Predicting Deeper into the Future of Semantic Segmentation

Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, Yann Lecun

Predicting Deeper into the Future of Semantic Segmentation

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

[1] Facebook AI Research

[2] Inria Grenoble, Laboratoire Jean Kuntzmann, Université Grenoble Alpes

[3] New York University

{paulineluc,nneverova,coupriec,yann}@fb.com jakob.verbeek@inria.fr

Abstract

The ability to predict and therefore to anticipate the future is an important attribute of intelligence. It is also of utmost importance in real-time systems, e.g. in robotics or autonomous driving, which depend on visual scene understanding for decision making. While prediction of the raw RGB pixel values in future video frames has been studied in previous work, here we introduce the novel task of predicting semantic segmentations of future frames. Given a sequence of video frames, our goal is to predict segmentation maps of not yet observed video frames that lie up to a second or further in the future. We develop an autoregressive convolutional neural network that learns to iteratively generate multiple frames. Our results on the Cityscapes dataset show that directly predicting future segmentations is substantially better than predicting and then segmenting future RGB frames. Prediction results up to half a second in the future are visually convincing and are much more accurate than those of a baseline based on warping semantic segmentations using optical flow.

1. Introduction

Prediction and anticipation of future events is a key component of intelligent decision-making [36]. Building smarter robotic systems and autonomous vehicles implies making decisions based on the analysis of the current situation and hypotheses made on what could happen next [8]. While humans can predict vehicle or pedestrian trajectories effortlessly and at the reflex level, it remains an open challenge for current computer vision systems. Besides the long term goal of learning a good representation allowing machines to reason about future events, an application which directly benefits from our work is autonomous driving. In this domain, approaches are either based on a number of semantic decompositions such as road and obstacle detection, or directly learn a mapping from visual input to driving instructions end-to-end. Recent work from Mobileye [34] demonstrated an advantage of the semantic abstraction approach in lowering the required amount of training data and decreasing the probability of failure. Other work [33] uses future prediction to facilitate long-term planning problems and forms a direct motivation for our work.

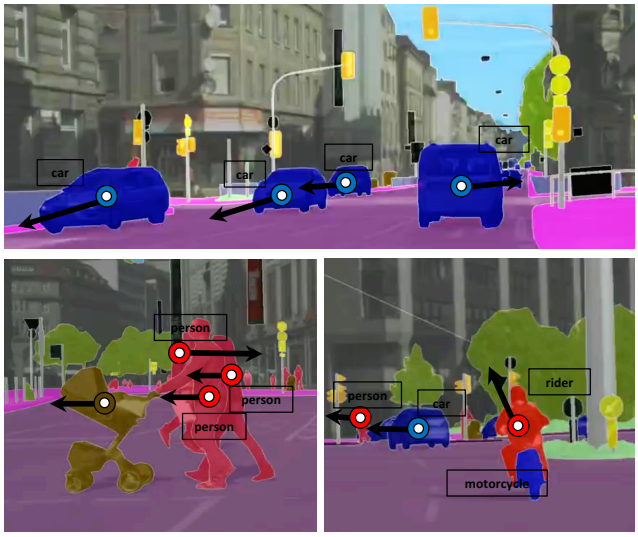

Figure 1: Our models learn semantic-level scene dynamics to predict semantic segmentations of unobserved future frames given several past frames.

The task of predicting future RGB video frames given preceding ones is interesting to assess if current vision systems are able to reason about future events, and it has recently received significant attention [19, 28, 32, 35]. Modeling raw RGB intensities is, however, overly complicated as compared to predicting future high-level scene properties, while the latter is sufficient for many applications. Such high-level future prediction has been studied in vari-

*These authors contributed equally

ous forms, *e.g.* by explicitly forecasting trajectories of people and other objects in future video frames [1, 12, 15, 21, 23, 29]. In our work we do not explicitly model objects or other scene elements, but instead model the dynamics of semantic segmentation maps of object categories with convolutional neural networks. Semantic segmentation is one of the most complete forms of visual scene understanding, where the goal is to label each pixel with the corresponding semantic label (*e.g.*, *tree*, *pedestrian*, *car*, *etc.*). In our work, we build upon the recent progress in this area [10, 25, 5, 46, 31, 30, 24], and develop models to predict the semantic segmentation of future video frames, given several preceding frames. See Figure 1 for an illustration.

The pixel-level annotations needed for semantic segmentation are expensive to acquire, and this is even worse if we need annotations for each video frame. To alleviate this issue we rely on state-of-the-art semantic image segmentation models to label all frames in videos, and then learn our future segmentation prediction models from these automatically generated annotations.

We systematically study the effect of using RGB frames and/or segmentations as inputs and targets for our models, and the impact of various loss functions. Our experiments on the Cityscapes dataset [6] suggest that it is advantageous to directly predict future frames at the abstract semantic-level, rather than to predict the low-level RGB appearance of future frames and then to apply a semantic segmentation model on these. By moving away from raw RGB predictions and modeling pixel-level object labels instead, the network's modeling capacity seems better allocated to learn basic physics and object interaction dynamics.

In this work we make two contributions:

- We introduce the novel task of predicting future frames in the space of semantic segmentation. Compared with prediction of the RGB intensities, we show that we can predict further into the future, and hence model more interesting distributions.
- We propose an autoregressive model which convincingly predicts segmentations up to 0.5 seconds into the future. The mean IoU of our predictions reaches two thirds of the ones obtained by the method used to automatically generate the dense video annotations used for training [46].

Our approach does not require extremely costly temporally dense video annotation and its genericity allows different architectures for still-image segmentation and future segmentation prediction to be swapped in.

2. Related work

Here we discuss the most relevant related work on video forecasting and on disambiguating learning under uncertainty, in particular using adversarial training.

Video forecasting. Several authors developed methods related to our work to improve the temporal stability of semantic video segmentation. Jin *et al*. [18] train a model to predict the semantic segmentation of the immediate next image from the preceding input frames, and fuse this prediction with the segmentation computed from the next input frame. Nilsson and Sminchisescu [30] use a convolutional RNN model with a spatial transformer component [17] to accumulate the information from past and future frames in order to improve prediction of the current frame segmentation. In a similar spirit, Patraucean *et al*. [31] employ a convolutional RNN to implicitly predict the optical flow, and use these to warp and aggregate per-frame segmentations. In contrast, our work is focused on predicting future segmentations without seeing the corresponding frames. Most importantly, we target a longer time horizon than a single frame.

A second line of related work focuses on generative models for future video frame forecasting. Ranzato *et al*. [32] introduced the first baseline of next video frame prediction. Srivastava *et al*. [35] developed a Long Short Term Memory (LSTM) [16] architecture for the task, and demonstrated a gain in action classification using the learned features. Mathieu *et al*. [28] improved the predictions using a multi-scale convolutional architecture, adversarial training [13], and a gradient difference loss. A similar training strategy was employed for future frame predictions in time-lapse videos [47]. To reduce the number of parameters to estimate, several authors reparameterize the problem to predict frame transformations instead of raw pixels [11, 37]. Luo *et al*. [27] employ a convolutional LSTM architecture to predict sequences of up to eight frames of optical flow in RGB-D videos. The video pixel network of Kalchbrenner *et al*. [19] combine LSTMs for temporal modeling with spatial autoregressive modeling. Rather than predicting pixels or flows, Vondrick *et al*. [39] instead predict features in future frames. They predict the activations of the last hidden layer of AlexNet [22] in future frames, and use these to anticipate objects and actions.

Learning under uncertainty. Generative adversarial networks (GANs) [13] and variational autoencoders (VAEs) [20] are deep latent variable models that can be used to deal with the inherent uncertainty in future-prediction tasks. An interesting approach using GANs for unsupervised image representation learning was simultaneously proposed in [7] and [9], where the generative model is trained along with an inference model that maps images to their latent representations. Vondrick *et al*. [40] showed that GANs can be applied to video generation. They use a two-stream generative model: one stream generates a static background, while the other generates a dynamic foreground sequence which is pasted on the background. Yang *et al*. [45] use similar ideas to develop an iterative im-

age generation model where objects are sequentially pasted on the image canvas using a recurrent GAN. Xue *et al*. [44] predict future video frames from a single given frame using a VAE approach. Similarly, Walker *et al*. [41] perform forecasting with a VAE, predicting feature point trajectories from still images.

3. Predicting future frames and segmentations

We start by presenting different scenarios to predict RGB pixel values and/or segmentations of the next video frame. In Section 3.2 we describe two extensions of the single-frame prediction model to predict further into the future.

3.1. Single-frame prediction models

Pixel-level supervision is laborious to acquire for semantic image segmentation, and even more so for its video counterpart. To circumvent the need for datasets with per-frame annotations, we use the state-of-the-art Dilation10 semantic image segmentation network [46] to provide input and target semantic segmentations for all frames in each video. We use the resulting temporally dense segmentation sequences to learn our models.

Let us denote with X_i the i-th frame of a video sequence and denote the sequence of frames from X_t to X_T as $X_{t:T}$. We denote by S_i the semantic segmentation of frame X_i given the Dilation10 network. We represent the segmentations S_i using the final softmax layer's pre-activations, rather than the probabilities it produces. This is motivated by recent observations in network distillation that the softmax pre-activations carry more information [3, 14]. For single-frame future prediction, we consider five different models that differ in whether they take RGB frames and/or segmentations as their inputs and targets: model X2X takes $X_{1:t}$ and predicts X_{t+1}, model S2S takes $S_{1:t}$ and predicts S_{t+1}, models XS2X and XS2S take $(X_{1:t}, S_{1:t})$ and predict respectively X_{t+1} and S_{t+1}, and finally model XS2XS takes $(X_{1:t}, S_{1:t})$ and predicts (X_{t+1}, S_{t+1}).

Architectures. Model X2X is a next frame prediction model, for which we use the multi-scale network of Mathieu *et al*. [28] with two spatial scales. Noting C the number of output channels, each scale module is a four-layer convolutional network alternating convolutions and ReLU operations, outputting feature maps with 128, 256, 128, C channels each, and filters of size 3 for the smaller scale, and 5, 3, 3, 5 for the larger scale. The last non-linear function is a hyperbolic tangent, to ensure that the predicted RGB values lie in the range $[-1, 1]$. The output at a coarser scale is upsampled, and used in input to the next scale module together with a copy of the input at that scale.

For models that predict segmentations S_{t+1}, we removed the last hyperbolic tangent non-linearities for the corresponding output channels, since the softmax pre-activations

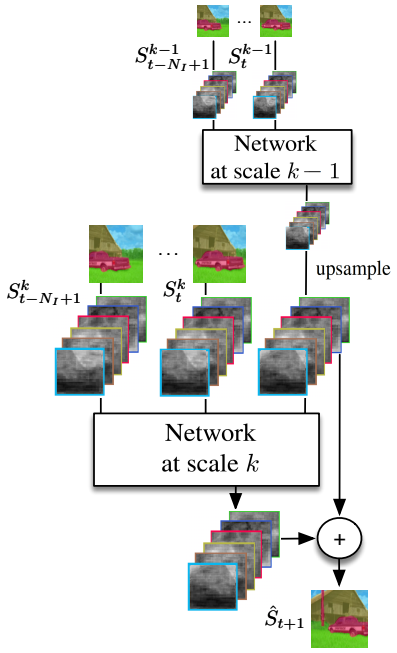

Figure 2: Multi-scale architecture of the S2S model that predicts the semantic segmentation of the next frame given the segmentation maps of the N_I previous frames.

are not limited to a fixed range. Apart from this difference, the S2S model, that predicts the next segmentation from past ones, has the same architecture as the X2X model. The multi-scale architecture of the S2S model is illustrated in Figure 2. The other models (XS2X, XS2S, and XS2XS), which take both RGB frames and segmentation maps as input, also use the same internal architecture.

Loss function. Following [28], for all models, the loss function between the model output $\hat{Y}$ and the target output Y is the sum of an ℓ_1 loss and a gradient difference loss:

$$\mathcal{L}(\hat{Y}, Y) = \mathcal{L}_{\ell_1}(\hat{Y}, Y) + \mathcal{L}_{\text{gdl}}(\hat{Y}, Y). \quad (1)$$

Using Y_{ij} to denote the pixel elements in Y, and similarly for $\hat{Y}$, the losses are defined as:

$$\mathcal{L}_{\ell_1}(\hat{Y}, Y) = \sum_{i,j} \left|Y_{ij} - \hat{Y}_{ij}\right|, \quad (2)$$

$$\mathcal{L}_{\text{gdl}}(\hat{Y}, Y) = \sum_{i,j} \Big|\left|Y_{i,j} - Y_{i-1,j}\right| - \left|\hat{Y}_{i,j} - \hat{Y}_{i-1,j}\right|\Big| + \Big|\left|Y_{i,j-1} - Y_{i,j}\right| - \left|\hat{Y}_{i,j-1} - \hat{Y}_{i,j}\right|\Big|, \quad (3)$$

where $|\cdot|$ denotes the absolute value function. The ℓ_1 loss tries to match all pixel predictions independently to their corresponding target values. The gradient difference loss, instead, penalizes errors in the gradients of the prediction. This loss is relatively insensitive to low-frequency

mismatches between prediction and target (*e.g.*, adding a constant to all pixels does not affect the loss), and is more sensitive to high-frequency mismatches that are perceptually more significant (*e.g.* errors along the contours of an object). We present a comparison of this loss with a multi-class cross entropy loss in Section 4.

Adversarial training. As shown by Mathieu *et al*. [28] in the context of raw images, introducing an adversarial loss allows the model to disambiguate between modes corresponding to different turns of events, and reduces blur associated with this uncertainty. Luc *et al*. [26] demonstrated the positive influence of adversarial training for semantic image segmentation.

Our formulation of the adversarial loss term is based on the recently introduced Wasserstein GAN [2], with some modifications for the semantic segmentation application. In the case of the S2S model, the parameters θ of the discriminator $\mathcal{D}_\theta$ are trained to maximize the absolute difference between its output for ground truth sequences $(S_{1:t}, S_{t+1})$ and sequences $(S_{1:t}, \hat{S}_{t+1})$ predicted by our model:

$$\max_\theta \left| \sigma\left(\mathcal{D}_\theta\left(S_{1:t}, S_{t+1}\right)\right) - \sigma\left(\mathcal{D}_\theta\left(S_{1:t}, \hat{S}_{t+1}\right)\right)\right|. \quad (4)$$

The outputs produced by the predictive model are softmax pre-activation maps with unbounded values. In the Wasserstein GAN they are encouraged to grow indefinitely. To avoid this and stabilize training, we employ an additional sigmoid non-linearity σ at the output of the discriminator, and set explicit targets for two kinds of outputs: 0 for generated sequences and α for real training sequences, set to 0.9 to prevent saturation.

The adversarial regularization term for our predictive model (*i.e.* the "generator") then takes the following form:

$$\mathcal{L}_{\text{adv}}(S_{1:t}, \hat{S}_{t+1}) = \lambda \left| \sigma\left(\mathcal{D}_\theta\left(S_{1:t}, \hat{S}_{t+1}\right)\right) - \alpha \right|. \quad (5)$$

The structure of the discriminator network is derived from the two-scale architecture described above. Additional details are provided in the supplementary material.

3.2. Predicting deeper into the future

We consider two extensions of the previous models to predict further into the future than a single frame. The first is to expand the output of the network to comprise a batch of m frames, *i.e.* to output $X_{t+1:t+m}$ and/or $S_{t+1:t+m}$. We refer to this as the "batch" approach. The drawback of this approach is that it ignores the recurrent structure of the problem. That is, it ignores the fact that S_{t+1} depends on $S_{1:t}$ in the same manner as S_{t+2} depends on $S_{2:t+1}$. As a result, the capacity of the model is split to predict the m output frames, and the number of parameters in the last layer scales linearly with the number of output frames.

In our second approach, we leverage the recurrence property, and iteratively apply a model that predicts a single step into the future, using its prediction for time $t+1$ as an input to predict at time $t+2$, and so on. This allows us to predict arbitrarily far into the future in an autoregressive manner, without resources scaling with the number of time-steps we want to predict. We refer to this approach as "autoregressive". See Figure 3 for a schematic illustration of the two extensions for multiple time-step predictions.

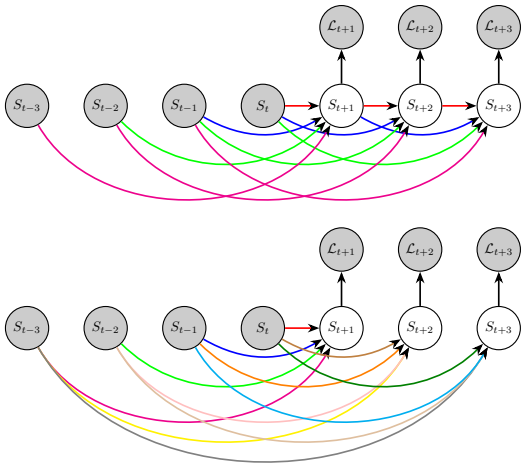

Figure 3: Illustration of the autoregressive (top) and batch (bottom) models. The autoregressive model shares parameters over time; dependency links are colored accordingly.

In the autoregressive mode, we first evaluate the models trained for single-frame prediction, then we fine-tune these models using backpropagation through time [43], to account for the fact that mistakes at each time-step affect all later time-steps.

4. Experiments

Before presenting our experimental results, we first describe the dataset and evaluation metrics in Section 4.1. We then present results on short-term (i.e. single-frame) prediction, mid-term prediction (0.5 sec.), and long-term prediction (10 sec.).

4.1. Dataset and evaluation metrics

The Cityscapes dataset [6] contains 2,975 training, 500 validation and 1,525 testing video sequences of 1.8 second. Each sequence consists of 30 frames, and a ground-truth semantic segmentation is available for the 20-th frame. The segmentation outputs of the Dilation10 network [46] are produced at a resolution of 128×256 and we perform all experiments at this resolution. For this purpose, we also downsample RGB frames and ground truth to this resolution. We report performance of our models on the Cityscapes validation set, and refer to the supplementary material for results on the test set.

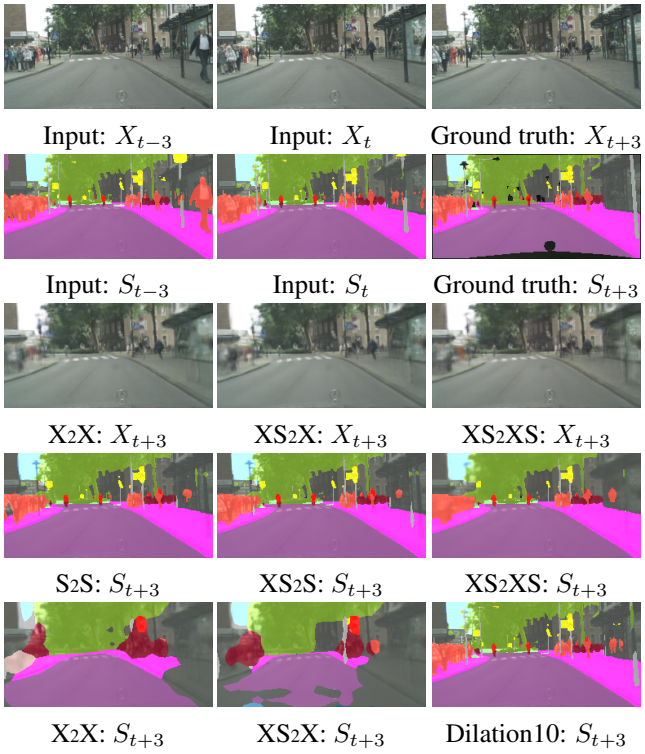

Figure 4: Short-term predictions of RGB frame X_{t+3} and segmentation S_{t+3} using our different models, compared to ground truth, and Dilation10 oracle that has seen X_{t+3}.

We assess performance using the standard mean Intersection over Union (IoU) measure, computed with respect to the ground truth segmentation of the 20-th frame in each sequence (IoU GT). We also compute the IoU measure with respect to the segmentation produced using the Dilation10 network [46] for the 20-th frame (IoU SEG). The IoU SEG metric allows us to validate our models with respect to the target segmentations from which they are trained. Finally, we compute the mean IoU across categories that can move in the scene: *person, rider, car, truck, bus, train, motorcycle, and bicycle* (IoU-MO, for "moving objects").

To evaluate the quality of the frame RGB predictions, we compute the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) measures [42]. The SSIM measures similarity between two images, ranging between -1 for very dissimilar inputs to $+1$ when the inputs are the same. It is based on comparing local patterns of pixel intensities normalized for luminance and contrast.

Unless specified otherwise, we train our models using a frame interval of 3, and taking 4 frames and/or segmentations as input. That is, the input sequence consists of frames $\{X_{t-9}, X_{t-6}, X_{t-3}, X_t\}$, and similarly for segmentations. We performed patch-wise training with 64×64 patches for the largest scale resolution, enabling equal class frequency sampling as in [10], using mini-batches of four patches and a learning rate of 0.01.

Method	PSNR	SSIM	IoU GT (SEG)	IoU-MO GT (SEG)
Copy last input	20.6	0.65	49.4 (54.6)	43.4 (48.2)
Warp last input	23.7	0.76	59.0 (**67.3**)	54.4 (**63.3**)
Model X2X	24.0	**0.77**	23.0 (22.3)	12.8 (11.4)
Model S2S	—	—	58.3 (64.9)	53.8 (59.8)
Model XS2X	**24.2**	**0.77**	22.4 (22.5)	10.8 (10.0)
Model XS2S	—	—	58.2 (64.6)	53.7 (59.9)
Model XS2XS	24.0	0.76	55.5 (61.1)	50.7 (55.8)
Model S2S-adv.	—	—	58.3 (65.0)	53.9 (60.2)
Model S2S-dil	—	—	**59.4** (66.8)	**55.3** (63.0)

Table 1: Short-term prediction accuracy of baselines and of our models taking either RGB frames (X) and/or segmentations (S) as input and output. For reference: the 59.4 IoU corresponds to 91.8% per pixel accuracy.

4.2. Short-term prediction

In our first experiment, we compare the five different input-output representations. For models that do not directly predict future segmentations, we generate segmentations using the Dilation10 network based on the predicted RGB frames. We also include two baselines. The first baseline copies the last input frame to the output. The second baseline estimates the optical flow between the last two inputs, and warps the last input using the estimated flow. Further details are given in the supplementary material. Comparison with tracking-based approaches is difficult since (i) segmentation is performed densely and lacks the notion of object instances used by object trackers, and (ii) "stuff" categories (road, vegetation, *etc*.), useful for drivable area detection in the context of autonomous driving, are not suitable for modeling with tracking-based approaches.

In Figure 4, we show qualitative results of the predictions for one of the validation sequences. From the quantitative result in Table 1 we make several observations. First, in terms of RGB frame prediction (PSNR and SSIM), the performance is comparable for the three models X2X, XS2X, and XS2XS, and improves over the two baselines. This shows that our models learn non-trivial scene dynamics in the RGB pixel space, and that adding semantic segmentations either at input and/or output does not have a substantial impact on this ability.

Second, in terms of the IoU segmentation metrics, the models that directly predict future segmentations (S2S, XS2S, XS2XS) perform much better than the models that only predict the RGB frames. This suggests that artifacts in the RGB frame predictions degrade the performance of the Dilation10 network. See also the corresponding RGB frame

Model	IoU GT	IoU SEG	IoU-MO GT
Dilation10 oracle	68.8	100	64.7
S2S, 2 scales, ℓ_1+gdl	**58.3**	**64.9**	**53.8**
S2S, 1 scale, ℓ_1+gdl	57.7	63.9	52.6
S2S, 2 scales, ℓ_1	57.6	64.0	53.2
S2S, 2 scales, MCE	55.5	60.9	49.7

Table 2: Ablation study with the S2S model, and comparison to a Dilation10 oracle that predicts the future segmentation using the future RGB frame as input.

predictions in Figure 4.

Third, the XS2XS model, which predicts both segmentations and RGB frames performs somewhat worse than the models that only predict segmentations (S2S and XS2S), suggesting that some of the modeling capacity is compromised by jointly predicting the RGB frames.

Fourth, we find that fine-tuning the S2S model using adversarial training (S2S-adv) does not lead to a significant improvement over normal training.

Table 2 presents results of an ablation study of the S2S model, assessing the impact of the different loss functions, as well as the impact of using one or two scales. We include the results obtained using the Dilation10 model as an "oracle", that predicts the future segmentation based on the future RGB frame, which is not accessible to our other models. This oracle result gives the maximum performance that could be expected, since this oracle was used to provide the training data - we can thus only expect our models to have at best comparable performance with this oracle. All variants of the S2S model were trained during about 960,000 iterations, taking about four days of training on a single GPU. The results show that using two scales improves the performance, as does the addition of the gradient difference loss. Training with the ℓ_1 and/or gdl loss on the softmax pre-activations gives better results as compared to training using the multi-class cross-entropy (MCE) loss on the segmentation labels. This is in line with observations made in network distillation [3, 14].

Finally, we perform further architecture exploration for the S2S model, which performed best. We propose a simpler, deeper, and more efficient architecture with dilated convolutions [46], to expand the field of view while retaining accurate localization for the predictions. We call this model S2S-dil, and provide details in the supplementary material. This model gives best overall results, reported in Table 1.

4.3. Mid-term prediction

We now address the more challenging task of predicting the mid-term future, *i.e*. the next 0.5 second. In these experiments we take in input frames 2, 5, 8, and 11, and predict

	Frame 14		Frame 20	
Model	PSNR	SSIM	PSNR	SSIM
Copy last input	20.4	0.64	18.0	0.55
Warp last input	23.5	**0.76**	19.4	0.59
X2X, AR	**23.9**	**0.76**	19.2	0.61
XS2XS, AR	23.8	**0.76**	19.3	0.61
X2X, batch	23.8	**0.76**	20.6	**0.65**
XS2X, batch	**23.9**	**0.76**	**20.7**	**0.65**
XS2XS, batch	23.8	**0.76**	**20.7**	0.64

Table 3: Mid-term RGB frame prediction results for frame 20 using different models in batch and autoregressive mode.

Model	IoU GT	IoU SEG	IoU-MO GT
Copy last input	36.9	39.2	26.8
Warp last input	44.3	47.2	37.0
S2S, AR	45.3	47.2	36.4
S2S, AR, fine-tune	46.7	49.7	39.3
XS2XS, AR	39.3	40.8	27.4
S2S, batch	42.1	44.2	32.8
XS2S, batch	42.3	44.6	33.1
XS2XS, batch	41.2	43.5	31.4
S2S-adv, AR	45.1	47.2	37.3
S2S-dil, AR	46.5	48.6	38.8
S2S-dil, AR, fine-tune	**47.8**	**50.4**	**40.8**

Table 4: Mid-term segmentation prediction results. For reference: the 47.8 IoU corresponds to 87.9% per pixel accuracy.

outputs for frames 14, 17 and 20. We compare different strategies: batch models, autoregressive models (AR), and models with autoregressive fine-tuning (AR fine-tune). We compare these strategies to our two baselines consisting in copying the last input, and the second one relying on optical flow. For the optical flow baseline, after the first prediction, we also warp the flow field so that the flow is applied to the correct locations at the next time-step, and so on. Qualitative prediction results are shown in Figure 5. For models XS2X and XS2S, the autoregressive mode is not used because either the frame or the segmentation input are missing for predicting from the second output on.

The results for RGB frame prediction in Table 3 show that for frame 14, all models give comparable results, consistently improve over the copy baseline and perform somewhat better than the warping baseline. For frame 20, the batch models perform best. On the contrary, when predicting segmentations, we find that the autoregressive models perform better than the batch models, as reported in Table 4. This is probably due to the fact that the single-

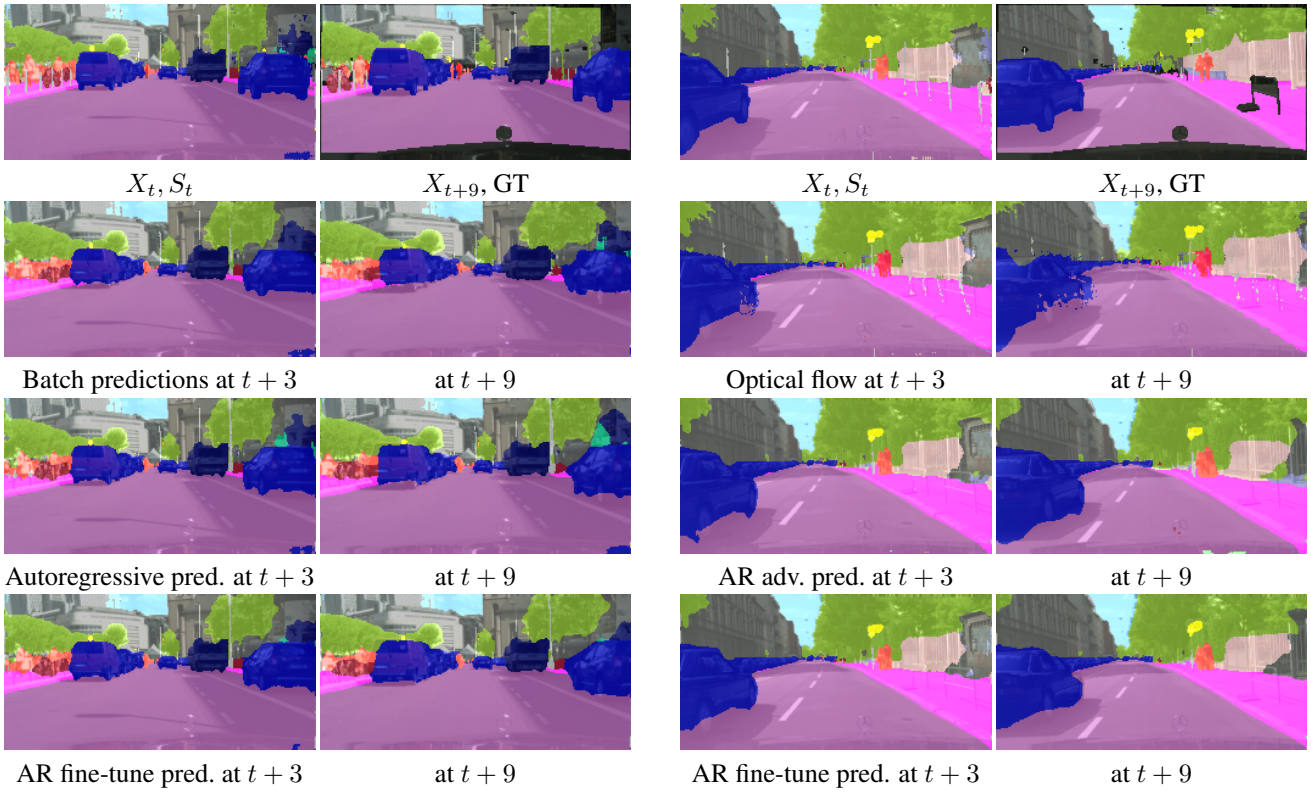

Figure 5: Optical flow baseline, S2S autoregressive and S2S batch predictions for two sequences (first sequence left, second sequence right). First row: last input and ground truth. Other rows show predictions overlayed with the true future frames. The full results are provided in the supplementary material.

step predictions are more accurate for segmentation, which makes them more suitable for autoregressive modeling. For RGB frame prediction, errors accumulate quickly, leading to degraded autoregressive predictions. Among the batch models, using the images as input (XS2S model) slightly helps. Predicting both the images and segmentation (XS2XS model) performs worst, the image prediction task presumably taking up resources otherwise available for modeling the dynamics of the sequence.

Model S2S is the most effective, as it can be applied in autoregressive mode, and outperforms XS2XS in this setting. In Figure 5 we compare different versions of this model. Visually, the first sequence shows some improvements using the autoregressive fine-tuned model, by more accurately matching contours of the moving cars than the other strategies. The second sequence displays typical failures of the optical flow baseline, where certain values cannot be estimated because they correspond to points that were not present in the input, *e.g.* those at the back of the incoming car, and must be filled using a standard region filling algorithm. This sequence also displays some improvements of the adversarial fine-tuning on the car contours. More examples are present in the supplementary material, where we can observe that difficult cases for our method include dealing with occlusions and with fast ego-motion.

4.4. Long-term prediction

To evaluate the limits of our S2S autoregressive models on arbitrarily long sequences, we use them to make predictions of up to ten seconds into the future. To this end, we evaluate our models on ten sequences on 238 frames extracted from the long Frankfurt sequence of the Cityscapes validation set. Given four segmentation frames with a frame interval of 17 images, corresponding to exactly one second, we apply our models to predict the ten next ones. In Figure 7 we report the IoU SEG performance as a function of time. In this extremely challenging setting, the predictive performance quickly drops over time. Fine-tuning the model in autoregressive mode improves its performance, but only gives a clear advantage over the input-copy baseline for predictions at one and two seconds ahead. We also applied our model with a frame interval of 3 to predict up to 55 steps ahead, but found this to perform much worse. Figure 6 shows an example of predictions compared to the actual future segmentations. The visualization shows that our model averages the different classes into an average future,

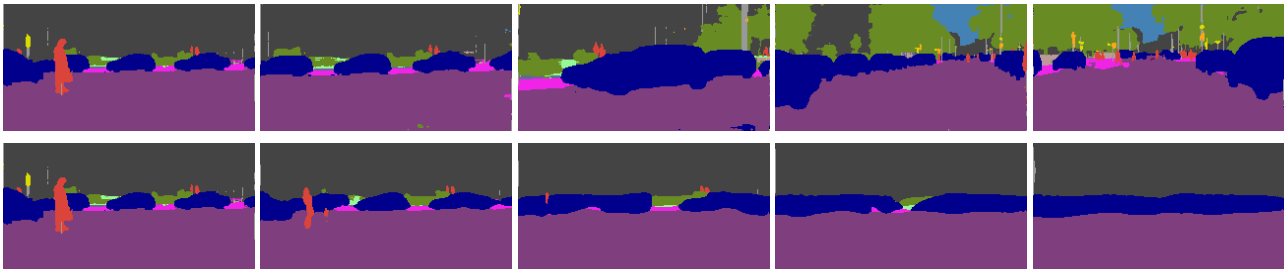

Figure 6: Last input segmentation, and ground truth segmentations at 1, 4, 7, and 10 seconds into the future (top row), and corresponding predictions of the autoregressive S2S model trained with fine-tuning (bottom row).

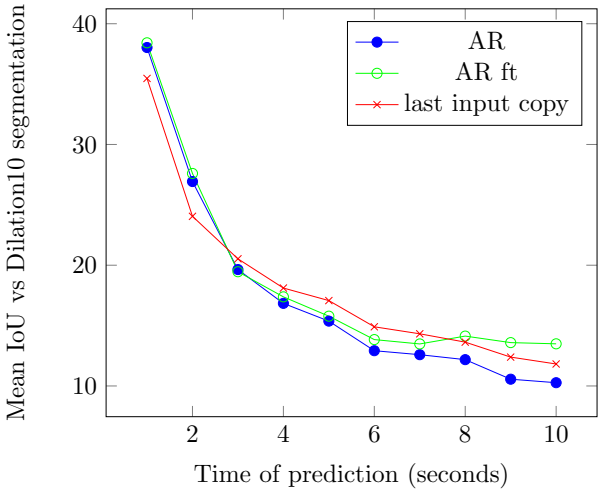

Figure 7: Mean IoU SEG of long-term segmentation prediction for the AR and AR fine-tune S2S models.

which is perhaps not entirely surprising. Sampling different possible futures using a GAN or VAE approach might be a way to resolve this issue.

4.5. Cross-dataset generalization

To evaluate the generalization capacity of our approach,we test our S2S model on the Camvid dataset [4], specifically on the test set of 233 images with 11 classes grouping employed in [38]. Ground truth segmentations are provided for every second on 30 fps video sequences. We first generate the Dilation10 segmentations - without fine-tuning the oracle to the CamVid dataset - using a frame interval of 5, roughly corresponding to a frame interval of 3 on Cityscapes. We note that the class correspondence between Cityscapes and CamVid is not perfect; for instance we associate the class "tree" to "vegetation". As reported in Table 5, our models have very good mid-term performance on this dataset, considering the oracle results. For reference, [46] reports an IoU of 65.3 using a fine-tuned Dilation8.

	Dilation10 oracle	Copy last input	Warp last input	S2S AR ft
IoU GT	55.4	40.8	43.7	**46.8**

Table 5: IoU of oracle and mid-term predictions on Camvid

5. Conclusion

We introduced a new visual understanding task of predicting future semantic segmentations. For prediction beyond a single frame, we considered batch models that predict all future frames at once, and autoregressive models that sequentially predict the future frames. While batch models were more effective in the RGB intensities space because of otherwise large error propagation, the more desirable autoregressive mode was more accurate in the semantic segmentation space, supporting with experimental evidence our motivation for this new task. The autoregressive mode lends itself naturally to predicting sequences of arbitrary length, thanks to which we can aim to model more interesting distributions.

In this respect, there is still room for improvement. Where the Dilation10 network for semantic image segmentation gives around 69 IoU, this drops to about 59 when predicting 0.18s ahead and to about 48 for 0.5s. Most predicted object trajectories are reasonable, but do not always correspond to the actual observed trajectories. GAN or VAE models may be useful to address the inherent uncertainty in the prediction of future segmentations. We open-source our Torch-based implementation, and invite the reader to watch videos of our predictions at https://thoth.inrialpes.fr/people/pluc/iccv2017.

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 Michael Mathieu, Matthijs Douze, Hervé Jegou, Larry Zitnick, Moustapha Cisse, Gabriel Synnaeve and anonymous reviewers for their precious comments.

References

[1] A. Alahi, V. Ramanathan, and L. Fei-Fei. Socially-aware large-scale crowd forecasting. In *CVPR*, 2014. 2

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In *ICML*, 2017. 4

[3] L. Ba and R. Caruana. Do deep nets really need to be deep? In *NIPS*, 2014. 3, 6

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 8

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 4

[7] J. Donahue, P. Krahenbuhl, and T. Darrell. Adversarial feature learning. In *ICLR*, 2017. 2

[8] A. Dosovitskiy and V. Koltun. Learning to act by predicting the future. In *ICLR*, 2017. 1

[9] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. In *ICLR*, 2017. 2

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8):1915–1929, 2013. 2, 5

[11] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016. 2

[12] D. Fouhey and C. Zitnick. Predicting object dynamics in scenes. In *CVPR*, 2014. 2

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *NIPS*, 2014. 2

[14] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2014. 3, 6

[15] M. Hoai and F. De la Torre. Max-margin early event detectors. *IJCV*, 107(2):191–202, 2014. 2

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. 2

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 2

[18] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, J. Feng, and S. Yan. Video scene parsing with predictive feature learning. *arXiv:1612.00119*, 2016. 2

[19] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv:1610.00527*, 2016. 1, 2

[20] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014. 2

[21] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. 2

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[23] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014. 2

[24] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 2

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[26] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic segmentation using adversarial networks. In *NIPS Workshop on Adversarial Training*, 2016. 4

[27] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *CVPR*, 2017. 2

[28] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 1, 2, 3, 4

[29] M.Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. In *ICCV*, 2011. 2

[30] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. arXiv:1612.08871, 2016. 2

[31] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR Workshop*, 2016. 2

[32] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014. 1, 2

[33] S. Shalev-Shwartz, N. Ben-Zrihem, A. Cohen, and A. Shashua. Long-term planning by short-term prediction. *arXiv:1602.01580*, 2016. 1

[34] S. Shalev-Shwartz and A. Shashua. On the sample complexity of end-to-end training vs. semantic abstraction training. *arXiv:1604.06915*, 2016. 1

[35] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015. 1, 2

[36] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. 1

[37] J. Van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala. Transformation-based models of video sequences. *arXiv:1701.08435*, 2017. 2

[38] B. Vijay, K. Alex, and C. Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv:1511.00561*, 2015. 8

[39] C. Vondrick, P. Hamed, and A. Torralba. Anticipating the future by watching unlabeled video. In *CVPR*, 2016. 2

[40] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 2

[41] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 3

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 5

[43] P. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.*, 1(4):339–356, 1988. 4

[44] T. Xue, J. Wu, K. Bouman, and W. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 3

[45] J. Yang, A. Kannan, D. Batra, and D. Parikh. LR-GAN: Layered recursive generative adversarial networks for image generation. In *ICLR*, 2017. 2

[46] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2, 3, 4, 5, 6, 8

[47] Y. Zhou and T. L. Berg. Learning temporal transformations from time-lapse videos. In *ECCV*, 2016. 2