Predictive Feature Learning for Future Segmentation Prediction

Zihang Lin, Jiangxin Sun, Jian-Fang Hu, Qizhi Yu, Jian-Huang Lai, Wei-Shi Zheng

Sun Yat-sen University, China, Zhejiang Laboratory, China, Pengcheng Laboratory, China, Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China, 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.

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 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-
commonly existed in previous future segmentation pre-

ting reliable future prediction. It is a critical weakness
learning discriminative segmentation features and learn-
marized as: 1) we point out the contradiction between
a considerable margin.

dition prediction and outperforms all counterparts with
achieves new state-of-the-art performances on future seg-
predictor. Combining the above designs, the proposed method
we further introduce a reconstruction constraint in the pre-
the suppressed details during recurrent feature prediction,
predictive feature. In order to reduce the risk of vanishing
the encoder and reconstruct the segmentation feature from
feature more predictive. The decoder is correspondingly
developed to recover the detailed information suppressed in
the low-uncertainty global information and suppress the
blocks. The rescaling operations are designed to enhance
the encoder and reconstructions together with rescaling
spatio-temporal correlation between neighboring frames.

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. Specif-
ically, we develop an encoder-decoder (form an autoen-
coder) which serves as the bridge between the segmenta-
tion feature and the predictive feature. And the prediction
is performed in the latent space of the autoencoder. The en-
coder consists of some convolutions together with rescaling
features for future segmentation prediction. Extensive experi-
ments 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.

2. Related Work

2.1. Future Segmentation Prediction

Future segmentation prediction aims to predict the seg-
mantation 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 seg-
mentation. 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 ConvL-
STM [27] or attention module [5] and combining multi-
modal 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 rep-
 resentation. Saric et al. [30] enhanced the feature predic-
tion with flow-based forecasting and explicitly modeling the
spatio-temporal correlation between neighboring frames.

For instance segmentation prediction, F2F [21] em-
ployed 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 segmenta-
tion 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 dis-
criminative segmentation feature for each pixel. The recent
approaches for learning better segmentation features can be

1Experiment details are provided in the supplementary material.
coarsely divided into two categories: learning context to enhance local discrimination and learning high-resolution representations. For context learning, there are two mainstreams. PSPNet [33], 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 encodes 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...
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 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 $h \times w$ and the target resolution $\frac{h}{2^l} \times \frac{w}{2^l}$ where $h$ and $w$ are the height and width of the input feature, respectively. The rescaling block can be formulated as:

$$Rb_{l \to k} = g(Rs_{l \to k}())$$ \hspace{1cm} (1)

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

$$Rs_{l \to k}() = \begin{cases} f_c \circ f_c \circ \cdots \circ f_c(\cdot), & k > l \\ \text{Identity}, & k = l \\ f_d \circ f_d \circ \cdots \circ f_d(\cdot), & k < l \end{cases}$$ \hspace{1cm} (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 $3 \times 3$.

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 $3 \times 3$ 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 $\mathcal{F} = \{P^1_t, P^2_t, ..., P^L_t\}$ ($L$ scales in total) as the pyramid feature of frame $t$ generated by the feature extractor, and denote $\tilde{\mathcal{F}} = D[E(\mathcal{F})]$ as the multi-scale features of frame $t$ produced by the decoder. The encoder and the decoder are trained with the following reconstruction loss:

$$L_{rec} = \sum_{i=1}^{L} \sum_{t=1}^{T} \|P^i_t - Q^i_t\|_F^2,$$ \hspace{1cm} (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 a 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,
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:

\[
\hat{R}_t = R_{t-1} + \Theta_f([R_1, R_2, ..., R_{t-1}]),
\]

\[
\hat{R}^b_t = \hat{R}_{t+1} + \Theta_b([\hat{R}_T, \hat{R}_{T-1}, ..., \hat{R}_{t+1}]),
\]

where \(R_t\) is the input feature representation for frame \(t\), \(\hat{R}_t\) is the feature of frame \(t\) predicted by the forward ConvLSTM, and \(\hat{R}^b_t\) is the output of backward ConvLSTM. Mappings \(\Theta_f\) and \(\Theta_b\) 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:

\[
L_{pred} = \sum_{t=2}^{T} \|R_t - \hat{R}_t\|^2_F + \sum_{t=1}^{T} \|R_t - \hat{R}^b_t\|^2_F,
\]

where the first term is employed to measure the prediction loss corresponding to forward ConvLSTM, and the second term is a reconstruction loss, which constrains 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:

\[
L = \lambda_{rec}L_{rec} + \lambda_{pred}L_{pred} + \lambda_{seg}L_{seg},
\]

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...
batch size is set to 8. In the first training stage, we trained the SGD algorithm with a Nesterov momentum of 0.9. The train our approaches using the stochastic gradient descent (SGD) algorithm with a ResNet-50-FPN backbone as our segmentation model. We dict segmentation features, by a considerable margin for upper bound an of our system. The comparison results of our system.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Short-term</th>
<th>Mid-term</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP50 AP</td>
<td>AP50 AP</td>
</tr>
<tr>
<td>Mask R-CNN [11] oracle</td>
<td>74.2 30.9</td>
<td>74.2 30.9</td>
</tr>
<tr>
<td>Copy-last segmentation</td>
<td>30.5 16.1</td>
<td>17.3 7.6</td>
</tr>
<tr>
<td>Optical flow - F2F [21]</td>
<td>43.9 20.7</td>
<td>25.8 12.1</td>
</tr>
<tr>
<td>CPCConvLSTM [34]</td>
<td>49.6 24.2</td>
<td>32.4 15.9</td>
</tr>
<tr>
<td>APANet† [13]</td>
<td>52.0 25.7</td>
<td>35.5 18.1</td>
</tr>
<tr>
<td>Ours</td>
<td>52.9 26.3</td>
<td>36.1 18.4</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Short-term</th>
<th>Mid-term</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALL MO</td>
<td>ALL MO</td>
</tr>
<tr>
<td>Semantic FPN [17] Oracle</td>
<td>75.9 75.1</td>
<td>75.9 75.1</td>
</tr>
<tr>
<td>Copy-last segmentation</td>
<td>53.5 48.5</td>
<td>38.9 29.8</td>
</tr>
<tr>
<td>3Dconv-F2F [7]</td>
<td>57.0 40.8</td>
<td></td>
</tr>
<tr>
<td>Dil10-S2S [22]</td>
<td>59.4 47.8</td>
<td>40.8</td>
</tr>
<tr>
<td>F2F [21]</td>
<td>61.2 41.2</td>
<td></td>
</tr>
<tr>
<td>FeatReproj3D [37]</td>
<td>61.5 45.4</td>
<td></td>
</tr>
<tr>
<td>Bayesian S2S [1]</td>
<td>65.1 51.2</td>
<td></td>
</tr>
<tr>
<td>DeformF2F [29]</td>
<td>65.5 53.6</td>
<td>49.9</td>
</tr>
<tr>
<td>LSTM AM S2S [5]</td>
<td>65.8 / 51.3/</td>
<td></td>
</tr>
<tr>
<td>APANet [13]</td>
<td>64.9 / 51.4/</td>
<td></td>
</tr>
<tr>
<td>LSTM M2M [35]</td>
<td>67.1 51.5</td>
<td>46.3</td>
</tr>
<tr>
<td>F2MF [30]</td>
<td>69.6 57.7</td>
<td>54.6</td>
</tr>
<tr>
<td>APANet‡ [13]</td>
<td>71.1 69.2</td>
<td>60.3 56.7</td>
</tr>
</tbody>
</table>

The results demonstrate that the proposed predictive feature auto-encoder can 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 sub-sampled 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 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], CPCConvLSTM [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.
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Short-term</th>
<th></th>
<th>Mid-term</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP50</td>
<td>AP</td>
<td>AP50</td>
<td>AP</td>
</tr>
<tr>
<td>Ours</td>
<td>48.7</td>
<td>24.9</td>
<td>30.5</td>
<td>14.8</td>
</tr>
<tr>
<td>w/o Multi-scale Fusion</td>
<td>47.5</td>
<td>24.1</td>
<td>28.8</td>
<td>13.6</td>
</tr>
<tr>
<td>w/o Rescaling</td>
<td>45.7</td>
<td>23.0</td>
<td>26.4</td>
<td>11.7</td>
</tr>
<tr>
<td>w/o Autoencoder</td>
<td>44.2</td>
<td>22.1</td>
<td>24.7</td>
<td>10.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Short-term</th>
<th></th>
<th>Mid-term</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP50</td>
<td>AP</td>
<td>AP50</td>
<td>AP</td>
</tr>
<tr>
<td>Ours</td>
<td>48.7</td>
<td>24.9</td>
<td>30.5</td>
<td>14.8</td>
</tr>
<tr>
<td>w/o Reconstruction Loss</td>
<td>47.9</td>
<td>24.1</td>
<td>29.4</td>
<td>13.6</td>
</tr>
<tr>
<td>w/o Residual Prediction</td>
<td>47.0</td>
<td>23.4</td>
<td>28.5</td>
<td>13.2</td>
</tr>
</tbody>
</table>

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 $3 \times 3$ 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.

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 $h \times w$. 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-resolution 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


