Predicting Future Instance Segmentations by Forecasting Convolutional Features

Pauline Luc, Camille Couprie, Yann Lecun, Jakob Verbeek

Predicting Future Instance Segmentations by Forecasting Convolutional Features

Pauline Luc[1,2] Camille Couprie[1] Yann LeCun[1,3] Jakob Verbeek[2]

[1] Facebook AI Research
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*, LJK, 38000 Grenoble, France
[3] New York University
{paulineluc,coupriec,yann}@fb.com jakob.verbeek@inria.fr

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 baselines based on optical flow.

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 [1]. 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 [2]. Most work in video prediction has focused on predicting the RGB values of future video frames [2,3,4,5].

Predictive models have important applications in decision-making contexts, such as autonomous driving, where rapid control decisions can be of vital importance [6,7]. 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.* [8] 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.

* Institute of Engineering Univ. Grenoble Alpes

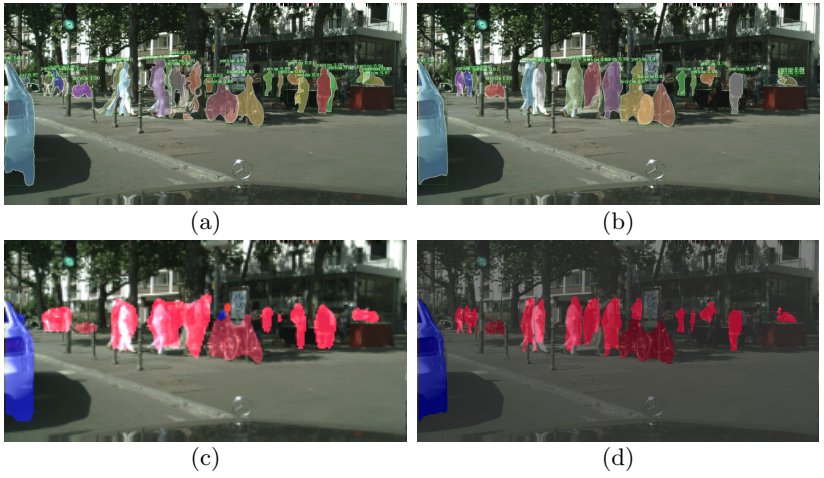


Fig. 1: Predicting 0.5 sec. into the future. Instance segmentations with (a) optical flow baseline and (b) our approach. Semantic segmentation (c) from [8] and (d) derived from our instance semantic segmentation approach. Instance modeling significantly improves the segmentation accuracy of the individual pedestrians.

Although spatially detailed, semantic segmentation does not account for individual objects, but rather lumps them together by assigning them to the same category label. See, *e.g.*, the pedestrians in Figure 1(c). Instance segmentation overcomes this shortcoming by additionally associating with each pixel an instance label, as show in Figure 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.* Figure 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*. [8] 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 associating 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 abstract 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 interme-

diate 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 prediction, 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 optical flow baselines.

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 [5], adversarial training [2], and recurrent networks [3,4,11]. 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 & 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 abstract CNN features of future video frames to anticipate actions and object appearances in video. Their work is similar in spirit to ours, but where they only predict image-level labels, we consider the more complex task of predicting spatially detailed future instance segmentations. To this end, we forecast spatially dense convolutional features, where Vondrick *et al*. were predicting the activations of more compact fully connected CNN layers.

Luc *et al*. [8] predicted future semantic segmentations 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 the case of 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. 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 Segmentations

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 for 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, regardless of its size. 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 the "detection head". The full model is trained end-to-end from images with pre-segmented object instances.

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

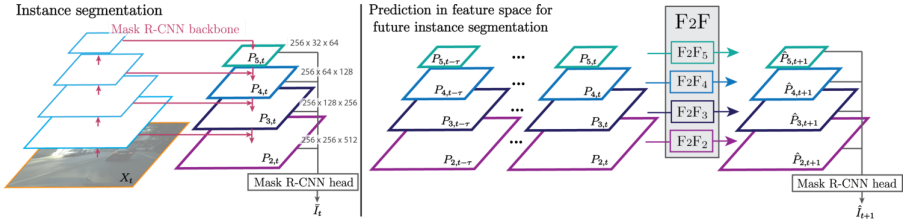


Fig. 2: Left: Features in the FPN backbone are obtained by upsampling features in the top-down path, and combining them with features from the bottom-up path at the same resolution. Right: For future instance segmentation, we extract FPN features from frames $t-\tau$ to t, and predict the FPN features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

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 Figure 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 [2,3,4,5] and future semantic segmentations [8], 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. The per-level networks are trained and function completely independently from each other. 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 Figure 2.

Each of the $\text{F}2\text{F}_l$ networks is implemented by a resolution-preserving CNN, *i.e.* where input and output have the same resolution. Each network is itself multi-scale as in [2,8], to efficiently enlarge the field of view while preserving high-resolution details. More precisely, for a given level l, $\text{F}2\text{F}_l$ consists itself of s_l subnetworks $\text{F}2\text{F}_l^s$, where $s \in \{1, ..., s_l\}$. The network $\text{F}2\text{F}_l^{s_l}$ first processes the input downsampled by a factor 2^{s_l-1}. Its output is up-sampled by a factor two, and concatenated to the input downsampled by a factor 2^{s_l-2}. This concatenation constitutes the input of $\text{F}2\text{F}_l^{s_l-1}$ which predicts a refinement of the initial coarse prediction. The same procedure is repeated until the final scale subnetwork $\text{F}2\text{F}_l^1$.

The design of subnetworks $\text{F}2\text{F}_l^s$ is inspired by the one of [8], 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 compute the coarsest P_5 feature level off-line, and train the $\text{F}2\text{F}_5$ model efficiently from these pre-computed features. Due to memory constraints, we cannot pre-compute and store the features as the higher resolution P_4, P_3 and P_2 levels. However, since the features of the different FPN levels are fed to the same recognition head network, these features are similar to the P_5 ones. Hence, we initialize the weights of $\text{F}2\text{F}_4$, $\text{F}2\text{F}_3$, and $\text{F}2\text{F}_2$ with the ones learned for $\text{F}2\text{F}_5$, and fine-tune them using features computed on the fly. Each of the $\text{F}2\text{F}_l$ networks is trained using an ℓ_2 loss on the predicted feature values.

For multiple time step prediction, we can finetune each subnetwork $\text{F}2\text{F}_l$ autoregressively using back propagation through time, similar to [8]. Unlike the typical "teacher forced" training of recurrent networks, this approach takes into account error accumulation over time. This is possible in our scenario since we predict in a continuous space, rather than in a discrete space as is commonly the case for recurrent networks, *e.g.* for language modeling. In this case, given a single sequence of input feature maps, we train with a separate ℓ_2 loss on all the future frames for which we predict. 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 positive impact 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 second each, recorded from a car driving in urban environments. Each sequence consists of 30 frames of resolution 1024×2048, and complete ground-truth semantic and

instance segmentation for every pixel 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. The coarsest FPN level P5 has resolution 32×64, and the finest level P2 has resolution 256×512.

Following [8], we train our models using a frame interval of three, and taking 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 predicting X_{t+3} as *short-term* and use *mid-term* prediction to denote predicting up to X_{t+9}, corresponding to predicting up to 0.17 sec. and 0.5 sec. respectively.

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 all pixels in the semantic segmentation 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 $c < \theta$ and accept it otherwise, where in our experiments we set $\theta = 0.5$. For accepted instances, we update the semantic segmentation at spatial positions corresponding to mask m with label k. This step potentially replaces labels set by an instance 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 as instance segmentation. 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 (PRI) [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. We cross validate the number of scales, the optimization algorithm, and the parameters per level of the pyramid. This leads to the choice of single scale network for each level of the pyramid, except for F_2F_2, where we employ three scales. The F_2F_5 network is trained for 60K iterations of SGD with Nesterov momentum 0.9, with learning rate 0.01, and batch size of four images. It is used to initialize the other subnetworks, which are trained for 80K iterations of SGD with Nesterov momentum 0.9 with batch size of one image, and learning rates of 5×10^{-3} for F_2F_4 and 0.01 for F_2F_4. For F_2F_2, which is much deeper, we used Adam with learning rate 5×10^{-5} and default parameters.

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. We also use a trivial copy baseline that returns the segmentation of the last input RGB frame.

Optical flow baselines. We designed two baselines based on optical flow field $\mathbf{F}_{t\to t-1}$, from RGB frame t to $t-1$, and the instance segmentation $\mathbf{I}_t$ predicted at frame t. 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 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.

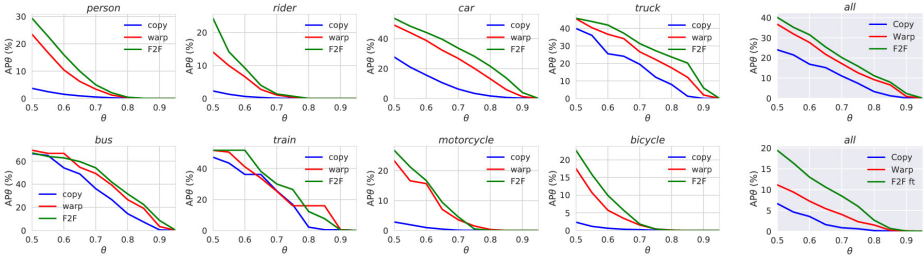
Future semantic segmentation using discrete label maps. For comparison with the future semantic segmentation approach of [8], which ignores instance-level labels, we train their S2S model on the label maps produced by Mask R-CNN. Following their approach, we down-sample the Mask R-CNN label maps to 128×256. Unlike the soft-label maps from the Dilated-10 network [31] used in [8], 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 pixel. 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, which allows us to apply backpropagation through time.

4.3 Quantitative results

Future instance segmentation. In Table 1 we present instance segmentation results of our future feature prediction approach (F2F) and compare it to the per-

	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
F2F w/o ar. fine tuning	**40.2**	19.0	17.5	6.2
F2F	39.9	**19.4**	**19.4**	**7.7**

Table 1: Instance segmentation accuracy on the Cityscapes val. set



(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).

formance of the oracle, copy, and optical flow baselines. From the results we draw several conclusions. First of all, 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 optical flow baselines are much better. While *Shift* and *Warp* perform comparably for short-term prediction (37.0% *vs*. 36.8% AP50 respectively), the *Warp* approach performs best for mid-term prediction (9.7% *vs*. 11.1% AP50 respectively). Our F2F approache gives the best results overall, reaching more than 74% of relative improvement over our best mid term baseline.

While for our F2F autoregressive finetuning 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). For short-term prediction, our F2F improves over the *Warp* baseline by 3.1% AP50, from 36.8% to 39.9%. For long-term prediction the difference is more pronounced: our F2F improves over the *Warp* baseline by 8.3% AP50, from 11.1% to 19.4%.

In Figure 3, 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

	Short term				Mid term			
	IoU	RI	VoI	GCE	IoU	RI	VoI	GCE
Oracle [8]	64.7	—	—	—	64.7	—	—	—
S2S [8]	55.3	—	—	—	40.8	—	—	—
Oracle	73.3	94.0	0.208	2.3	73.3	94.0	0.208	2.3
Copy	45.7	92.2	0.290	3.5	29.1	90.6	0.338	4.2
Shift	56.7	92.9	0.255	2.9	36.7	91.1	0.305	3.3
Warp	58.8	93.1	0.252	3.0	41.4	91.5	0.310	3.8
S2S	55.4	92.8	0.258	2.9	**42.4**	91.8	0.297	3.4
F2F	**61.2**	**93.1**	**0.248**	**2.8**	41.2	**91.9**	**0.288**	**3.1**

Table 2: Short and mid term semantic segmentation of moving objects (8 classes) performance on the Cityscapes val. dataset.

few exceptions. From the last two panels, which average results over all classes for short-term and mid-term prediction, we see that F2F consistently improves over the baselines across all thresholds. The improvements over the optical flow baseline are particularly important for mid-term prediction.

Future semantic segmentations. In addition to future instance prediction abilities, we provide a comparative evaluation on semantic segmentation in Table 2. First, we observe that our discrete implementation of the S2S model performs slightly better than the best results obtained by [8], thanks to our better underlying segmentation model (Mask R-CNN *vs*. the Dilation-10 model [31]). Second, the advantage of the *Warp* baseline over the *Shift* again appears clearly, with a 5 points boost in mid-term IoU. Finally, we find that our F2F obtains clear improvements in IoU over all methods for short-term segmentation, ranking first with an IoU of 61.2%. The *Warp* baseline reaches the second position with an IoU of 58.8%. 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.

4.4 Qualitative results

Figure 4 shows representative results of our approach, similar to the ones in Figure 1, both in terms of instance and semantic segmentation prediction, as well as results from the *Warp* baseline for instance segmentation and the S2S model for semantic segmentation. 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 of objects better than the baseline. As expected, the predicted masks are also much more precise than those of the S2S, which is not instance-aware. This observation is also highlighted by the better VoI, RI and GCE score obtained by our model F2F over S2S.

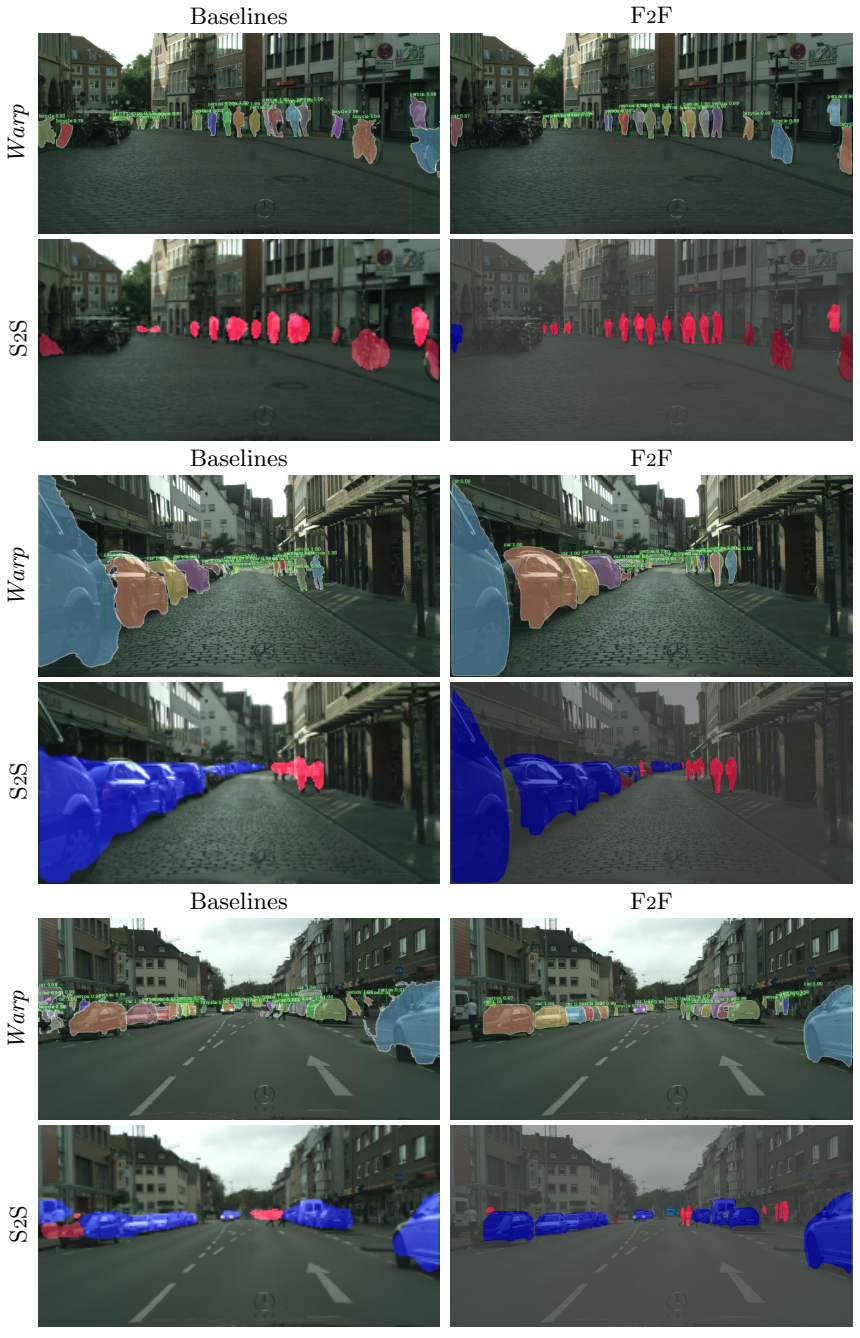


Fig. 4: Mid-term predictions (0.5 sec. future) for three sequences. For each case we show (clock-wise order, from top left): instance segmentations using the *Warp* baseline and F2F, and semantic segmentation using F2F and S2S model.

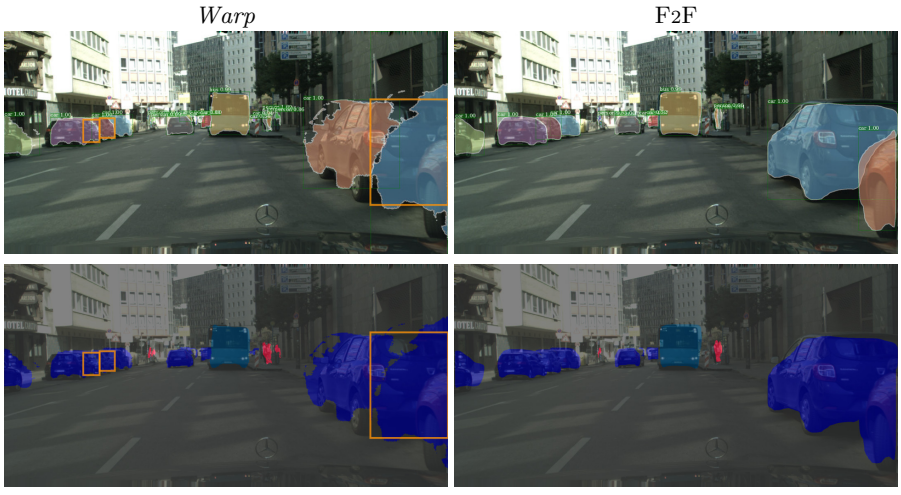


Fig. 5: Instance and semantic segmentations for mid-term prediction obtained with *Warp* baseline and our F2F model. Inaccurate instance segmentations can result in accurate semantic segmentations areas, see orange rectangle highlights.

In Figure 5 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 masks are split across multiple objects, this will affect the AP measure much more than it will affect 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.

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 in Figure 6. In the first example in Figure 6(a), the white car is missed because in all of the preceding frames that the model considers it is entirely occluded by another vehicle. Such cases are unavoidable, unless the object is visible in some earlier frames, in which case long-term memory mechanisms might avoid the error.

In Figure 6(b), 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.

Finally, 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

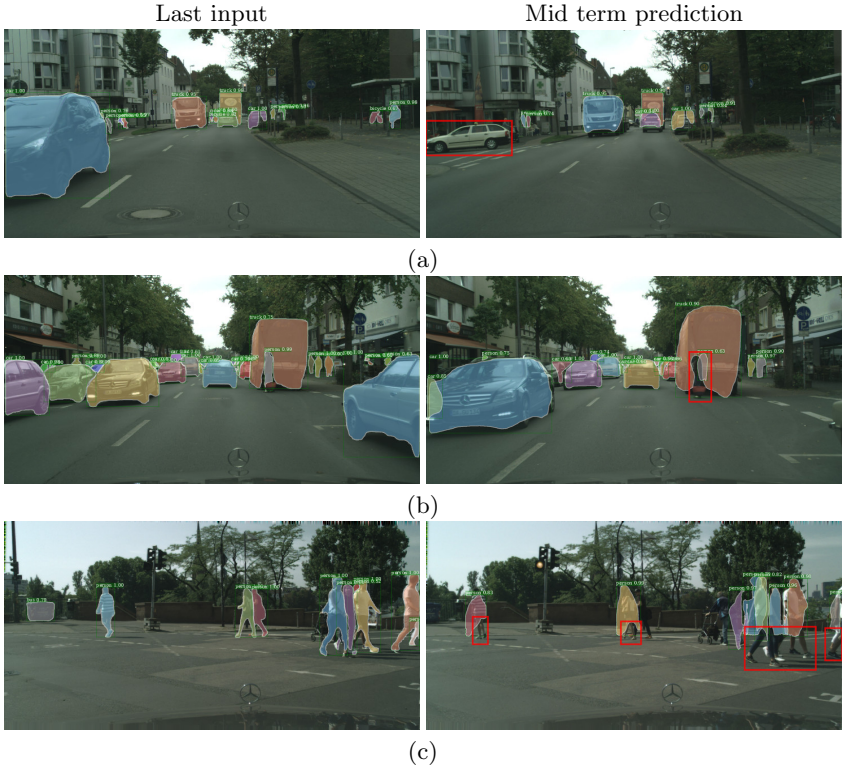


Fig. 6: Failure modes of mid-term prediction with the F2F ft model. Left: last input frame, right: mid term prediction. The red boxes highlight different issues: missed detections (a), incoherent masks (b), lack of detail in highly deformable object regions, such as feet of pedestrians (c).

legs of pedestrians in Figure 6(c), 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 [5,32,33].

Qualitative examples of long term prediction. In Figure 7 we show predictions with F2F up to 1.5 seconds in the future. We show results on two sequences of the long Frankfurt video of the Cityscapes validation set, where frames were extracted with an interval of three 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 postprocessing step. We define the linking score as the sum of confidence scores of subsequent instances $\bar{I}_t$, $\bar{I}_{t+1}$ and of their IoU. We then compute shortest paths between instances using the Viterbi algorithm. Using this post-processing, we obtain object tracks along the (unseen) future video frames. Some object trajectories are forecasted reasonably well up to a

second, such as the red parked car in the first example and the motorcylist in second example, while others are lost by that time such as the motorbike in the second example. A few objects are predicted well upto 1.5 sec. such as the gray car in the center in the first example and the blue car in the second example.

Fig. 7: Long-term predictions (1.5 seconds) from our F2F model.

5 Conclusion

We introduced a new anticipated recognition task: predicting instance segmentations for future video frames. This task is defined at a semantically meaningful level rather the level of raw RGB pixel values, and adds instance-level as compared to predicting future semantic segmentation. We proposed a generic and self-supervised approach for anticipated recognition based on predicting the convolutional features for 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 dimensions, and apply the "detection head" on these features to produce a variable number of instance segmentations. Our results show that future instance segmentations can be predicted much better than naively copying the segmentations from the last observed frame, and that our future feature prediction approach significantly outperforms a strong baseline that warps segmentations based on optical-flow. When evaluated for the more basic task of semantic segmentation without instance-level detail, our approach yields performance quantitatively comparable to earlier approaches, while having qualitative advantages.

While our results are very encouraging, we believe they may be further improved by explicitly modeling the temporal consistency of instance segmentations, 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 and Thomas Lucas for their precious comments.

References

1. Sutton, R., Barto, A.: Reinforcement learning: An introduction. MIT Press (1998)
2. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR. (2016)
3. 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)
4. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using LSTMs. In: ICML. (2015)
5. Kalchbrenner, N., van den Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., Kavukcuoglu, K.: Video pixel networks. In: ICML. (2017)
6. Shalev-Shwartz, S., Ben-Zrihem, N., Cohen, A., Shashua, A.: Long-term planning by short-term prediction. arXiv 1602.01580 (2016)
7. Shalev-Shwartz, S., Shashua, A.: On the sample complexity of end-to-end training vs. semantic abstraction training. arXiv 1604.06915 (2016)
8. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: ICCV. (2017)
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: ECCV. (2016)
14. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: ECCV. (2014)
15. Kitani, K., Ziebart, B., Bagnell, J., Hebert, M.: Activity forecasting. In: ECCV. (2012)
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., Xiao, H., Shen, X., Yang, J., Lin, Z., Chen, Y., Jie, Z., Feng, J., Yan, S.: Predicting scene parsing and motion dynamics in the future. In: NIPS. (2017)
20. Romera-Paredes, B., Torr, P.: Recurrent instance segmentation. In: ECCV. (2016)
21. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: CVPR. (2017)

22. Pinheiro, P., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: ECCV. (2016)
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., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)
26. Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.: Microsoft COCO: common objects in context. In: ECCV. (2014)
27. Yang, A., Wright, J., Ma, Y., Sastry, S.: Unsupervised segmentation of natural images via lossy data compression. CVIU **110**(2) (2008) 212–225
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., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: 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)
35. Chen, Q., Koltun, V.: Full flow: Optical flow estimation by global optimization over regular grids. In: CVPR. (2016)

A Future instance segmentation results on the test set

We provide instance prediction results on the Cityscapes test set in Table 3 for mid term predictions, as obtained from the online evaluation server. For reference, we also computed the Mask R-CNN oracle results (prediction using future RGB frames), and the Warp optical flow baseline results. The results are comparable to those on the validation set, and we again observe that the results of our F2F model are far more accurate than those of the flow baseline.

	Mid term	
	AP50	AP
Mask R-CNN oracle	58.1	31.9
Optical flow – *Warp*	11.8	4.3
F2F	**17.5**	**6.7**

Table 3: Mid-term instance segmentation results on the Cityscapes test set

B Details on optical flow baselines

To obtain the optical flow estimates, we employed the full flow method [35] using the default parameters given by the authors on the MPI Sintel Flow Dataset.

B.1 Ablation study for the post processing on *Warp*

Prior to any post-processing, the *Warp* baseline generates some artifacts, as shown in Figure 8(a), in particular when objects are moving fast towards the camera. In this case, the optical flow should lead the predicted mask to become larger. But by construction, the number of pixels composing the masks can only stay equal or decrease in the warping process. Masks are therefore broken in parts corresponding to uniform areas of the flow field $\mathbf{F}_{t\to t-1}$, and this phenomenon worsens with the number of steps.

In order to remove these artifacts, we employ mathematical morphology operators to post-process the predictions. First we employ a morphological closing, followed by a closing of holes on the masks. This addresses the problem in an effective manner, as shown in Figure 8(b).

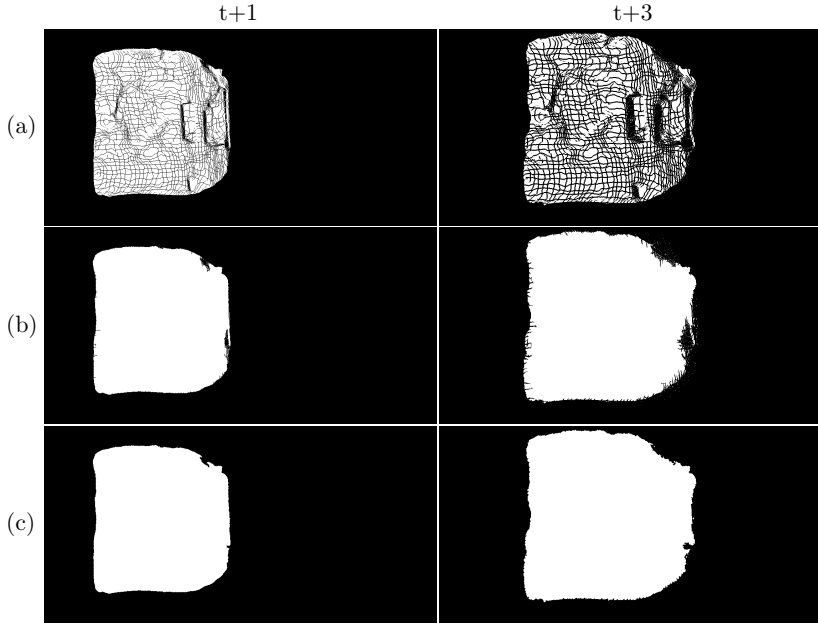


Fig. 8: Qualitative comparison of masks obtained using the *Warp* approach: (a) *w.o.* post-processing, (b) with closing operations, and (c) with full post-processing.

For mid term predictions, we perform these operations on the output before it is used as an input, at each time step. We use bilinear interpolation to estimate flow values at the added positions of the binary mask. This post-processing of the flow adds small spurious artifacts at the border of the the masks, visible in particular in Figure 8(b), right. These are easily removed using morphological openings, see Figure 8(c).

In Table 4 we report the performance of the *Warp* baseline corresponding to the illustrations in Figure 8: (a) before any post-processing is applied (*Warp w.o.* post-processing), (b) with closing operations only (*Warp w.o.* opening), and (c) with full post-processing (*Warp*). These results show that the post processing operations we employ significantly improve performance.

	Mid term		
	AP50	AP	IoU
Warp w.o. post-processing	5.7	1.6	32.2
Warp w.o. opening	10.9	4.0	40.6
Warp	11.1	4.1	41.4

Table 4: Ablation study on the Cityscapes val. set for post-processings on *Warp* optical flow baseline.

B.2 Qualitative comparison with *Shift*

The *Shift* optical flow baseline leads to qualitatively better masks in cases where the optical flow field is not accurate enough. This approach, however, is unable to scale the objects. We illustrate this in Figure 9, in an example where a train is approaching the camera. At the first prediction, the mask predicted by *Shift* has nicer contours than that of *Warp*. However, one can already see that the *Warp* mask is a bit larger. By the third prediction, we see that this has become much more accentuated. As a consequence, *Shift* does not reach the performance of the *Warp* approach, as reported in the main paper.

Disentangling the camera motion from that of the instances and incorporating additional geometric priors to additionally scale masks might improve the results of the *Shift* approach, but is outside the scope of this work.

C F2F architecture design

We recall that our F2F model is composed of four networks: F2F_l, where $l \in \{2, 3, 4, 5\}$, to forecast features at varying scales. Each network may be itself multiscale and is composed in this case of s_l subnetworks F2F_l^s, where $s \in \{1, ..., s_l\}$. Each subnetwork is fed with an input having a channel dimension $n \times p$,

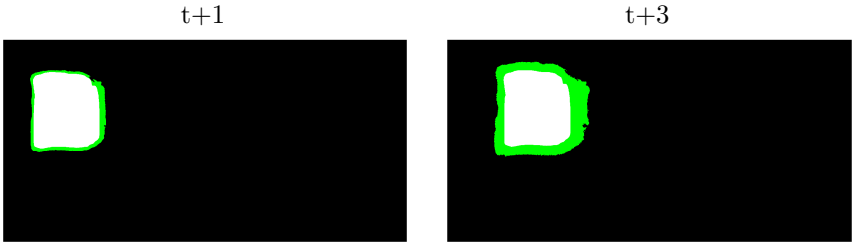


Fig. 9: Comparison between the masks predicted by *Shift*, in white, and *Warp*, the union of the white and green zones. Predictions are shown for short and mid term.

where n is the number of input frames, including the coarse prediction output by the previous subnetwork, and p is the channel dimension of the input and target feature space. In our experiments we have $n = 4$ (or $n = 5$ including the previous coarse prediction), and $p = 256$. To facilitate comparison, our architecture closely follows that of [8], modifying the number of layers and dilation parameters to scale the architecture to the high dimension of our input and target feature space. More sophisticated designs have the potential to improve performance, but they are not the focus of this work. We summarize both architectures in Figure 10.

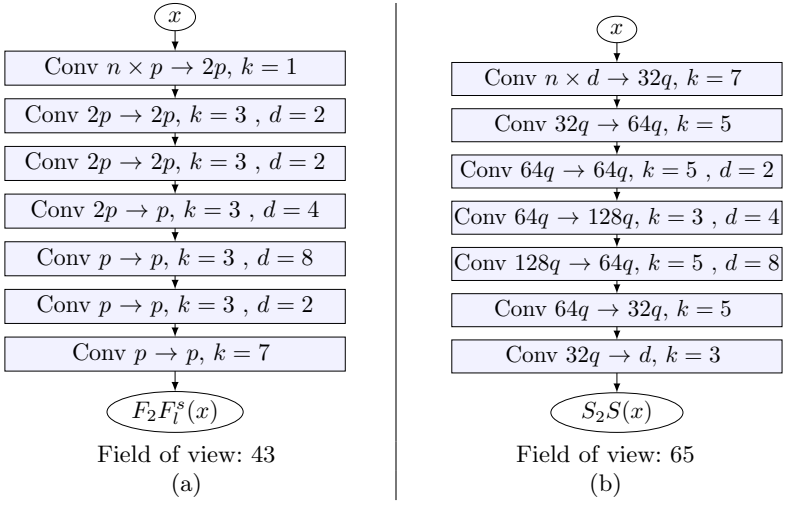


Fig. 10: Architecture design of (a) F_2F_l^s and (b) S2S from [8]. Each convolutional layer except the final one is followed by a ReLU. Stride is always one, padding is chosen so as to maintain the size of the input. The parameter q of S2S was set to 1.5 as in [8].