Predictive Feature Learning for Future Segmentation Prediction

Zihang Lin[1]*, Jiangxin Sun[1]*, Jian-Fang Hu[1,4,5]†, Qizhi Yu[2], Jian-Huang Lai[1,4], Wei-Shi Zheng[1,3,5]
[1]Sun Yat-sen University, China [2]Zhejiang Laboratory, China [3]Pengcheng Laboratory, China
[4]Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China
[5] Key Laboratory of Machine Intelligence and Advanced Computing, MOE
{linzh59, sunjx5}@mail2.sysu.edu.cn, hujf5@mail.sysu.edu.cn
qyu@ieee.org, stsljh@mail.sysu.edu.cn, wszheng@ieee.org

Abstract

Future segmentation prediction aims to predict the segmentation masks for unobserved future frames. Most existing works addressed it by directly predicting the intermediate features extracted by existing segmentation models. However, these segmentation features are learned to be local discriminative (with rich details) and are always of high resolution/dimension. Hence, the complicated spatio-temporal variations of these features are difficult to predict, which motivates us to learn a more predictive representation. In this work, we develop a novel framework called Predictive Feature Autoencoder. In the proposed framework, we construct an autoencoder which serves as a bridge between the segmentation features and the predictor. In the latent feature learned by the autoencoder, global structures are enhanced and local details are suppressed so that it is more predictive. In order to reduce the risk of vanishing the suppressed details during recurrent feature prediction, we further introduce a reconstruction constraint in the prediction module. Extensive experiments show the effectiveness of the proposed approach and our method outperforms state-of-the-arts by a considerable margin.

1. Introduction

Future segmentation prediction aims to predict the segmentation masks for unobserved future frames. It serves as a prerequisite for a broad set of applications with decision-making intelligent systems such as autonomous driving, visual surveillance, and robot designing. For instance, self-driving cars can avoid hitting pedestrians if they can forecast possible collisions by predicting the masks of pedestrians in the future. Besides the benefits for potential applications, future segmentation prediction is also closely related to learning better representation for future reasoning, which motivates us to study this problem in this work.

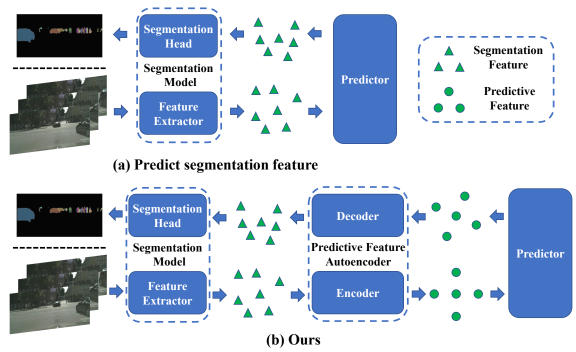

Figure 1. Previous methods addressed future segmentation prediction by predicting the segmentation featureswhich are of high resolution/dimension and contains rich details. However, learning to predict the spatio-temporal variations of these features is difficult. We propose to learn a predictive feature by developing an autoencoder and perform prediction on the learned predictive feature.

Previous work [22] found that directly predicting the segmentation features is much more effective than first predict raw RGB values of future images and then segment. It is now the mainstream pipeline (Figure 1 (a)) in the community and most recent works [27, 5, 29, 30, 34, 13] focused on improving the prediction of these features. However, by revisiting the recent advances of segmentation methods, we discover a conflict between learning discriminative segmentation features and learning reliable future prediction.

The mainstream of learning a strong segmentation model is to learn discriminative feature representation for each pixel. Existing works [2, 38, 43, 10, 41] attempted to achieve this goal by learning resolution preserved representations and aggregating context to enhance local discrimination. The feature learned in this way is of high-resolution and contains rich local details. Although increasing the feature resolution and local details can improve the segmentation performance, it also greatly increase the difficulty of learning accurate future prediction. As the resolution in-

*Equal contributions, † Corresponding author.

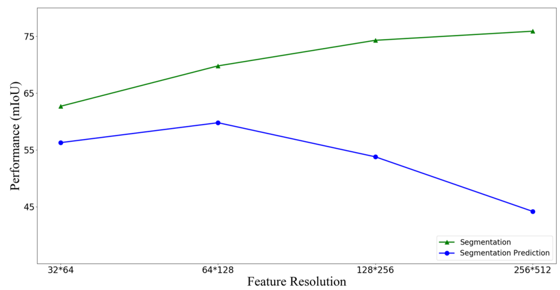

Figure 2. Influence of the feature resolution for segmentation and future segmentation prediction.

creasing, the fitting of the high-uncertainty or even unpredictable local details will gradually dominate the learning process of the prediction model, which can hinder the learning of global motion and leads to degradation of prediction accuracy. To verify the above analysis, we conducted an experiment[1] to investigate the feature resolution's influence on segmentation and future segmentation prediction. As shown in Figure 2, increasing the resolution helps improve the segmentation performance, but it greatly affects the prediction accuracy. This means that learning better segmentation features is contradictory to learning precise future prediction.

In order to address this problem, we seek to learn a predictive feature and perform prediction on this feature space rather than the segmentation feature space. To this end, we construct a novel framework called Predictive Feature Autoencoder as illustrated in Figure 1. Specifically, we develop an encoder-decoder (form an autoencoder) which serves as the bridge between the segmentation feature and the predictive feature. And the prediction is performed in the latent space of the autoencoder. The encoder consists of some convolutions together with rescaling blocks. The rescaling operations are designed to enhance the low-uncertainty global information and suppress the high-uncertainty local details, which makes the outputted feature more predictive. The decoder is correspondingly developed to recover the detailed information suppressed in the encoder and reconstruct the segmentation feature from predictive feature. In order to reduce the risk of vanishing the suppressed details during recurrent feature prediction, we further introduce a reconstruction constraint in the predictor. Combining the above designs, the proposed method achieves new state-of-the-art performances on future segmentation prediction and outperforms all counterparts with a considerable margin.

Overall, the main contributions of this work can be summarized as: 1) we point out the contradiction between learning discriminative segmentation features and learning reliable future prediction. It is a critical weakness commonly existed in previous future segmentation prediction approaches; 2) we propose a simple yet effective autoencoder-based framework which learns predictive features for future segmentation prediction. Extensive experiments have been conducted to demonstrate the effectiveness of each proposed component; 3) our proposed approach achieves new state-of-the-art results and outperforms other methods by a considerable margin on both future instance segmentation and future semantic segmentation prediction.

[1] Experiment details are provided in the supplementary material.

2. Related Work

2.1. Future Segmentation Prediction

Future segmentation prediction aims to predict the segmentation results of the unobserved future frames. It has attracted more and more attention in recent years and many approaches have been proposed. Existing methods mainly focus on the prediction of semantic segmentation [22, 1, 29, 30] and instance segmentation [21, 34, 13].

For semantic segmentation prediction, early works focus on learning a mapping from past segmentation to future segmentation. Luc *et al.* [22] proposed an encoder-decoder to extract features from masks and designed a CNN predictor to forecast the extracted features. Great progress has been made by modeling the temporal relationship using ConvLSTM [27] or attention module [5] and combining multimodal features with variational inference [1]. Recent study shows that predicting intermediate segmentation features is more effective. Vsaric *et al.* [29] employed deformable convolutions to model varied motion patterns. Chiu *et al.* [7] introduced teacher-student learning to learn a better representation. Saric *et al.* [30] enhanced the feature prediction with flow-based forecasting and explicitly modeling the spatio-temporal correlation between neighboring frames.

For instance segmentation prediction, F2F [21] employed several convolutions to predict the pyramid features extracted by FPN [19]. Following the pipeline in [21] to predict pyramid segmentation features, Sun *et al.* [34, 13] proposed to predict the pyramid feature of varied pyramid levels jointly so that the complex structural connections among them can be explicitly explored.

Overall, recent works mainly address future segmentation prediction by predicting the segmentation features. In this work, we figure out the weakness of directly predicting the segmentation features and focus on learning a predictive feature representation for future segmentation prediction.

2.2. Image Segmentation

Image segmentation is a fundamental computer vision problem which aims to assign a label to each pixel. Most recent works develop deep neural networks to address it as a pixel-wise classification task and the key is to learn discriminative segmentation feature for each pixel. The recent approaches for learning better segmentation features can be

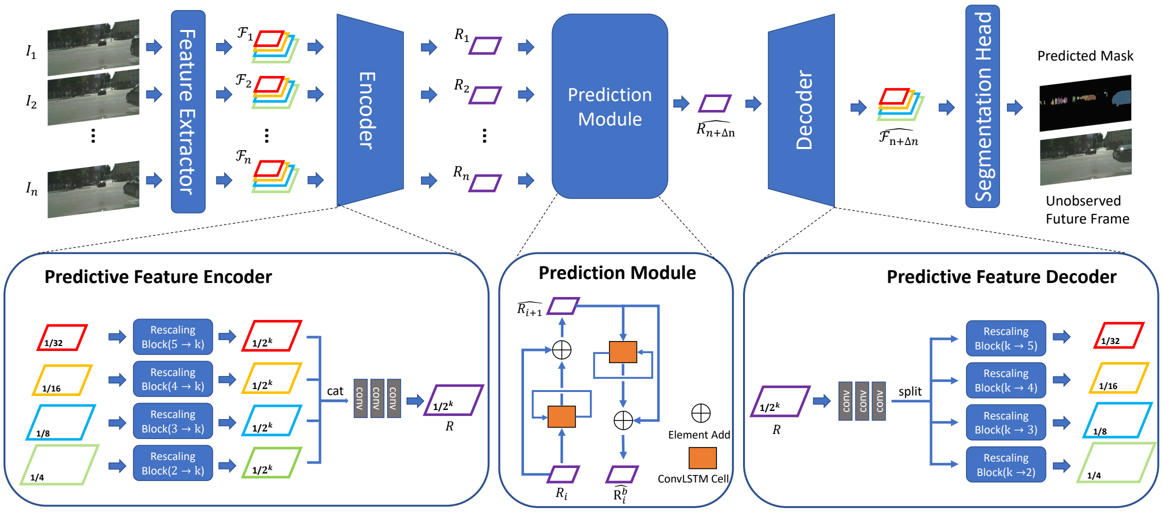

Figure 3. An overview of the proposed framework. We first encode the pyramid segmentation features to a low-resolution predictive representation for each observed frame and then predict this representation for unobserved future frame with the prediction module. The predicted feature is then decoded back to pyramid features, which are finally fed into a segmentation head to generate segmentation results.

coarsely divided into two categories: learning context to enhance local discrimination and learning high-resolution representations. For context learning, there are two mainstreams. PSPNet [43], DeepLab series [2, 4] are developed to learn multi-scale context. DANet [10], CCNet [14], OCNet [42], ANNet [44], and OCRNet [41] learn context according to self-similarity in the feature space. For learning high-resolution representations, deconvolutions with skip-connections [28] and dilated convolutions [3, 40] are adopted in many segmentation models. Recently, HRNet [38] proposed to learn high-resolution feature representations and it becomes a popular backbone for segmentation models. In short, learning high-resolution feature representation together with context aggregation is the mainstream for improving image segmentation in recent years. However, we observe that the segmentation features learned in this way are not suitable for future segmentation prediction. Therefore, we seek to learn a predictive feature from the segmentation features.

2.3. Video Prediction

The goal of video prediction is to synthesize future frames according to observed past video sequences [25]. Early works focused on directly predicting raw pixel values. Enormous mechanisms (including patch clusters [26], autoencoder [33], adversarial training [23], bidirectional flow [18] and 3D convolution [39]) have been introduced to improve the accuracy of prediction. However, performing prediction in the original pixel space is difficult since the dimension and uncertainty are high and there are some unpredictable noises. Recently, great progress has been achieved by simplifying the prediction task. They tried to factorize the prediction space. These works explicitly modeled the variability as transformations between frames by introducing spatial transformer [16], dynamic neural advection [9], object-centric representation [6] or separated motion from content [36]. In summary, the existing literature tried to find a prediction space where the uncertainty is low so that more reliable predictions can be made. In this work, we also seek to make the prediction more reliable and propose to learn a predictive feature in a new feature space.

3. Method

In this work, we develop a novel framework called **Predictive Feature Autoencoder**, which intends to learn a predictive feature for improving future segmentation prediction. The flowchart of our framework is illustrated in Figure 3. As shown, our framework contains three blocks: a feature encoder, a bidirectional prediction module, and a feature decoder. Specifically, the feature encoder encodes the pyramid segmentation feature to a unified low-resolution feature which is predictive. The prediction module predicts this feature for future frame and the feature decoder decodes the predicted feature to segmentation feature which is fed into the segmentation head for producing segmentation masks. In the following, we will introduce the major components in the proposed model.

3.1. Predictive Feature Encoder and Decoder

We propose a feature encoder and decoder to learn a predictive representation for feature prediction. Following previous works [21, 34], we construct our approach based on an advanced segmentation model, which extracts pyramid segmentation features for segmentation. Instead of directly predicting these features, we propose to predict the feature

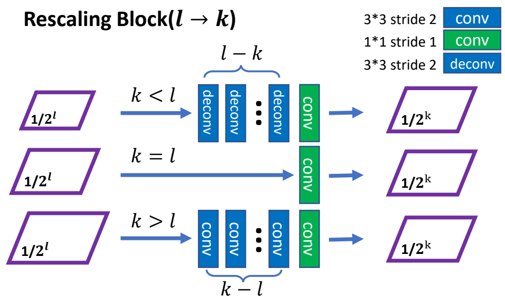

Figure 4. The architecture of the rescaling block.

outputted by the encoder. Specifically, our encoder intends to learn a predictive feature in which the local details are suppressed and the global structures are enhanced. Correspondingly, our decoder is developed to recover the detailed information suppressed in the encoder and reconstruct the segmentation features from the predictive feature. The proposed feature encoder and decoder form an autoencoder to learn predictive features from the segmentation features without losing meaningful information. The detailed architectures for our encoder-decoder are illustrated in Figure 3.

In the encoder, in order to suppress the local details and to better capture the global structure, we develop a rescaling block to rescale the segmentation features to a proper resolution (a relatively low resolution in practice). Our rescaling block is defined as a set of stacked convolutions or deconvolutions operators, as illustrated in Figure 4. Considering an input feature map with resolution $\frac{h}{2^l} \times \frac{w}{2^l}$ and the target resolution $\frac{h}{2^k} \times \frac{w}{2^k}$ where h and w are the height and width of the input feature, respectively. The rescaling block can be formulated as:

$$Rb_{l \to k}(\cdot) = g(Rs_{l \to k}(\cdot)), \tag{1}$$

where $g(\cdot)$ is a 1×1 convolution with stride 1 and $Rs_{l \to k}(\cdot)$ is the rescaling operation defined as follows:

$$Rs_{l \to k}(\cdot) = \begin{cases} \underbrace{f_c \circ f_c \circ \cdots \circ f_c}_{k-l}(\cdot), & k > l \\ Identity, & k = l \\ \underbrace{f_d \circ f_d \circ \cdots \circ f_d}_{l-k}(\cdot), & k < l \end{cases} \tag{2}$$

where $f_c(\cdot), f_d(\cdot)$ are convolution and deconvolution with stride 2 followed by a BN layer [15], respectively. The kernel sizes of convolution and deconvolution are set as 3×3.

Following the rescaling block, we concatenate the rescaled feature maps at the channel dimension. Then three stacked convolutions with Batch Normalization [15] and ReLU [24] activation are applied to aggregate information from the features rescaled from different pyramid levels, forming our predictive feature representation.

The input of our feature decoder is the predictive feature forecasted by our prediction module, which has the same shape as the output of our encoder. As shown in Figure 3, we first employ three stacked 3×3 convolution operations to disentangle the input into several feature maps of the same shape. Similar to the encoder, we employ rescaling blocks (see Equation (1)) to rescale these feature maps to reconstruct pyramid segmentation features.

The proposed feature encoder $E(\cdot)$ and decoder $D(\cdot)$ form an autoencoder to learn predictive features from segmentation features. Let us denote $\mathcal{F} = \{\mathbf{P}_1^t, \mathbf{P}_2^t, ..., \mathbf{P}_L^t\}$ (L scales in total) as the pyramid feature of frame t generated by the feature extractor, and denote $\hat{\mathcal{F}} = D\left[E(\mathcal{F})\right] = \{\mathbf{Q}_1^t, \mathbf{Q}_2^t, ..., \mathbf{Q}_L^t\}$ as the multi-scale features of frame t produced by the decoder, the encoder and the decoder are trained with the following reconstruction loss:

$$L_{rec} = \sum_t \sum_{i=1}^{L} \left\| \mathbf{P}_i^t - \mathbf{Q}_i^t \right\|_F^2, \tag{3}$$

which is defined such that the decoder can exactly reconstruct the original pyramid feature from the predictive feature produced by the encoder. It means that the information loss is explicitly minimized and the details hidden in the encoding stage can be reconstructed in the decoding stage.

To illustrate the effect of our autoencoder more intuitively, we performed Fourier transform on the original segmentation feature, the predictive feature produced by our encoder and the reconstructed pyramid feature outputted by our decoder. As shown in Figure 5, the original segmentation feature contains many high-frequency components. The encoder hides some high-frequency components and produces a latent representation containing more low-frequency components. However, it does not mean that the high-frequency components corresponding to local details are discarded. Our decoder would reconstruct them back in the decoding stage, as shown in the comparison with (b) and (d) in Figure 5. Intuitively, the encoder weakens the local details and enhances the global structure in the latent space for producing a more predictive feature representation.

3.2. Prediction Module

The prediction module is employed to forecast the predictive feature (outputted by feature encoder) for future unobserved frames. The detailed architecture of our prediction module is presented in Figure 6, which takes the predictive features of the observed frames as inputs and outputs the feature prediction of the unobserved future frames. We develop our prediction module based on the ConvLSTM framework [32]. Since some detailed information is suppressed in the predictive representation, there is a high risk that the detailed information could be lost due to information vanishing during the recurrent feature prediction. In order to mitigate the information vanishing problem, we introduce a reconstruction constraint in the prediction module and formulate it as a combination of two ConvLSTMs,

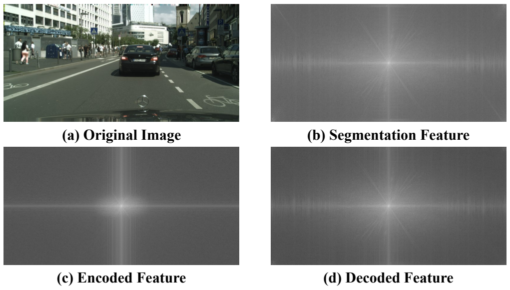

Figure 5. Visualized results of applying Fourier transform on segmentation feature, encoded predictive feature and decoded feature.

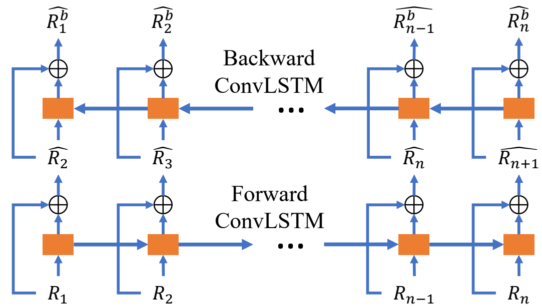

Figure 6. The detailed architecture of the prediction module.

i.e., forward ConvLSTM and backward ConvLSTM. The forward ConvLSTM is defined to predict the features of the unobserved future frames. In contrast, the backward ConvLSTM is used to reconstruct the features of observed past frames. For both the forward and backward ConvLSTM, we add a skip connection in each ConvLSTM cell (as shown in Figure 6), which means that the employed ConvLSTMs mainly predict the feature difference between temporal neighboring frames. Formally, the prediction procedure can be formulated as:

$$\begin{aligned} \hat{\mathbf{R}}_t &= \mathbf{R}_{t-1} + \mathbf{\Theta}_f([\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_{t-1}]), \\ \hat{\mathbf{R}}_t^b &= \hat{\mathbf{R}}_{t+1} + \mathbf{\Theta}_b([\hat{\mathbf{R}}_T, \hat{\mathbf{R}}_{T-1}, ..., \hat{\mathbf{R}}_{t+1}]), \end{aligned} \tag{4}$$

where $\mathbf{R}_t$ is the input feature representation for frame t, $\hat{\mathbf{R}}_t$ is the feature of frame t predicted by the forward ConvLSTM, and $\hat{\mathbf{R}}_t^b$ is the output of backward ConvLSTM. Mappings Θ_f and Θ_b represent the forward and the backward ConvLSTMs without skip connection, respectively. T represents the maximum temporal length.

To train our prediction module, we minimize the following prediction loss:

$$L_{pred} = \sum_{t=2}^{T} \left\| \mathbf{R}_t - \hat{\mathbf{R}}_t \right\|_F^2 + \sum_{t=1}^{T} \left\| \mathbf{R}_t - \hat{\mathbf{R}}_t^b \right\|_F^2, \tag{5}$$

where the first term is employed to measure the prediction loss corresponding to forward ConvLSTM, and the second term is a reconstruction loss, which constraints the prediction module not to forget the input information. Intuitively, if some input information is lost, it is impossible for the backward ConvLSTM to correctly reconstruct the original input, which leads to a large reconstruction loss.

3.3. Model Training and Inference

Here, we present our training and testing procedure.

Training. We train our model with a three-stage optimization strategy. The first and second stages are employed to pre-train the parameters of our encoder-decoder and prediction module, respectively. Specifically, in the first stage, we discard the prediction module and train the encoder-decoder with the reconstruction loss L_{rec} defined in Equation (3). In the second stage, we fix the encoder and decoder, and only train the prediction module with the loss L_{pred} defined in Equation (5). Finally, we train the whole system jointly in the third stage by minimizing the following loss:

$$L = \lambda_{rec} L_{rec} + \lambda_{pred} L_{pred} + \lambda_{seg} L_{seg}, \tag{6}$$

where $\lambda_{rec}, \lambda_{pred}, \lambda_{seg}$ are weights to control the contribution of different loss terms. L_{rec} and L_{pred} are the losses defined previously. L_{seg} is a loss corresponding to the employed segmentation model. Specifically, for future instance segmentation prediction, we choose Mask R-CNN [12] as our segmentation model and L_{seg} consists of a classification loss, a bounding box regression loss and a segmentation loss as defined in [12]. For future semantic segmentation prediction, we employ Semantic FPN [17] as the segmentation model and L_{seg} is the pixel-wise cross-entropy loss between the predicted mask and ground-truth.

Inference. The inference is quite straightforward. We first use the feature extractor to extract pyramid segmentation features and feed them into our feature encoder, to obtain the corresponding predictive feature representations. We then feed them into the forward ConvLSTM of our prediction module, the feature decoder and the segmentation head to generate segmentation prediction results.

4. Experiments

We conduct experiments on two benchmark sets for future instance segmentation prediction and future semantic segmentation prediction tasks.

4.1. Experimental Settings

Datasets. We conduct experiments on the Cityscapes [8] and Inria 3DMovie Dataset v2 [31], both of which are specifically collected for the research of video-based segmentation. Cityscapes contains a total of 5000 sequences, in which 2975, 500 and 1525 sequences are used for model training, validation and testing, respectively. Each sequence contains 30 image frames and the 20-th frame are manually

Table 1. Comparison results for future instance segmentation prediction on the Cityscapes validation set. †: concurrent work.

	Short-term		Mid-term	
	AP50	AP	AP50	AP
Mask R-CNN [11] oracle	65.8	37.3	65.8	37.3
Copy-last segmentation	24.1	10.1	6.6	1.8
Optical flow - Shift [21]	37.0	16.0	9.7	2.9
Optical flow - Warp [21]	36.8	16.5	11.1	4.1
Mask H2F [21]	25.5	11.8	14.2	5.1
F2F [21]	39.9	19.4	19.4	7.7
CPConvLSTM [34]	44.3	22.1	25.6	11.2
APANet† [13]	46.1	23.2	29.2	12.9
Ours	**48.7**	**24.9**	**30.5**	**14.8**

Table 2. Comparison results for future instance segmentation prediction on the Inria 3DMovie Dataset v2 set. †: concurrent work.

	Short-term		Mid-term	
Method	AP50	AP	AP50	AP
Mask R-CNN [11] oracle	74.2	30.9	74.2	30.9
Copy-last segmentation	30.5	16.1	17.3	7.6
F2F [21]	43.9	20.7	25.8	12.1
CPConvLSTM [34]	49.6	24.2	32.4	15.9
APANet† [13]	52.0	25.7	35.5	18.1
Ours	**52.9**	**26.3**	**36.1**	**18.4**

annotated with masks for both semantic segmentation and instance segmentation. Inria 3DMovie Dataset v2 is collected for performing video instance segmentation, which consists of 27 video clips corresponding to 2476 frames in total. Masks of 632 person instances are provided in this set. Following the settings in [34], we split this set into a training set (7 clips) and a validation set (20 clips).

Evaluation settings. Same as [8], we measure the performance of our method using the metrics AP50 and AP for future instance segmentation prediction and mIoU (mean intersection over union) for future semantic segmentation prediction. Following settings in [21], we temporally subsampled all sequences by a factor of three and frames $\{I_{t-9}, I_{t-6}, I_{t-3}, I_t\}$ form the input of our method. Both short-term and mid-term prediction are conducted to predict segmentation of future frame $\{I_{t+3}\}$ (about 0.17 second later) and $\{I_{t+9}\}$ (about 0.5 second later), respectively. For mid-term prediction, we perform it with 3 auto-regressive forecasting steps, i.e. predict $I_{t+3}, I_{t+6}, I_{t+9}$.

Implementation details. For future instance segmentation prediction, we follow the implementations in [21, 34] and employ the Mask R-CNN [12] pre-trained on the MS-COCO dataset [20] with ResNet-50-FPN backbone as our segmentation model. For future semantic segmentation prediction, we employ the Semantic FPN [17] model with a ResNet-50-FPN backbone as our segmentation model. We train our approaches using the stochastic gradient descent (SGD) algorithm with a Nesterov momentum of 0.9. The batch size is set to 8. In the first training stage, we trained

Table 3. Comparison results for future semantic segmentation prediction on the Cityscapes validation set using mIoU as the evaluation metric. ALL: all classes. MO: moving objects.

	Short-term		Mid-term	
Method	ALL	MO	ALL	MO
Semantic FPN [17] Oracle	75.9	75.1	75.9	75.1
Copy-last segmentation	53.5	48.5	38.9	29.8
3Dconv-F2F[7]	57.0	/	40.8	/
Dil10-S2S[22]	59.4	55.3	47.8	40.8
F2F[21]	/	61.2	/	41.2
FeatReproj3D[37]	61.5	/	45.4	/
Bayesian S2S[1]	65.1	/	51.2	/
DeformF2F[29]	65.5	63.8	53.6	49.9
LSTM AM S2S[5]	65.8	/	51.3	/
APANet[13]	/	64.9	/	51.4
LSTM M2M[35]	67.1	65.1	51.5	46.3
F2MF[30]	69.6	67.7	57.9	54.6
Ours	**71.1**	**69.2**	**60.3**	**56.7**

the encoder and decoder with a learning rate of 0.01 for 4 epochs. In the second stage, we trained the prediction module with learning rate 0.01 for 4 epochs. In the third stage, we fine-tuned the whole framework jointly using different learning rates for different blocks for 18 epochs. Specifically, the learning rate for our encoder, prediction and decoder modules is set as 0.01 and decreased to 0.001 after 10 epochs, while the learning rate for segmentation blocks (i.e., the FPN feature extractor and task-specific segmentation head) is set as 0.0001. The parameters in the loss (Equation (6)) is set as $\lambda_{seg} = 0.1$, $\lambda_{rec} = 1$ and $\lambda_{pred} = 1$. The total training time is 2 days using one V100 GPU.

4.2. Main Results

Here, we report our results for future instance segmentation and future semantic segmentation prediction.

4.2.1 Results for Instance Segmentation Prediction

We first conduct experiments for future instance segmentation prediction on the Cityscapes [8]. We compare our method with state-of-the-art approaches including F2F [21], CPConvLSTM [34] and four baselines (copy-last segmentation baseline, Optical flow-Shift, Optical flow-Warp, Mask H2F) developed in [21]. We also report the performance of Mask R-CNN oracle which performs future segmentation prediction by feeding the corresponding ground truth frames to Mask R-CNN. This performance can be seen as an **upper bound** of our system. The comparison results are presented in Table 1. Our method consistently outperforms all the competitors, which intends to directly predict segmentation features, by a considerable margin for both short-term (+2.6% AP50, +1.7%AP) and mid-term (+1.3% AP50, +1.9%AP) predictions. The results demonstrate that the proposed predictive feature auto-encoder can

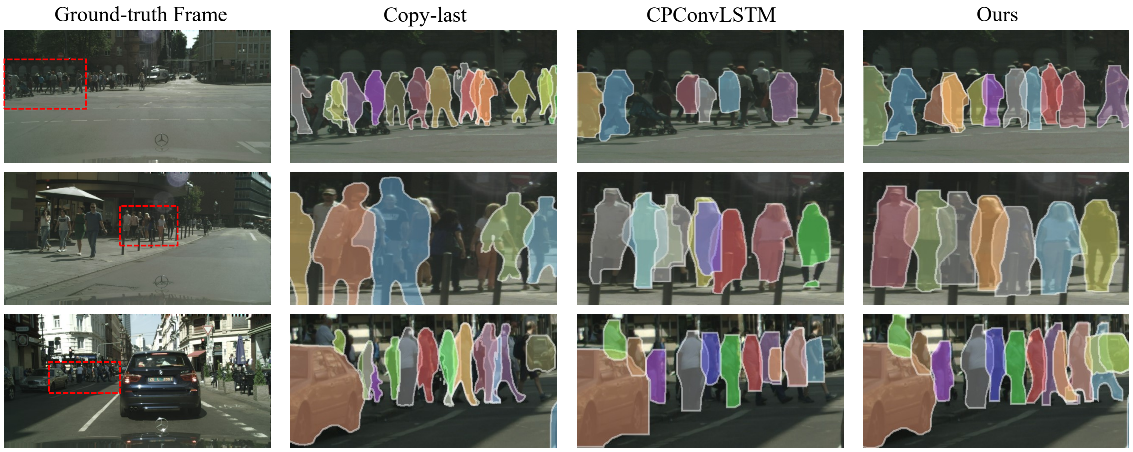

Figure 7. Some visualized results for the mid-term future instance segmentation prediction. Best viewed in color.

better capture the feature variations for future prediction. The results also verify that predicting the predictive features with fewer local details can achieve better performance.

We report our results on the Inria 3DMovie dataset v2 [31] in Table 2. As shown, our method achieves state-of-the-art performances for both short-term and mid-term future instance segmentation prediction. Specifically, for the short-term prediction, our method achieves 52.9% AP50 and 26.3% AP, which are 3.3% and 2.1% higher than the results reported in CPConvLSTM [34] in terms of AP50 and AP, respectively. For the mid-term prediction, our approach outperforms CPConvLSTM [34] by 3.7% AP50 and 2.5% AP. Our method also achieves a higher performance than the concurrent work APANet [13]. These results further demonstrate the effectiveness of our method.

We further present some visualization results in Figure 7. As shown, our model gains substantial improvements over CPConvLSTM[34]. By examining the results in the first row, we can find that our method can successfully predict the pedestrians with occlusion, most of which are missed by CPConvLSTM. The results in the second row show that the masks predicted by our method are more complete and they can cover the whole body of the pedestrians. In the last row, we observe that the boundary of the car is accurately predicted, which demonstrates that although our approach intends to construct predictive features with fewer local image details in the encoding stage for better feature prediction, the hidden details can be recovered in the decoding stage. Overall, the interesting observations in the visualization results demonstrate that the proposed encoder-decoder can effectively produce a predictive representation for future instance segmentation prediction.

4.2.2 Results for Semantic Segmentation Prediction

To further validate the effectiveness of our approach, we conduct experiments for future semantic segmentation prediction. Here, we exactly follow the settings in [22] and conduct experiments on the Cityscapes dataset [8]. The results are presented in Table 3, where the mIoU scores for all classes (termed ALL) and 8 certain classes with moving objects (termed MO) [22, 21] are reported. As shown, for the short-term prediction, our method outperforms F2MF [30] by a margin of 1.5%. It is worth noting that the performance gap between our method and Semantic FPN oracle is already quite small, which performs segmentation based on the ground-truth image data for the frame to be predicted. For mid-term prediction, our method outperforms F2MF [30] by a larger margin of 2.4%. We attribute this to that the proposed prediction module together with the encoder-decoder can capture more feature variation information.

4.3. Ablation Study

In this section, we conduct extensive ablation experiments for future instance segmentation prediction on the Cityscapes dataset [8] to study the influence of each component in our framework.

4.3.1 Analysis on the Proposed Autoencoder

Here, we provide in-depth analysis of the proposed autoencoder by gradually removing components in the autoencoder. The results are summarized in Table 4.

We first simplify the architecture of the autoencoder by discarding the multi-scale feature fusion in the autoencoder, i.e., removing the concatenate operation in the encoder and the split operation in the decoder. In this experiment, since the pyramid features are not fused in the encoder, we need to employ 4 prediction modules to predict features of each pyramid level independently. The results in Table 4 that the performance degrades by 1.2% and 1.7% in terms of AP50 for short-term and mid-term prediction, respectively. This indicates that fusing the multi-scale feature is necessary, especially for mid-term prediction. We conjecture that

Table 4. Evaluation on the autoencoder and other components.

Method	Short-term AP50	Short-term AP	Mid-term AP50	Mid-term AP
Ours	**48.7**	**24.9**	**30.5**	**14.8**
w/o Multi-scale Fusion	47.5	24.1	28.8	13.6
w/o Rescaling	45.7	23.0	26.4	11.7
w/o Autoencoder	44.2	22.1	24.7	10.6

Table 5. Evaluation on the effectiveness of our prediction module.

Method	Short-term AP50	Short-term AP	Mid-term AP50	Mid-term AP
Ours	**48.7**	**24.9**	**30.5**	**14.8**
w/o Reconstruction Loss	47.9	24.1	29.4	13.6
w/o Residual Prediction	47.0	23.4	28.5	13.2

this is because independently predicting features of different scales could ignore the structural information in pyramid features and thus produce inconsistent prediction for the features of different pyramid levels.

We further remove the rescaling blocks to validate the effectiveness of re-scaling in the autoencoder. Now both the encoder and the decoder consist of 3 stacked 3×3 convolutions. As shown in the third row in Table 4, this further decreases the performance by 1.8% and 2.4% in terms of AP50 for short-term and mid-term predictions, respectively. The results demonstrate that forcing the autoencoder to hide some details with the re-scaling blocks can help learn better feature for prediction. It is also worth noting that with such a simplified autoencoder, our method can still perform better than the model without autoencoder by a margin of around 1.5% AP50 for both short-term and mid-term prediction, which indicates that building an autoencoder architecture is a simple yet effective way to learn predictive features for future segmentation prediction.

Overall, the proposed autoencoder brings significant gains for both short-term (+4.5%AP50, +2.8%AP) and mid-term (+5.8%AP50, +4.2%AP) predictions. The promising improvement demonstrates that the proposed autoencoder can effectively learn a more predictable feature representation for better prediction of future features.

4.3.2 Evaluation on the Prediction Module

In this section, we evaluate the effectiveness of the prediction module and tabulate the results in Table 5. In this work, we propose to formulate the prediction module bidirectional by adding a backward ConvLSTM and calculating a reconstruction loss as formulated in Equation (5). We also develop the prediction module to predict the feature residual between neighboring frames rather than directly predict the feature of next frame. As shown in Table 5, both the reconstruction loss and the residual prediction can bring some gains to the system performance. This is because the prediction module can mitigate the information vanishing problem in recurrent predictions.

Table 6. Evaluation on the influence of feature resolution of predictive feature.

The value of k	Short-term AP50	Short-term AP	Mid-term AP50	Mid-term AP
$k = 2$	43.4	21.3	23.5	10.1
$k = 3$	45.1	22.8	26.3	12.2
$k = 4$	**48.7**	**24.9**	**30.5**	**14.8**
$k = 5$	47.9	24.3	29.4	14.1

4.3.3 Study on the Resolution of Predictive Feature

We investigate the influence of the resolution of the learned predictive feature in this section. In the proposed autoencoder, the encoder transforms the pyramid segmentation features to predictive feature with a unified resolution of $\frac{h}{2^k} \times \frac{w}{2^k}$. We evaluate different values of k and the results are tabulated in Table 6. As shown, the proposed method achieves best performance with $k = 4$. Noting that when $k = 2$, the performance degrade significantly, which indicates that high-resolutional features contain too many details, which hinders the learning of the feature predictor. When $k = 5$, the performance is also not good enough due to excessive loss of detailed information, indicating that choosing a proper resolution for the predictive feature is quite important for future segmentation prediction.

5. Conclusion

In this work, we have addressed the problem of future segmentation prediction by learning predictive features. Specifically, we have proposed a novel framework (Predictive Feature Autoencoder) containing a feature encoder, a prediction module, and a decoder. The encoder is employed to learn a predictive feature from segmentation feature. The decoder is defined to reconstruct segmentation features from the predictive features. We have further introduced residual prediction and reconstruction constraint to reduce the risk of information vanishing during recurrent feature prediction. Extensive experiments on two video-based segmentation sets show that our method outperforms the state-of-the-art methods by a considerable margin.

Acknowledgement

This work was supported partially by the NSFC (U1911401, U1811461, 62076260, 61772570), Guangdong Natural Science Funds Project (2020B1515120085, 2018B030312002), Guangdong NSF for Distinguished Young Scholar (2018B030306025), Guangzhou Research Project (201902010037), Research Projects of Zhejiang Lab (2019KD0AB03), and the Key-Area Research and Development Program of Guangzhou (202007030004).

References

[1] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Bayesian prediction of future street scenes using synthetic likelihoods. In *International Conference on Learning Representations*, 2018.

[2] 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*, 40(4):834–848, 2018.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018.

[5] Xin Chen and Yahong Han. Multi-timescale context encoding for scene parsing prediction. In *IEEE International Conference on Multimedia and Expo*, pages 1624–1629, 2019.

[6] Xiongtao Chen, Wenmin Wang, Jinzhuo Wang, and Weimian Li. Learning object-centric transformation for video prediction. In *Proceedings of the ACM international conference on Multimedia*, pages 1503–1512, 2017.

[7] Hsu-kuang Chiu, Ehsan Adeli, and Juan Carlos Niebles. Segmenting the future. *IEEE Robotics and Automation Letters*, 5(3):4202–4209, 2020.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[9] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 64–72, 2016.

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[11] K He, G Gkioxari, P Dollar, and R Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2018.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):386–397, 2020.

[13] Jian-Fang Hu, Jiangxin Sun, Zihang Lin, Jian-Huang Lai, Wenjun Zeng, and Wei-Shi Zheng. Apanet: Auto-path aggregation for future instance segmentation prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[14] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612, 2019.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the 28tInternational Conference on Neural Information Processing Systems*, pages 2017–2025, 2015.

[17] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[18] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014.

[21] Pauline Luc, Camille Couprie, Yann Lecun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *Proceedings of the European Conference on Computer Vision*, pages 584–599, 2018.

[22] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 648–657, 2017.

[23] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations*, 2016.

[24] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

[25] Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[26] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[27] Mrigank Rochan et al. Future semantic segmentation with convolutional lstm. *arXiv preprint arXiv:1807.07946*, 2018.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[29] Josip Šarić, Marin Oršić, Tonći Antunović, Sacha Vražić, and Siniša Šegvić. Single level feature-to-feature forecasting with deformable convolutions. In *German Conference on Pattern Recognition*, pages 189–202, 2019.

[30] Josip Saric, Marin Orsic, Tonci Antunovic, Sacha Vrazic, and Sinisa Segvic. Warp to the future: Joint forecasting of features and feature motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10648–10657, 2020.

[31] Guillaume Seguin, Piotr Bojanowski, Remi Lajugie, and Ivan Laptev. Instance-level video segmentation from object tracks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[32] Xingjian Shi, Zhourong Chen, Hao Wang, Dit Yan Yeung, Wai Kin Wong, and Wangchun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, 2015.

[33] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning*, pages 843–852, 2015.

[34] Jiangxin Sun, Jiafeng Xie, Jian-Fang Hu, Zihang Lin, Jianhuang Lai, Wenjun Zeng, and Wei-shi Zheng. Predicting future instance segmentation with contextual pyramid convlstms. In *Proceedings of the ACM International Conference on Multimedia*, page 2043–2051, 2019.

[35] Adam Terwilliger, Garrick Brazil, and Xiaoming Liu. Recurrent flow-guided semantic forecasting. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1703–1712, 2019.

[36] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations*, 2017.

[37] Suhani Vora, Reza Mahjourian, Soeren Pirk, and Anelia Angelova. Future semantic segmentation using 3d structure. 2018.

[38] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[39] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations*, 2018.

[40] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016.

[41] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[42] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6230–6239, 2017.

[44] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 593–602, 2019.