

Enhancing Traffic Scene Predictions with Generative Adversarial Networks

Peter König¹, Sandra Aigner¹ and Marco Körner¹

Abstract—We present a new two-stage pipeline for predicting frames of traffic scenes where relevant objects can still reliably be detected. Using a recent video prediction network, we first generate a sequence of future frames based on past frames. A second network then enhances these frames in order to make them appear more realistic. This ensures the quality of the predicted frames to be sufficient to enable accurate detection of objects, which is especially important for autonomously driving cars. To verify this two-stage approach, we conducted experiments on the Cityscapes dataset. For enhancing, we trained two image-to-image translation methods based on generative adversarial networks, one for blind motion deblurring and one for image super-resolution. All resulting predictions were quantitatively evaluated using both traditional metrics and a state-of-the-art object detection network showing that the enhanced frames appear qualitatively improved. While the traditional image comparison metrics, *i.e.*, MSE, PSNR, and SSIM, failed to confirm this visual impression, the object detection evaluation resembles it well. The best performing prediction-enhancement pipeline is able to increase the average precision values for detecting cars by about 9% for each prediction step, compared to the non-enhanced predictions.

I. INTRODUCTION

Predicting possible future trajectories of objects in traffic scenes, such as cars and pedestrians, plays an essential role in anticipatory driving. Only by having knowledge about the type of object and its possible movement patterns, we are able to make safe decisions as a human driver. Having predictions as an additional input to a driver assistance system or an autonomous driving system would be beneficial to its internal decision-making process. Such a system could make faster and possibly more informed decisions regarding the control of the vehicle, which leads to an increase in safety.

Predicting the future frames of videos of street scenes is one way to anticipate the movement of objects. However, to support a system such as an autonomously driving car, the quality of the predicted frames must be high enough to enable the reliable detection of relevant objects. Depending on the identified object, the decision process will vary greatly. State-of-the-art object detection software produces good results on real videos of street scenes. Thus, if a prediction looks as similar to the real data as possible, we can assume that detecting objects correctly will be easier.

Due to the success of neural networks on a variety of computer vision tasks, we test the capabilities of neural network-based methods for generating enhanced video predictions that

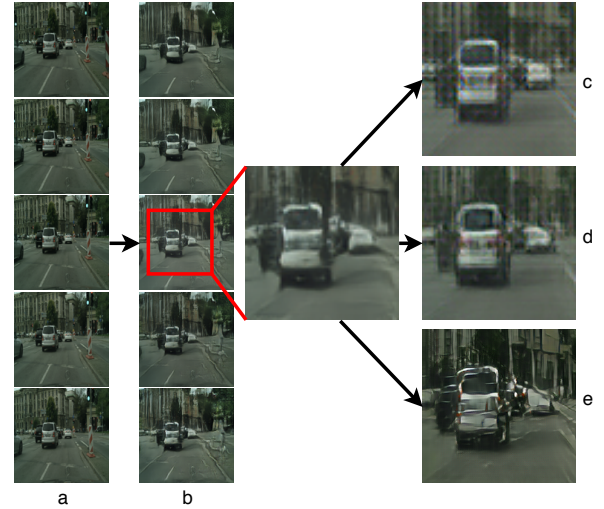


Fig. 1. Example predictions of the different prediction-enhancement pipelines. a: Input Sequence, b: FutureGAN [1], c: DeblurGAN [22] (transposed convolutions), d: DeblurGAN (NN-upsampling + convolution), e: SRGAN [24].

allow for the accurate detection of objects. Particularly, we build on our previous video prediction network, *FutureGAN* [1], and predict five future frames of a street scene based on five input frames. The original results on the *Cityscapes* dataset [7] suggest that the network has learned reasonably good movement representations. However, for complex input data, such as natural street scenes, the predicted frames suffer from blurring effects and other unrealistic artifacts. Therefore, we test several additional methods to enhance the predicted frames, thus making them appear more realistic, see figure 1 for example predictions. For enhancing, we utilize *generative adversarial networks (GANs)* [13]. In order to make the predictions more realistic and increase object detection results, we test two different GAN-based methods. The first one is an image super-resolution approach, the *SRGAN* [24], and the second one is a blind motion deblurring approach, the *DeblurGAN* [22]. In both cases, the frame enhancement is treated as an *image-to-image translation* problem, where GANs have led to good results.

In this paper, we provide a reliable pipeline for predicting traffic scenes. To prove the effectiveness of our prediction-enhancement pipeline, we evaluate all resulting predictions using the state-of-the-art object detection network *YOLOv3* [35]. Our final model is able to produce predictions of both good visual quality and high detection accuracy. The *average precision (AP)* values for the object class "car" can be increased by about 9% for each prediction step, compared

¹TUM Department of Civil, Geo and Environmental Engineering, Technical University of Munich, Germany {peter.koenig, sandra.aigner, marco.koerner}@tum.de

to the non-enhanced predictions.

II. RELATED WORK

Ranzato *et al.* [33] first introduced a baseline for video prediction using deep neural networks. Since then, the deep learning-based prediction of future traffic sequences has become a widely researched topic in computer vision, especially in the autonomous driving community.

Due to the uncertainty in predicting the future, generating high-quality predictions of natural traffic scenes is a very complex task. This is why some approaches simplify this task and focus on predicting semantic segmentation masks, rather than generating the pixel values of the frames [28], [17], [18], [30], [5]. Many of these approaches, as well as approaches that directly generate the pixel values, use recurrent neural network structures [8], [27], [10], [37], [23], [6], [30], [39]. Lotter *et al.* [27], for example, utilized long short-term memory (LSTM) [16] units to generate the pixel values of the frame one time step ahead. Bhattacharjee *et al.* [3] generate predictions using a multi-stage GAN that takes input frames at different scales. Using GANs [4], [1], [3], or a combination of GANs and recurrent modules [25], are further common methods to predict video frames of traffic scenes.

Despite the recent advances in this field, the resulting frames often lack realism. To make predictions occur more realistic, others tackled the problem by learning separate representations for the static and dynamic components of a video. This is done either by incorporating motion conditions, such as optical flow information [12], [34], [15], [17], [25], or by learning sparse features that represent pixel dynamics [26]. Decomposing the video into static and non-static components allows the network to simply reproduce the values of the static part for the majority of pixels. Transformations are then only performed on the non-static pixels. This leads to the problem of occluded and new objects not being properly modeled, especially in long-term predictions.

Our approach builds on the idea of enhancing each prediction directly using a second network. Recently, related ideas without an application for traffic scenes were introduced [41], [38]. These approaches use two-stage networks to first generate subsequent frames from structure and content conditions, and then refine the frames using temporal signals or motion dynamics. We, on the other hand, use the learned motion representations of a GAN-based model to predict a set of future frames from a set of input frames. We then use a separate second model, an image-to-image translation GAN, to eliminate the artifacts and blurring effects caused by the transformations of the first network.

III. ENHANCEMENT OF PREDICTED VIDEO FRAMES

The methods used in this paper are based on GANs. In an adversarial setting, a generator network is trained to model the data distribution of the training data. During training, a second network, the discriminator, provides feedback to the generator about the similarity between the modeled and the observed data distribution. This results in a minimax

game. The discriminator D tries to maximize its score of correctly classifying the samples it observes as real or fake. The generator G tries to fool the discriminator by minimizing the difference of the modeled and the data distribution, *i.e.*, by optimizing

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_r} [\log(D(x))] + \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [\log(1 - D(\tilde{x}))], \quad (1)$$

where \mathbb{P}_r is the data distribution, \mathbb{P}_g is the model distribution, implicitly defined by $\tilde{x} = G(z)$, and z is the input sampled from a random distribution $P(z)$. During training, this approach gradually enforces the generator to produce samples that appear more and more similar to the training data.

However, there are problems with GAN-based approaches. First, they are hard to train and the highly unstable training process often leads to non-convergence. Secondly, there is the mode collapse effect. This means, the generator learns to fool the discriminator by producing samples of a limited set of modes, thus produces samples that lack diversity. The generator fails to sufficiently model the variation in the real data distribution.

For generating the traffic scene predictions we make use of our recent GAN-based approach, FutureGAN, that avoids these problems. We then evaluate how the predictions can be enhanced in order to improve the object detection results on the predicted frames. The methods to enhance the predictions are all based on variants of the *conditional GAN (cGAN)* [29], where enhancing is treated as an image-to-image translation problem. In the following, we describe the approaches used in this paper in more detail.

A. FutureGAN

To predict the future frames of the traffic sequences, we use FutureGAN. This network predicts multiple output frames from a set of input frames. It is trained using the *progressively growing of GANs* technique, introduced by Karras *et al.* [20]. During training, layers are added progressively to both the generator and the discriminator network to increase the frame resolution gradually. Many architectures were particularly designed to overcome the GAN-related training issues, such as non-convergence and mode collapse [32], [2]. The progressive growing training strategy helps to further improve the GAN training. Additionally, the authors used feature normalization and a *Wasserstein GAN with gradient penalty (WGAN-GP)* [14] loss to increase the training stability of the network. For details on the network structure and architectural design, we refer the reader to the original paper [1].

B. DeblurGAN

As a first enhancement method, we chose DeblurGAN [22]. DeblurGAN is a blind motion deblurring method based on cGANs. The DeblurGAN framework treats motion deblurring as an image-to-image translation problem. Rather than to estimate a motion kernel, the network is trained to directly translate the image from a blurry version to an unblurred one.

The DeblurGAN loss function

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda \mathcal{L}_X \quad (2)$$

consists of two components, a WGAN-GP loss term \mathcal{L}_{GAN} and a content loss term \mathcal{L}_X , with λ as a balancing factor. The content loss was introduced in addition to the adversarial loss term to increase the perceptual quality of the generated images. In contrast to the standard L_1 (MAE) or L_2 (MSE) losses, which are based on the differences of the raw pixel values, this *perceptual loss* [19] is based on the differences in feature space. In particular,

$$\mathcal{L}_X = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^S)_{x,y} - \phi_{i,j}(G_\theta(I^B))_{x,y})^2 \quad (3)$$

is the L_2 difference between the feature maps of the ground truth and the deblurred image of a specific layer in the VGG-19 [36] network, where $\phi_{i,j}$ is the feature map obtained before the i -th max-pooling layer and after the j -th convolutional layer of the VGG-19 network trained on ImageNet [9], and I^S and I^B are the sharp ground truth and the blurry predicted frame. $W_{i,j}$ and $H_{i,j}$ are the width and height dimensions of the feature maps, respectively. In this case, we used the $VGG_{3,3}$ convolutional layer, since the general image content is typically captured in the lower layers of such a network [40].

The original results of Kopyn *et al.* [22] show that motion blur and artifacts similar to those of the FutureGAN street scene predictions can be removed effectively. After training DeblurGAN on our data, we observed that the network generates a checkerboard pattern on the deblurred test images (*cf.* Figure 2). Following the findings by Odena *et al.* [31], we assume these patterns to be caused by the transposed convolutional layers in the upsampling part of the generator network. Transposed convolutions can produce this type of pattern because of the overlap that occurs when the kernel sizes are not divisible by the strides. To avoid such undesired patterns in the deblurred predictions, we designed a different version of DeblurGAN. We replaced each of the transposed convolutional layers in the original DeblurGAN architecture with a nearest-neighbor upsampling layer followed by a regular convolutional layer. The resulting generator structure can be seen in Figure 2. For completeness and comparability, we conducted separate experiments using both DeblurGAN versions.

C. SRGAN

The second method we used to enhance the frames predicted by FutureGAN is a GAN-based approach for image super-resolution, the SRGAN [24]. Image super-resolution means that a low resolution (LR) image is upsampled to its high-resolution (HR) version. The SRGAN was designed to generate HR images of high perceptual quality, which are upsampled by a factor of 4 from the LR images. An increased resolution of the traffic scene predictions might also have positive effects on the object detection results because of the increased number of details in the high-resolution image.

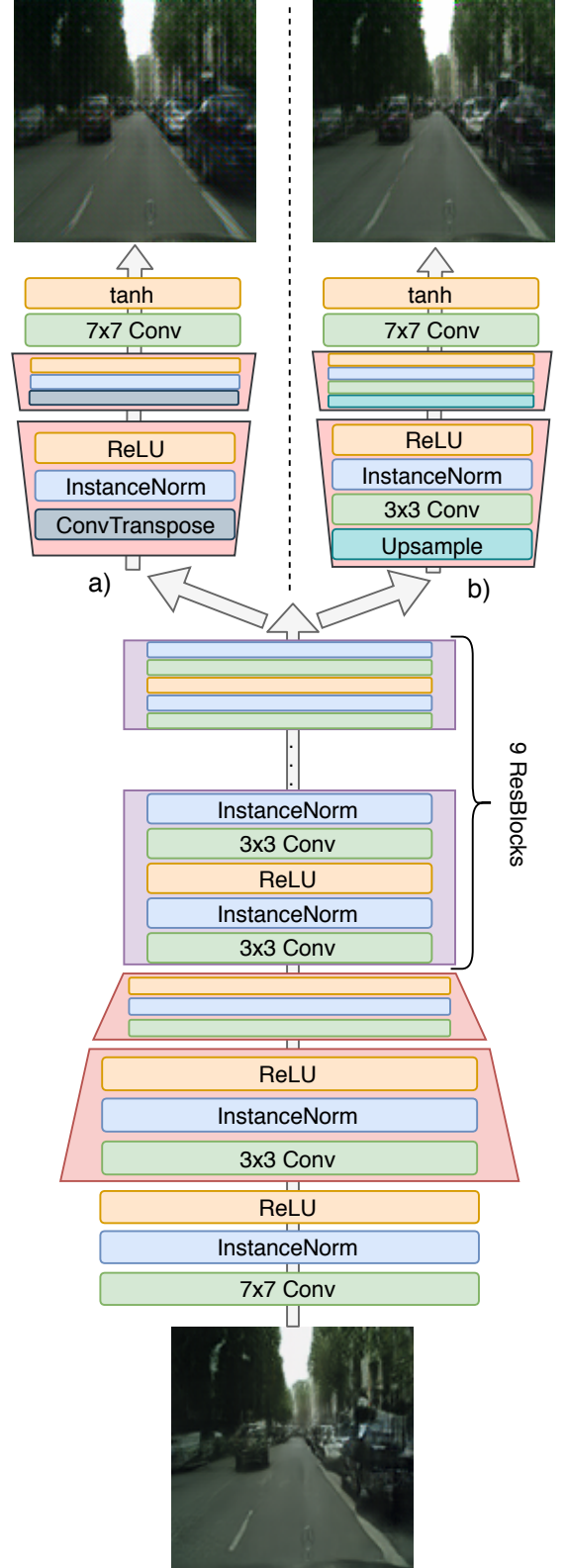


Fig. 2. Generator structure of the two different DeblurGAN-based enhancement architectures. a) Regular structure as introduced by Kopyn *et al.* [22]. b) Modified structure according to the suggestions of Odena *et al.* [31].

The SRGAN loss is similar to the DeblurGAN loss (cf. Eq. 2). It also contains both an adversarial and a content loss term. The content loss is defined as in Eq. 3 as the L_2 difference of the feature maps of a specific VGG-19 layer. In image super-resolution, the task is to recover high-frequency components, therefore Ledig *et al.* [24] chose a deeper VGG-19 convolutional layer ($VGG_{5,4}$) to calculate the content loss. Similar as for DeblurGAN, the VGG-19 based content loss was chosen to learn perceptually meaningful representations for generating images of high visual quality. In the original experiments, SRGAN was able to recover high-level details in the images quite well and achieved high human rating based *mean-opinion scores (MOS)*. For the detailed structure, we refer the reader to the original paper [24].

IV. EXPERIMENTS AND EVALUATION

To evaluate the different enhancement methods for traffic scene prediction, the networks were trained on the Cityscapes dataset [7]. This dataset consists of 30 frame long 16 bit color videos, which were recorded with a frame rate of 17 fps in 50 different German cities. The training split contains 2975 videos, the test split 1525.

For generating our initial predictions, we first trained FutureGAN according to the procedure described by Aigner and Körner [1]. The network was trained to predict five output frames from five input frames, thus the training and test sets contained 8924 and 4574 sequences, respectively. To avoid any overlap between the training and test split for all further experiments on the enhancement methods, we continued using only the Cityscapes test split as a database. We separated the new dataset into an 80:20 train-test split, leading to 3659 training sequences and 915 test sequences. The original input frames of size 2048×1024 px were downsampled bicubically to 128×128 px in all cases except for the ground truth frames for the SRGAN experiments, which were downsampled bicubically to 512×512 px. All networks are implemented in either PyTorch or Tensorflow for Python.

The training was performed on a single NVIDIA TITAN X Pascal GPU with 12 GB of RAM separately for each network. We used the ADAM optimizer [21] for all networks. FutureGAN trained for 140 epochs with a gradually decaying learning rate of initially $l = 0.001$ and $\beta_1 = 0.0$. Both DeblurGAN versions trained for 300 epochs with $\beta_1 = 0.5$ and an initial learning rate of $l = 0.0001$ which gradually decayed to zero after 150 epochs. SRGAN was trained for 10 initialization epochs using the content loss and then for 300 full epochs with a learning rate of 0.0001 and $\beta_1 = 0.9$.

After training, we evaluated the different prediction methods on our test split. For the plain predictions, we used the trained FutureGAN network to generate a set of five future frames from a set of five input frames. To test the different enhancement methods, we generated five predictions using FutureGAN and then enhanced each of the five frames using the different image-to-image translation networks. In total, we evaluated four different prediction pipelines: plain FutureGAN (no enhancement), FutureGAN + DeblurGAN (transposed

convolution), FutureGAN + DeblurGAN (upsample + convolution), and FutureGAN + SRGAN. In order to get an estimate of the inference time that it takes for predicting five future frames with each of the prediction pipelines, the following list provides the average values over the whole test set on an NVIDIA GeForce RTX 2070 GPU with 8 GB of RAM:

- FutureGAN: 0.011 s
- FutureGAN + DeblurGAN (trconv): 0.019 s
- FutureGAN + DeblurGAN (ups+conv): 0.020 s
- FutureGAN + SRGAN: 0.707 s

The SRGAN needs the most time to generate five predictions, most likely due to the increased frame size of the outputs.

A. Qualitative Results

Figure 3 shows a qualitative comparison of the prediction results for two different video sequences. For each of the two sequences, we display the input frames, the corresponding ground truth predictions, the prediction results of FutureGAN without any enhancement, and the results for the three different enhancement approaches. The differences between the enhancement methods are clearly visible. When using the original DeblurGAN architecture to enhance the predicted frames, the checkerboard pattern mentioned in section III-B can be seen in Figure 3 d. Using the modified DeblurGAN with nearest-neighbor upsampling followed by regular convolutional layers reduces this pattern in the enhanced frames. In general, both DeblurGAN versions lead to an improved object appearance in all frames. Although there still remain unclear object boundaries after enhancing the frames, especially the cars and lane markings appear smoothed and straightened in comparison to the plain FutureGAN predictions. We further observed that the DeblurGAN architecture learned to generate object-specific features, such as the red colored taillights of cars (see figure 3). In contrast to that, the SRGAN-enhancement does not seem to produce a more realistic version of the predictions. The SRGAN learned to add high-frequency details to the image which do not match the original details. This effect is probably caused by the content loss that is calculated with deeper VGG-19 feature maps. Even though SRGAN also generates object-specific features such as red taillights, the overall visual quality of the predicted frames seems best after the enhancement with the modified DeblurGAN (see Figure 3 e).

B. Quantitative Results: Traditional Metrics

A traditional way to quantitatively evaluate the enhanced predictions is to calculate image comparison metrics, such as the *mean squared error (MSE)*, the *peak signal-to-noise ratio (PSNR)*, and the *structural similarity index (SSIM)*. For these evaluations, the resulting images are compared with the ground truth image of size 128×128 px, except for the case of SRGAN, where the comparison is on the increased size of 512×512 px. The average values over all five frames are provided in Table I. Additionally, we plotted the trends of the MSE, PSNR and SSIM values per predicted frame in figure 4.

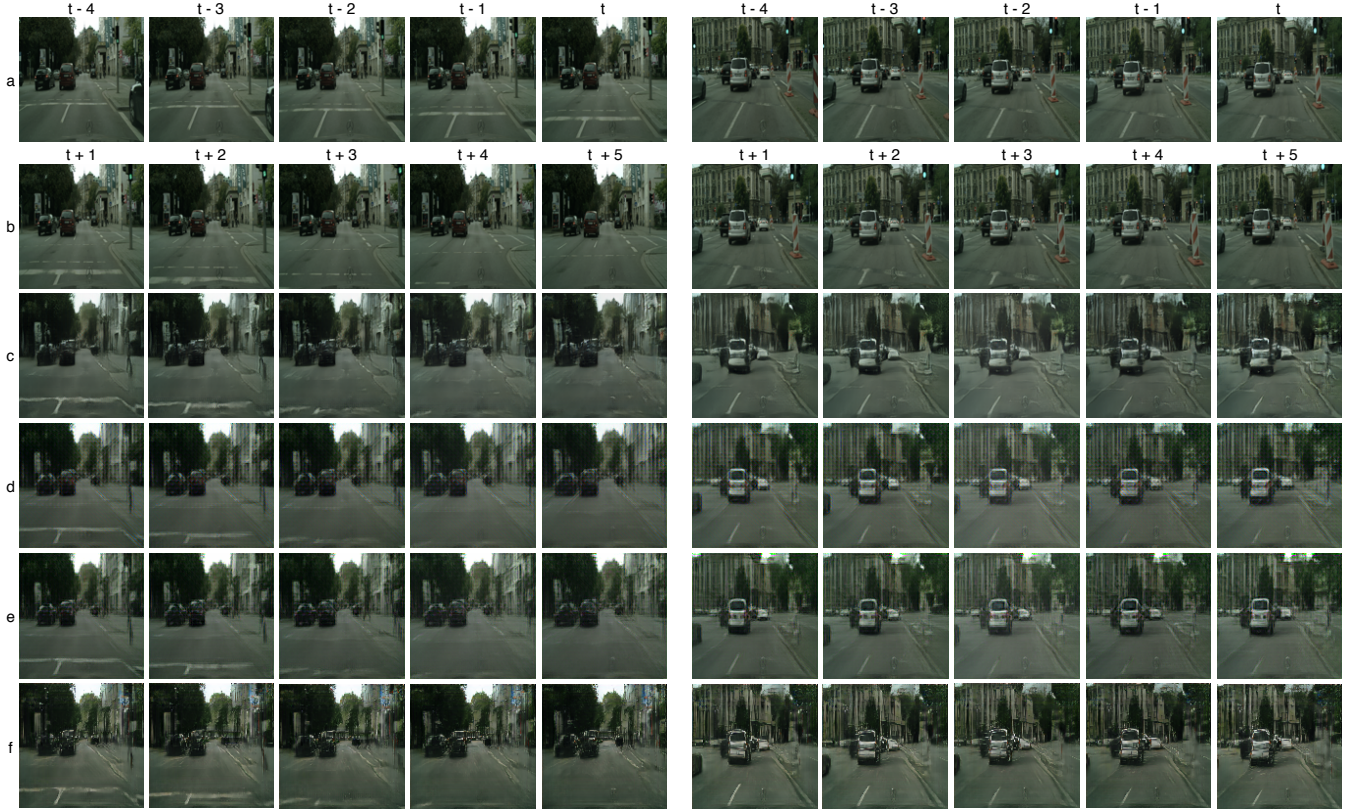


Fig. 3. Prediction results for the Cityscapes test sequences. a: Input, b: Ground Truth, c: FutureGAN [1], d: DeblurGAN [22] (transposed convolutions), e: DeblurGAN (NN-upsampling + convolution), f: SRGAN [24].

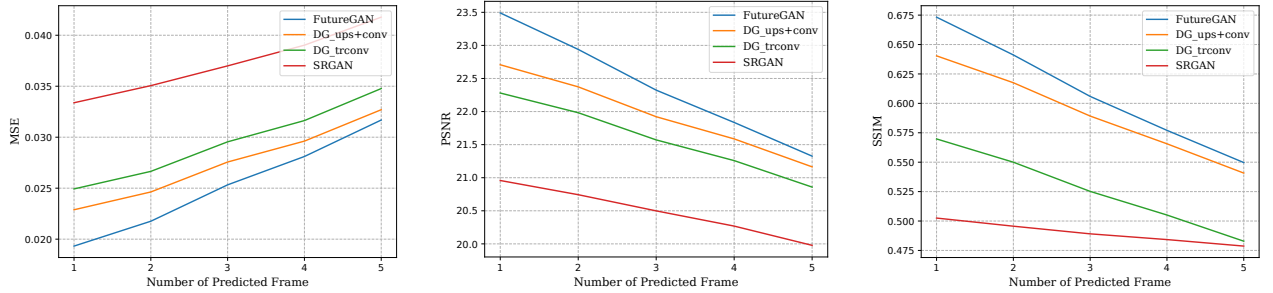


Fig. 4. Quantitative results per predicted frame for all enhancement methods (DG = DeblurGAN).

TABLE I
AVERAGE RESULTS OVER 5 FRAMES FOR ALL ENHANCEMENT METHODS
(BEST RESULTS IN BOLD)

	MSE	PSNR	SSIM
FutureGAN [1]	0.0252	22.3829	0.6094
DeblurGAN [22] (trconv)	0.0295	21.5902	0.5266
DeblurGAN (ups+conv)	0.0275	21.9507	0.5907
SRGAN [24]	0.0372	20.4893	0.4900

In general, the MSE, PSNR, and SSIM show worse results the higher the frame number. This was expected since a higher frame number represents a prediction further into the

future. Looking in detail at the values, the non-enhanced FutureGAN predictions yield the best values (lower for MSE and higher for PSNR and SSIM). Enhancing the predictions using the modified DeblurGAN version (see Figure 2 b) gives the second best results for all three metrics. When comparing the results of these traditional metrics, they seem contrary to the visual impression of the frames in figure 3. The traditional metrics can apparently not represent the human perception of improvement. Figure 3 shows this, when comparing the plain FutureGAN predictions (see Figure 3 c) to the enhanced predictions of the modified DeblurGAN (see Figure 3 e).

C. Quantitative Results: Object Detection

To quantify the perceived visual improvement of the enhancement methods, especially also in the context of traffic

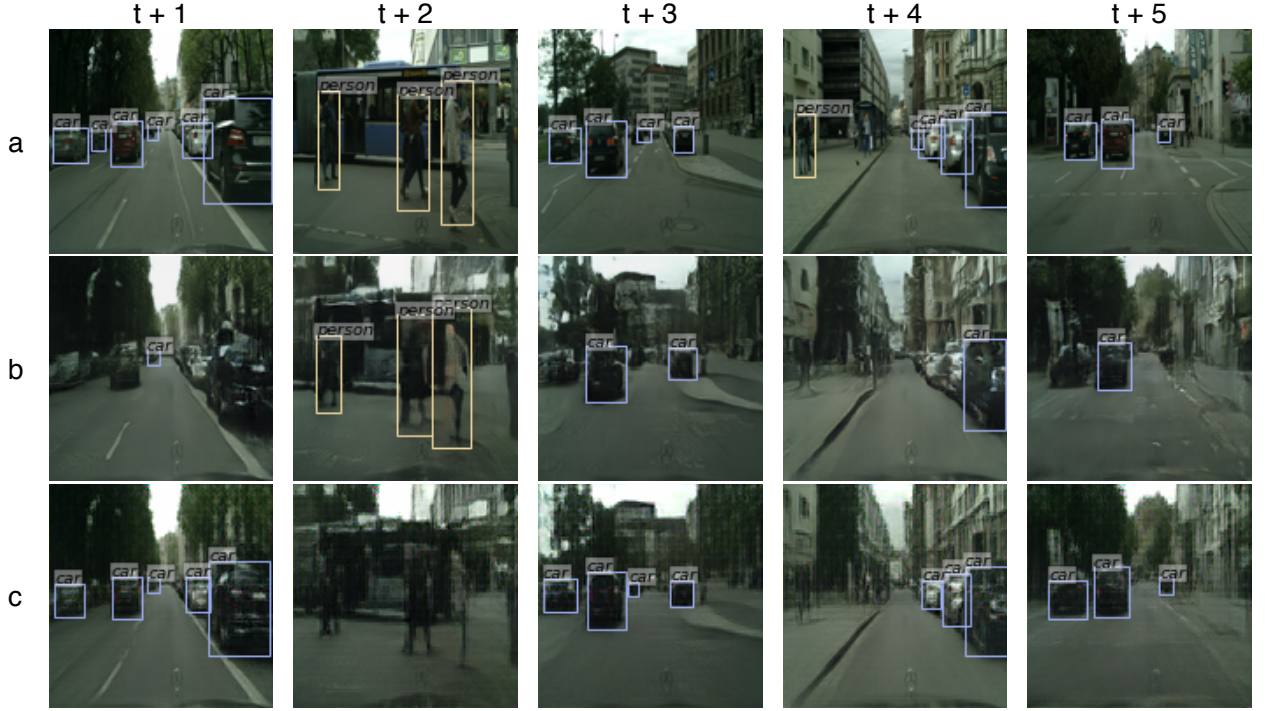


Fig. 5. Object detection results of YOLOv3 [35] from different video sequences at the five different time steps. a: Ground truth, b: plain FutureGAN [1], c: DeblurGAN [22] (upsample+convolution) enhanced prediction.

scene prediction, we evaluated the images using a state-of-the-art object detection network, YOLOv3 [35]. The network outputs bounding boxes and corresponding class labels. For evaluating the precision and recall of the object detection we take the detections on the ground truth frames as ground truth bounding boxes. This means the evaluation is relative with respect to the detection results of the algorithm on the ground truth images.

Figure 5 shows the qualitative results of the object detection network for four images. For brevity, we now only show the best performing enhancement method, the modified DeblurGAN (upsample + convolution), the plain FutureGAN predictions, and the ground truth frames. In these examples, one can see that, especially for the object class "car", the number of detections increases for the enhanced predictions in comparison with the non-enhanced predictions. However, the object detection network has problems detecting the class "person" in the enhanced images. An example of this is also shown in figure 5.

Since the class "car" is by far the most common class in our dataset, we specifically look at the average precision (AP) values of this class. We calculate the AP as defined in [11] with an IoU threshold of 50% counting as correctly detected. Figure 6 shows the development of the values per predicted frame for each of the prediction methods. For all methods, the general trend is a declining AP for an increased number of prediction steps, but the slow decrease suggests that cars are preserved well in the predicted frames. Additionally, the qualitatively best performing enhancement method, DeblurGAN (upsample + convolution), shows the

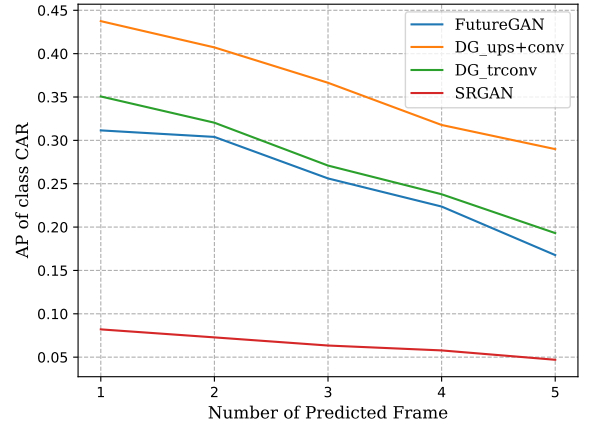


Fig. 6. Average precision of the class "car" per predicted frame for all enhancement methods (DG = DeblurGAN).

highest AP values for all five frames, which confirms the visual impression of the results. The SRGAN enhancement exhibits the lowest values throughout, which is also in accordance with the visual impression of the prediction results.

V. CONCLUSIONS AND DISCUSSION

In this paper, we evaluated the capabilities of GAN-based methods, SRGAN and DeblurGAN, to enhance video frame predictions of another generative model, FutureGAN. While, in general, motion representations and the difference between the movement of foreground and background objects are

learned by FuturGAN, the predictions suffer from blurring effects on the moving objects, leading to irregular shapes and over-smoothed object details. In order to correct these effects, we used established image-to-image translation models to generate enhanced versions of the predicted frames. The networks were trained on the Cityscapes dataset to use them for traffic scene prediction. We evaluated the different enhancement methods especially regarding their positive effects on object detection results, using a state-of-the-art object detector, the YOLOv3.

The visual quality of the enhanced predictions varies greatly between the enhancement methods. DeblurGAN shows a straightening effect, especially on car shapes and lane markings, leading to visually more realistic results. Additionally, the network learns to include object-specific features such as car taillights, which initially were averaged out in the prediction results of FutureGAN. In contrast, SRGAN mainly learns to add high-frequency features to the enhanced image, which results in unrealistic edges and patterns in the objects. These differences are most likely caused by the different content losses of DeblurGAN and SRGAN. While both networks use a very similar approach to calculate the content loss, DeblurGAN uses an earlier VGG-19 layer, SRGAN uses a deeper layer. With the low frame resolution in mind, a lower VGG-19 layer might be better for capturing the general content of the image.

We evaluated the enhanced frame predictions using traditional image comparison metrics, but they failed to resemble the visual impression and showed no improvement for any of the enhancement methods. The evaluation of the object detection capabilities with YOLOv3, on the other hand, confirms the visual impression. The enhancement method that produced the qualitatively best predictions, DeblurGAN (upsample + convolution), yielded the best detection performance. Even though pedestrians got over-smoothed by the enhancement network, possibly due to the low resolution of the images or the small number of training examples, the positive enhancement effect on cars is substantial. The average precision for detecting cars could be increased significantly in the enhanced predictions compared with the regular predictions. This verifies the application of image-to-image translation models for enhancing predictions of traffic scenes.

REFERENCES

- [1] S. Aigner and M. Körner. FutureGAN: Anticipating the Future Frames of Video Sequences using Spatio-Temporal 3d Convolutions in Progressively Growing GANs. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, XLII-2/W16:3–11, 2019.
- [2] M. Arjovsky, S. Chitala, and L. Bottou. Wasserstein Generative Adversarial Networks. In *ICML*, volume 70, pages 214–223, 2017.
- [3] P. Bhattacharjee and S. Das. Temporal Coherency based Criteria for Predicting Video Frames using Deep Multi-stage Generative Adversarial Networks. In *NeurIPS*, 2017.
- [4] P. Bhattacharjee and S. Das. Context Graph based Video Frame Prediction using Locally Guided Objective. In *ECCV: Workshop on Anticipating Human Behavior*, 2018.
- [5] A. Bhattacharyya, M. Fritz, and B. Schiele. Bayesian Prediction of Future Street Scenes using Synthetic Likelihoods. In *ICLR*, 2019.
- [6] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos. Fully Context-Aware Video Prediction. In *ECCV*, 2018.
- [7] 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*, pages 3213–3223, 2016.
- [8] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool. Dynamic Filter Networks. In *NeurIPS*, 2016.
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, pages 248–255, 2009.
- [10] N. Elsayed, A. S. Maida, and M. Bayoumi. Reduced-Gate Convolutional LSTM Using Predictive Coding for Spatiotemporal Prediction. *CoRR*, abs/1810.07251, 2018.
- [11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [12] H. Gao, H. Xu, Q.-Z. Cai, R. Wang, F. Yu, and T. Darrell. Disentangling Propagation and Generation for Video Prediction. *CoRR*, abs/1812.00452, 2018.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *NeurIPS*, pages 2672–2680, 2014.
- [14] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. In *NeurIPS*, pages 5767–5777, 2017.
- [15] Z. Hao, X. Huang, and S. Belongie. Controllable Video Generation with Sparse Trajectories. In *CVPR*, 2018.
- [16] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.
- [17] 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. In *ICCV*, 2017.
- [18] X. Jin, H. Xiao, X. Shen, J. Yang, Z. Lin, Y. Chen, Z. Jie, J. Feng, and S. Yan. Predicting Scene Parsing and Motion Dynamics in the Future. In *NeurIPS*, 2017.
- [19] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*, 2016.
- [20] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR*, 2018.
- [21] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, 2015.
- [22] O. Kopyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. In *CVPR*, 2018.
- [23] A. R. Kosiorek, H. Kim, I. Posner, and Y. W. Teh. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In *NeurIPS*, 2018.
- [24] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *CVPR*, 2016.
- [25] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual Motion GAN for Future-Flow Embedded Video Prediction. In *ICCV*, 2017.
- [26] W. Liu, A. Sharma, O. Camps, and M. Szaier. DYAN: A Dynamical Atoms-Based Network For Video Prediction. In *ECCV*, 2018.
- [27] W. Lotter, G. Kreiman, and D. Cox. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. In *ICLR*, 2017.
- [28] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun. Predicting Deeper Into the Future of Semantic Segmentation. In *ICCV*, 2017.
- [29] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *CoRR*, abs/1411.1784, 2014.
- [30] S. S. Nabavi, M. Roohan, and Y. Wang. Future Semantic Segmentation with Convolutional LSTM. In *BMVC*, 2018.
- [31] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and Checkerboard Artifacts. *Distill*, 2016.
- [32] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR*, 2016.
- [33] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (Language) Modeling: A Baseline for generative Models of natural Videos. *CoRR*, abs/1412.6604, 2014.
- [34] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro. SDC-Net: Video prediction using spatially-displaced convolution. In *ECCV*, 2018.
- [35] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *CoRR*, abs/1804.02767, 2018.
- [36] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

- [37] H. Wei, X. Yin, and P. Lin. Novel Video Prediction for Large-scale Scene using Optical Flow. *CoRR*, abs/1805.12243, 2018.
- [38] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo. Learning to Generate Time-Lapse Videos Using Multi-Stage Dynamic Generative Adversarial Networks. In *CVPR*, 2018.
- [39] J. Xu, B. Ni, Z. Li, S. Cheng, and X. Yang. Structure Preserving Video Prediction. In *CVPR*, 2018.
- [40] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV*, pages 818–833. Springer, 2014.
- [41] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. Metaxas. Learning to Forecast and Refine Residual Motion for Image-to-Video Generation. In *ECCV*, 2018.