APANet: Auto-Path Aggregation for Future Instance Segmentation Prediction

Jian-Fang Hu*, Jiangxin Sun*, Zihang Lin, Jianhuang Lai, Wenjun Zeng, and Wei-Shi Zheng

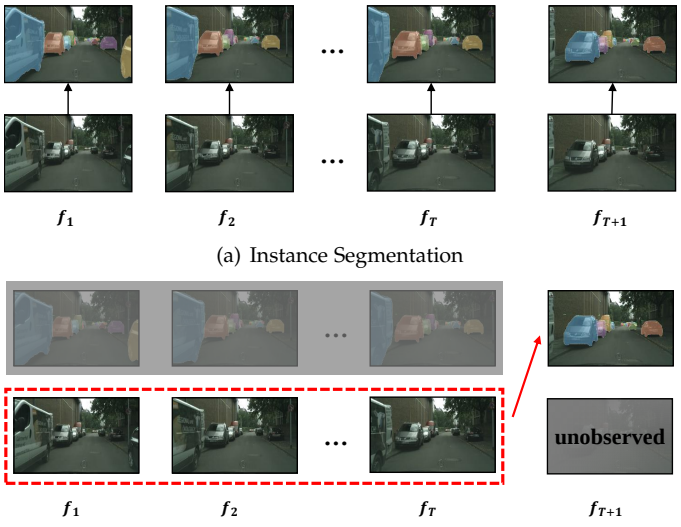
Abstract—Despite the remarkable progress achieved in conventional instance segmentation, the problem of predicting instance segmentation results for unobserved future frames remains challenging due to the unobservability of future data. Existing methods mainly address this challenge by forecasting features of future frames. However, these methods always treat features of multiple levels (e.g. coarse-to-fine pyramid features) independently and do not exploit them collaboratively, which results in inaccurate prediction for future frames; and moreover, such a weakness can partially hinder self-adaption of a future segmentation prediction model for different input samples. To solve this problem, we propose an adaptive aggregation approach called Auto-Path Aggregation Network (APANet), where the spatio-temporal contextual information obtained in the features of each individual level is selectively aggregated using the developed "auto-path". The "auto-path" connects each pair of features extracted at different pyramid levels for task-specific hierarchical contextual information aggregation, which enables selective and adaptive aggregation of pyramid features in accordance with different videos/frames. Our APANet can be further optimized jointly with the Mask R-CNN head as a feature decoder and a Feature Pyramid Network (FPN) feature encoder, forming a joint learning system for future instance segmentation prediction. We experimentally show that the proposed method can achieve state-of-the-art performance on three video-based instance segmentation benchmarks for future instance segmentation prediction.

Index Terms—Future prediction, Future instance segmentation prediction, Instance segmentation, auto-path aggregation.

1 Introduction

Instance segmentation, which requires segmentation of all object instances that appear in given images/videos, has received increasing attention over the past few years. Recently, the deep learning-based approaches have achieved remarkable success in instance segmentation. Most existing methods have been developed for after-the-fact instance segmentation, in which the images/frames to be segmented are accessible to the system. However, in many practical cases, instance segmentation must be performed before the corresponding images/frames are observed; see Figure 1 for details. The problem of predicting instance segmentation results for unobserved future frames, i.e. future instance segmentation prediction, is considerably more important than after-the-fact instance segmentation in certain real-world applications, such as human-machine interaction and autonomous driving etc. For example, in the case of autonomous driving, many accidents could be avoided if the system was able to predict possible collisions with other cars or pedestrians.

Predicting future instance segmentation from observed past frames is very challenging due to the uncertainty associated with the appearance variations caused by object movement, occlusion between objects, and changes of viewpoint. For example, for the video shown in Figure 1, the car that appears in the first T frames has almost disappeared in later frames (e.g., the $(T+1)$-th frame). Existing methods [1], [2] mainly rely on developing models to capture the appearance variations in the spatial and temporal dimensions. More specifically, the work in [2] attempts to forecast



(a) Instance Segmentation

(b) Future Instance Segmentation Prediction

Fig. 1. Video-based instance segmentation vs. future instance segmentation prediction. (a) presents the process of instance segmentation, in which instance segmentation results are generated for the observed frames. (b) shows the process of future instance segmentation prediction, the goal of which is to produce instance segmentation results for unobserved future frames based on the observed past frames.

- *J.-F. Hu, J. Sun, Z. Lin, J. Lai, and W.-S. Zheng are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. J.-F. Hu is also with the Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China. W.-S. Zheng is also with the Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China, and Peng Cheng Laboratory, Shenzhen, China. (Corresponding author: Wei-Shi Zheng) E-mail:hujf5@mail.sysu.edu.cn, {sunjx5, linzh59}@mail2.sysu.edu.cn, stsljh@mail.sysu.edu.cn, and wszheng@ieee.org*
- *W. Zeng is with the Microsoft Research Asia. E-mail:wezeng@microsoft.com*
- *Jian-Fang Hu and Jiangxin Sun are the co-first authors of the article.*

the convolutional pyramid features of future frames based on the pyramid features extracted from the observed past frames. However, in this model, the pyramid features at each individual level are predicted independently without collaboration, which means that the intrinsic relationships among the features at different pyramid levels are ignored.

In fact, collaboratively rather than separately predicting the multi-level pyramid features at different temporal locations helps to identify the intrinsic cues that are effective for the future prediction task. It is probably that features of different levels (such as pyramid features) serve differently at different time steps for the future instance segmentation prediction task. Not all of them are sensitive or effective for predicting the segmentation results of different object instances. This could demand selecting rather than equally making use of all levels of features for the future prediction task. However, this is not a simple selection work, because the object instances to be segmented could vary greatly in different videos and frames, which requires that the selection among the features of different levels could adaptively change for different inputs and different time steps. Unfortunately, previous methods are universal models without multi-level feature selection for predicting the future instance segmentation of different inputs.

In this work, for the future instance segmentation prediction, we propose a novel adaptive aggregation framework called Auto-Path Aggregation Network (APANet), which aims to model the future instance segmentation prediction by collaboratively predicting multi-level pyramid features. Our APANet builds on a set of convolutional long short-term memory (ConvLSTM) units, each of which is employed to capture the intra-level spatio-temporal contexts depicted in the features of a certain pyramid level with different temporal locations. We design auto-path connections between each pair of ConvLSTM units to ensure that the contextual information extracted by the ConvLSTM at a certain level can be aggregated to other ConvLSTMs, thus allowing the inter-level spatio-temporal dependencies to be further modeled to capture cross-level contexts. Our method allows the interaction between intra-level and inter-level spatio-temporal contexts to predict the features of different pyramid levels collaboratively, and thus more task-specific hierarchical contextual information can be embedded into the predicted multi-level pyramid features, which leads to a better system performance on future instance segmentation prediction.

In order to adaptively aggregate the information obtained on each individual ConvLSTM for each video sample, we formulate our auto-path connection in the framework of neural architecture search (NAS) [3], so that the architectures of the auto-path connections can change for different input videos and frames at varied time steps. Therefore, our method has the evolutionary capability to adapt the network architecture to each individual video to better capture the relevant appearance variations. This is beneficial for segmentation prediction because videos with different object instances are expected to contain different kinds of appearance variations, and thus the optimal network architectures should differ for different videos and time steps to better capture these diverse appearance variations.

To efficiently optimize the proposed APANet framework with auto-path architecture search, we make a continuous relaxation on the search space and formulate it as a linear combination of candidate operations, thus allowing the efficient search of the architectures by learning a set of continuous variables. Based on the continuous relaxation, a three-stage optimization approach is proposed to efficiently train the APANet from scratch, with gradient descent. We further illustrate that the proposed APANet can be optimized jointly with the mask region-convolution neural network (Mask R-CNN) head and a Feature Pyramid Network (FPN) feature extractor to form a joint learning system, which can be optimized jointly in order to aggregate more discriminative spatio-temporal pyramid contexts for the future prediction.

We evaluate our method using three benchmark datasets: the Cityscapes Dataset [4], the Inria 3DMovie Dataset v2 [5], and the BDD100K Dataset [6]. Our results demonstrate that collaboratively and adaptively learning the task-specific hierarchical contextual structures among pyramid features with different resolutions using proposed auto-path connections is beneficial for future feature forecasting, and enables our model to substantially outperform the state-of-the-art models for both short-term and mid-term future instance segmentation prediction.

In summary, our main contributions are threefold: 1) a flexible auto-path aggregation network (APANet) for collaboratively predicting multi-level pyramid features, which can selectively and adaptively aggregate the task-specific hierarchical spatio-temporal contextual information obtained on the features of each individual level; 2) a three-stage optimization approach to train the proposed APANet; and 3) a joint learning system that consists of feature extraction, feature prediction and segmentation generation for adaptively predicting future instance segmentation results from observed past frames. We also present an extensive experimental analysis of three benchmark datasets to illustrate the effectiveness of the proposed approach.

2 Related Work

Our work is closely related to context aggregation for segmentation, instance segmentation, video prediction and neural architecture search, which have been extensively investigated in the community. In the following sections, we will provide a brief review of these works.

2.1 Context Aggregation for Segmentation

Context is essential for segmentation because it provides rich information about the objects and their surroundings. Recent studies [7], [8], [9] show that properly aggregating context is of great importance for robust video-based or image-based based segmentation.

For image-based segmentation, explicitly modelling the relations among the representations of different pixels in a single-level feature map can aggregate useful spatial contexts [9], [10], [11], [12], [13]. The spatial context can also be aggregated using convolutions of different scales [7], [14], [15], [16], [17]. Recently, [8], [18], [19], [20], [21], [22], [23],

[24] intend to aggregate the information captured in multi-level features. For example, [8] aggregates information obtained in paths of different network layers by enhancing the feature hierarchy. Among these methods, [24] is the most related to our work, which also intends to automatically select paths for aggregating contexts captured in feature maps with different scales (resolutions). However, in [24], the operations for transmitting information between different feature maps are manually designed without adaptive learning.

Recent studies [25], [26], [27], [28], [29] show that further aggregating temporal context can also improve video-based segmentation. For instance, Fayyaz et al. employ a context module to aggregate the information embedded in the extracted spatio-temporal features [25]. Wang et al. employ ConvLSTM to aggregate contexts from both spatial and temporal directions [26]. Lin et al. formulate a spatio-temporal RNN to aggregate spatio-temporal contexts for video object segmentation [27]. However, the above approaches are developed to aggregate contexts from a single-level feature rather than features of different pyramid levels and temporal locations.

2.2 Instance Segmentation

The goal of instance segmentation is to detect and segment each distinct object of interest. Numerous methods have been proposed to address this problem in recent years [8], [30], [31], [32]. There are two mainstreams in this research field. The first one is built on semantic segmentation. These methods [33], [34], [35], [36], [37], [38], [39] firstly employ a semantic segmentation model to obtain pixel-wise classification results and then cluster the pixels to different object instances. The second one is related to object detection. In this line of works, early methods mainly perform instance segmentation in a two-stage pipeline, i.e., detect then segment [8], [30], [40], [41], [42], [43]. For example, Mask R-CNN [30] added a segmentation branch into the Faster R-CNN [44] for generating instance masks according to the detection results. Another pipeline for the object detection related approaches is to develop a one-stage instance segmentation system. These methods simultaneously detect object locations and predict mask representations [32], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54]. In addition to the segmentation-based and detection-related approaches, some methods intend to address instance segmentation by classifying mask proposals [55] or developing dense sliding window methods [31], [56], [57], [58]. These works are specially developed for segmenting instances from observed images. In this work, we focus on predicting future instance segmentation for future unobserved images/frames.

2.3 Video Prediction

Predicting future information is very important in many real-world applications. Most existing approaches [59], [60], [61], [62], [63], [64], [65] intend to generate future RGB frames based on deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). For example, Mathieu et al. propose a CNN model for predicting one or several future frames from concatenated multi-scale input frames [59]. Oh et al. apply encoding-transformation-decoding network architectures constructed from CNNs and RNNs to directly generate the RGB values of pixels in future video frames [63]. Wang et al. derive a spatio-temporal LSTM (ST-LSTM) unit for the prediction of future frames [64]. Chen et al. propose an object-centric video prediction model that learns local motion transformations for key objects to improve the performance of RGB frame prediction [65]. These approaches mainly predict the values of the pixels in each video frame.

The prediction of more abstract representations, such as object trajectories and human actions, has been investigated in the community as well. For instance, Morris et al. propose the use of Gaussian mixture modeling, hidden Markov models and maximum likelihood regression for trajectory learning and activity understanding in live videos [66]. Bhattacharyya et al. propose a new model to jointly predict future ego-motion and person trajectories over long-term on-board horizons [67]. Shi et al. use a radial basis function (RBF) kernelized feature mapping RNN to predict future human actions [68]. Vondrick et al. attempt to anticipate the visual features of future video frames to predict actions in observed partial videos [69]. Xie et al. propose an intelligent agent-based method of localizing objects for predicting human intentions and trajectories in surveillance videos [70]. These approaches are developed for the prediction of high-level abstract representations of future frames rather than pixel-wise segmentation.

The prediction of future semantic segmentation and future instance segmentation has also gained increasing attention recently [1], [2], [71], [72], [73]. Most methods perform future segmentation prediction in videos by predicting features for future unobserved frames from the observed past frames. The feature prediction is explicitly modeled in the framework of CNN [1], [2], [72], [73] or ConvLSTM [71]. For instance, [2] proposes a framework containing four resolution-preserving CNN sub-networks. It is worth noting that F2F [2] and [71] intend to predict pyramid features at each individual level independently. In contrast, our approach aims to predict the pyramid features collaboratively by uniquely exploring the interaction among the features of different pyramid levels and temporal locations. More importantly, in our approach, more task-specific hierarchical contextual information can be embedded into the predicted multi-level pyramid features, which leads to better performance for future instance segmentation prediction.

2.4 Neural Architecture Search

Recent studies show that automatically learning network architectures can obtain more powerful image/video representations than manually designed architectures, and thus lead to a better performance in practice [74], [75], [76], [77]. Early researches intend to directly search architecture for the entire network [78], [79], [80], [81], [82], [83], [84], which is quite time-consuming and space-consuming. Recent studies show that designing the network as a stack of repeated cells and searching architectures for the cells can reduce the search cost [74], [85], [86]. Most existing approaches mainly search the architecture in a pre-defined discrete

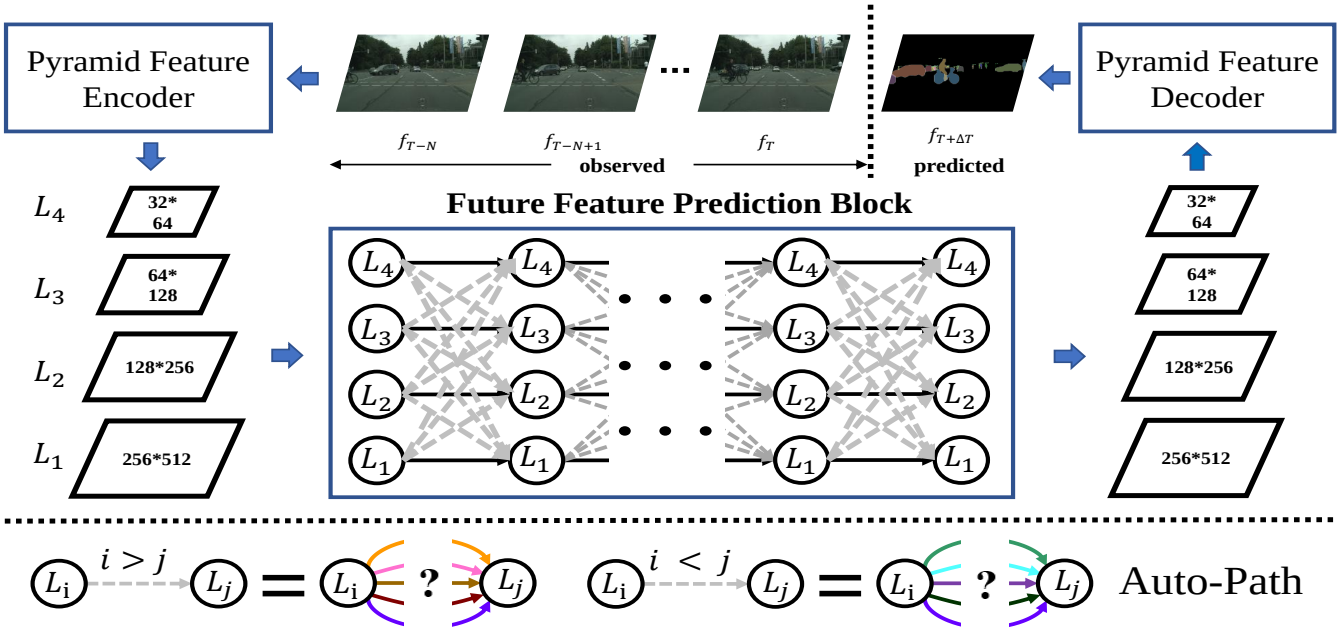


Fig. 2. An overview of the proposed framework for future instance segmentation prediction. Our framework consists of three blocks: a pyramid feature encoder, a future feature prediction block, and a pyramid feature decoder. We use gray dash lines to indicate the auto-path connections. The operations employed in each auto-path connection will be automatically determined by neural architecture search (NAS) [3]. The sets of possible operations for the auto-path connections in top-down and bottom-up directions are different, as shown at the bottom of the figure. Different candidate operations are indicated by different colors.

space using different search strategies, including random search [76], [87], bayesian optimization [88], [89], [90], reinforcement learning [74], [85], [86], evolutionary algorithms [75], [91], [92], etc. The researches in [3], [77], [93] show that the network architecture can be efficiently obtained with a gradient-descent based approach. In these methods, the discrete space is mathematically modeled with binary variables, which can be relaxed to continuous ones. Our work follows the gradient-descent based neural architecture search (NAS) formulation and develops a video-adaptive network, in order to adaptively change the network architecture to capture the complex appearance variations in each individual video.

A preliminary version of the current work was reported in [94], in which the paths for the information aggregation were manually designed. In this work, we have significantly extended our framework in the following three aspects. Firstly, a more advanced future instance segmentation prediction method is formulated by using the NAS mechanism to automatically design the network architecture for information aggregation, which enables the network to 1) selectively aggregate the information obtained on each ConvLSTM, and 2) automatically adapt its architecture to each individual video to better capture the complex appearance variations contained in different videos. Secondly, we have reported a more extensive comparative analysis of our approach to show the additional benefits of automatically determining the architectures of auto-path connections via the NAS mechanism. Thirdly, we have conducted further experiments and reported better experimental results on the Cityscapes and Inria 3DMovie datasets, where the performance improvements are greater than 1.8% AP50 for short-term prediction and 3.6% AP50 for mid-term prediction on the Cityscapes set. We also report extensive comparisons for one additional dataset (BDD100K).

3 Our Approach

In this work, we address the future instance segmentation prediction problem by collaboratively predicting the pyramid features of future unobserved frames. More specifically, we formulate a novel network called Auto-Path Aggregation Network (APANet) to selectively and adaptively aggregate the spatio-temporal information embedded in the pyramid features of varied temporal locations. Our APANet is quite flexible and can automatically change the network architecture to adapt to different input videos. A graphic illustration of our system is presented in Figure 2. The proposed APANet is modeled as a set of ConvLSTMs densely connected by auto-path connections. Each ConvLSTM predicts the features of a particular pyramid level. The auto-path connections among each pair of ConvLSTMs are employed to exploit the inter-level spatio-temporal contexts. The operations employed in each auto-path connection will be automatically determined in a data-driven manner, in order to selectively and adaptively aggregate hierarchical spatio-temporal contextual information.

Before delving deeper into the proposed framework, we first briefly describe the problem of future feature prediction. Then, we introduce our framework for multi-level feature prediction with auto-path aggregation. With the proposed APANet, we ultimately propose a joint learning system for the segmentation prediction, which consists of three parts: the Feature Pyramid Network (FPN) [21] as a pyramid feature encoder to represent the observed video frames as pyramid features, a future feature prediction block to predict pyramid features for future frames (i.e. our

APANet), and the Mask R-CNN head [30] as a pyramid feature decoder to generate segmentation results.

3.1 Problem Statement

The goal of future pyramid feature prediction is to learn a mapping $\mathbf{\Theta}$ between the convolutional pyramid features extracted from observed past video frames and the features extracted from unobserved future frames. More specifically, the future feature prediction can be formulated as

$$\mathcal{F}_{T+\triangle T} = \mathbf{\Theta}(\mathcal{F}_{T-N}, \mathcal{F}_{T-N+1}, ..., \mathcal{F}_{T}), \tag{1}$$

where the inputs to the mapping $\mathbf{\Theta}$ are the multi-level pyramid features $\{\mathcal{F}_{T-N}, \mathcal{F}_{T-N+1}, ..., \mathcal{F}_{T}\}$ extracted from the observed past $N+1$ frames. Here, $\mathcal{F}_t$ denotes the multi-level pyramid features $\{\mathbf{P}_t^1, \mathbf{P}_t^2, ..., \mathbf{P}_t^L\}$ for the t-th frame, which has a total of L pyramid levels. $\mathbf{P}_t^l$ is the feature of the t-th frame at the l-th pyramid level. The features at different pyramid levels describe different aspects of an observed frame with various resolutions and receptive fields. In general, the features of higher pyramid levels have lower resolution and larger receptive fields. The output of $\mathbf{\Theta}$ consists of the pyramid features $\mathcal{F}_{T+\triangle T}$ predicted for the future frame.

Notably, the existing method F2F [2] intends to decompose mapping $\mathbf{\Theta}$ into L independent sub-mappings $\{\mathbf{\Theta}_l\}_{l=1,2,...,L}$, each of which corresponds to the prediction of features at a certain level. This means that F2F has to train a total of L prediction networks independently and does not make use of them collaboratively, which results in inaccurate feature prediction. Moreover, the ignored feature collaboration further partially hinders the self-adaption of F2F for segmenting instances from different inputs. In the following, we propose an adaptive approach to collaboratively predict multi-level pyramid features, which could explicitly exploit the task-specific hierarchical contextual structures among the features of different pyramid levels and time steps for prediction.

3.2 Model Architecture

We describe our method for predicting future pyramid features $\mathcal{F}_{T+\triangle T}$, which adaptively aggregates spatio-temporal contextual information among the features of different pyramid levels and time steps, $\{\mathcal{F}_{T-N}, \mathcal{F}_{T-N+1}, ..., \mathcal{F}_{T}\}$. The spatio-temporal information is adaptively aggregated along the auto-path across different pyramid levels and time steps, which is automatically determined using NAS in a data-driven manner. Thus, we name our network the Auto-Path Aggregation Network (APANet).

To model the spatio-temporal dependencies among the features of each level, we formulate our APANet based on the ConvLSTM [95], which has shown good performance in capturing spatio-temporal contexts among different temporal features in recent studies [95], [96]. For each individual pyramid level, we employ a ConvLSTM to capture the intra-level spatio-temporal information among the features.

In addition to the intra-level spatio-temporal pyramid contexts, we also explore the inter-level spatio-temporal pyramid information among the ConvLSTMs of varied pyramid levels, by selectively connecting the cells (hidden states) of different ConvLSTMs. In this way, the L connected ConvLSTMs form our mapping $\mathbf{\Theta}$ for collaboratively predicting future pyramid features.

Now, we combine the modeling of both intra-level and inter-level spatio-temporal pyramid contexts by the following proposed "auto-path" connection. As shown in the illustration of one cell in our framework (the left of Figure 3), the hidden state $\mathbf{C}_t^l$ of the l-th ConvLSTM at time step t can selectively aggregate the contextual information embedded in the hidden states $\{\mathbf{C}_{t-1}^v\}_{v=1,2,...,L}$ of the ConvLSTMs at the previous time step. More specifically, the information aggregated at $\mathbf{C}_t^l$ can be formulated as

$$APA(\{\mathbf{C}_{t-1}^v\}_{v=1,2,...,L}, l) = \mathbf{W}_{APA}^l * \sum_{v=1}^{L} AP(\mathbf{C}_{t-1}^v, l), \tag{2}$$

where $*$ is the convolution operation and $AP(\mathbf{C}_{t-1}^v, l)$ encodes the spatio-temporal pyramid contextual information propagated from hidden state $\mathbf{C}_{t-1}^v$ to $\mathbf{C}_t^l$, which is referred to as an *auto-path connection*. $\mathbf{W}_{APA}^l$ is the convolution kernel used to control the information aggregation. The auto-path connections are employed to adaptively aggregate spatio-temporal contextual information obtained on each ConvLSTM, which is expressed as

$$AP(\mathbf{C}_{t-1}^v, l) = \begin{cases} \mathbf{C}_{t-1}^v & \text{if } v = l, \\ \sum_{\psi \in \mathbf{\Psi}_{v \to l}} \alpha_{v \to l}^{t,\psi} \psi(\mathbf{C}_{t-1}^v) & \text{if } v \neq l, \end{cases} \tag{3}$$

where ψ is an operation used to propagate information from the hidden state $\mathbf{C}_{t-1}^v$ to the hidden state of the l-th ConvLSTM. The parameter $\alpha_{v \to l}^{t,\psi}$ is employed to control the information propagated by each operation ψ. It is defined as a binary (0-1) variable under constraint $\alpha_{v \to l}^{t,\psi}$: $\sum_{\psi \in \mathbf{\Psi}_{v \to l}} \alpha_{v \to l}^{t,\psi} = 1$. When the value of $\alpha_{v \to l}^{t,\psi}$ is 0, the corresponding operation ψ is not active for information aggregation, i.e., the information from the v-th ConvLSTM propagated by ψ is not aggregated to the l-th ConvLSTM at the t-th time step. Otherwise, the operation ψ would be active if $\alpha_{v \to l}^{t,\psi}$ is 1.

The auto-path connection is defined in such a way that both the intra-level and inter-level spatio-temporal pyramid information can be exploited for multi-level pyramid feature prediction. The combination of intra-level and inter-level spatio-temporal pyramid contexts allows more task-specific hierarchical information to be embedded in the predicted pyramid features. In general, the auto-path connections propagate intra-level and inter-level spatio-temporal pyramid information when $v = l$ and $v \neq l$, respectively. For the connection between hidden states of the same level (i.e., $v = l$), we define it as the information transmission operation widely used in ConvLSTM, whose effectiveness for capturing intra-level spatio-temporal contextual information has been demonstrated in the literature [95]. For the connection between hidden states of different levels (i.e., $v \neq l$), we formulate our auto-path connection (i.e., Eq. (3)) in the mechanism of neural architecture search (NAS) and attention, so that it can adaptively and selectively aggregate multi-level contextual information from previous time steps.

Auto-path Architecture Search. Considering that the object instances to be segmented could vary greatly in different videos and frames, we seek to formulate our approach as a video-adaptive network. To this end, we refer to some ideas in NAS [3], [77] and treat $\alpha_{v \to l}^{t,\psi}$ as the architecture

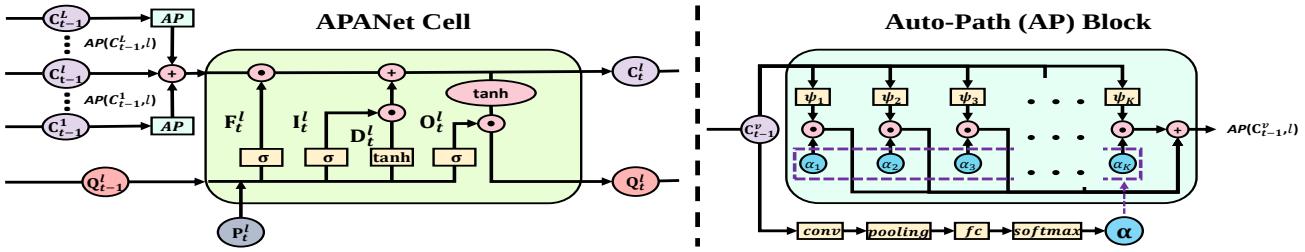


Fig. 3. Our proposed auto-path aggregation network (APANet) with one cell. The left of the figure shows a cell in our APANet and the right of the figure represents the architecture of our auto-path connections (i.e. the AP block in the APANet cell). $\mathbf{P}_t^l$, $\mathbf{Q}_t^l$, and $\mathbf{C}_t^l$ denote the input feature, output feature, and hidden state, respectively, for the l-th ConvLSTM at time step t. AP denotes an auto-path connection between two different ConvLSTM cells (see Eq. (3)). $\odot$ denotes element-wise multiplication, and $+$ is the addition operation. The parameter $\boldsymbol{\alpha}$ is determined according to the input hidden state, which enables our method to adaptively adjust the architecture for aggregation in accordance with the observed videos/frames.

TABLE 1
Candidates of "Operation X" for different pathways.

Type	Candidates	
Top-down Pathways	3×3 deconvolution	5×5 deconvolution
	3×3 depthwise-separable deconvolution	5×5 depthwise-separable deconvolution
	3×3 atrous deconvolution with rate 2	5×5 atrous deconvolution with rate 2
	bilinear interpolation	no connection (zero)
Bottom-up Pathways	3×3 convolution	5×5 convolution
	3×3 depthwise-separable convolution	5×5 depthwise-separable convolution
	3×3 atrous convolution with rate 2	5×5 atrous convolution with rate 2
	pooling	no connection (zero)

parameter to be searched, which would be determined with gradient descent algorithm [3]. A graphic illustration of it is presented in Figure 2. Our approach would automatically learn a proper operation from a set of possible operations for each auto-path connection.

The set of possible operations $\boldsymbol{\Psi}_{v\to l}$ differs for different relations between v and l. Specifically, we define the operation set $\boldsymbol{\Psi}_{v\to l}$ in the form of "Operation X" and "Operation X + Attention". For the case of $v > l$, i.e., auto-path connections along the top-down direction, which needs to transmit information from lower-resolution feature map to higher-resolution feature map. For the case of $v < l$, i.e., the auto-path connections in the bottom-up direction, which needs to transmit information from higher-resolution feature map to lower-resolution feature map.

The candidates of "Operation X" for the top-down and bottom-up pathways are presented in Table 1. The operations are employed following the implementations of [3], [74], [76], [77], [85], and all of them are widely used in the modern CNN architectures to transmit information between features of varied resolutions. Our approach could search operations with different convolution kernel sizes (3×3 or 5×5) and types (i.e. traditional convolution, atrous convolution and depthwise-separable convolution). $\boldsymbol{\Psi}_{v\to l}$ is defined differently for different cases (top-down and bottom-up) because the hidden states at different pyramid levels have different resolutions. In general, the higher the pyramid level is, the lower the feature resolution (see Figure 2 for details) is contained. Thus, upsampling or downsampling operations are required for scaling the information obtained in the ConvLSTMs at different pyramid levels to allow the information to be directly aggregated. Specifically, for the case of $v < l$ (bottom-up aggregation), downsampling operations such as convolution and max pooling are employed to scale the input hidden state, while for the case of $v > l$ (top-down aggregation), upsampling operations such as deconvolution and bilinear interpolation are used. The operation "no connection (zero)" means that the corresponding path for information propagation is blocked. Since the "no connection (zero)" operation with attention and without attention are the same in practice, the total number of candidate operations for each auto-path connection is $K = 15$.

The advantages of introducing NAS into our framework are twofold. Firstly, the NAS mechanism enables our approach to learn a suitable combination of the operations for aggregating spatio-temporal pyramid information in a data-driven manner. Secondly, it introduces additional flexibility into our prediction system and thus allows the system to more easily adapt to newly observed data.

Attention mechanism. We employ an attention map in the computation of some information propagation operations (i.e., $\psi(\mathbf{C}_{t-1}^v)$ in Eq. (3)), such as "3×3 Deconvolution + Attention" and "3×3 Convolution + Attention", in order to inhibit the information that is irrelevant to instances. Specifically, the involved attention map $\mathbf{A}^{v\to l}$ is computed from $\mathbf{C}_{t-1}^v$ by applying a self-attention mechanism, which can be formulated as

$$\begin{aligned} \mathbf{C}^{v\to l} &= \mathbf{A}^{v\to l} \odot \bar{\mathbf{C}}_{t-1}^v, \\ \mathbf{A}^{v\to l} &= \sigma(\mathbf{W}^{v\to l} * \bar{\mathbf{C}}_{t-1}^v + \mathbf{B}^{v\to l}), \end{aligned} \tag{4}$$

where $\odot$ means element-wise product. $\bar{\mathbf{C}}_{t-1}^v$ is obtained by applying the corresponding "Operation X" to the hidden state $\mathbf{C}_{t-1}^v$. $\mathbf{W}^{v\to l}$ is a parameter used to encode the information propagated from ConvLSTM-v to ConvLSTM-l and $\mathbf{B}^{v\to l}$ is a bias term, which will be learned in the training stage. $\mathbf{C}^{v\to l}$ means the selective information prop-

agated from the v-th level to the l-th level by $\psi(\mathbf{C}_{t-1}^{v})$. The employed attention mechanism allows our method to selectively aggregate the information that is closely related to the instances of interest. The learned attention maps are diverse for different samples and auto-path connections, which will be discussed in detail in Section4.2.1.

Overall, the information propagation in the t-th ConvLSTM cell for the features at the l-th pyramid level in our APANet can be formulated as follows:

$$\begin{aligned}\mathbf{C}_t^l = \mathbf{F}_t^l \odot (APA(\{\mathbf{C}_{t-1}^v\}_{v=1,2,...,L}, l)) + \mathbf{I}_t^l \odot \mathbf{D}_t^l,\\ \mathbf{Q}_t^l = \mathbf{O}_t^l \odot tanh(\mathbf{C}_t^l),\end{aligned} \tag{5}$$

where $\mathbf{Q}_t^l$ is the model output (i.e., predicted feature) for the l-th level at the t-th time step. $\mathbf{I}_t^l$, $\mathbf{F}_t^l$, $\mathbf{O}_t^l$ and $\mathbf{D}_t^l$ are the input gates, forget gates, output gates, and the information propagated from the inputs for the ConvLSTM at the t-th time step, respectively. They are formulated as follows:

$$\begin{aligned}\mathbf{I}_t^l &= \sigma(\mathbf{W}_i^l * \mathbf{P}_t^l + \mathbf{H}_i^l * \mathbf{Q}_{t-1}^l + \mathbf{B}_i^l),\\ \mathbf{F}_t^l &= \sigma(\mathbf{W}_f^l * \mathbf{P}_t^l + \mathbf{H}_f^l * \mathbf{Q}_{t-1}^l + \mathbf{B}_f^l),\\ \mathbf{O}_t^l &= \sigma(\mathbf{W}_o^l * \mathbf{P}_t^l + \mathbf{H}_o^l * \mathbf{Q}_{t-1}^l + \mathbf{B}_o^l),\\ \mathbf{D}_t^l &= tanh(\mathbf{W}_{\mathbf{D}}^l * \mathbf{P}_t^l + \mathbf{H}_{\mathbf{D}}^l * \mathbf{Q}_{t-1}^l + \mathbf{B}_d^l),\end{aligned} \tag{6}$$

where $*$ denotes convolution operation. $\mathbf{W}_\bullet^l \in R^{k\times k\times c}$ and $\mathbf{H}_\bullet^l \in R^{k\times k\times c}$ are convolution kernels that control the propagation of information along the input-state direction and output-state direction, respectively. Here, c is the number of channels, and k represents the size of the kernel. The kernel size k is a hyperparameter that needs to be tuned in practice. In general, more local spatial contextual information can be taken into account by using a larger kernel. $\mathbf{B}_\bullet^l$ is the corresponding bias term. $\mathbf{W}_\bullet^l$, $\mathbf{H}_\bullet^l$, and $\mathbf{B}_\bullet^l$ are model parameters to be learned in the training stage. All these parameters are shared across different time steps to form a recurrent architecture. σ is the sigmoid operation. Note that the resolution of output $\mathbf{Q}_t^l$ is equal to the resolution of the corresponding input feature $\mathbf{P}_t^l$, which means that the proposed APANet is a resolution-preserving mapping.

3.3 Model Learning

Objective function. Our objective in future feature prediction is to learn both the model parameters and the architecture parameters such that the gap between the predicted features and the ground-truth features is minimized. To this end, we minimize the following prediction loss:

$$L_p = \frac{1}{T}\sum_{t=1}^{T}\sum_{l=1}^{L}\frac{1}{n_l}\left\|\mathbf{Q}_t^l - \mathbf{P}_{t+1}^l\right\|_F^2, \tag{7}$$

where n_l is the number of elements in $\mathbf{Q}_t^l$, which is the set of convolutional pyramid features predicted for a future frame (the $(t+1)$-th frame) using our APANet from the observed past t frames, and $\mathbf{P}_{t+1}^l$ is the corresponding set of features extracted from the ground-truth ($(t+1)$-th) video frame.

Continuous relaxation. Directly optimizing the proposed APANet with a discrete NAS process is difficult. Most of the existing methods mainly solve the NAS process by reinforcement learning [74], [82], [85] or evolutionary algorithms [75], [83], [91], which is computationally intensive in practice. Here, we follow the main idea in DARTS [3] and propose to solve the whole framework by an efficient gradient-descent approach. Specifically, we introduce a continuous relaxation of the discrete architecture space and then discuss how to perform optimization via gradient descent. Following [3], we relax the binary constraint $\sum_{\psi\in\mathbf{\Psi}_{v\to l}}\alpha_{v\to l}^{t,\psi} = 1, \alpha_{v\to l}^{t,\psi} \in \{0,1\}$, as follows: $\sum_{\psi\in\mathbf{\Psi}_{v\to l}}\alpha_{v\to l}^{t,\psi} = 1, 0 \le \alpha_{v\to l}^{t,\psi} \le 1$, which can be easily implemented by the softmax function. The main advantage of relaxing the constraint in this way is that the architecture search space is now differentiable and can be embedded into a differentiable computation graph for efficient optimization. Specifically, the α can be computed from the hidden state of the corresponding ConvLSTM by a stack of a convolutional layer, a global pooling layer, a fully connected layer and a softmax layer, as illustrated in the right of Figure 3. We define α in such a way that it is a sample-dependent parameter. Thus, our network can automatically adapt its architecture to each individual video to better capture the relevant appearance variations contained in different inputs.

Three-stage optimization. Even with the continuous relaxation, training the proposed APANet with L densely connected ConvLSTMs is not easy. Here, we employ a three-stage optimization approach to solve this problem. In the first stage (see Figure4 (a)), we mainly pre-train the parameters of each ConvLSTM without considering the auto-path connections among each pair of ConvLSTMs. In the second stage (see Figure4 (b)), we turn on the influence of auto-path connections and train the entire network with auto-path connections using DARTS [3]. Here, each individual ConvLSTM is initialized with the parameters determined in the first step. In the third stage (see Figure4 (c)), we perform network architecture decoding. Specifically, we decode the dense network architecture learned in the second stage and finetune the network for future pyramid feature prediction.

ConvLSTM pre-train (first stage). In the first stage, we turn off the influence of the inter-level spatio-temporal pyramid contexts by blocking all corresponding auto-path connections and training each ConvLSTM independently, which means that only the intra-level spatio-temporal contexts are considered for the information propagation in this stage.

Auto-path search process (second stage). In the second stage, we adopt the first-order approximation presented in [3] to search a suitable dense network architecture for multi-level feature prediction. Specifically, we divide the training data into two disjoint sets trainA and trainB, then apply the following bilevel optimization procedure:

1) Update the network parameters ω_n based on set trainA, denoted by $\bigtriangledown_{\omega_n} L_{p,trainA}(\omega_n, \omega_\alpha)$.
2) Update the architecture parameters ω_α based on set trainB, denoted by $\bigtriangledown_{\omega_\alpha} L_{p,trainB}(\omega_n, \omega_\alpha)$.

Here, the architecture parameters ω_α and network parameters ω_n refer to $\{\boldsymbol{\alpha}_{v\to l}^{t,\psi}\}$ and $\{\mathbf{W}_\bullet^l, \mathbf{H}_\bullet^l, \mathbf{B}_\bullet^l, \mathbf{W}^{v\to l}, \mathbf{B}^{v\to l}, \mathbf{W}_{APA}^l\}$, respectively. Once the convergence is reached after several iterations, we can obtain a dense network architecture, in which all the candidate operations $\mathbf{\Psi}_{v\to l}$ in each auto-path connection are associated with a set of learned weights.

Network architecture decoding (third stage). Following [3], we decode the dense network architecture by choosing

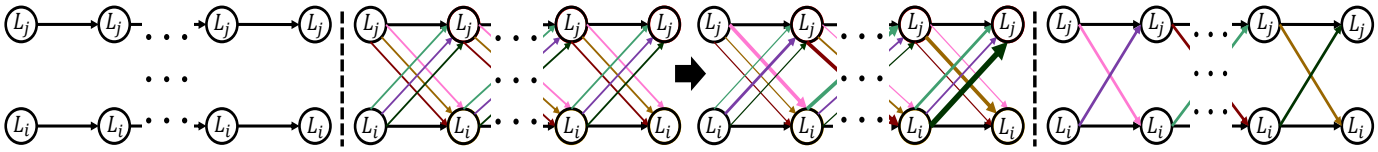


Fig. 4. A graphic illustration of the employed 3-stage training strategy. In this figure, different operations are marked with different colors. For better illustration, we only present the aggregation of 2-level features. Best viewed in color.

the most likely operations, i.e., the operations with the maximum connection strengths. The decoded architecture is determined as follows:

$$\psi^{\star}(v,l,t) = \text{argmax}_{\psi}\ \boldsymbol{\alpha}_{v\rightarrow l}^{t,\psi}. \tag{8}$$

After this step, we can obtain a sparse network architecture that specifies the auto-path connections for the various ConvLSTMs and time steps (v,l,t). Then, we finetune the obtained sparse network on the training set to obtain an improved model for future feature prediction. Note that the architecture parameters are determined by the hidden states of the corresponding ConvLSTMs, which means that the architectures of the auto-path connections can differ for different video samples and time steps. Thus, our APANet framework is quite flexible, and the architecture can be adapted to suit the needs of different input videos.

3.4 Joint learning system for future instance segmentation prediction

Our APANet collaboratively predicts pyramid features for unobserved future frames, which are then fed into the Mask R-CNN head [30] to generate instance segmentation results for future unobserved frames. Accordingly, we implement a joint learning framework consisting of our APANet, a feature encoding block (FPN [21]) and a feature decoding block (the Mask R-CNN head [30]) and train the whole system in the joint learning manner. In this implementation, the parameters for future feature prediction are pre-trained using the three-stage optimization method described above. We minimize the following loss:

$$L = L_p + \lambda L_{MaskR-CNN}, \tag{9}$$

where L_p is the feature prediction loss defined in Eq. 9 and $L_{MaskR-CNN}$ consists of a classification loss, a bounding box regression loss and a segmentation loss as defined in Mask R-CNN [30]. λ is a parameter used to control the balance between the losses for feature prediction and instance segmentation prediction. Its influence will be studied in Section 4.3.7.

Overall, our system for future instance segmentation prediction consists of three parts: the Feature Pyramid Network (FPN) [21] as a feature encoder to represent the observed video frames as pyramid features, a future feature prediction block to predict pyramid features for future frames (i.e. our APANet), and the Mask R-CNN head [30] as a feature decoder to generate segmentation results. Our proposed adaptive aggregation framework called Auto-Path Aggregation Network (APANet) predicts the multi-level pyramid features collaboratively by explicitly exploiting the hierarchical contextual interactions among the features of different resolutions and time steps, which enables our system to adaptively aggregate pyramid features in accordance with different videos/frames.

4 Experiments

We evaluate our method on three video-based instance segmentation datasets: the Cityscapes dataset [4], Inria 3DMovie Dataset v2 [5] and BDD100K dataset [6].

4.1 Experimental Settings

Evaluation Metrics. We use the metrics AP50 and AP defined in [4] to measure the performance of instance segmentation prediction. For the AP50 metric, segmentation for a given instance is considered correct if it has an intersection over union (IoU) of at least 50% with the corresponding ground-truth instance. The AP metric is defined as the mean of the average precision values obtained for ten equally spaced IoU thresholds from 50% to 95%. On the Cityscapes and BDD100K datasets, the performance is measured across the eight object classes with available ground-truth annotations: person, rider, car, truck, bus, train, motorcycle, and bicycle. On the Inria 3DMovie Dataset v2, the performance is measured only on the person class. Following the implementation in [2], each video is temporally subsampled by a factor of three, and the clips with four frames $\{X_{t-9}, X_{t-6}, X_{t-3}, X_t\}$ are employed as the input to our model. Both **short-term** and **mid-term** predictions are considered in our experiments, where the instance segmentation for the future frames X_{t+3} (approximately **0.17** seconds later) and X_{t+9} (approximately **0.5** seconds later) are predicted, respectively.

Implementation Details. Following [2], we employ the Mask R-CNN model with a ResNet-50-FPN backbone, which has been pre-trained on the MS-COCO dataset [97] and then fine-tuned on the corresponding training sets, for the extraction of pyramid features. The pyramid features extracted from the observed past frames are then fed into our APANet to predict corresponding future features, which are subsequently processed by the Mask R-CNN head to generate instance segmentation results. Four levels of pyramid features are obtained from the FPN feature extractor, with resolutions of 256×512 (L_1), 128×256 (L_2), 64×128 (L_3) and 32×64 (L_4). These features correspond to P_2, P_3, P_4, and P_5, respectively, in the FPN. For the first stage of the training process, in which the auto-path connections are blocked, the ConvLSTM for each pyramid level is trained separately using the stochastic gradient descent (SGD) algorithm with a Nesterov momentum of 0.9. The batch size is set to 4.

TABLE 2
Comparison results on the Cityscapes validation set.

	Short-term		Mid-term	
	AP50	AP	AP50	AP
Mask R-CNN oracle	65.8	37.3	65.8	37.3
Copy-last segmentation	24.1	10.1	6.6	1.8
Optical flow - shift [2]	37.0	16.0	9.7	2.9
Optical flow - warp [2]	36.8	16.5	11.1	4.1
Mask H2F [2]	25.5	11.8	14.2	5.1
F2F [2]	39.9	19.4	19.4	7.7
Ours	**46.1**	**23.2**	**29.2**	**12.9**

The learning rate is initialized at 0.01 and is decreased to 0.001. For the second stage of training, we randomly select half of the training samples to form trainA and used the rest as trainB. Then we perform bilevel optimization as previously discussed. In each iteration, we load a batch of samples from set trainA to optimize ω_n and then load a batch of samples from trainB to optimize ω_α. The learning rate is set to 0.001 for ω_n and 0.01 for ω_α. For the third stage of training, the learning rate of 0.001 is used, and the weight decay is set to 0.0005. For the joint training of the whole system, different learning rates are employed for different blocks. APANet is trained with a learning rate of 0.001, while the segmentation blocks (i.e., the FPN feature extractor and Mask R-CNN head) are trained with a lower learning rate (0.0001). The size of the convolution kernel is 3×3. We apply depthwise-separable convolutions on the auto-path. The weight λ for controlling the balance between the future prediction loss and Mask R-CNN loss is set to 0.1. Its influence is investigated in Section 4.3.7. It took about 10 days to train our system on the Cityscapes dataset using 4 GPUs.

4.2 Comparison with previous state-of-the-art

Here, we compare the results of our method with those of other state-of-the-art approaches on the Cityscapes dataset, the Inria 3DMovie Dataset v2 and the BDD100K dataset.

4.2.1 Results on the Cityscapes Dataset

We use the Cityscapes dataset [4] to evaluate our approach. It is a large-scale dataset for video-based instance segmentation containing 2,975 training videos, 500 validation samples and 1,525 test sequences. All sequences in this set were recorded on urban streets in 50 different cities. Each sequence in this set consists of 30 image frames with a resolution of 1024×2048. Ground-truth segmentation annotations are provided for the 20-th frame in each video.

Compared Methods. In this experiment, we compare our method with the following existing models: optical flow - shift, optical flow - warp, Mask H2F and F2F [2]. Both the optical flow - shift and optical flow - warp models use optical flow for future instance segmentation prediction. In the shift approach, each mask is shifted by the average flow vector computed across the mask. In the warp approach, the mask of each instance is independently warped using the flow field inside the mask. Mask H2F is a variant of Mask R-CNN that accepts four successive RGB frames as input and generates instance segmentation results for the future frame. F2F is a CNN-based feature prediction network that achieved state-of-the-art performance on this dataset. In addition to the results of these methods, we also report the accuracy of the Mask R-CNN oracle, which corresponds to simply feeding the ground-truth future RGB frames into Mask R-CNN. This accuracy can be regarded as an upper bound for the performance of our system. We also report the accuracy achieved by directly using the pyramid features extracted from the last observed frame (denoted by Copy-last segmentation), which serves as a lower bound for the performance of our approach.

Comparison results. The detailed comparison results are presented in Table 2. As shown, our system achieves the best performance for both short-term and mid-term instance segmentation prediction. Specifically, for short-term prediction, our model achieves a performance of 46.1% in terms of AP50, representing a clear improvement by a large margin of 6.2% over the state-of-the-art model F2F [2]. With respect to the AP metric, our model also clearly outperforms the F2F model, by a margin of 3.8%. The performance improvement demonstrates that our approach performs better than all the competitors in capturing the spatio-temporal contextual information contained in the observed frames for future instance segmentation prediction. Regarding the mid-term prediction results, similar observations can be obtained: our approach again has a clear advantage over its competitors, with performance results of 29.2% and 12.9% in terms of AP50 and AP, respectively. The results demonstrate that collaboratively predicting multi-level pyramid features with the proposed APANet benefits clearly for future instance segmentation prediction. We can also observe that the improvement of our system over F2F is much larger for mid-term prediction than that for short-term prediction. We attribute this to the ability of our model to capture cross-level spatio-temporal contexts, which plays a more important role in predicting segmentation with longer time steps.

Visualization results for the learned auto-path. In our APANet model, we have employed auto-path connections to explicitly aggregate information among the temporal features of different pyramid levels. The architectures of auto-path connections are learned adaptively according to the observed videos and frames in the neural architecture search framework. By exactly examining the learned architecture structures in Figure 5, we can conclude two interesting observations. Firstly, the auto-path connections learned for the same video differ slightly at different time steps. This is as expected, as the dependencies among consecutive video frames could slightly change over time due to the limited variety of object motion, illumination, visual angle, etc. Secondly, the learned architecture could be vastly different for different video samples. For the video with large objects such as cars, our network tends to use more top-down auto-path connections for the context aggregation. While for the video containing many small, densely distributed and deformed objects like pedestrians, our network is more likely to select bottom-up auto-path connections for aggregation. The reason could be that the higher-level features with lower resolution contain more abstract information about the image, which is quite useful for large object

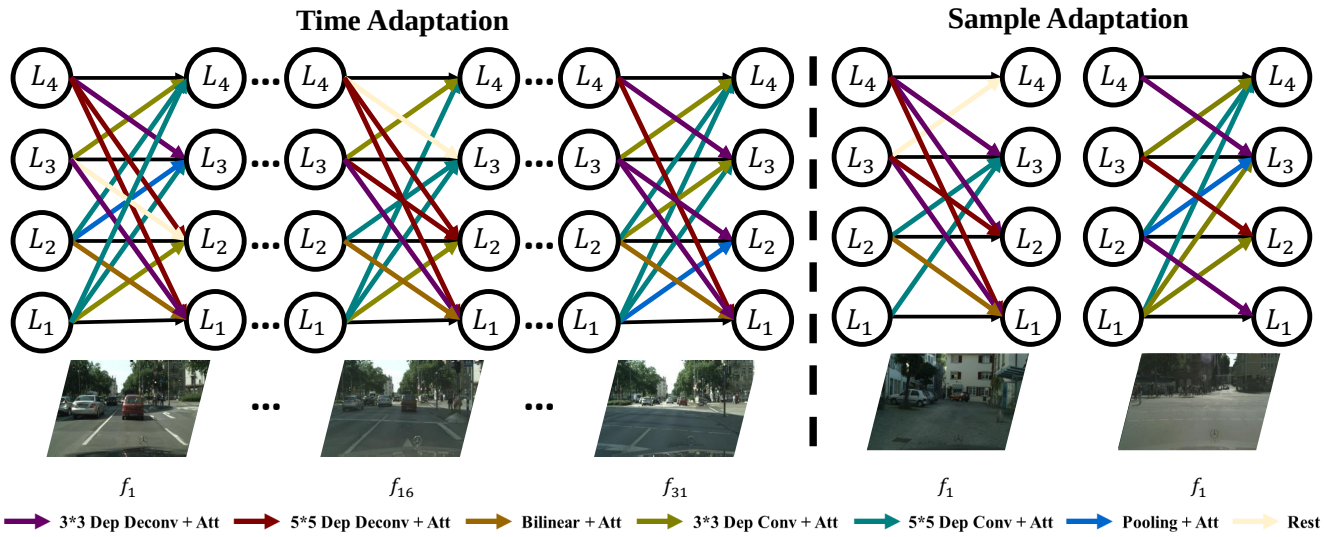
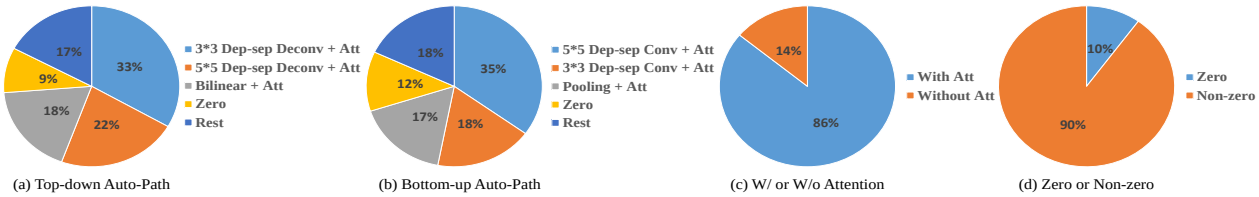


Fig. 5. Some visualization results of the learned auto-path architectures. The left of the figure presents the architecture for the same video at varied time steps and the right gives the architecture for different samples at the same time step. As shown, the architecture of the learned auto-path connections differs slightly for the same sample with different time steps. In contrast, it differs significantly for different samples. Different operations are marked with different colors for better visualization. When zero operation is selected, we do not present any connection between the two ConvLSTMs in the figure. Best viewed in color.



(a) Top-down Auto-Path (b) Bottom-up Auto-Path (c) W/ or W/o Attention (d) Zero or Non-zero

Fig. 6. The statistical information for the operations selected in our APANet on the Cityscapes set. Figure (a) and (b) provide the top 4 operations that have been selected in our top-down and bottom-up auto-path connections, respectively. (c) shows the statistics for with/without attention for non-zero operations. (d) presents the statistics for zero/non-zero operations. Best viewed in color.

segmentation. While for the small objects, some details such as object boundaries become excessively blurred when the resolution is small.

We also summarize the statistical information of the operations selected in our APANet after the auto-path architecture search. The detailed results are presented in Figure 6. From this figure, we can draw three notable observations. Firstly, the operations with attention mechanism are more likely to be used (see Figure 6 (c)). Secondly, the depthwise-separable convolution/deconvolution operations are more likely to be employed as compared with the conventional convolution/deconvolution operations (see Figure 6 (a) and (b)). We attribute this to the fact that the conventional convolution operations with more parameters could introduce more uncertainty to the model learning, which makes it easier to fall into local minima. Lastly, we can observe that most of the auto-path connections between layers are preserved after neural architecture search, i.e., only a small set of connections select zero operations (Figure 6 (d)). This observation demonstrates that selectively and adaptively learning the task-specific hierarchical connections among the features of varied pyramid levels is beneficial for predicting future instance segmentation.

Visualization results for the segmentation prediction. We further visualize some instance segmentation prediction results in Figure 7, where predictions with confidence scores greater than 0.9 are visualized. We can observe that our approach achieves a substantial improvement over the other methods for predicting the segmentation of overlapping and deformed objects. As shown, our approach makes great progress on instance segmentation prediction with better boundaries and more clear contours. These results demonstrate that the proposed method for collaboratively predicting multi-level pyramid features can successfully learn more spatio-temporal task-specific hierarchical contexts from observed past frames for predicting the segmentation of moving objects.

Visualization results for the learned attention map. Figure 8 presents some visualization results about the learned attention maps. We can observe that the attention weights are much larger in the regions containing instances to be segmented. However, the regions with large attention weights differ for different auto-path connections. Also, we can observe that the undeformed objects like cars can obtain relatively high responses, while the deformed and occluded moving objects such as pedestrians are likely to obtain relatively low responses. The results indicate that the employed attention mechanism allows the auto-path connections to selectively aggregate contexts that are closely related to the instances of interest.

4.2.2 *Results on the Inria 3DMovie Dataset*

To evaluate the generalizability of our approach, we further conduct experiments on the Inria 3DMovie Dataset v2 [5],

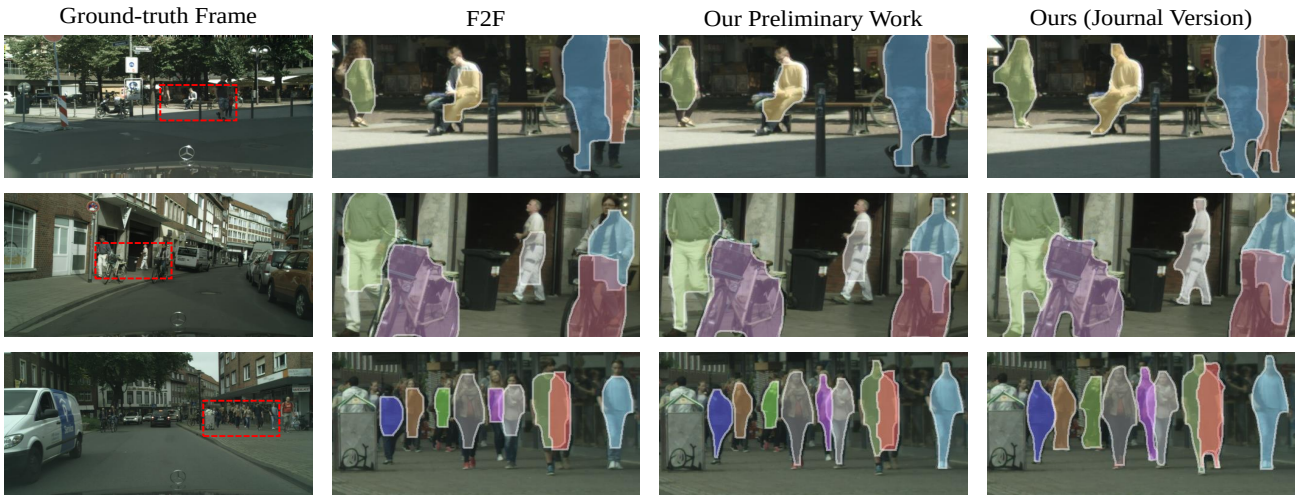


Fig. 7. Some visualized results for the mid-term future instance segmentation prediction. From left to right: the ground-truth frames, the prediction results for the regions indicated by red-dashed boxes obtained by the F2F [2], our preliminary work [94] and our method, respectively. As shown, our method produces the best future instance segmentation prediction results for deformed and occluded moving objects, some of which are highlighted with red dashed boxes in ground-truth frames. The results illustrate that collaboratively predicting multi-level pyramid features with selective and adaptive information aggregation is beneficial for future instance segmentation prediction. Best viewed in color.

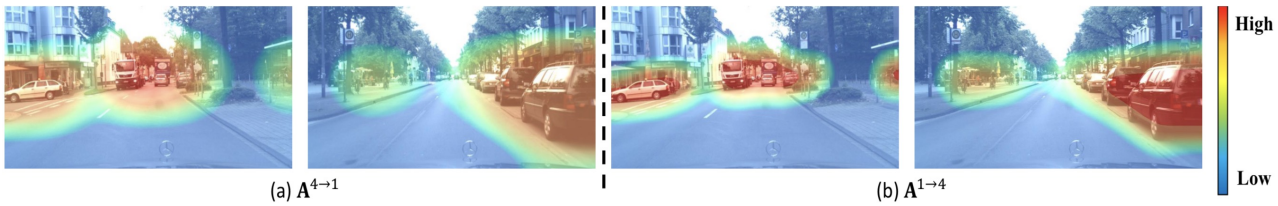


Fig. 8. Qualitative results for the learned attention maps $\mathbf{A}^{v\to l}$, as expressed in Eq. (4). The colors used to visualize the attention values range from red to blue in rainbow order, where red color indicates a high attention value and blue color indicates a low attention value. Best viewed in color.

which was specifically collected for research on instance-level video segmentation. This set consists of 27 video clips, corresponding to a total of 2476 frames, for which masks for 632 person instances are provided. All video clips are obtained from the 3D feature film *StreetDance 3D*. This dataset is an improved version of the Inria 3D Movie Dataset [98] and is challenging for future instance segmentation prediction for the following reasons: 1) the people depicted in some videos appear in very complicated poses (such as dancing and jumping), and 2) self-occlusions often occur in the video clips. Following [2], [5], we split the dataset into a set of 7 clips for training and a set of 20 clips for evaluation. All video sequences are subsampled by a factor of three.

The detailed comparison results are presented in Table 3. Our system again achieves the best performance for both short-term and mid-term instance segmentation prediction. Specifically, for short-term prediction, our model achieves an excellent performance of 52.0% in terms of AP50, which represents an improvement by a large margin of 8.1% over the state-of-the-art model F2F [2]. In terms of the AP metric, our model also clearly outperforms the F2F model, by a margin of 5.0%. This performance improvement demonstrates that our approach performs better than F2F in capturing the spatio-temporal contextual information contained in the observed frames for future instance segmentation prediction. Regarding mid-term future prediction, the table shows similar results: our approach again has an obvious advantage over F2F, with performance results of 35.5% and 18.1% in terms of AP50 and AP, respectively, representing improvements by a margin of 9.7% for AP50 and a margin of 6.0% for AP. A close examination of the mid-term prediction results supports the previous assertion, namely, that our model can capture cross-level spatio-temporal contexts by aggregating spatio-temporal contextual information gained in the multi-level pyramid features with both selection and adaption considered. These promising results confirm that collaboratively predicting multi-level pyramid features with the proposed approach is very helpful for future instance segmentation prediction.

TABLE 3
Comparison results on the Inria 3DMovie Dataset v2.

Method	Short-term		Mid-term	
	AP50	AP	AP50	AP
Mask R-CNN oracle	74.2	30.9	74.2	30.9
Copy-last segmentation	30.5	16.1	17.3	7.6
F2F [2]	43.9	20.7	25.8	12.1
Ours	**52.0**	**25.7**	**35.5**	**18.1**

Interestingly, we can observe that each model achieves higher performance on the Inria 3DMovie Dataset than that on the Cityscapes dataset. We attribute this to the fact that the Cityscapes dataset contains many small instances, which complicates the prediction. Additionally, some classes in the Cityscapes dataset contain few samples, which is not

TABLE 4
Comparison results on the BDD100K dataset.

Method	Short-term		Mid-term	
	AP50	AP	AP50	AP
Mask R-CNN oracle	53.2	29.1	53.2	29.1
Copy-last segmentation	15.3	6.4	4.5	0.7
F2F [2]	28.6	10.3	11.2	4.8
Ours	**32.8**	**13.5**	**16.2**	**8.7**

sufficient for training the APANet. In contrast, the Inria 3DMovie Dataset only requires the segmentation of the human class, which is large and occupies many pixels in the video. Thus, our approach achieves better performance on the Inria 3DMovie Dataset than on the Cityscapes dataset.

4.2.3 Results on the BDD100K Dataset

We also conduct experiments on the BDD100K dataset [6]. This dataset contains 100k raw video sequences with a resolution of 1280×720 and a frame rate of 30 fps, which represents more than 1000 hours of driving and corresponds to more than 100 million images. Like that in the Cityscapes dataset, one image is selected from each video clip for manual annotation. Therefore, a total of 100k images are annotated at the bounding box level, and 10k images are annotated at the pixel level. The videos in this set were captured under various weather conditions (including sun, rain, snow and fog) and different times (including daytime, dawn and night), which makes it much challenging for the future instance segmentation prediction task.

We compare the performance of our method to that of the state-of-the-art future instance segmentation prediction model F2F. The results are presented in Table 4. As shown, our model still achieves the best performance for both short-term and mid-term instance segmentation prediction tasks. Specifically, for the short-term prediction, our model achieves a performance of 32.8% in terms of AP50, with an improvement of 4.2% over the state-of-the-art model F2F. In terms of the AP metric, our model also obviously outperforms the F2F model, by a margin of 3.2%. This indicates that our APANet performs better in capturing the spatio-temporal contextual information among features at different pyramid levels than F2F. For mid-term prediction, our approach has a clear advantage over F2F, with performances of 16.2% and 8.7% in terms of the AP50 and AP metrics, respectively, representing improvements by a margin of 5.0% for AP50 and a margin of 3.9% for AP. These results demonstrate that selectively and adaptively propagating information among features at different pyramid levels and temporal locations with our APANet is substantially beneficial for the future instance segmentation prediction. The employed auto-path aggregation can reduce the domain inconsistencies caused by the captured weather conditions and times among varied samples, by adaptively adjusting the learned auto-path architectures based on the observed video frames. We also note that all the models obtain worse results on the BDD100K dataset than that on the other benchmarks. This somehow indicates that this dataset is more challenging for future instance segmentation prediction.

4.3 Ablation Results

Here, we present extensive ablation studies of the proposed approach on the Cityscapes validation set [4], which is widely used in the literature for the evaluation of instance segmentation-based methods.

4.3.1 Effect of auto-path connections

In this paper, we have introduced auto-path connections among the ConvLSTMs for the features of different pyramid levels to collaboratively predict multi-level pyramid features, which can selectively and adaptively aggregate more task-specific hierarchical spatio-temporal contextual information in videos. Here, we study the benefits of introducing auto-path for future instance segmentation prediction. In our experiments, we first report the results of the baseline "Ours (w/o Path)", which blocks all the connections among the ConvLSTMs in our APANet. Furthermore, we test four different basic settings for the connections, in which the architecture is manually designed rather than automatically learned. i) A network with top-down connections between neighboring pyramid levels, named "Ours (with TD-path)". This network contains three connections, $\{\mathbf{L}_{l+1} \rightarrow \mathbf{L}_l\}_{l=1,2,3}$. ii) A network with bottom-up connections between neighboring pyramid levels, named "Ours (with BU-path)". This network contains three connections, $\{\mathbf{L}_l \rightarrow \mathbf{L}_{l+1}\}_{l=1,2,3}$. iii) A dense extension of "Ours (with TD-path)" that consists of top-down connections among all pyramid levels, named "Ours (with DTD-path)". This network contains 6 top-down connections, $\{\mathbf{L}_l \rightarrow \mathbf{L}_k\}_{1\leq k<l\leq 4}$. iv) A dense extension of "Ours (with BU-path)" that consists of bottom-up connections among all pyramid levels, named "Ours (with DBU-path)". This network contains 6 bottom-up connections, $\{\mathbf{L}_l \rightarrow \mathbf{L}_k\}_{1\leq l<k\leq 4}$.

The results presented in Table 5 show that explicitly aggregating information among the features of different pyramid levels substantially improves the prediction performance for both short-term prediction and mid-term prediction. As expected, our APANet (i.e., "Ours (with Auto-path)") performs better than all the competitors, including the method "Ours (with DTDBU-path)" with manually designed dense connections in both top-down and bottom-up directions, for both short-term prediction (1.2% AP50 improvement) and mid-term prediction (2.9% AP50 improvement). This finding indicates that the architecture defined based on human experience is not the best for aggregating spatio-temporal pyramid contexts. The proposed auto-path connections for selectively and adaptively aggregating information among features of varied pyramid levels can explore more task-specific hierarchical contexts, which is beneficial for predicting future instance segmentation.

4.3.2 Evaluation on the adaptive learning

In our APANet, we have employed a set of parameters $\boldsymbol{\alpha}_{v\rightarrow l}^{t,\psi}$ to quantify the architectures for auto-path connection, which are adaptively determined according to the hidden states of the ConvLSTMs. Specifically, we have defined the parameters $\boldsymbol{\alpha}_{v\rightarrow l}^{t,\psi}$ in Eq. (8) as $\boldsymbol{\alpha}_{v\rightarrow l}^{t,\psi} = f(X_n, C_t^v)$, where C_t^v denotes the hidden state for the v-th layer at the t-th time step for the n-th sample X_n, which is adaptive to the input sample and time steps. Indeed, we can also

TABLE 5
The benefits of introducing connections. Please refer to Section 4.3.1 for more details about the denotations in the table.

Method	Connection				Short-term		Mid-term	
	Top-down	Bottom-up	Dense	Auto	AP50	AP	AP50	AP
Ours (w/o Path)	×	×	×	×	41.9	20.8	22.7	9.2
Ours (with TD-path)	✓	×	×	×	43.1	21.7	24.2	10.3
Ours (with BU-path)	×	✓	×	×	42.3	21.3	23.0	9.7
Ours (with DTD-path)	✓	×	✓	×	44.3	22.1	25.6	11.2
Ours (with DBU-path)	×	✓	✓	×	43.6	21.8	24.7	10.5
Ours (with DTDBU-path)	✓	✓	✓	×	44.9	22.4	26.3	11.7
Ours (with Auto-path)	✓	✓	✓	✓	**46.1**	**23.2**	**29.2**	**12.9**

TABLE 6
Evaluation on the influence of adaptive learning. Please refer to Section 4.3.2 for more details about the denotations in the table.

Method	Short-term		Mid-term	
	AP50	AP	AP50	AP
Ours (w/o Path)	41.9	20.8	22.7	9.2
Ours (w/o sample & time adaption)	44.8	22.5	26.5	11.7
Ours (w/o sample adaption)	45.6	22.7	27.1	12.3
Ours (w/o time adaption)	45.8	22.9	28.4	12.6
Ours	**46.1**	**23.2**	**29.2**	**12.9**

Fig. 9. Visualization results of our approaches with/without adaption.

implement the architecture parameters in other ways. First, we can learn a common architecture parameter $\boldsymbol{\alpha}$ to segment all the samples and time steps, which is a common setting in the existing neural architecture search methods [3], [77]. Here, we define the architecture parameter as $\boldsymbol{\alpha}_{v\to l}^{t,\psi} = \frac{1}{N}\sum_{n=1}^{N} f(X_n, C_1^v)$, where N denotes the number of samples. This is a setting without both sample adaption and time adaption and we denote it as "Ours (w/o sample & time adaption)". Secondly, we can also define the parameters as $\boldsymbol{\alpha}_{v\to l}^{t,\psi} = \frac{1}{N}\sum_{n=1}^{N} f(X_n, C_t^v)$, which means that the model architectures can differ for different time steps but remain the same for different samples (denoted by "Ours (w/o sample adaption)"). Finally, we can also set the parameters varying across different samples while remaining the same for various time steps, e.g., $\boldsymbol{\alpha}_{v\to l}^{t,\psi} = f(X_n, C_1^v)$. Thus, we denote this setting as "Ours (w/o time adaption)".

The comparison results are presented in Table 6. As shown, the model "Ours (w/o sample & time adaption)" using a single common architecture across different samples and different time steps performs the worst in our experiments, with only a 2.9% AP50 improvement for short-term prediction and a 3.8% AP50 improvement for mid-term prediction with respect to the baseline "Ours (w/o Path)". The model "Ours (w/o sample adaption)", which shares auto-path connections among different samples while varies across different time steps, achieves slightly better performance, with a 3.7% AP50 gain for short-term prediction and a 4.4% AP50 gain for mid-term prediction compared to the baseline "Ours (w/o Path)". The performance would be further improved by varying the auto-path connections across different samples but remaining the same at all the time steps (i.e., "Ours (w/o time adaption)"). The proposed APANet, in which the auto-path connections differ for different samples and different time steps, achieves the best prediction performances, with gains of 4.2% AP50 for short-term prediction and 6.5% AP50 for mid-term prediction relative to the baseline "Ours (w/o Path)". The visualization results in Figure 9 also illustrate that adaptive learning is necessary for segmenting video samples with different appearance variations. We can also observe that adapting the architecture parameter to different samples or different time steps improves the system's performance in practice. The performance gap between sample-adaptive methods and sample-inadaptive methods is substantially larger than the time-adaptive and time-inadaptive ones. This is to be expected, as some time-varying information has been encoded by the employed ConvLSTM, which renders it less obvious for future segmentation prediction in the methods using time-adaptive auto-path architectures.

TABLE 7
Comparison results with our preliminary work [94].

Dataset	Method	Short-term		Mid-term	
		AP50	AP	AP50	AP
Cityscapes	Preliminary Work	44.3	22.1	25.6	11.2
	Ours	**46.1**	**23.2**	**29.2**	**12.9**
Inria 3d	Preliminary Work	49.6	24.2	32.4	15.9
Movie v2	Ours	**52.0**	**25.7**	**35.5**	**18.1**
BDD100K	Preliminary Work	30.4	11.7	13.6	6.9
	Ours	**32.8**	**13.5**	**16.2**	**8.7**

4.3.3 *Comparison with our preliminary work*

Here, we provide the comparison results of our approach and our preliminary work [94] in Table 7. As shown, our APANet consistently outperforms the preliminary work on all the datasets, with a margin of more than 1.8% and 2.6%

TABLE 8
Comparison results on the Cityscapes validation set for semantic segmentation prediction.

Method	mIoU	
	Short-term	Mid-term
Mask R-CNN oracle	73.3	73.3
Copy-last segmentation	45.7	29.1
Optical flow - shift [2]	56.7	36.7
Optical flow - warp [2]	58.8	41.4
Mask H2F [2]	46.2	30.5
S2S [1]	55.4	42.4
F2F [2]	61.2	41.2
DeformF2F [73]	63.8	49.9
Our Preliminary Work [94]	63.2	48.6
Ours	**64.9**	**51.4**

in the term of AP50 for short-term prediction and mid-term prediction, respectively. The results indicate that selectively and adaptively aggregating contextual information embedded in the features is substantially beneficial for future instance segmentation prediction. With the introduced adaptive information aggregation, our extension model can adaptively change the network architecture to capture the appearance variations in each individual video.

4.3.4 *Comparison results for semantic segmentation prediction.*

Besides instance segmentation prediction, our approach can also be used to predict semantic segmentation results. For direct comparison to the related work [1], [2], [73], we followed the implementation of F2F [2] and converted our predicted instance segmentation to semantic segmentation. The mIoU metric is employed to evaluate the semantic segmentation prediction results. The detailed results are presented in Table 8. As shown, our approach can still obtain the best performance for the future semantic segmentation prediction task and outperform the competitors that are specially designed for semantic segmentation prediction S2S [1] and DeformF2F [73], with a margin of more than 1.1%. The results indicate that collaboratively predicting multi-level pyramid features with the proposed APANet can also benefit the future semantic segmentation prediction task.

4.3.5 *Influence of the number of pyramid feature levels*

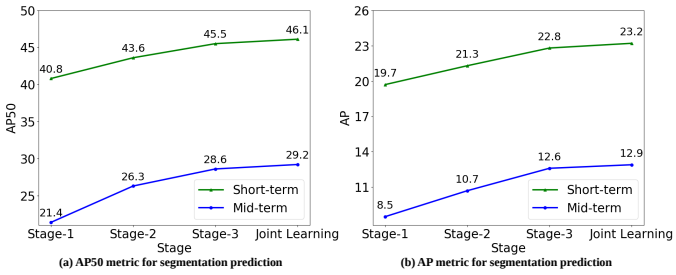
In most of our implementations, our APANet intends to predict the segmentation results by collaboratively forecasting four-level pyramid features (i.e., $\{L_1, L_2, L_3, L_4\}$ outputted by FPN feature extractor [21] with a resolution from high to low). Here, we study the influence of the number of feature levels. The detailed comparison results are presented in Table 9. As shown, aggregating contextual information from more pyramid levels can produce better segmentation results, which demonstrates that the proposed framework for pyramid contextual information aggregation is beneficial for future instance segmentation prediction.

4.3.6 *More evaluations of the optimization*

In this work, we have proposed a three-stage optimization approach to pre-train the APANet and then optimize the

TABLE 9
Influence of the number of pyramid feature levels. Please refer to Section 4.3.5 for more details about the denotations in the table.

Method	Short-term		Mid-term	
	AP50	AP	AP50	AP
L_1	36.9	17.3	18.5	6.8
$L_1 + L_2$	40.5	19.3	22.5	10.1
$L_1 + L_2 + L_3$	42.9	21.4	26.1	11.7
$L_1 + L_2 + L_3 + L_4$	**46.1**	**23.2**	**29.2**	**12.9**



(a) AP50 metric for segmentation prediction (b) AP metric for segmentation prediction

Fig. 10. Evaluation on the optimization for our future instance segmentation prediction system.

APANet, FPN, and Mask R-CNN head in a joint learning manner. Here, we study the influence of each stage in the three-stage optimization. The results are presented in Figure 10. As shown, each stage in our optimization contributes clearly to the system performance. Jointly learning the proposed APANet, feature encoder (FPN), and feature decoder (Mask R-CNN head) obtains better results for the future instance segmentation prediction.

4.3.7 *Influence of λ*

When training the proposed system for future instance segmentation prediction, our objective is to simultaneously minimize both the prediction loss and the segmentation loss with a parameter λ to control the balance between them (see Eq. (9) for details). Here, we investigate its influence by setting it to different values (0.01, 0.1, 1, and 10). The results are presented in Table 10. As shown, our system is quite robust to different values of λ, although it achieves its best performance when a proper λ is employed, e.g., $\lambda = 0.1$. We experimentally find that in the case of $\lambda = 0.1$, the values of the prediction loss and the segmentation loss are similar in most of our experiments, which means that they contribute almost equivalently to the loss. A larger or smaller λ forces the system to focus more on either segmentation or feature prediction, which yields inferior performance for future instance segmentation prediction.

4.3.8 *Prediction with single-frame vs. multi-frame annotations*

Since manually annotating the objects of interest with masks is quite expensive, most existing video-based segmentation datasets only provide a single mask annotation for each video clip. Thus, we have to train our APANet based on the single-frame annotation setting in our experiments. However, we would like to point out that our method can be easily extended to address the multi-frame annotation

TABLE 10
Effects of the parameter λ(Eq. (9)), which is used to control the balance between the prediction loss and the segmentation loss.

	Short-term		Mid-term	
λ value	AP50	AP	AP50	AP
10	45.1	22.4	27.8	11.9
1	45.4	22.6	28.1	12.4
0.1	**46.1**	**23.2**	**29.2**	**12.9**
0.01	44.5	22.0	26.9	11.6

TABLE 11
Evaluation on the system performance based on single-frame vs. multi-frame annotations.

	Short-term		Mid-term	
Method	AP50	AP	AP50	AP
Single-frame Annotation	52.0	25.7	35.5	18.1
Multi-frame Annotation	**52.9**	**26.4**	**37.2**	**18.9**

setting. Specifically, we can just replace the feature prediction loss for each time step with segmentation loss without any other modifications. To obtain multi-frame annotation with low cost, we just use the results of Mask R-CNN model, pre-trained on the MS-COCO dataset [97], to annotate each frame. Here, we conduct experiments on the Inria 3D Movie Dataset v2[1], as the pre-trained Mask R-CNN model can produce precise segmentation results on this set. The detailed results are presented in Table 11. As shown, the use of additional annotations can improve the accuracy of both short-term and mid-term predictions. These results also imply that the system performance can be improved using pre-trained instance segmentation models to annotate multiple frames, without increasing the burden of manual annotation.

5 Conclusion

In this paper, we have addressed the problem of future instance segmentation prediction by collaboratively predicting multi-level pyramid features. Specifically, we have proposed a novel adaptive framework called APANet to selectively and adaptively aggregate the task-specific hierarchical spatio-temporal information gained in the features of different pyramid levels and different temporal locations. Our framework is quite flexible and can adaptively change its network architecture to predict future instance segmentation results for different input samples. We have evaluated the effectiveness of our method on three video-based instance segmentation benchmarks and obtained state-of-the-art results for both the short-term and mid-term prediction. An attempt of improving our method is to further consider learning architecture inside ConvLSTM cells, which requires an extra insight development of LSTM.

Acknowledgment

This work was supported partially by the NSFC(U1911401, U1811461, 62076260), Guangdong NSF Project (No. 2020B1515120085, No. 2018B030312002), Guangzhou Research Project (201902010037), and Research Projects of Zhejiang Lab (No. 2019KD0AB03), and the Key-Area Research and Development Program of Guangzhou (202007030004). The corresponding author for this paper is Wei-Shi Zheng.

[1] We conducted experiments on this set, as the Cityscapes and BDD100K sets are much challenging for the Mask R-CNN model to obtain reliable segmentation results.

References

[1] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *IEEE International Conference on Computer Vision*, 2017, pp. 648–657.
[2] P. Luc, C. Couprie, Y. Lecun, and J. Verbeek, "Predicting future instance segmentation by forecasting convolutional features," in *European Conference on Computer Vision*, 2018, pp. 584–599.
[3] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations*, 2019.
[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
[5] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev, "Instance-level video segmentation from object tracks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3678–3687.
[6] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2636–2645.
[7] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.
[8] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.
[9] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *European Conference on Computer Vision*, 2020.
[10] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.
[11] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 548–557.
[12] Y. Yuan and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.
[13] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "Acfnet: Attentional class feature network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2019, pp. 6798–6807.
[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6230–6239.
[15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Cision*, 2018, pp. 801–818.
[16] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692.
[17] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7519–7528.
[18] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 04, pp. 640–651, 2017.
[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[22] Q. Zhou, W. Yang, G. Gao, W. Ou, H. Lu, J. Chen, and L. J. Latecki, "Multi-scale deep context convolutional neural networks for semantic segmentation," *World Wide Web*, vol. 22, no. 2, pp. 555–570, 2019.

[23] H. Ding, X. Jiang, B. Shuai, A. Qun Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2393–2402.

[24] Y. Li, L. Song, Y. Chen, Z. Li, X. Zhang, X. Wang, and J. Sun, "Learning dynamic routing for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8553–8562.

[25] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, R. Klette, and F. Huang, "Stfcn: spatio-temporal fcn for semantic video segmentation," *arXiv preprint arXiv:1608.05971*, 2016.

[26] Y. Wang, B. Luo, J. Shen, and M. Pantic, "Face mask extraction in video sequence," *International Journal of Computer Vision*, vol. 127, no. 6-7, pp. 625–641, 2019.

[27] Z. Lin, J. Xie, C. Zhou, J.-F. Hu, and W.-S. Zheng, "Interactive video object segmentation via spatio-temporal context aggregation and online learning," *The 2019 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2019.

[28] K. Xu, L. Wen, G. Li, L. Bo, and Q. Huang, "Spatiotemporal cnn for video object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1379–1388.

[29] Y. Wang, M. Dong, J. Shen, Y. Wu, S. Cheng, and M. Pantic, "Dynamic face video segmentation via reinforcement learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6959–6969.

[30] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[31] X. Chen, R. Girshick, K. He, and P. Dollar, "Tensormask: A foundation for dense object segmentation," in *IEEE International Conference on Computer Vision*, 2019, pp. 2061–2069.

[32] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 193–12 202.

[33] M. D. Rodriguez and M. Shah, "Detecting and segmenting humans in crowded scenes," in *ACM International Conference on Multimedia*, 2007, pp. 353–356.

[34] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan, "Proposal-free network for instance-level object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2978–2991, 2018.

[35] D. Neven, B. D. Brabandere, M. Proesmans, and L. Van Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8829–8837.

[36] A. Arnab and P. H. S. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 879–888.

[37] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2858–2866.

[38] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5008–5017.

[39] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "Sgn: Sequential grouping networks for instance segmentation," in *IEEE International Conference on Computer Vision*, 2017, pp. 3516–3524.

[40] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[41] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.

[42] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6409–6418.

[43] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[45] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *IEEE International Conference on Computer Vision*, 2019, pp. 9157–9166.

[46] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," *arXiv preprint arXiv:1912.04488*, 2019.

[47] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[48] Y. Wang, Z. Xu, H. Shen, B. Cheng, and L. Yang, "Centermask: Single shot instance segmentation with point representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[49] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *European Conference on Computer Vision*, 2020.

[50] K. Sofiiuk, O. Barinova, and A. Konushin, "Adaptis: Adaptive instance selection network," in *IEEE International Conference on Computer Vision*, 2019, pp. 7355–7363.

[51] F. Wei, X. Sun, H. Li, J. Wang, and S. Lin, "Point-set anchors for object detection, instance segmentation and pose estimation," in *European Conference on Computer Vision*, 2020.

[52] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou, "Deep snake for real-time instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[53] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun, "Polytransform: Deep polygon transformer for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[54] R. Zhang, Z. Tian, C. Shen, M. You, and Y. Yan, "Mask encoding for single shot instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[55] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*, 2014, pp. 297–312.

[56] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[57] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*, 2016, pp. 75–91.

[58] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *European Conference on Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016.

[59] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *International Conference on Learning Representations*, 2016.

[60] Y. Kwon and M. Park, "Predicting future frames using retrospective cycle gan," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1811–1820.

[61] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos, "Contextvp: Fully context-aware video prediction," in *European Conference on Computer Vision*, 2018, pp. 781–797.

[62] H. Gao, H. Xu, Q. Cai, R. Wang, F. Yu, and T. Darrell, "Disentangling propagation and generation for video prediction," in *IEEE International Conference on Computer Vision*, 2019, pp. 9005–9014.

[63] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Advances in Neural Information Processing Systems*, 2015, pp. 2863–2871.

[64] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," in *Advances in Neural Information Processing Systems*, 2017, pp. 879–888.

[65] X. Chen, W. Wang, J. Wang, and W. Li, "Learning object-centric transformation for video prediction," in *ACM International Conference on Multimedia*, 2017, pp. 1503–1512.

[66] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2287–2301, 2011.

[67] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4194–4202.

[68] Y. Shi, B. Fernando, and R. Hartley, "Action anticipation with rbf kernelized feature mapping rnn," in *European Conference on Computer Vision*, 2018, pp. 301–317.

[69] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating the future by watching unlabeled video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[70] D. Xie, T. Shu, S. Todorovic, and S. Zhu, "Learning and inferring dark matter and predicting human intents and trajectories in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1639–1652, 2018.

[71] M. Rochan *et al.*, "Future semantic segmentation with convolutional lstm," *arXiv preprint arXiv:1807.07946*, 2018.

[72] H.-k. Chiu, E. Adeli, and J. C. Niebles, "Segmenting the future," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4202–4209, 2020.

[73] J. Šarić, M. Oršić, T. Antunović, S. Vražić, and S. Šegvić, "Single level feature-to-feature forecasting with deformable convolutions," in *German Conference on Pattern Recognition*. Springer, 2019, pp. 189–202.

[74] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710.

[75] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, 2019, pp. 4780–4789.

[76] L.-C. Chen, M. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens, "Searching for efficient multi-scale architectures for dense image prediction," in *Advances in neural information processing systems*, 2018, pp. 8699–8710.

[77] C. Liu, L. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 82–92.

[78] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *International Conference on Learning Representations*, 2017.

[79] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proceedings of the genetic and evolutionary computation conference*, 2017, pp. 497–504.

[80] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, "Efficient architecture search by network transformation," in *Proceedings of the aaai conference on artificial intelligence*, 2018.

[81] T. Elsken, J.-H. Metzen, and F. Hutter, "Simple and efficient architecture search for convolutional neural networks," *arXiv preprint arXiv:1711.04528*, 2017.

[82] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *International Conference on Learning Representations*, 2017.

[83] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *International Conference on Machine Learning*, 2017, pp. 2902–2911.

[84] B. Baker, O. Gupta, R. Raskar, and N. Naik, "Accelerating neural architecture search using performance prediction," *arXiv preprint arXiv:1705.10823*, 2017.

[85] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," in *International Conference on Machine Learning*, 2018, pp. 4092–4101.

[86] Z. Zhong, J. Yan, W. Wu, J. Shao, and C.-L. Liu, "Practical block-wise neural network architecture generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2423–2432.

[87] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *International Conference on Machine Learning*, 2018, pp. 550–559.

[88] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, "Neural architecture search with bayesian optimisation and optimal transport," in *Advances in neural information processing systems*, 2018, pp. 2016–2025.

[89] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1946–1956.

[90] L. Ma, J. Cui, and B. Yang, "Deep neural architecture search with deep graph bayesian optimization," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 2019, pp. 500–507.

[91] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *European Conference on Computer Vision*, 2018, pp. 19–34.

[92] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via lamarckian evolution," in *International Conference on Learning Representations*, 2018.

[93] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," in *International Conference on Learning Representations*, 2018.

[94] J. Sun, J. Xie, J.-F. Hu, Z. Lin, J. Lai, W. Zeng, and W.-S. Zheng, "Predicting future instance segmentation with contextual pyramid convlstms," in *ACM International Conference on Multimedia*, 2019, pp. 2043–2051.

[95] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.

[96] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *European Conference on Computer Vision*, 2018, pp. 715–731.

[97] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[98] G. Seguin, K. Alahari, J. Sivic, and I. Laptev, "Pose estimation and segmentation of multiple people in stereoscopic movies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1643–1655, 2015.

Jian-Fang Hu is now an associate professor with Sun Yat-sen University. He received the PhD and B.S. degrees from the School of Mathematics, Sun Yat-Sen University, Guangzhou, China, in 2016 and 2010, respectively. His research interests include human-object interaction modeling, 3D face modeling, and RGB-D action recognition. He has published several scientific papers in the international conferences and journals including ICCV, CVPR, ECCV, IEEE TPAMI, IEEE TCSVT, and PR.

Jiangxin Sun received the bachelors degree in computer science from Sun Yat-Sen University in 2020. He is now a M.S. student in the School of Computer Science and Engineering in Sun Yat-Sen University. His research interests include instance segmentation and 3D human motion.

Zihang Lin received the bachelors degree in computer science from Sun Yat-Sen University in 2020. He is now a M.S. student in the School of Computer Science and Engineering in Sun Yat-Sen University. His research interests include video object segmentation and weakly supervised learning.

Jian-Huang Lai is now a full Professor with Sun Yat-sen University. He received his Ph.D. in Basic Mathematics from Sun Yat-Sen University in 1999. His current research interests are in the areas of computer vision, pattern recognition, machine learning and its applications. He has published over 100 scienti?c papers in international journals and conferences including IEEE TPAMI, IEEE TNN, IEEE TIP, PR, ICCV, CVPR, IJCAI and AAAI. He won the first prize of Guangdong Science and Technology Award for Natural Science (2018 ranking 1). Jian-Huang Lai is also a Fellow of the Chinese Society of Image and Graphics, and serves as vice president of the Chinese Society of Image and Graphics.

Wenjun (Kevin) Zeng (M97-SM03-F12) is a Sr. Principal Research Manager and a member of the senior leadership team at Microsoft Research Asia. He has been leading the video analytics research empowering the Microsoft Cognitive Services, Azure Media Analytics Services, Office, and Windows Machine Learning since 2014. He was with Univ. of Missouri from 2003 to 2016, most recently as a Full Professor. Prior to that, he had worked for PacketVideo Corp., Sharp Labs of America, Bell Labs, and Panasonic Technology. Wenjun has contributed significantly to the development of international standards (ISO MPEG, JPEG2000, and OMA). He received his B.E., M.S., and Ph.D. degrees from Tsinghua Univ., the Univ. of Notre Dame, and Princeton Univ., respectively. His current research interests include mobile-cloud media computing, computer vision, and multimedia communications and security. He is on the Editorial Board of International Journal of Computer Vision. He was an Associate Editor-in-Chief of IEEE Multimedia Magazine, and was an AE of IEEE Trans. on Circuits & Systems for Video Technology, IEEE Trans. on Info. Forensics & Security, and IEEE Trans. On Multimedia (TMM). He was on the Steering Committee of IEEE Trans. on Mobile Computing and IEEE TMM. He served as the Steering Committee Chair of IEEE ICME in 2010 and 2011, and has served as the General Chair or TPC Chair for several IEEE conferences (e.g., ICME2018, ICIP2017). He was the recipient of several best paper awards. He is a Fellow of the IEEE.

Wei-Shi Zheng is now a full Professor with Sun Yat-sen University. Dr. Zheng received his Ph.D. degree in Applied Mathematics from Sun Yat-sen University in 2008. His research interests include person/object association and activity understanding in visual surveillance, and the related large-scale machine learning algorithm. Especially, Dr. Zheng has active research on person re-identification in the last five years. He has ever joined Microsoft Research Asia Young Faculty Visiting Programme. He has ever served as area chairs of CVPR, ICCV, BMVC and IJCAI. He is an IEEE MSA TC member. He is an associate editor of the Pattern Recognition Journal. He is a recipient of the Excellent Young Scientists Fund of the National Natural Science Foundation of China, and a recipient of the Royal Society-Newton Advanced Fellowship of the United Kingdom.