Predicting Future Instance Segmentation by Forecasting Convolutional Features

Pauline Luc[1,2(✉)], Camille Couprie[1], Yann LeCun[3,4], and Jakob Verbeek[2]

[1] Facebook AI Research, Paris, France
{paulineluc,coupriec}@fb.com
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), LJK, 38000 Grenoble, France
jakob.verbeek@inria.fr
[3] New York University, New York, USA
[4] Facebook AI Research, New York, USA
yann@fb.com

Abstract. Anticipating future events is an important prerequisite towards intelligent behavior. Video forecasting has been studied as a proxy task towards this goal. Recent work has shown that to predict semantic segmentation of future frames, forecasting at the semantic level is more effective than forecasting RGB frames and then segmenting these. In this paper we consider the more challenging problem of future instance segmentation, which additionally segments out individual objects. To deal with a varying number of output labels per image, we develop a predictive model in the space of fixed-sized convolutional features of the Mask R-CNN instance segmentation model. We apply the "detection head" of Mask R-CNN on the predicted features to produce the instance segmentation of future frames. Experiments show that this approach significantly improves over strong baselines based on optical flow and repurposed instance segmentation architectures.

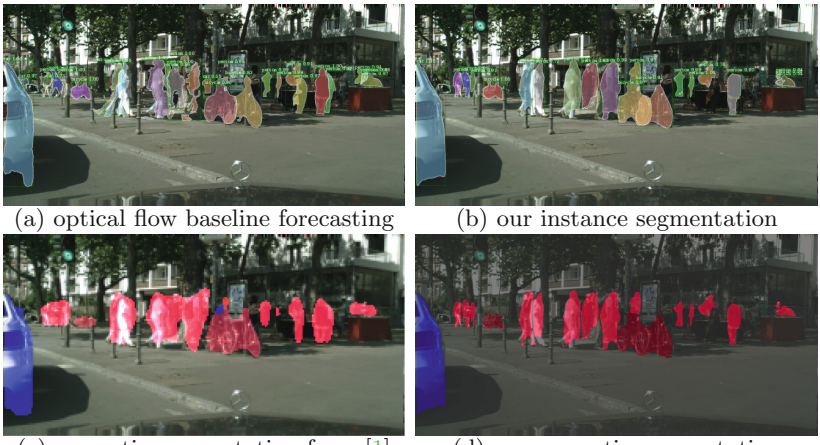
Keywords: Video prediction · Instance segmentation
Deep learning · Convolutional neural networks

1 Introduction

The ability to anticipate future events is a key factor towards developing intelligent behavior [2]. Video prediction has been studied as a proxy task towards pursuing this ability, which can capitalize on the huge amount of available unlabeled video to learn visual representations that account for object interactions and interactions between objects and the environment [3]. Most work in video prediction has focused on predicting the RGB values of future video frames [3–6].

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01240-3_36) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11213, pp. 593–608, 2018.
https://doi.org/10.1007/978-3-030-01240-3_36



(a) optical flow baseline forecasting (b) our instance segmentation

(c) semantic segmentation from [1] (d) our semantic segmentation

Fig. 1. Predicting 0.5 s. into the future. Instance modeling significantly improves the segmentation accuracy of the individual pedestrians.

Predictive models have important applications in decision-making contexts, such as autonomous driving, where rapid control decisions can be of vital importance [7,8]. In such contexts, however, the goal is not to predict the raw RGB values of future video frames, but to make predictions about future video frames at a semantically meaningful level, *e.g.* in terms of presence and location of object categories in a scene. Luc *et al.* [1] recently showed that for prediction of future semantic segmentation, modeling at the semantic level is much more effective than predicting raw RGB values of future frames, and then feeding these to a semantic segmentation model.

Although spatially detailed, semantic segmentation does not account for individual objects, but rather lumps them together by assigning them to the same category label, *e.g.* the pedestrians in Fig. 1(c). Instance segmentation overcomes this shortcoming by additionally associating with each pixel an instance label, as show in Fig. 1(b). This additional level of detail is crucial for down-stream tasks that rely on instance-level trajectories, such as encountered in control for autonomous driving. Moreover, ignoring the notion of object instances prohibits by construction any reasoning about object motion, deformation, *etc.* Including it in the model can therefore greatly improve its predictive performance, by keeping track of individual object properties, *c.f.* Fig. 1(c) and (d).

Since the instance labels vary in number across frames, and do not have a consistent interpretation across videos, the approach of Luc *et al.* [1] does not apply to this task. Instead, we build upon Mask R-CNN [9], a recent state-of-the-art instance segmentation model that extends an object detection system by predicting with each object bounding box a binary segmentation mask of the object. In order to forecast the instance-level labels in a coherent manner, we predict the fixed-sized high level convolutional features used by Mask R-CNN.

We obtain the future object instance segmentation by applying the Mask R-CNN "detection head" to the predicted features.

Our approach offers several advantages: (i) we handle cases in which the model output has a variable size, as in object detection and instance segmentation, (ii) we do not require labeled video sequences for training, as the intermediate CNN feature maps can be computed directly from unlabeled data, and (iii) we support models that are able to produce multiple scene interpretations, such as surface normals, object bounding boxes, and human part labels [10], without having to design appropriate encoders and loss functions for all these tasks to drive the future prediction. Our contributions are the following:

- the introduction of the new task of future instance segmentation, which is semantically richer than previously studied anticipated recognition tasks,
- a self-supervised approach based on predicting high dimensional CNN features of future frames, which can support many anticipated recognition tasks,
- experimental results that show that our feature learning approach improves over strong baselines, relying on optical flow and repurposed instance segmentation architectures.

2 Related Work

Future Video Prediction. Predictive modeling of future RGB video frames has recently been studied using a variety of techniques, including autoregressive models [6], adversarial training [3], and recurrent networks [4,5,11]. Villegas *et al*. [12] predict future human poses as a proxy to guide the prediction of future RGB video frames. Instead of predicting RGB values, Walker *et al*. [13] predict future pixel trajectories from static images.

Future prediction of more abstract representations has been considered in a variety of contexts in the past. Lan *et al*. [14] predict future human actions from automatically detected atomic actions. Kitani *et al*. [15] predict future trajectories of people from semantic segmentation of an observed video frame, modeling potential destinations and transitory areas that are preferred or avoided. Lee *et al*. predict future object trajectories from past object tracks and object interactions [16]. Dosovitskiy and Koltun [17] learn control models by predicting future high-level measurements in which the goal of an agent can be expressed from past video frames and measurements.

Vondrick *et al*. [18] were the first to predict high level CNN features of future video frames to anticipate actions and object appearances in video. Their work is similar in spirit to ours, but while they only predict image-level labels, we consider the more complex task of predicting future instance segmentation, requiring fine spatial detail. To this end, we forecast spatially dense convolutional features, where Vondrick *et al*. were predicting the activations of much more compact fully connected CNN layers. Our work demonstrates the scalability of CNN feature prediction, from 4K-dimensional to 32M-dimensional features, and yields results with a surprising level of accuracy and spatial detail.

Luc *et al*. [1] predicted future semantic segmentation in video by taking the softmax pre-activations of past frames as input, and predicting the softmax pre-activations of future frames. While their approach is relevant for future semantic segmentation, where the softmax pre-activations provide a natural fixed-sized representation, it does not extend to instance segmentation since the instance-level labels vary in number between frames and are not consistent across video sequences. To overcome this limitation, we develop predictive models for fixed-sized convolutional features, instead of making predictions directly in the label space. Our feature-based approach has many advantages over [1]: segmenting individual instances, working at a higher resolution and providing a framework that generalizes to other dense prediction tasks. In a direction orthogonal to our work, Jin *et al*. [19] jointly predict semantic segmentation and optical flow of future frames, leveraging the complementarity between the two tasks.

Instance Segmentation Approaches. Our approach can be used in conjunction with any deep network to perform instance segmentation. A variety of approaches for instance segmentation has been explored in the past, including iterative object segmentation using recurrent networks [20], watershed transformation [21], and object proposals [22]. In our work we build upon Mask R-CNN [9], which recently established a new state-of-the-art for instance segmentation. This method extends the Faster R-CNN object detector [23] by adding a network branch to predict segmentation masks and extracting features for prediction in a way that allows precise alignment of the masks when they are stitched together to form the final output.

3 Predicting Features for Future Instance Segmentation

In this section we briefly review the Mask R-CNN instance segmentation framework, and then present how we can use it for anticipated recognition by predicting internal CNN features of future frames.

3.1 Instance Segmentation with Mask R-CNN

The Mask R-CNN model [9] consists of three main stages. First, a convolutional neural network (CNN) "backbone" architecture is used to extract high level feature maps. Second, a region proposal network (RPN) takes these features to produce regions of interest (ROIs), in the form of coordinates of bounding boxes susceptible of containing instances. The bounding box proposals are used as input to a *RoIAlign* layer, which interpolates the high level features in each bounding box to extract a fixed-sized representation for each box. Third, the features of each RoI are input to the detection branches, which produce refined bounding box coordinates, a class prediction, and a fixed-sized binary mask for the predicted class. Finally, the mask is interpolated back to full image resolution within the predicted bounding box and reported as an instance segmentation for the predicted class. We refer to the combination of the second and third stages as the "detection head".

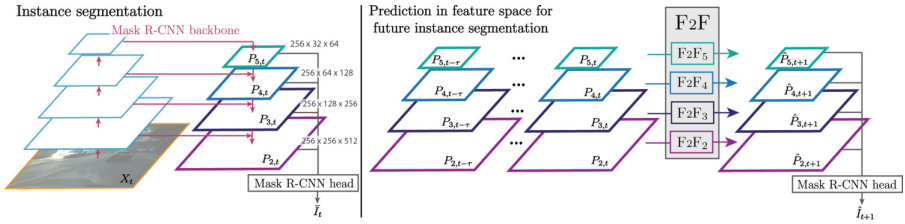


Fig. 2. Left: Features in the FPN backbone are obtained by upsampling features in the top-down path, and combining them with features from the bottom-up path at the same resolution. Right: For future instance segmentation, we extract FPN features from frames $t-\tau$ to t, and predict the FPN features for frame $t+1$. We learn separate feature-to-feature prediction models for each FPN level: F2F_l denotes the model for level l.

He *et al*. [9] use a Feature Pyramid Network (FPN) [24] as backbone architecture, which extracts a set of features at several spatial resolutions from an input image. The feature pyramid is then used in the instance segmentation pipeline to detect objects at multiple scales, by running the detection head on each level of the pyramid. Following [24], we denote the feature pyramid levels extracted from an RGB image X by P_2 through P_5, which are of decreasing resolution $(H/2^l \times W/2^l)$ for P_l, where H and W are respectively the height and width of X. The features in P_l are computed in a top-down stream by up-sampling those in P_{l+1} and adding the result of a 1×1 convolution of features in a layer with matching resolution in a bottom-up ResNet stream. We refer the reader to the left panel of Fig. 2 for a schematic illustration, and to [9,24] for more details.

3.2 Forecasting Convolutional Features

Given a video sequence, our goal is to predict instance-level object segmentations for one or more future frames, *i.e*. for frames where we cannot access the RGB pixel values. Similar to previous work that predicts future RGB frames [3–6] and future semantic segmentations [1], we are interested in models where the input and output of the predictive model live in the same space, so that the model can be applied recursively to produce predictions for more than one frame ahead. The instance segmentations themselves, however, do not provide a suitable representation for prediction, since the instance-level labels vary in number between frames, and are not consistent across video sequences. To overcome this issue, we instead resort to predicting the highest level features in the Mask R-CNN architecture that are of fixed size. In particular, using the FPN backbone in Mask R-CNN, we want to learn a model that given the feature pyramids extracted from frames $X_{t-\tau}$ to X_t, predicts the feature pyramid for the unobserved RGB frame X_{t+1}.

Architecture. The features at the different FPN levels are trained to be input to a shared detection head, and are thus of similar nature. However, since the

resolution changes across levels, the spatio-temporal dynamics are distinct from one level to another. Therefore, we propose a multi-scale approach, employing a separate network to predict the features at each level, of which we demonstrate the benefits in Sect. 4.1. The per-level networks are trained and function completely independently from each other. This allows us to parallelize the training across multiple GPUs. Alternative architectures in which prediction across different resolutions is tied are interesting, but beyond the scope of this paper. For each level, we concatenate the features of the input sequence along the feature dimension. We refer to the "feature to feature" predictive model for level l as F2F_l. The overall architecture is summarized in the right panel of Fig. 2.

Each of the F2F_l networks is implemented by a resolution-preserving CNN. Each network is itself multi-scale as in [1,3], to efficiently enlarge the field of view while preserving high-resolution details. More precisely, for a given level l, F2F_l consists of s_l subnetworks F2F_l^s, where $s \in \{1, ..., s_l\}$. The network $\text{F2F}_l^{s_l}$ first processes the input downsampled by a factor of 2^{s_l-1}. Its output is upsampled by a factor of 2, and concatenated to the input downsampled by a factor of 2^{s_l-2}. This concatenation constitutes the input of $\text{F2F}_l^{s_l-1}$ which predicts a refinement of the initial coarse prediction. The same procedure is repeated until the final scale subnetwork F2F_l^1. The design of subnetworks F2F_l^s is inspired by [1], leveraging dilated convolutions to further enlarge the field of view. Our architecture differs in the number of feature maps per layer, the convolution kernel sizes and dilation parameters, to make it more suited for the larger input dimension. We detail these design choices in the supplementary material.

Training. We first train the F2F_5 model to predict the coarsest features P_5, precomputed offline. Since the features of the different FPN levels are fed to the same recognition head network, the next levels are similar to the P_5 features. Hence, we initialize the weights of F2F_4, F2F_3, and F2F_2 with the ones learned by F2F_5, before fine-tuning them. For this, we compute features on the fly, due to memory constraints. Each of the F2F_l networks is trained using an ℓ_2 loss.

For multiple time step prediction, we can fine-tune each subnetwork F2F_l autoregressively using backpropagation through time, similar to [1] to take into account error accumulation over time. In this case, given a single sequence of input feature maps, we train with a separate ℓ_2 loss on each predicted future frame. In our experiments, all models are trained in this autoregressive manner, unless specified otherwise.

4 Experimental Evaluation

In this section we first present our experimental setup and baseline models, and then proceed with quantitative and qualitative results, that demonstrate the strengths of our F2F approach.

4.1 Experimental Setup: Dataset and Evaluation Metrics

Dataset. In our experiments, we use the Cityscapes dataset [25] which contains 2,975 train, 500 validation and 1,525 test video sequences of 1.8 s each, recorded

from a car driving in urban environments. Each sequence consists of 30 frames of resolution 1024×2048. Ground truth semantic and instance segmentation annotations are available for the 20-th frame of each sequence.

We employ a Mask R-CNN model pre-trained on the MS-COCO dataset [26] and fine-tune it in an end-to-end fashion on the Cityscapes dataset, using a ResNet-50-FPN backbone. The coarsest FPN level P5 has resolution 32×64, and the finest level P2 has resolution 256×512.

Following [1], we temporally subsample the videos by a factor three, and take four frames as input. That is, the input sequence consists of feature pyramids for frames $\{X_{t-9}, X_{t-6}, X_{t-3}, X_t\}$. We denote by *short-term* and *mid-term* prediction respectively predicting X_{t+3} only (0.17 s) and through X_{t+9} (0.5 s). We additionally evaluate *long-term* predictions, corresponding to X_{t+27} and 1.6 s ahead on the two long Frankfurt sequences of the Cityscapes validation set.

Conversion to Semantic Segmentation. For direct comparison to previous work, we also convert our instance segmentation predictions to semantic segmentation. To this end, we first assign to all pixels the *background* label. Then, we iterate over the detected object instances in order of ascending confidence score. For each instance, consisting of a confidence score c, a class k, and a binary mask m, we either reject it if it is lower than a threshold θ and accept it otherwise, where in our experiments we set $\theta = 0.5$. For accepted instances, we update the spatial positions corresponding to mask m with label k. This step potentially replaces labels set by instances with lower confidence, and resolves competing class predictions.

Evaluation Metrics. To measure the instance segmentation performance, we use the standard Cityscapes metrics. The average precision metric AP50 counts an instance as correct if it has at least 50% of intersection-over-union (IoU) with the ground truth instance it has been matched with. The summary AP metric is given by average AP obtained with ten equally spaced IoU thresholds from 50% to 95%. Performance is measured across the eight classes with available instance-level ground truth: *person, rider, car, truck, bus, train, motorcycle,* and *bicycle*.

We measure semantic segmentation performance across the same eight classes. In addition to the IoU metric, computed w.r.t. the ground truth segmentation of the 20-th frame in each sequence, we also quantify the segmentation accuracy using three standard segmentation measures used in [27], namely the Probabilistic Rand Index (RI) [28], Global Consistency Error (GCE) [29], and Variation of Information (VoI) [30]. Good segmentation results are associated with high RI, low GCE and low VoI.

Implementation Details and Ablation Study. We cross-validate the number of scales, the optimization algorithm and hyperparameters per level of the pyramid. For each level of the pyramid a single scale network was selected, except for F_2F_2, where we employ 3 scales. The F_2F_5 network is trained for 60 K iterations of SGD with Nesterov Momentum of 0.9, learning rate 0.01, and batch size of 4 images. It is used to initialize the other networks, which are trained for 80 K iterations of SGD with Nesterov Momentum of 0.9, batch size of 1 image

Table 1. Ablation study: short-term prediction on the Cityscapes val. set.

Levels	P_5	P_4–P_5	P_3–P_5	P_2–P_5	P_5 //
IoU	15.5	38.5	54.7	**60.7**	38.7
AP50	2.2	10.2	24.8	**40.2**	16.7

and learning rates of 5×10^{-3} for F_2F_4 and 0.01 for F_2F_3. For F_2F_2, which is much deeper, we used Adam with learning rate 5×10^{-5} and default parameters. Table 1 shows the positive impact of using each additional feature level, denoted by $P_i - P_5$ for $i = 2, 3, 4$. We also report performance when using all features levels, predicted by a model trained on the coarsest P_5 features, shared across levels, denoted by P_5 //. The drop in performance w.r.t. the column P_2–P_5 underlines the importance of training specific networks for each feature level.

4.2 Baseline Models

As a performance upper bound, we report the accuracy of a Mask R-CNN oracle that has access to the future RGB frame. As a lower bound, we also use a trivial copy baseline that returns the segmentation of the last input RGB frame. Besides the following baselines, we also experiment with two weaker baselines, based on nearest neighbor search and on predicting the future RGB frames, and then segmenting them. We detail both baselines in the supplementary material.

Optical Flow Baselines. We designed two baselines using the optical flow field computed from the last input RGB frame to the second last, as well as the instance segmentation predicted at the last input frame. The *Warp* approach consists in warping each instance mask independently using the flow field inside this mask. We initialize a separate flow field for each instance, equal to the flow field inside the instance mask and zero elsewhere. For a given instance, the corresponding flow field is used to project the values of the instance mask in the opposite direction of the flow vectors, yielding a new binary mask. To this predicted mask, we associate the class and confidence score of the input instance it was obtained from. To predict more than one time-step ahead, we also update the instance's flow field in the same fashion, to take into account the previously predicted displacement of physical points composing the instance. The predicted mask and flow field are used to make the next prediction, and so on. Maintaining separate flow fields allows competing flow values to coexist for the same spatial position, when they belong to different instances whose predicted trajectories lead them to overlap. To smoothen the results of this baseline, we perform post-processing operations at each time step, which significantly improve the results and which we detail in the supplementary material.

Warping the flow field when predicting multiple steps ahead suffers from error accumulation. To avoid this, we test another baseline, *Shift*, which shifts each mask with the average flow vector computed across the mask. To predict T time steps ahead, we simply shift the instance T times. This approach, however, is

Table 2. Instance segmentation accuracy 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*	37.0	16.0	9.7	2.9
Optical flow – *Warp*	36.8	16.5	11.1	4.1
Mask H2F*	25.5	11.8	14.2	5.1
F2F w/o ar. fine tuning	**40.2**	19.0	17.5	6.2
F2F	39.9	**19.4**	**19.4**	**7.7**

* Separate models were trained for short-term and mid-term predictions.

unable to scale the objects, and is therefore unsuitable for long-term prediction when objects significantly change in scale as their distance to the camera changes.

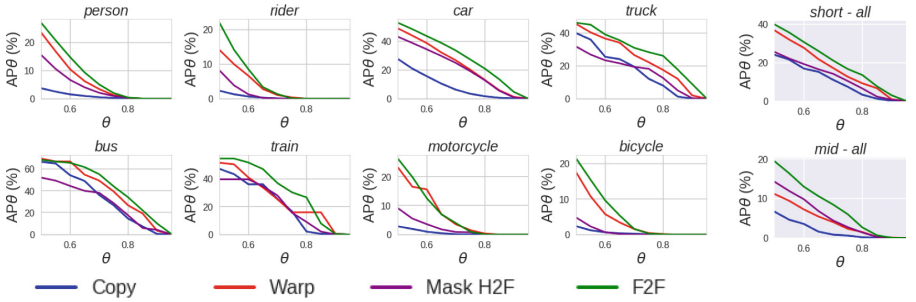
Future Semantic Segmentation Using Discrete Label Maps. For comparison with the future semantic segmentation approach of [1], which ignores instance-level labels, we train their S2S model on the label maps produced by Mask R-CNN. Following their approach, we down-sample the Mask R-CNN label maps to 128×256. Unlike the soft label maps from the Dilated-10 network [31] used in [1], our converted Mask R-CNN label maps are discrete. For autoregressive prediction, we discretize the output by replacing the softmax network output with a one-hot encoding of the most likely class at each position. For autoregressive fine-tuning, we use a softmax activation with a low temperature parameter at the output of the S2S model, to produce near-one-hot probability maps in a differentiable way, enabling backpropagation through time.

Future Segmentation Using the Mask R-CNN Architecture. As another baseline, we fine-tune Mask R-CNN to predict mid-term future segmentation given the last 4 observed frames, denoted as the Mask H2F baseline. As initialization, we replicate the weights of the first layer learned on the COCO dataset across the 4 frames, and divide them by 4 to keep the features at the same scale.

4.3 Quantitative Results

Future Instance Segmentation. In Table 2 we present instance segmentation results of our future feature prediction approach (F2F) and compare it to the performance of the oracle, copy, optical flow and Mask H2F baselines. The copy baseline performs very poorly (24.1% in terms of AP50 *vs*. 65.8% for the oracle), which underlines the difficulty of the task. The two optical flow baselines perform comparably for short-term prediction, and are both much better than the copy baseline. For mid-term prediction, the *Warp* approach outperforms *Shift*. The Mask H2F baseline performs poorly for short-term prediction, but its results

degrade slower with the number of time steps predicted, and it outperforms the *Warp* baseline for mid-term prediction. As Mask H2Foutputs a single time step prediction, either for short or mid-term predictions, it is not subject to accumulation of errors, but each prediction setting requires training a specific model. Our F2F approach gives the best results overall, reaching more than 37% of relative improvement over our best mid-term baseline. While our F2F autoregressive fine-tuning makes little difference in case of short-term prediction (40.2% *vs*. 39.9% AP50 respectively), it gives a significant improvement for mid-term prediction (17.5% *vs*. 19.4% AP50 respectively).



(a) Short-term - Individual classes scores (b) Overall

Fig. 3. Instance segmentation APθ across different IoU thresholds θ. (a) Short-term prediction per class; (b) Average across all classes for short-term (top) and mid-term prediction (bottom).

In Fig. 3(a), we show how the AP metric varies with the IoU threshold, for short-term prediction across the different classes and for each method. For individual classes, F2F gives the best results across thresholds, except for very few exceptions. In Fig. 3(b), we show average results over all classes for short-term and mid-term prediction. We see that F2F consistently improves over the baselines across all thresholds, particularly for mid-term prediction.

Future Semantic Segmentation. We additionally provide a comparative evaluation on semantic segmentation in Table 3. First, we observe that our discrete implementation of the S2S model performs slightly better than the best results obtained by [1], thanks to our better underlying segmentation model (Mask R-CNN *vs*. the Dilation-10 model [31]). Second, we see that the Mask H2F baseline performs weakly in terms of semantic segmentation metrics for both short and mid-term prediction, especially in terms of IoU. This may be due to frequently duplicated predictions for a given instance, see Sect. 4.4. Third, the advantage of *Warp* over *Shift* appears clearly again, with a 5% boost in mid-term IoU. Finally, we find that F2F obtains clear improvements in IoU over all methods for short-term segmentation, ranking first with an IoU of 61.2%. Our F2F mid-term IoU is comparable to those of the S2S and *Warp* baseline, while being much more accurate in depicting contours of the objects as shown by consistently better RI, VoI and GCE segmentation scores.

Table 3. Short and mid-term semantic segmentation of moving objects (8 classes) performance on the Cityscapes validation set.

	Short-term				Mid-term			
	IoU	RI	VoI	GCE	IoU	RI	VoI	GCE
Oracle [1]	64.7	–	–	–	64.7	–	–	–
S2S [1]	55.3	–	–	–	40.8	–	–	–
Oracle	73.3	94.0	20.8	2.3	73.3	94.0	20.8	2.3
Copy	45.7	92.2	29.0	3.5	29.1	90.6	33.8	4.2
Shift	56.7	92.9	25.5	2.9	36.7	91.1	30.5	3.3
Warp	58.8	**93.1**	25.2	3.0	41.4	91.5	31.0	3.8
Mask H2F*	46.2	92.5	27.3	3.2	30.5	91.2	31.9	3.7
S2S	55.4	92.8	25.8	2.9	**42.4**	91.8	29.7	3.4
F2F	**61.2**	**93.1**	**24.8**	**2.8**	41.2	**91.9**	**28.8**	**3.1**

* Separate models were trained for short-term and mid-term predictions.

4.4 Qualitative Results

Figures 4 and 5 show representative results of our approach, both in terms of instance and semantic segmentation prediction, as well as results from the *Warp* and Mask H2F baselines for instance segmentation and S2S for semantic segmentation. We visualize predictions with a threshold of 0.5 on the confidence of masks. The Mask H2F baseline frequently predicts several masks around objects, especially for objects with ambiguous trajectories, like pedestrians, and less so for more predictable categories like cars. We speculate that this is due to the loss that the network is optimizing, which does not discourage this behavior, and due to which the network is learning to predict several plausible future positions, as long as they overlap sufficiently with the ground-truth position. This does not occur with the other methods, which are either optimizing a per-pixel loss or are not learned at all. F2F results are often better aligned with the actual layouts of the objects than the *Warp* baseline, showing that our approach has learned to model dynamics of the scene and objects more accurately than the baseline. As expected, the predicted masks are also much more precise than those of the S2S model, which is not instance-aware.

In Fig. 6 we provide additional examples to better understand why the difference between F2F and the *Warp* baseline is smaller for semantic segmentation metrics than for instance segmentation metrics. When several instances of the same class are close together, inaccurate estimation of the instance masks may still give acceptable semantic segmentation. This typically happens for groups of pedestrians and rows of parked cars. If an instance mask is split across multiple objects, this will further affect the AP measure than the IoU metric. The same example also illustrates common artifacts of the *Warp* baseline that are due to error accumulation in the propagation of the flow field.

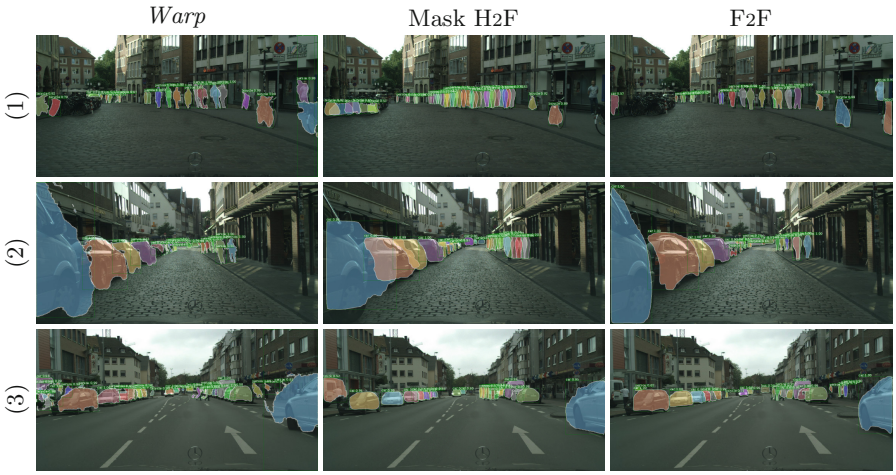


Fig. 4. Mid-term instance segmentation predictions (0.5 s future) for 3 sequences, from left to right: *Warp* baseline, Mask H2F baseline and F2F.

4.5 Discussion

Failure Cases. To illustrate some of the remaining challenges in predicting future instance segmentation we present several failure cases of our F2F model in Fig. 7. In Fig. 7(a), the masks predicted for the truck and the person are incoherent, both in shape and location. More consistent predictions might be obtained with a mechanism for explicitly modeling occlusions. Certain motions and shape transformations are hard to predict accurately due to the inherent ambiguity in the problem. This is, *e.g.*, the case for the legs of pedestrians in Fig. 7(b), for which there is a high degree of uncertainty on the exact pose. Since the model is deterministic, it predicts a rough mask due to averaging over several possibilities. This may be addressed by modeling the intrinsic variability using GANs, VAEs, or autoregressive models [6,32,33].

Long Term Prediction. In Fig. 8 we show a prediction of F2F up to 1.5 s. in the future in a sequence of the long Frankfurt video of the Cityscapes validation set, where frames were extracted with an interval of 3 as before. To allow more temporal consistency between predicted objects, we apply an adapted version of the method of Gkioxari *et al.* [34] as a post-processing step. We define the linking score as the sum of confidence scores of subsequent instances and of their IoU. We then greedily compute the paths between instances which maximize these scores using the Viterbi algorithm. We thereby obtain object tracks along the (unseen) future video frames. Some object trajectories are forecasted reasonably well up to a second, such as the rider, while others are lost by that time such as the motorbike. We also compute the AP with the ground truth of the long Frankfurt video. For each method, we give the best result of either predicting 9 frames with a frame interval of 3, or the opposite. For Mask H2F, only the latter

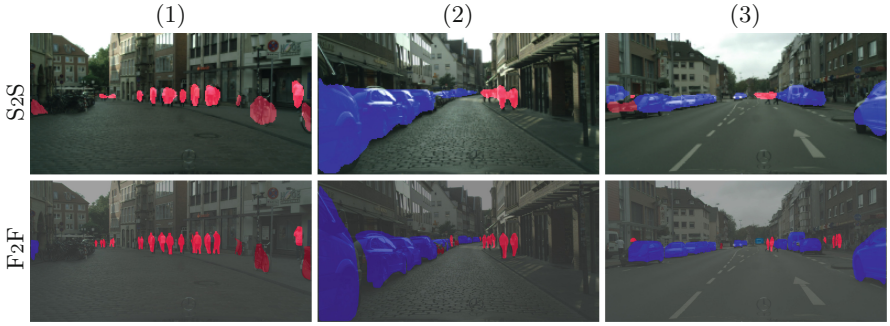


Fig. 5. Mid-term semantic segmentation predictions (0.5 s) for 3 sequences. For each case we show from top to bottom: S2S model and F2F model.

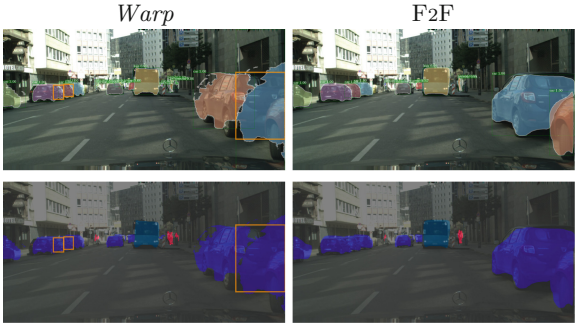


Fig. 6. Mid-term predictions of instance and semantic segmentation with the *Warp* baseline and our F2F model. Inaccurate instance segmentations can result in accurate semantic segmentation areas; see orange rectangle highlights. (Color figure online)

is possible, as there are no such long sequences available for training. We obtain an AP of 0.5 for the flow and copy baseline, 0.7 for F2F and 1.5 for Mask H2F. All methods lead to very low scores, highlighting the severe challenges posed by this problem.

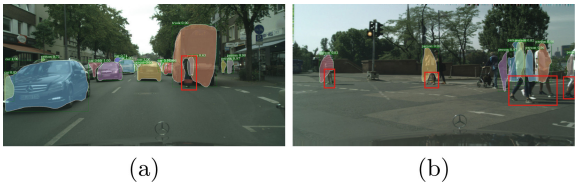


Fig. 7. Failure modes of mid-term prediction with the F2F model, highlighted with the red boxes: incoherent masks (a), lack of detail in highly deformable object regions, such as legs of pedestrians (b). (Color figure online)

Fig. 8. Long-term predictions (1.5 s) from our F2F model.

5 Conclusion

We introduced a new anticipated recognition task: predicting instance segmentation of future video frames. This task is defined at a semantically meaningful level rather the level of raw RGB values, and adds instance-level information as compared to predicting future semantic segmentation. We proposed a generic and self-supervised approach for anticipated recognition based on predicting the convolutional features of future video frames. In our experiments we apply this approach in combination with the Mask R-CNN instance segmentation model. We predict the internal "backbone" features which are of fixed dimension, and apply the "detection head" on these features to produce a variable number of predictions. Our results show that future instance segmentation can be predicted much better than naively copying the segmentations from the last observed frame, and that our future feature prediction approach significantly outperforms two strong baselines, the first one relying on optical-flow-based warping and the second on repurposing and fine-tuning the Mask R-CNN architecture for the task. When evaluated on the more basic task of semantic segmentation without instance-level detail, our approach yields performance quantitatively comparable to earlier approaches, while having qualitative advantages.

Our work shows that with a feed-forward network we are able to obtain surprisingly accurate results. More sophisticated architectures have the potential to further improve performance. Predictions may be also improved by explicitly modeling the temporal consistency of instance segmentation, and predicting multiple possible futures rather than a single one.

We invite the reader to watch videos of our predictions at http://thoth.inrialpes.fr/people/pluc/instpred2018.

Acknowledgment. This work has been partially supported by the grant ANR-16-CE23-0006 "Deep in France" and LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01). We thank Matthijs Douze, Xavier Martin, Ilija Radosavovic and Thomas Lucas for their precious comments.

References

1. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: ICCV (2017)
2. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

3. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
4. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv 1412.6604 (2014)
5. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using LSTMs. In: ICML (2015)
6. Kalchbrenner, N., et al.: Video pixel networks. In: ICML (2017)
7. Shalev-Shwartz, S., Ben-Zrihem, N., Cohen, A., Shashua, A.: Long-term planning by short-term prediction. arXiv 1602.01580 (2016)
8. Shalev-Shwartz, S., Shashua, A.: On the sample complexity of end-to-end training vs. semantic abstraction training. arXiv 1604.06915 (2016)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
10. Kokkinos, I.: UberNet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: CVPR (2017)
11. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. In: ICLR (2017)
12. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: ICML (2017)
13. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: forecasting from static images using variational autoencoders. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 835–851. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_51
14. Lan, T., Chen, T.-C., Savarese, S.: A hierarchical representation for future action prediction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 689–704. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_45
15. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 201–214. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_15
16. Lee, N., Choi, W., Vernaza, P., Choy, C., Torr, P., Chandraker, M.: DESIRE: distant future prediction in dynamic scenes with interacting agents. In: CVPR (2017)
17. Dosovitskiy, A., Koltun, V.: Learning to act by predicting the future. In: ICLR (2017)
18. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating the future by watching unlabeled video. In: CVPR (2016)
19. Jin, X., et al.: Predicting scene parsing and motion dynamics in the future. In: NIPS (2017)
20. Romera-Paredes, B., Torr, P.H.S.: Recurrent instance segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 312–329. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_19
21. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: CVPR (2017)
22. Pinheiro, P.O., Lin, T.-Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 75–91. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_5

23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
24. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
25. Cordts, M., et al.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
26. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
27. Yang, A., Wright, J., Ma, Y., Sastry, S.: Unsupervised segmentation of natural images via lossy data compression. CVIU **110**(2), 212–225 (2008)
28. Parntofaru, C., Hebert, M.: A comparison of image segmentation algorithms. Technical report CMU-RI-TR-05-40, Carnegie Mellon University (2005)
29. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
30. Meilă, M.: Comparing clusterings: An axiomatic view. In: ICML (2005)
31. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
32. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
33. Kingma, D., Welling, M.: Auto-encoding variational Bayes. In: ICLR (2014)
34. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR (2015)