009.

[72] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia, "Video instance segmentation with a propose-reduce paradigm," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1739–1748.

[73] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11781–11794, 2021.

[74] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu, "Recurrent dynamic embedding for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1332–1341.

[75] Jiaming Zhang, Yutao Cui, Gangshan Wu, and Limin Wang, "Joint modeling of feature, correspondence, and a compressed memory for video object segmentation," *arXiv preprint arXiv:2308.13505*, 2023.

[76] Zongxin Yang and Yi Yang, "Decoupling features in hierarchical propagation for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36324–36336, 2022.

[77] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen, "Joint feature learning and relation modeling for tracking: A one-stream framework," in *European Conference on Computer Vision*. Springer, 2022, pp. 341–357.

Appendix A

Implementation and Reproducibility Details

All experiments in this work use PyTorch 1.13 and Python 3.9. Training, testing, and inference on all studies conducted were performed using two NVIDIA A6000 GPUs and a single NVIDIA GV100 GPU.