Two-Level Temporal Relation Model for Video Instance Segmentation

by

Çağan Selim Çoban

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering

KOÇ ÜNİVERSİTESİ

September 5, 2022

Two-Level Temporal Relation Model for Video Instance Segmentation

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Çağan Selim Çoban

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Dr. Fatma Güney (Advisor)

_____

Prof. Dr. Yücel Yemez

_____

Prof Dr. Hazım Kemal Ekenel

Date: _____

to my all beloved ones

ABSTRACT

Two-Level Temporal Relation Model for Video Instance Segmentation
Çağan Selim Çoban
Master of Science in Computer Science and Engineering
September 5, 2022

In Video Instance Segmentation (VIS), current approaches either focus on the quality of the results, by taking the whole video as input and processing it offline; or on speed, by handling it frame by frame at the cost of competitive performance.

In this work, we propose an online method that is on par with the performance of the offline counterparts. We introduce a message-passing graph neural network that encodes objects and relates them through time. We additionally propose a novel module to fuse features from the feature pyramid network with residual connections.

Our model, trained end-to-end, achieves state-of-the-art performance on the YouTube-VIS dataset within the online methods. Further experiments on DAVIS demonstrate the generalization capability of our model to the video object segmentation task. We also evaluate our work on autonomous driving setting and show comparable results in KITTI MOTS dataset.

ÖZETÇE

Video Örnek Segmentasyonu için İki Seviyeli İlişki Modeli
Çağan Selim Çoban
Bilgisayar Bilimleri ve Mühendisliği, Yüksek Lisans
5 Eylül 2022

Videolarda obje tespiti, segmentasyonu ve takibi alanında mevcut yaklaşımlar ya tüm videoyu girdi olarak alıp çevrimdışı olarak işleyerek sonuçların kalitesine odaklanmakta; ya da kare kare işleyerek performastan feragat ederek hıza odaklanmaktadır.

Bu çalışmada, çevrimdışı muadillerinin performansı ile yakın olan bir çevrimiçi yöntem öneriyoruz. Bu yöntemde özgün olarak nesneleri kodlayan ve onları zaman içinde ilişkilendiren, mesaj ileten bir grafik sinir ağı sunuyoruz. Ayrıca modelimizi, özellik piramidi ağındaki özellikleri artık bağlantılarla birleştirmek için yeni bir modül ile güçlendiriyoruz.

Uçtan uca eğitilmiş modelimiz, çevrimiçi yöntemler dahilinde YouTube-VIS veri setinde muadil modeller arasında en iyi performansı elde etti. DAVIS üzerinde yapılan diğer deneyler, modelimizin video nesnesi bölümleme görevine genelleme kabiliyetini göstermektedir. Ayrıca otonom sürüş ayarı konusundaki çalışmalarımızı değerlendiriyor ve KITTI MOTS veri setinde karşılaştırılabilir sonuçlar gösteriyoruz.

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. Fatma Güney. I am grateful to her for continuous support, mentorship and leadership. Also, I consider myself lucky since I got a chance to work with Jordi Pont-Tuset, and I am grateful to him for his guidance, valuable feedback, and trust in me. A special thanks to our intern, Oğuzhan Keskin, he is a hard working professional.

I sincerely thank all hardworking members of AVG: Kaan Akan, Sadra Safadoust, Nazir Nayal, Taher Anjary, Ege Onat Özsuer, and Mısra Yavuz. We were a great team and learned from each other. This work wouldn't be possible without your fellowship and friendship. I would like to thank all of the professors in our program, we receive a decent education, and this work wouldn't be possible without it. Also, I would like to thank Can Küçüksözen, my project partner in courses; I learned from him a lot.

I want to thank all of my friends; Yavuzhan Danışman, Onur Küçükoğlu, Cem Güney Torun, Tayyip Kaya, Can Berk Artan, Deniz Esenyel, Defne Çetiner, Ece Köksal for their support. Special thanks to Kerem Zaman for the valuable discussions during this thesis work.

Finally, I would like to thank my family: my mother Tülin, my father Atila and our cat Alex for their continuous support and believeing in me.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

VOS	Video Object Segmentation
VIS	Video Instance Segmentation
MOTS	Multi Object Tracking and Segmentation
DAVIS	Densely Annotated VIdeo Segmentation
SSVOS	Semi-Supervised Video Object Segmentation
UVOS	Unsupervised Video Object Segmentation
mAP	Mean Average Precision
CNN	Convolutional Neural Networks
GNN	Graph Neural Networks

Chapter 1

INTRODUCTION

Object detection in videos is one of the fundamental tasks in computer vision. While humans are extremely good at decomposing the world into objects, identifying, tracking, and resolving relations between them, it is not an easy task for machines. A myriad of applications would become possible if machines could achieve a similar decomposition and understand the structure of visual scenes; replicating this intelligence is an essential direction for computer vision.

In this study, we focus on two major directions for detecting, segmenting, and tracking objects in videos. First, a model needs to use temporal knowledge in videos efficiently to boost the performance in failure cases (e.g., large pose variation, camera defocus, motion blur) as shown in Fig. 1.1. Secondly, a model should learn the generalization of objects in a video in test time. We handle these two problems jointly and propose a methodology for each to improve the performance in object detection in videos.

There are several tasks targetting object detection in videos, most popular ones are Video Instance Segmentation (VIS), Video Object Segmentation (VOS), and Multi-Object Tracking and Segmentation (MOTS). Even though there are task-specific differences between them, these tasks share a similar setting: discovering objects and consecutively tracking them.

In this work, we propose an efficient online method that can reach the performance of offline methods in these tasks. We achieve that by modeling the changes to the representation of the objects at two levels to allow the flow of temporal information between frames: a Graph Neural Network (GNN) at the object level and a spatio-temporal feature encoder, coined *ResFuser*, at the feature level. We build on top of a one-stage instance segmentation network, CenterMask [Lee and Park, 2020] and at the feature level, we aggregate features

Figure 1.1: **Common hard cases in video object detection and segmentation.** Motion blur and occlusion. Image is from ImageNet-VID [Russakovsky et al., 2015] dataset.

with the proposed ResFuser module, which uses a skip connection to predict changes from one frame to the next. At the object level, we estimate the changes to the states of objects through time and model the interactions between them with a GNN [Kipf et al., 2019]: Each object becomes a node on the graph represented by its encoded state. The states of objects are updated via message passing between all the objects in consecutive frames. In the end, we associate the objects with an online tracker across multiple frames [Yang et al., 2019a]. Despite the two-level temporal aggregation, our network can be trained end-to-end.

Our comprehensive experiments on Youtube-VIS show the importance of both the feature level and the object level aggregation for the competitive results. Further experiments on DAVIS show our method's generalization capability to one more task: Video Object Segmentation (VOS).

1.1 Contributions and Novelties

We summarize our main contributions as follows:

- **ResFuser for feature sharing in the Feature Pyramid Network (FPN) [Lin et al., 2017] with residual connections:** This module enables to predict the changes to the

feature maps of consecutive frames at each level of the FPN to relate the two frames at the feature-level.

- **A message-passing GNN that encodes objects and relates them across time:** This module exploits the fact that an object is still the most similar to itself from one frame to the next, despite some changes which typically occur due to the objects interacting with each other and with the environment. These interactions lead to some changes in the appearance of the object which we learn to estimate with a GNN.

1.2 Publications

- Çağan Selim Çoban, Oğuzhan Keskin, Jordi Pont-Tuset, Fatma Güney, "Two Level Temporal Model: Two-Level Temporal Relation Model for Online Video Instance Segmentation", Under Review

1.3 Outline of the Thesis

The structure of the thesis is organized as follows:

Chapter 2 provides an overview of the prior works in several object discovery tasks in videos, specifically video object segmentation (VOS), video instance segmentation (VIS) and multi-object tracking and segmentation (MOTS).

Chapter 3 describes the model which is tailored to all of the tasks listed in Chapter 2.

Chapter 4 presents the results of the model.

Chapter 5 concludes the thesis with a brief discussion and possible future directions.

Chapter 2

RELATED WORK

In this section, I describe three tasks and methods in the literature related to our work from these perspectives: Video Instance Segmentation (VIS), Video Object Segmentation (VOS) and Multi-Object Tracking and Segmentation (MOTS).

2.1 Video Instance Segmentation (VIS)

2.1.1 Problem Definition

VIS considers simultaneous detection, classification, segmentation, and tracking of multiple object instances in videos [Yang et al., 2019b]. The task is in a closed-world setting, that is, the number of classes is pre-defined.

2.1.2 Datasets and Metrics

The YouTube-VIS 2019 [Yang et al., 2019a] (YouTube Video Instance Segmentation 2019) dataset is the first large benchmark introduced for the VIS task. It consists of 2883 high-quality YouTube videos containing 4883 unique objects annotated with approximately 131k object masks corresponding to 40 predetermined object categories. The task is to segment and classify each object instance while consistently tracking them across frames. The evaluation metrics are Average Precision (AP) and Average Recall (AR). These metrics are evaluated over 10 IoU thresholds from 50% to 95% at a step of 5%, and calculated firstly by category and finally averaged over the category set.

The metric is a modified version of mean average precision (mAP), with an adopted version for videos.The IoU scheme also considers spatial-temporal consistency of predictions. This implies that it is not enough to detect the object in frames for an algorithm, the predictions need to be tracked across frames. In other words, if the method fails to track the object even they are segmented successfully, it will get a low IoU.

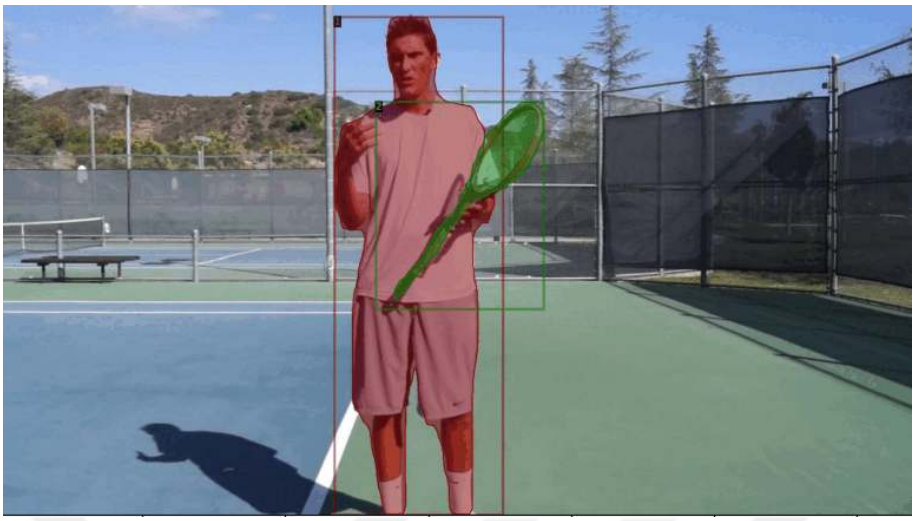

Figure 2.1: **Example from Youtube-VIS dataset.**

2.1.3 Methods

Methods in VIS can be categorized as online or offline, depending on whether they operate *frame-by-frame* or take the whole video as input. **Online** methods typically add a *tracking head* to a segmentation network such as Mask R-CNN [He et al., 2017]. In Mask-Track R-CNN [Yang et al., 2019a], tracking is formulated as a multi-class classification problem into one of the already identified instances or a new unseen instance. This methodology, called tracking-by-segmentation, extends an instance segmentation method with a matching algorithm and a history queue. QueryInst [Fang et al., 2021] treats instances as learnable queries with a multi-stage end-to-end framework and adopts the tracking method proposed in [Yang et al., 2019a]. SipMask [Cao et al., 2020] follows the same association strategy by utilizing a single-stage segmentation network. Single-stage methods for VIS have progressed quite significantly in recent years, leveraging spatio-temporal feature fusion techniques. CrossVIS [Yang et al., 2021] uses features of instances in a frame to localize the same instances in another frame at the pixel level. STMask [Li et al., 2021] employs a frame-level feature calibration module between predicted and ground-truth boxes and a temporal fusion module between consecutive frames to improve the inference of instance masks. SG-Net [Liu et al., 2021] proposes a one-stage framework based on FCOS [Tian et al., 2019] to improve mask quality by dividing target instances into sub-regions and

performing instance segmentation on each sub-region.

We propose an online method by using the single-stage CenterMask [Lee and Park, 2020] as the instance segmentation network. We extend the temporal context with two modules, ResFuser and GNN, at the feature and object levels, respectively. VisSTG [Wang et al., 2021a] also uses a GNN to relate the frames, but at the pixel level, which is too costly due to message passing between many pixels. This, in particular, entails that larger backbones cannot be utilized due to memory constraints. In contrast, our method relates objects with a GNN but still benefits from correlations at the feature level with the ResFuser, which is significantly less costly.

Offline methods operate on the whole video or on multiple frames to make better use of temporal information but this typically hinders efficiency and prevents their use in real-time scenarios. STEm-Seg [Athar et al., 2020], for example, models a video clip as a spatio-temporal volume where instances are represented as clusters inside the volume. Instead of costly 3D volume processing, we use a single-stage segmentation network in the detection phase. Based on DETR [Carion et al., 2020], VisTR [Wang et al., 2021b] uses transformers, but is not fully end-to-end trainable and has slow convergence as explained in [Wu et al., 2022]. The performance of EfficientVIS [Wu et al., 2022] heavily depends on the number of input frames. As shown in our experiments, our method taking just 2 frames as input outperforms the results of EfficientVIS with 9 frames.

2.2 Video Object Segmentation (VOS)

2.2.1 Problem Definition

Video Object Segmentation (VOS) is the task of segmenting and tracking arbitrary, novel objects without considering their semantic categories. Depending on the input at test time, VOS can be divided into semi-supervised (SSVOS or one-shot), where an initial mask of the object of interest is given [Maninis et al., 2018, Perazzi et al., 2017, Voigtlaender et al., 2019a, Wang et al., 2019a, Wang et al., 2019d], and unsupervised (UVOS or zero-shot), without any initial mask [Lu et al., 2019a, Ventura et al., 2019, Wang et al., 2019c, Zhou et al., 2020, Yang et al., 2019c]. Our method falls into the multiple-object zero-shot category. While some benchmarks consider only the dominant object in the scene for

(a) Single Object Case (b) Multi Object Case

Figure 2.2: **Examples from DAVIS dataset.**

segmenting and tracking [Perazzi et al., 2017], in the more generic case the videos have multiple objects [Pont-Tuset et al., 2017, Xu et al., 2018a, Caelles et al., 2019]. In summary, there are three sub-groups of VOS task:

- Multiple-Object Semi-Supervised Video Object Segmentation (SSVOS)
- Single-Object Unsupervised Video Object Segmentation (UVOS)
- Multiple-Object Unsupervised Video Object Segmentation (UVOS)

2.2.2 Datasets and Metrics

DAVIS (Densely Annotated VIdeo Segmentation) challenge is the first benchmark for this task, which covers high-resolution videos. The first version, DAVIS 2016, contains 50 videos with a **single** foreground object per video, which is suitable for both unsupervised and semi-supervised methods. Thereafter, 2017 is introduced with 90 videos with **multiple** objects per video, but this dataset is introduced for self-supervised methods only. On the top of 2017 version, 2019 version is published to benchmark multi-object unsupervised VOS models. Some examples from this dataset is given in Fig. 2.2

However, the latest version of DAVIS dataset has only 90 short video clips, 60 videos are used for training and 30 videos for validation; which is considerably small to train an algorithm. As a remedy, Youtube-VOS [Xu et al., 2018b] dataset is introduced to cover diverse set of objects consisting of 4453 videos. Still, DAVIS has both higher-quality of video resolution and annotations compared to YoutubeVOS since it covers with more

	DAVIS 16	**DAVIS 17 & 19**	**Youtube VOS**
Videos	50	90	4453
Objects	50	205	7755
Annotations	3440	13543	197272

Table 2.1: **Comparison between DAVIS and YouTube-VOS datasets.**

complex scenarios; including camera motion, interactions of objects and occlusions. A comparative table is given at Table 2.1.

For both datasets, the evaluation metrics are region similarity J and contour accuracy F. For a single object, **The Jaccard index (J)** for region similarity is defined as:

$$J = \frac{M \cap G}{M \cup G} \tag{2.1}$$

where M is the predicted segmentation for an object and G is the target ground-truth mask G. Note that, this formulation is similar to IoU measure, and it is not solely sufficient to benchmark an algorithm's performance for segmentation contours. Provided that each mask can be modeled as a set of closed contours, **contour accuracy F** of an object is defined as:

$$F = \frac{2P_c R_c}{P_c + R_c} \tag{2.2}$$

where P_c and R_c are precision and recall values for ground-truth and predicted contour points, aligned by bipartite-graph matching.

In multi-object case, the mean of the measures (J, F) over all object instances are computed. The overall performance metric is the arithmetic mean of J&F. Since the target object categories are unknown in unsupervised multi-object case, methods have to provide a pool of $N \leq 20$ non-overlapping video object proposals for every video sequence. In the next, the proposals and ground-truth objects are associated to compute the highest result with Hungarian algorithm.

2.2.3 Methods

Multiple-Object Semi-Supervised Video Object Segmentation: This task focuses on tracking object instances that are given in the first frame, and mostly evaluated on two datasets: DAVIS 2017 semi-supervised and YoutubeVOS. The methods for this task are strongly depend on the first-frame instances masks; for example in FEELVOS [Voigtlaender et al., 2019b] these masks are generally propagated to the end of the video. Alternatively a segmentation network is fine-tuned conditioned on the first frame in several methods: OSVOS [Caelles et al., 2017], OnAVOS [Voigtlaender and Leibe, 2017], PReMVOS [Luiten et al., 2018]. The seminal work PReMVOS links instance proposals by using similarity and temporal cues. The model's components are finetuned with the first-frame ground truth instances as a guidance objects to track. At inference time, the model performs online tracking by processing one frame at a time.

Single-Object Unsupervised Video Object Segmentation: This task requires to separate foreground/background regions by segmenting and grouping salient object(s) into one, and in contrast to semi-supervised case, no ground truth instance masks are provided at inference time. The algorithms are mostly evaluated on the DAVIS 2016 single-object benchmark and generally perform binary classification in the image level, and boost the performance by using additional supervisions such as optical-flow (LSMO) [Tokmakov et al., 2019], co-attention (COSNet) [Lu et al., 2019b] and graph neural networks (AGNN) [Wang et al., 2019b]. The most similar work to our model, AGNN builds a fully connected graph where nodes correspond to frames and a mask on each frame is related through message passing between them via an attention mechanism. COSNet [Lu et al., 2019b] is also trained on pairs of video frames with a co-attention equipped Siamese network.

Multi-Object Unsupervised Video Object Segmentation (UVOS): This task (also known as zero-shot VOS), is an extension of previous two tasks; the models are expected to discover foreground object(s) and continuously track them. The baseline model is selected as RVOS (Recurrent Video Object Segmentation) method in DAVIS 2019 Challenge for the UVOS task. RVOS consist of a few recurrent neural networks, one operates over the set of objects to generate tracks over the video. As of now, UnOVOST is the top-performing

method; it is based on Mask R-CNN proposals with a strongly engineered clip-level post-processing pipeline leverages motion (optical-flow) heuristics.

2.3 Multiple Object Tracking and Segmentation (MOTS)

2.3.1 Problem Definition

While VIS focuses on segmenting and tracking instances belonging to a diverse set of classes (or no class restriction as in VOS) MOTS was proposed around the same time to track larger number of instances belonging to pedestrian and car classes in driving scenes. KITTI MOTS [Voigtlaender et al., 2019c] as an extension to Multiple Object Tracking. In MOT, the benchmarks only provide bounding boxes and object IDs across time. Bounding boxes can be under-representative in occlusion cases, hence, using segmentation masks is a natural and better choice for evaluation.

2.3.2 Dataset and Metrics

KITTI MOTS is the main benchmark of this task. It consist of 21 videos (12 of them is for training, 9 of them is for validation), 65,213 pixel masks for 977 distinct objects (cars and pedestrians) in 10,870 video frames. An example from this dataset is given in Fig. 2.3. The metrics are derived from the original CLEAR MOT [Bernardina and Stiefelhagen., 2008] metrics, box-based IoU calculations are simply replaced with mask-based counterparts; simply MOTA is replaced with sMOTSA.

2.3.3 Methods

The baseline proposed with the dataset extends the Mask R-CNN by 3D convolutions and a tracking head for the association. [Porzi et al., 2020] propose a new data generation process for MOTS by using optical flow and improving the tracking head with a novel mask-pooling layer. STEm-Seg [Athar et al., 2020] also evaluates it's method in this dataset.

Figure 2.3: **Example from KITTI MOTS dataset**

Chapter 3

METHOD

Our framework *Two-Level Temporal Relation Model* consists of two main parts: (i) we modify an instance segmentation model to aggregate spatio-temporal information between two consecutive frames and (ii) we use an object encoder to represent the objects and then relate them with a GNN-based transition model to learn changes to the representation of the objects. Below we explain the details of each part in detail.

3.1 Spatio-Temporal Feature Aggregation (ResFuser)

We start with an anchor-free image object segmentation method, CenterMask [Lee and Park, 2020], based on a single-stage object detector, FCOS [Tian et al., 2019]. FCOS utilizes multi-level predictions with a Feature Pyramid Network (FPN) to detect objects of various sizes. Given an input frame at time t, FCOS extracts features $\mathbf{p}_i^t$ at each level i. For spatio-temporal feature aggregation, we introduce the ResFuser module with the following functionality:

$$\mathbf{p}_i^{(t+1)} = f_i(\mathbf{p}_i^{(t)}, \mathbf{p}_i^{(t+1)}) + \mathbf{p}_i^{(t+1)} \tag{3.1}$$

where each f_i is implemented as a two-layer CNN to learn residual changes at each level $i = \{3, \ldots, 7\}$ as given in Table B.5. The residual connections allow the fusing of missing information from the previous frame at each level.

A common approach in the design of segmentation networks is to add a separate head for segmentation in addition to the detection and the classification heads of an object detector, e.g. as in Mask R-CNN [He et al., 2017]. CenterMask adds a spatial attention-guided mask (SAG-Mask) branch to the anchor-free one-stage object detector FCOS [Tian et al., 2019]. Given the bounding boxes from the detector, the SAG-Mask branch predicts a segmentation mask for each box with the spatial attention map. The goal of the attention

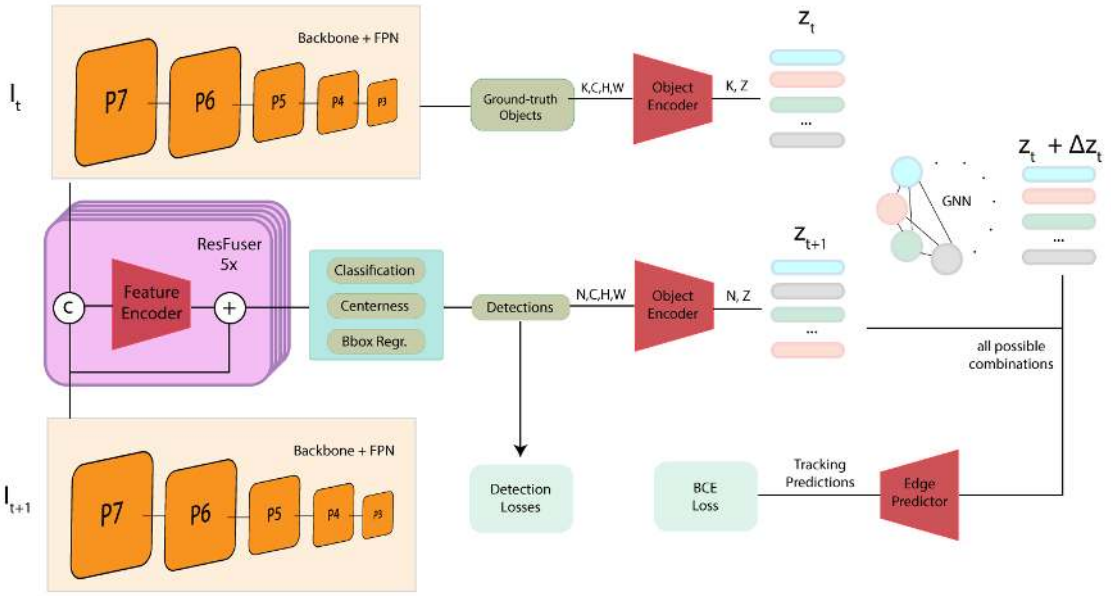

Figure 3.1: **Overall Framework.** Our method takes two consecutive frames as input and relates them temporally at two levels to better segment and track instances in videos: (i) feature level with ResFuser at each level of the feature pyramid and (ii) object level with a GNN after encoding objects with the object encoder.

map is to learn to focus on the pixels belonging to the object and learn to ignore the others around the object. The input to the SAG-Mask layer is the features inside the predicted region of interest (RoI) extracted by RoI Align [He et al., 2017]. Finally, the result of this operation is upsampled and used to predict class-specific masks. We use the same classification, regression, centerness, and mask losses as CenterMask. We use the result of RoI Align also to represent objects as explained next.

3.2 Encoding and Relating Objects

Object Encoder. Our object encoder takes the pooled object-centric features $\mathbf{o}_t^k$ of CenterMask as input for each object k at time t. These features are used to estimate the mask of the object but are high dimensional, e.g. $256 \times 14 \times 14$ with a ResNet-50 (the number of channels times the size of the spatial grid). We first encode these features into a

low-dimensional latent representation $\mathbf{z}_t^k$, for each object k on both frames t and $t+1$:

$$\mathbf{z}_t^k = \mathcal{E}(\mathbf{o}_t^k) \qquad \mathbf{z}_{t+1}^k = \mathcal{E}(\mathbf{o}_{t+1}^k) \tag{3.2}$$

with an object encoder $\mathcal{E}$ which is implemented as a simple two-layer CNN followed by an MLP, given in Table B.4. We relate the objects to each other in this latent representation as explained next.

Relating Objects. Inspired by C-SWM [Kipf et al., 2019], we relate the objects using a Graph Neural Network (GNN). GNN models the state transitions of objects in the latent space by considering pairwise interactions between them. While we cannot explicitly model the action on the object which causes the transition as in C-SWM, we take advantage of an object being the most similar to itself from one frame to the next.

We construct a fully-connected graph withe the objects as nodes. Our goal is to learn the state transitions, i.e. the change in the state representation of each object on the graph from one frame to the next, and we do so by passing messages between them. The key observation is that the state transition of a node in the GNN should match the residual in between the object representation in two consecutive frames. We use the same object encoder in the following frame and enforce the difference between the state representations of the first frame and the next to be the state transition estimated by the GNN.

We use the encoded latent representations $\mathbf{z}_t^k$ as node features for each object $k \in \{1, \dots, K\}$ where K is the number of objects segmented on frame t. Our goal is to learn the transition $\Delta\mathbf{z}_t^k$ to model the change from frame t to $t+1$:

$$\mathbf{z}_{t+1}^k = \mathbf{z}_t^k + \Delta\mathbf{z}_t^k \tag{3.3}$$

at each node/object k. The output of the GNN is the set of transitions for all K objects at frame t, $\Delta\mathbf{z}_t = \{\Delta\mathbf{z}_t^1, \dots, \Delta\mathbf{z}_t^K\}$.

While learning the transitions, GNN relates objects to each other by passing messages between them. The message passing operation is performed by iteratively updating the node representations:

$$\mathbf{e}_t^{i,j} = f_e(\mathbf{z}_t^i,\ \mathbf{z}_{t+1}^j) \qquad\qquad \Delta\mathbf{z}_t^j = f_n(\mathbf{z}_t^j,\ \sum_{i \neq j} \mathbf{e}_t^{i,j}) \tag{3.4}$$

where f_n and f_e are the node and the edge update functions, respectively; and $\mathbf{e}_t^{i,j}$ is the edge representation for the edge between the node i and j at time t. After one step of message passing, nodes become more aware of the other objects in the scene as well as their transitions.

We use the same edge and node networks as in C-SWM [Kipf et al., 2019] with a custom selection of the number of hidden units as 128 and node dimension as 512. The detailed architecture for edge and node models and edge predictor is given in Table B.1, Table B.2 and Table B.3 respectively.

We train the network a binary-cross-entropy loss for associating instances in a video. We define a positive edge between the ground-truth instance node of an object in the first frame and a proposal node that belongs to the same object in the next frame. During inference, we employ the same tracking scheme as MaskTrack R-CNN [Yang et al., 2019a] with the following modification on the score matrix: We use the predicted edge scores instead of the inner-product of object vectors as done in [Yang et al., 2019a].

3.3 Tracking Scheme

We use the same tracking scheme as MaskTrack-RCNN [Yang et al., 2019a] with the initial parameters for the similarity matrix. For matching an ID *n* to a candidate box *i*, the similarity matrix is formed as follows:

$$v_i(n) = \psi \, log p_i(n) + \alpha \log s_i + \beta \, \mathrm{IoU}(b_i, b_n) + \gamma \, \delta(c_i, c_n) \tag{3.5}$$

where p_i calculates the similarity score between the two object-vectors, s_i is the score, δ is the indicator function that is 1 when the two objects belong to the same class and 0 otherwise. We performed a grid search to adjust the coefficients ψ, α, γ in our setting.

We started with the initial parameters are $\psi = 1.0, \alpha = 1.0, \beta = 2.0, \gamma = 10$ for ResNet-50 based model, with an average precision of 35.23 on the Youtube-VIS 19 validation set. The fine-tuned parameters are $\psi = 0.5, \alpha = 1.5, \beta = 2.5, \gamma = 10.$, with the final reported performance of 35.82 mAP. We follow the same approach ResNet-101 backbone and obtain the values as $\psi = 1.25, \alpha = 1.0, \beta = 2.5, \gamma = 10$. We use 0.05 as the threshold in inference, which is also used in the original CenterMask detector. In

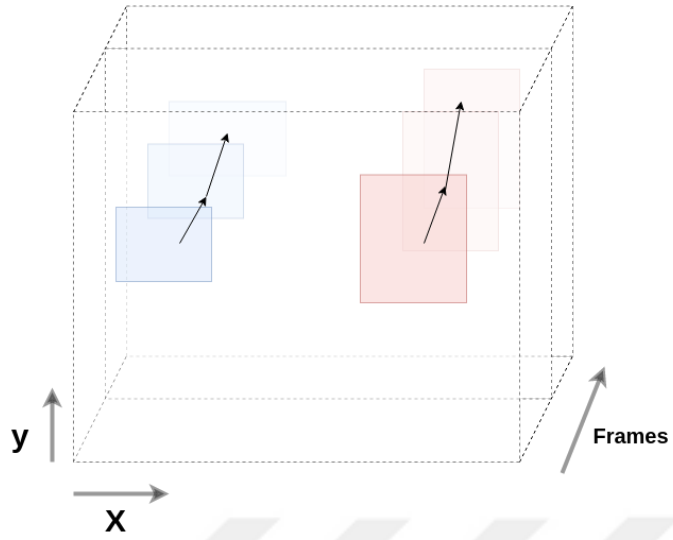

Figure 3.2: **IoU Tracker**

the experiments, we find that the model is prone to produce significantly more detections because of this low threshold, our tracking scheme can select the correct detections to filter the unnecessary ones.

As a baseline, we use our pretrained off-the-shelf CenterMask detector with a small change in the tracker. We set $\psi = 0$ since we are not using object vectors and keep other coefficients as tuned. The tracking is scheme simply a version of IoUTracker as shown in Fig. 3.2. Also, class information and detection score are used as an additional cue. This kind of tracker usually fails the cases where there are similar and occluded objects.

Chapter 4

EXPERIMENTS

4.1 Training Details

In the following sections, I will describe the training process of our model in two folds.

4.1.1 Pretraining with Images

Pretraining an object detector is very important to achieve a reliable starting point. Initialization with an ImageNet pretrained backbone is the golden standard for object detectors, especially for one of the most popular object detection benchmarks, COCO, which consists of 80 classes. In the video instance segmentation (VIS) task the number of classes is smaller than COCO; there are 40 classes where 19 of them overlaps with COCO. Therefore, an out of the box object detector that is trained on COCO is not compatible to VIS benchmark.

One can initialize the backbone as in object-detection case, and train an object detector as the pretraining process. However, in Youtube-VIS dataset, the number of unique objects are quite limited, which makes the models prone to overfit during the training process. As a remedy, extending the dataset with a large-scale object detection dataset is a natural choice.

To have a good starting point, we train CenterMask [Lee and Park, 2020] on a combination of COCO [Lin et al., 2014], YouTube-VIS [Yang et al., 2019a] and Open Images [Kuznetsova et al., 2020] for the 40 classes of YouTube-VIS 2019, similarly to [Luiten et al., 2020]. We used 19 overlapping classes of COCO, and we mapped 60 similar classes of Open Images to the 40 classes of YouTube-VIS i.e. similar classes are represented with samples from both datasets (please see Appendix A for the mapping table). The resulting dataset distribution is given Table 4.1.

As the train time augmentation, random horizontal flips and multi-scale training is used. In multi-scale training, an input image is randomly resized to a height of the following

person	299286	giant_panda	6039	lizard	6088
parrot	6422	skateboard	18837	sedan	187752
ape	5903	dog	26442	snake	3274
monkey	5542	hand	5997	rabbit	3896
duck	19319	cat	16425	cow	16306
fish	12737	train	13810	horse	15221
turtle	4472	bear	3662	motorbike	20421
giraffe	7180	leopard	4042	fox	2715
deer	1680	owl	2669	surfboard	8705
airplane	22121	truck	10594	zebra	6952
tiger	2436	elephant	8452	snowboard	3771
boat	15276	shark	1554	mouse	2599
frog	2864	eagle	2371	earless_seal	4028
tennis racket	7001				
total	**814861**				

Table 4.1: Our dataset's distribution of instances among all 40 categories

set: (640, 672, 704, 736, 768, 800). Random flip is applied with a probability of 0.5. We use the same losses and the learning rate as in CenterMask [Lee and Park, 2020] except MaskIoU [Huang et al., 2019], which we have observed to add little to the performance as shown in Table 4.2.

4.1.2 Training with Videos and Motion Hallucinated Images

For data augmentation, we apply random affine transformations and motion blur as in [Athar et al., 2020] to simulate video movement from static images as shown in Fig. 4.1

We train our method using Stochastic Gradient Descent (SGD) for 180K iterations with a batch size of 16 and an initial learning rate of $5e^{-3}$. We decrease the learning rate by a factor of 10 after 100K and 150K iterations.

MaskIoU	AP^{mask}	AP^{mask}_L	AP^{mask}_M	AP^{mask}_S	AP^{box}	AP^{box}_L	AP^{box}_M	AP^{box}_S
Off	36.71	52.27	39.65	18.13	41.77	53.23	45.60	25.04
On	37.83	53.22	40.79	19.33	42.07	53.54	45.76	26.10

Table 4.2: **MaskIoU Branch.** We ablate the MaskIoU branch with ResNet-50 backbone and multi-scale training.

Figure 4.1: **Motion Augmentation.** During training, our method takes a static image and processes it to have a slightly different version.

4.2 Quantitative Results

4.2.1 Video Instance Segmentation

Table 4.3 compares our method to online and real-time state-of-the-art VIS methods on YouTube-VIS 2019. Following the literature, we perform a separate comparison by using a small (ResNet-50 [He et al., 2016]) and a large (ResNet-101 [He et al., 2016]) backbone. The results show that our model outperforms the other models by 1.8 points with the ResNet-101 backbone while preserving a competitive performance with ResNet-50. Our model extends the capability of online models to benefit more from the temporal context with the two proposed modules, GNN and ResFuser. Furthermore, our model can better utilize a large number of parameters as shown in the SoTA performance with the large backbone. Since these two modules have additional parameters, more data is required, both in terms of quantity and variety as shown in our ablations (Section 4.2.4). The only other model that employs a GNN for feature aggregation is VisSTG [Wang et al., 2021a]

Method (ResNet-50)	**Aug.**	**FPS**	**AP**	**AP50**	**AP75**	**AR1**	**AR10**
MaskTrack R-CNN [Yang et al., 2019a]	-	33	30.3	51.1	32.6	31.0	35.5
STEm-Seg [Athar et al., 2020]	-	7	30.6	50.7	33.5	31.6	37.1
SipMask [Cao et al., 2020]	-	34	32.5	53.0	33.3	33.5	38.9
CompFeat [Fu et al., 2020]	-	<33	35.3	56.0	38.6	33.1	40.3
TraDeS [Wu et al., 2021]	-	26	32.6	52.6	32.8	29.1	36.6
CrossVIS [Yang et al., 2021]	-	40	34.8	54.6	37.9	34.0	39.0
STMask [Li et al., 2021]	DCN	29	33.5	52.1	36.9	31.1	39.2
SG-Net [Liu et al., 2021]	MS	23	34.8	56.1	36.8	35.8	40.8
Ours	MS	15	35.8	57.1	38.0	34.7	41.2
QueryInst [Fang et al., 2021]	MS	32	36.2	56.7	39.7	**36.1**	**42.9**
CrossVIS [Yang et al., 2021]	MS	40	36.3	56.8	38.9	35.6	40.7
VisSTG [Wang et al., 2021a]	MS	22	**36.5**	**58.6**	**39.0**	35.5	40.8

Method (ResNet-101)	**Aug.**	**FPS**	**AP**	**AP50**	**AP75**	**AR1**	**AR10**
MaskTrack R-CNN [Yang et al., 2019a]	-	29	31.9	53.7	32.3	32.5	37.7
SRNet [Ying et al., 2021]	-	35	32.3	50.2	34.8	32.3	40.1
STEm-Seg [Athar et al., 2020]	-	7	34.6	55.8	37.9	34.4	41.6
CrossVIS [Yang et al., 2021]	-	36	36.6	57.3	39.7	36.0	42.0
SipMask [Cao et al., 2020]	MS	24	35.8	56.0	39.0	35.4	42.4
STMask [Li et al., 2021]	DCN	23	36.8	56.8	38.0	34.8	41.8
SG-Net [Liu et al., 2021]	MS	20	36.3	57.1	39.6	35.9	43.0
Ours	MS	14	**38.1**	**61.9**	**40.6**	**37.0**	**44.4**

Table 4.3: **Results on YouTube-VIS 2019.** We compare our method to other online state-of-the-art methods on YouTube-VIS 2019 `val` set using ResNet-50 (top) and ResNet-101 (bottom) as backbones. "MS" denotes multi-scale training. DCN [Dai et al., 2017] denotes using Deformable Convolutions.

but their results are not reported with the ResNet-101 backbone, probably due to overly increased computational and memory requirements.

Note that offline methods which make use of the full sequence are not included in

Method	Online	E2E	J & F	Mean	Recall	Decay	Mean	Recall	Decay
RVOS [Ventura et al., 2019]		✓	41.2	36.8	40.2	0.5	45.7	46.4	**1.7**
KIS [Cho et al., 2019]	✓	-	59.9	-	-	-	-	-	-
Ours		✓	**61.9**	**60.7**	**70.3**	**-2.4**	**63.1**	**70.9**	1.9
AGNN [Wang et al., 2019b]		-	61.1	58.9	65.7	11.7	63.2	67.1	14.3
STEm-Seg [Athar et al., 2020]	-	✓	64.7	61.5	70.4	**-4**	67.8	75.5	1.2
UnOVOST [Luiten et al., 2020]		-	67.9	**66.4**	**76.4**	-0.2	69.3	**76.9**	**0.01**
Propose-Reduce [Lin et al., 2021]		✓	**68.3**	65.0	-	-	**71.6**	-	-

Table 4.4: **Results on DAVIS-19.** We compare our method to state-of-the-art methods, online at the top and offline at the bottom, on DAVIS-19 Unsupervised `val` set. All results are reported with the ResNet-101 backbone.

Table 4.3 for a fair comparison. For reference, VisTR [Wang et al., 2021b], a real-time and end-to-end trainable encoder-decoder transformer-based framework, reports a score of 35.6 mAP with ResNet-50 backbone and 38.6 mAP with ResNet-101 backbone with $T = 36$ frames as input, which is the largest amount of annotated frames on YouTube-VIS-19 dataset. Similarly, the state-of-the-art EfficientVIS [Wu et al., 2022] achieves 37.9 mAP with ResNet-50 and 39.8 mAP with ResNet-101 backbone using $T = 36$. These high scores can be attributed to rich information available on the whole video level as proven by a score of 35.3 mAP when reducing the temporal window to $T = 9$ input frames [Wu et al., 2022], while our method can obtain 35.8 mAP with just $T = 2$ frames. Although STEm-Seg [Athar et al., 2020] is also a whole-video-level method, we still include it in our comparisons because, to the best of our knowledge, it is the first method that reports results on both VIS and VOS tasks like us.

4.2.2 Video Object Segmentation

We evaluate our model's performance on the additional task of VOS in Table 4.4 without training on it. Our method considerably outperforms previous online methods RVOS [Ventura et al., 2019] and KIS [Cho et al., 2019], and achieves a slightly better $J\&F$ score with a noticeably less decay than offline AGNN [Wang et al., 2019b]. Our method is only 2.5 points below the offline STEm-Seg, despite the 14-frame video input used by their method.

With $T = 4$ input frames, STEm-Seg reports a $J\&F$ score of 62.2 which is similar to our method's performance (61.9) with only $T = 2$ input frames. Furthermore, STEm-Seg is trained on both image (COCO [Lin et al., 2014] and PASCAL [Everingham et al., 2010]) and video datasets (YouTube-VIS and DAVIS) whereas our method is not even trained on DAVIS, which shows a better generalization performance to the video object segmentation task. Although the offline UnOVOST [Luiten et al., 2020] method achieves an impressive score of 67.9, it cannot be trained end-to-end due to its complex post-processing based on heuristics that are specific to this benchmark. Also, it is significantly slower (1 vs. 14 FPS). Propose-Reduce [Lin et al., 2021], current state of the art on DAVIS-19, is another offline method based on performing segmentation on keyframes and then propagating it with a two-stage network. Both the performance and efficiency are highly dependent on the number of keyframes.

4.2.3 Multi Object Segmentation and Tracking

We extended our research to autonomous driving scenarios by adapting our work to KITTI MOTS dataset. Different from video instance segmentation and video object segmentation tasks, KITTI MOTS includes more longer clips and objects.

Cars	sMOTSA	MOTSA	MOTSP	IDS
UnOVOST	50.7	60.2	85.6	151
TrackRCNN	76.2	87.8	87.2	93
StemSeg	72.7	83.8	87.2	76
Ours	64.0	74.4	88.0	347
Pedestrians	sMOTSA	MOTSA	MOTSP	IDS
UnOVOST	33.4	47.7	76.0	68
TrackRCNN	46.8	65.1	75.7	78
StemSeg	50.4	66.1	77.7	14
Ours	32.1	47.4	79.8	331

Table 4.5: **MOTS Results**

	YTVIS-19 (mAP)	DAVIS-19 (J&F)
No Open Images	28.0	54.7
Open Images	34.3	59.1
Open Images + ResFuser	34.9	59.4
Open Images + ResFuser + GNN	**35.8**	**60.5**

Table 4.6: **Ablation Study.** We show the effect of pre-training on OpenImages [Kuznetsova et al., 2020], and our two contributions, ResFuser and GNN on YoutubeVIS and DAVIS. All results are presented with the ResNet-50 backbone.

We pre-trained our model in COCO for two classes with 270K iterations, then finetuned on Mapillary dataset with 50K iterations. Then, the whole network is trained end-to-end on KITTI MOTS dataset with 5K iterations. The results show that we are slightly underperforming than the baseline as UnOVOST. We argue that our algorithm is more aligned with video instance/object segmentation tasks rather than multi-object tracking and segmentation. Specifically, our method suffers significantly higher ID switches (IDS) than other methods in both classes. This is mostly because of our tracking scheme since we have an online setup and are processing two frames in one pass. Track R-CNN uses three frames, StemSeg uses 8, and UnOVOST gets the whole video as the input. Also, in pedestrian class, we perform the worse even though in car class we have comparably good performance. This is probably because our embedding strategy fails to discriminate similarly looking instances, and there is room for improvement in detection performance.

4.2.4 Ablation Study

Table 4.6 presents our ablation experiments to isolate the effect of each of our contributions. The first row is a baseline with CenterMask pre-trained only on YouTube-VIS-19 [Yang et al., 2019a] and COCO [Lin et al., 2014]. Here, we use an IoU-based tracker by removing the Kalman Filter from the well-known SORT algorithm [Bewley et al., 2016]. In the second row, we pre-train CenterMask [Lee and Park, 2020] additionally on Open Images [Kuznetsova et al., 2020] to see the effect of a larger-scale image instance segmen-

tation dataset on video segmentation problems. Although Open Images is not typically used for video segmentation problems, our framework greatly benefits from this additional data as it relies on instance detection and segmentation on the frame level. We observe a significant boost of +6.3 mAP on YouTube-VIS-19 and +4.4 $J\&F$ on DAVIS-19.

We then add our feature-sharing module ResFuser which is a simple feature-sharing procedure between the consecutive frames to improve detection and segmentation by exploiting similar feature representations of corresponding instances. Despite being conceptually very simple, ResFuser consistently improves the results on both datasets too (+0.6 mAP on YouTube-VIS-19 and +0.3 $J\&F$ on DAVIS-19).

Finally, the last row shows that linking objects across time with a GNN further improves the performance by a noticeable margin (+0.9 mAP in YouTube-VIS-19). The gain on the DAVIS-19 unsupervised validation set is even higher (+1.1 $J\&F$). This reinforces our claim that GNN enhances instance segmentation and tracking in multiple objects, which is the case for DAVIS-19 as it contains a large number of sequences with 2 or 3 objects (the average number of objects per sequence is 2.2 on the validation set [Caelles et al., 2019]). Our method of relating objects with a GNN shows promising improvements for both video segmentation tasks.

4.3 Qualitative Analysis

We show two qualitative examples in Fig. 4.2 comparing the results with our GNN module versus those without. In the case of VIS, our model with GNN can correctly segment and track the tennis racket and the two players despite frequent exiting and re-entering throughout the video. The model without GNN, on the other hand, assigns a different ID to each object when they re-enter the view, showing the benefits of our object-level reasoning in the GNN. In the VOS video, we demonstrate our model's capability to segment and track object instances with similar appearances and frequent occlusions. The two fish occluding each other on the first frame (1 and 5) can be separated correctly with the help of GNN while they are treated as a single object without GNN. In the next frame, a fish makes a fast turn and in the third frame, another fish swims at the bottom edge of the frame. These situations significantly affect object appearances, and GNN continues to detect and

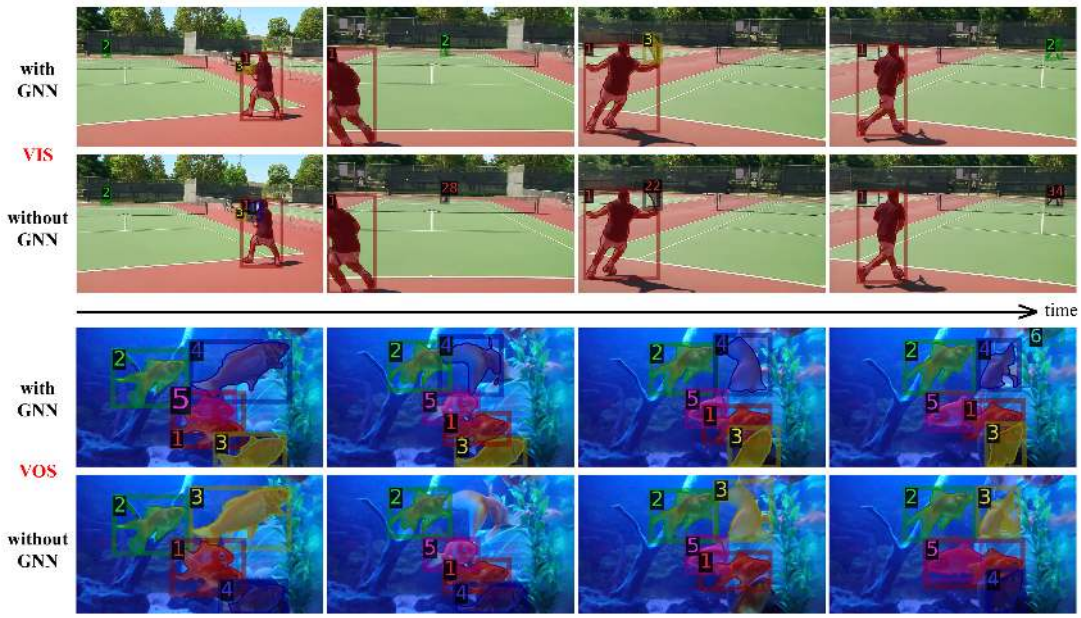

Figure 4.2: **Qualitative Results.**

track these fish as opposed to the model without GNN. Finally, in the last frame, our model with GNN can segment and identify the fish entering the frame (6) as a new object despite being heavily camouflaged.

Chapter 5

CONCLUSIONS

5.1 Summary

In this thesis, we present a new end-to-end trainable framework for simultaneously detecting, segmenting, classifying, and tracking objects in videos online. We introduced two key novelties: *an effective skip-connection module (ResFuser)* that allows feature sharing between consecutive frames and an *object-relating GNN*.

These two modules enable feature aggregation on two levels: feature level and object level. By the ResFuser module, feature aggregation is performed at each level of the FPN. This allows the model to fuse features at different scales in a residual fashion. By the GNN, the model is able to solve inter-relations between objects by message passing in the object level, where each node corresponds to an object in the graph.

The two together reach state-of-the-art results on the challenging YouTube-VIS dataset, with an AP of 38.1 at 14 frames per second. We showed that our framework generalizes well to the VOS task by achieving competitive results (61.9 J & F Mean score) without seeing the corresponding training data.

We also argue that Youtube-VIS dataset is not sufficient to learn and generalize object categories in video instance segmentation task due to lacking number of unique objects. To this end, we curated a dataset targeting video instance segmentation task by effectively combining Youtube-VIS, OpenImages and COCO datasets. This joint dataset allows the model to learn the class distribution well.

5.2 Limitations and Future Directions

We present our model to solve three main tasks of object detection in videos: video instance segmentation (VIS), video object segmentation (VOS), and multiple object tracking and segmentation (MOTS). In all of these tasks, our model can get competitive results even

though the method is online and uses only two frames. Using only two frames is a limitation since it restricts the availability of temporal information. Hence, a natural direction for extending this work is to increase the number of input frames of the network to enable the network to model temporal context better.

However, extending our formulation to multi-frame is not trivial because it requires additional effort to update our novel modules: ResFuser and GNN. First, ResFuser needs to be changed to incorporate more than two frames. In the current design, the ResFuser module predicts a delta for FPN features at multiple scales given the first frame features. In the multi-frame setting, the temporal feature sources for predicting the delta for the current frame can be one previous frame *(t-1)* and the next one *(t+1)* for predicting a delta at *t*. Using 3D convolutions or attention modules are the other alternatives; these all are experimental questions to be answered.

In addition, the graph neural network module must be updated to comply with the multi-frame scenario. Specifically, the graph structure should be re-engineered to construct temporal and spatial links over multiple frames. First, a fully-connected graph can be heavy due to increased temporal context and prone to accumulate unnecessary information during node and edge updates. Specifically, edge and node update functions can use temporal distance; in other words, an object node can pool more information from nearby frames in time, or the graph can be pruned depending on some additional cues; such as class information and spatial distance. At last, the direction of the node updates is also essential. In our two-frame version, we update the first frame features to make it similar to the next frame. In more than two frame scenarios, we need to update node features in multiple passes, possibly in a bi-directional fashion, in both forward and backward directions in time.Another extension to GNN can be adding an appropriate action to the graph nodes, such as instance-specific optical flow. With this adoption, the model is expected to use motion information as actions while updating nodes (objects) to model object-to-object interactions and can deliver promising results.

Also, our tracker is not fully adjusted for the MOTS benchmark. In MOTS experiments, our ID switches are high compared to other models, which indicates that our model can provide more performance with better post-processing and temporal modeling. We can try the extended multi-frame model in this dataset to see the performance gain and build a

new tracker tailored to the MOTS benchmark.

Lastly, our performance gain is also based on curating the dataset and including rare class examples within OpenImages; the gain is even more than the method itself. Our results show that OpenImages allows learning complex temporal relations where previous datasets fall short. Even though our performance is data-driven, we think that an ultimate goal can be to incorporate cues to learn object detection in a completely unsupervised fashion. For example, using motion information to segment moving objects and making a graph neural network to model object-to-object interactions and concurrently supervise the segmentation branch can be a promising future direction. However, several works in the unsupervised object discovery domain, including our base GNN model C-SWM, are limited to working on synthetic datasets. Even though segmenting salient regions can be done with today's models, grouping multiple objects into one or dissecting an object into several segments is still a unresolved problem in computer vision research. Moreover, the definition of an object is vague, it changes from context to context, and a widely accepted benchmark for unsupervised object discovery in real videos is still unavailable. On top of our work, one can start with building such a benchmark, then try the ideas on the new dataset with an appropriate definition of an object.

BIBLIOGRAPHY

[Athar et al., 2020] Athar, A., Mahadevan, S., Osep, A., Leal-Taixé, L., and Leibe, B. (2020). Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *Proc. of the European Conf. on Computer Vision (ECCV)*.

[Bernardina and Stiefelhagen., 2008] Bernardina, K. and Stiefelhagen., R. (2008). Learning to segment moving objects.

[Bewley et al., 2016] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In *Proc. IEEE International Conf. on Image Processing (ICIP)*.

[Caelles et al., 2017] Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Van Gool, L. (2017). One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

[Caelles et al., 2019] Caelles, S., Pont-Tuset, J., Perazzi, F., Montes, A., Maninis, K.-K., and Van Gool, L. (2019). The 2019 DAVIS challenge on VOS: Unsupervised multi-object segmentation. *arXiv:1905.00737*.

[Cao et al., 2020] Cao, J., Anwer, R. M., Cholakkal, H., Khan, F. S., Pang, Y., and Shao, L. (2020). Sipmask: Spatial information preservation for fast image and video instance segmentation. In *Proc. of the European Conf. on Computer Vision (ECCV)*.

[Carion et al., 2020] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*.

[Cho et al., 2019] Cho, D., Hong, S., Kang, S., and Kim, J. (2019). Key instance selection for unsupervised video object segmentation. *arXiv preprint arXiv:1906.07851*.

[Dai et al., 2017] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338.

[Fang et al., 2021] Fang, Y., Yang, S., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., and Liu, W. (2021). Instances as queries. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Fu et al., 2020] Fu, Y., Yang, L., Liu, D., Huang, T. S., and Shi, H. (2020). Compfeat: Comprehensive feature aggregation for video instance segmentation. *arXiv preprint arXiv:2012.03400*.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Huang et al., 2019] Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X. (2019). Mask scoring r-cnn. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Kipf et al., 2019] Kipf, T., van der Pol, E., and Welling, M. (2019). Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*.

[Kuznetsova et al., 2020] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. (2020). The open images dataset v4. *International Journal of Computer Vision (IJCV)*, 128(7):1956–1981.

[Lee and Park, 2020] Lee, Y. and Park, J. (2020). Centermask : Real-time anchor-free instance segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Li et al., 2021] Li, M., Li, S., Li, L., and Zhang, L. (2021). Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lin et al., 2021] Lin, H., Wu, R., Liu, S., Lu, J., and Jia, J. (2021). Video instance segmentation with a propose-reduce paradigm. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Lin et al., 2017] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Proc. of the European Conf. on Computer Vision (ECCV)*.

[Liu et al., 2021] Liu, D., Cui, Y., Tan, W., and Chen, Y. (2021). Sg-net: Spatial granularity network for one-stage video instance segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lu et al., 2019a] Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., and Porikli, F. (2019a). See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lu et al., 2019b] Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., and Porikli, F. (2019b). See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Luiten et al., 2018] Luiten, J., Voigtlaender, P., and Leibe, B. (2018). Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*.

[Luiten et al., 2020] Luiten, J., Zulfikar, I. E., and Leibe, B. (2020). Unovost: Unsupervised offline video object segmentation and tracking. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[Maninis et al., 2018] Maninis, K.-K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Gool, L. V. (2018). Video object segmentation without temporal information. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*.

[Perazzi et al., 2017] Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., and Sorkine-Hornung, A. (2017). Learning video object segmentation from static images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Pont-Tuset et al., 2017] Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., and Van Gool, L. (2017). The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*.

[Porzi et al., 2020] Porzi, L., Hofinger, M., Ruiz, I., Serrat, J., Bulò, S. R., and Kontschieder, P. (2020). Mots: Multi-object tracking and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Tian et al., 2019] Tian, Z., Shen, C., Chen, H., and He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Tokmakov et al., 2019] Tokmakov, P., Schmid, C., and Alahari, K. (2019). Learning to segment moving objects. *Int. J. Comput. Vision*, 127(3):282–301.

[Ventura et al., 2019] Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., and Giro-i Nieto, X. (2019). Rvos: End-to-end recurrent network for video object segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Voigtlaender et al., 2019a] Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., and Chen, L.-C. (2019a). Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Voigtlaender et al., 2019b] Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., and Chen, L.-C. (2019b). FEELVOS: Fast end-to-end embedding learning for video object segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Voigtlaender et al., 2019c] Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., and Leibe, B. (2019c). Mots: Multi-object tracking and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Voigtlaender and Leibe, 2017] Voigtlaender, P. and Leibe, B. (2017). Online adaptation of convolutional neural networks for video object segmentation. In *British Machine Vision Conference*.

[Wang et al., 2019a] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., and Torr, P. H. (2019a). Fast online object tracking and segmentation: A unifying approach. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wang et al., 2021a] Wang, T., Xu, N., Chen, K., and Lin, W. (2021a). End-to-end video instance segmentation via spatial-temporal graph neural networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Wang et al., 2019b] Wang, W., Lu, X., Shen, J., Crandall, D. J., and Shao, L. (2019b). Zero-shot video object segmentation via attentive graph neural networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Wang et al., 2019c] Wang, W., Song, H., Zhao, S., Shen, J., Zhao, S., Hoi, S. C. H., and Ling, H. (2019c). Learning unsupervised video object segmentation through visual attention. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wang et al., 2021b] Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., and Xia, H. (2021b). End-to-end video instance segmentation with transformers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wang et al., 2019d] Wang, Z., Xu, J., Liu, L., Zhu, F., and Shao, L. (2019d). RANet: Ranking attention network for fast video object segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wu et al., 2021] Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., and Yuan, J. (2021). Track to detect and segment: An online multi-object tracker. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wu et al., 2022] Wu, J., Yarram, S., Liang, H., Lan, T., Yuan, J., Eledath, J., and Medioni, G. (2022). Efficient video instance segmentation via tracklet query and proposal. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Xu et al., 2018a] Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., and Huang, T. (2018a). YouTube-VOS: Sequence-to-sequence video object segmentation. In *Proc. of the European Conf. on Computer Vision (ECCV)*.

[Xu et al., 2018b] Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., and Huang, T. S. (2018b). Youtube-vos: A large-scale video object segmentation benchmark. *ArXiv*, abs/1809.03327.

[Yang et al., 2019a] Yang, L., Fan, Y., and Xu, N. (2019a). Video instance segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Yang et al., 2019b] Yang, L., Fan, Y., and Xu, N. (2019b). Video instance segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Yang et al., 2021] Yang, S., Fang, Y., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., and Liu, W. (2021). Crossover learning for fast online video instance segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Yang et al., 2019c] Yang, Z., Wang, Q., Bertinetto, L., Hu, W., Bai, S., and Torr, P. H. (2019c). Anchor diffusion for unsupervised video object segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

[Ying et al., 2021] Ying, X., Li, X., and Chuah, M. C. (2021). Srnet: Spatial relation network for efficient single-stage instance segmentation in videos. In *Proceedings of the 29th ACM International Conference on Multimedia*.

[Zhou et al., 2020] Zhou, T., Li, J., Wang, S., Tao, R., and Shen, J. (2020). MATNet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Trans. on Image Processing (TIP)*, 29.

Appendix A

OPENIMAGES TO YOUTUBE-VIS CLASS MAPPING

Table A.1: Mapping of Classes from OpenImages to Youtube-VIS.

OpenImages Class ID	OpenImages Class Name	YoutubeVIS Class Name
/m/0cmf2	Airplane	airplane
/m/0k5j	Aircraft	airplane
/m/01dws	Bear	bear
/m/01dxs	Brown bear	bear
/m/0633h	Polar bear	bear
/m/01btn	Barge	boat
/m/0ph39	Canoe	boat
/m/01yrx	Cat	cat
/m/01xq0k1	Cattle	cow
/m/0dbzx	Mule	cow
/m/0bt9lr	Dog	dog
/m/09ddx	Duck	duck
/m/0dbvp	Goose	duck
/m/0dftk	Swan	duck
/m/09csl	Eagle	eagle
/m/02l8p9	Harbor seal	earless_seal
/m/0gd36	Sea lion	earless_seal
/m/0bwd_0j	Elephant	elephant
/m/03fj2	Goldfish	fish
/m/0ch_cf	Fish	fish
/m/0306r	Fox	fox

/m/09ld4	Frog	frog
/m/03bj1	Panda	giant_panda
/m/06l9r	Red panda	giant_panda
/m/03bk1	Giraffe	giraffe
/m/0174n1	Glove	hand
/m/03grzl	Baseball glove	hand
/m/03k3r	Horse	horse
/m/0449p	Jaguar (Animal)	leopard
/m/04g2r	Lynx	leopard
/m/0c29q	Leopard	leopard
/m/0cd4d	Cheetah	leopard
/m/04m9y	Lizard	lizard
/m/08pbxl	Monkey	monkey
/m/04_sv	Motorcycle	motorbike
/m/03qrc	Hamster	mouse
/m/04rmv	Mouse	mouse
/m/09d5_	Owl	owl
/m/0gv1x	Parrot	parrot
/m/01bl7v	Boy	person
/m/01g317	Person	person
/m/03bt1vf	Woman	person
/m/04yx4	Man	person
/m/05r655	Girl	person
/m/06mf6	Rabbit	rabbit
/m/0k4j	Car	sedan
/m/0pg52	Taxi	sedan
/m/0by6g	Shark	shark
/m/06_fw	Skateboard	skateboard
/m/078jl	Snake	snake

/m/019w40	Surfboard	surfboard
/m/05_5p_0	Table tennis racket	tennis_racket
/m/0h8my_4	Tennis racket	tennis_racket
/m/07dm6	Tiger	tiger
/m/07jdr	Train	train
/m/07r04	Truck	truck
/m/011k07	Tortoise	turtle
/m/0120dh	Sea turtle	turtle
/m/09dzg	Turtle	turtle
/m/0898b	Zebra	zebra

Appendix B

ARCHITECTURE DETAILS

You can find the architecture details in this section.

	Blocks	**ch-in**	**ch-out**	**Activation**
1	Linear	1024	128	ReLU
2	Linear	128	128	-
3	LayerNorm	128	128	ReLU
4	Linear	128	128	-

Table B.1: **The Details of Edge MLP.**

	Blocks	**ch-in**	**ch-out**	**Activation**
1	Linear	640	128	ReLU
2	Linear	128	128	-
3	LayerNorm	128	128	ReLU
4	Linear	128	512	-

Table B.2: **The Details of Node MLP.**

	Blocks	**ch-in**	**ch-out**	**Activation**
1	Linear	1024	128	ReLU
2	Linear	128	1	Sigmoid

Table B.3: **The Details of Edge Predictor.**

	Blocks	**ch-in**	**ch-out**	**k**	**s**	**p**	**Activation**
1	Conv	256	512	3	1	1	ReLU
2	MaxPool	512	512	2	2	0	-
3	Conv	512	128	3	1	1	ReLU
4	Conv	128	32	3	1	1	ReLU
5	Linear	1568	512	-	-	-	-

Table B.4: **The implementation details of Object Encoder.** k: kernel size, s: stride, p: padding

	Blocks	**ch-in**	**ch-out**	**k**	**s**	**p**	**Activation**
1	Conv	512	256	3	1	1	ReLU
2	Conv	256	256	3	1	1	-

Table B.5: **The implementation details of ResFuser.** k: kernel size, s: stride, p: padding