A Transfer-Learning Approach for Few Shot Video Synthesis

Trenton Chang
tchang97@stanford.edu

Roshan Toopal
rtoopal@stanford.edu

Akash Modi
akmodi@stanford.edu

Introduction

NVIDIA has created a Vid2Vid model using GANs to take video input and generate video which matches the input in the style of training videos. [1] Vid2Vid lacks the ability to generalize to arbitrary city scenes though, as it only learns how to draw one of the particular styles seen in training. Rather than retraining the network each time with representative city images, we propose a method inspired by neural transfer. Given an input segmentation map, we minimize some distance metric between the ground truth image and the generated image.

Data

We used Cityscapes and KITTI, which are loaded into the model with a fixed width of 1024 pixels. [4, 5] All videos are 3-channel RGB. The data was preprocessed using NVIDIA's 35-class image segmentation algorithm to create segmentation masks as inputs into Vid2Vid. [6]

Fig 1. Sample Image from the KITTI Dataset [5]

Features

The original Vid2Vid model is trained using an adversarial training scheme typical to generative adversarial networks (GANs), in which the generator attempts to learn the distribution of real images given segmentation masks, optical flow between images, and previous frames, and the discriminator attempts to separate generated from real images.

In our transfer learning approach, we are refining the generator, so we use the same input features as Vid2Vid. The architecture for optical flow was already included within Vid2Vid; we found a pretrained segmentation network to generate those features.

Fig 2. Sample segmentation mask generated from Cityscapes dataset

Results

Fig 3.Vid2Vid output with Cityscapes segmentation masks as input (left); Vid2Vid output with KITTI segmentation masks as input (right), pretrained network on Cityscapes. [1]

Fig 4. Vid2Vid output using Pytorch structural Similarity (SSIM) loss function, trained for 10 epochs

Fig 5. Vid2Vid output using Pytorch Mean Squared Error (MSE) loss function, trained for 10 epochs

The above result demonstrates that our scheme of using SSIM and MSE resulted in severe image degradation and checkerboard artifacts across the image. However, frame-to-frame object tracking remains fairly realistic.

Models

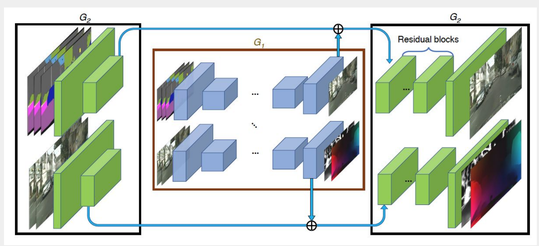

Fig. 6: Vid2Vid Generator Architecture [1]

$$p(\hat{x}_1^T \mid \mathbf{s}_1^T) = \prod_{i=1}^{T} (\hat{x}^T \mid \hat{x}_{T-L}^{T-1}, \mathbf{s}_{T-L}^{T-1})$$

Fig. 7: Markov assumption of generator

$$\underset{\theta}{argmin}\ J(G(\mathbf{s}_{T-L}^{T-1}), x^T)$$

Fig 8: Our optimization problem

Wang et. al. adopts a coarse-to-fine approach, training generators at multiple scales [1]. The outputs of smaller networks are concatenated with a selected layer in the larger-scale network. (Fig. 6)

The generator G attempts to learn this distribution **p(x | s),** where x is the generated image and s is the segmentation mask, using the Markov assumption that a frame at timestep T is dependent only on the segmentation masks and generated images L timesteps into the past. (Fig 7) We thus model G as a function x' = G(X, S): given segmentation masks S, synthesize next frame x'. We compare it to ground truth image x using our distance metric J, and optimize using ADAM based on this problem. (Fig. 8)

Discussion

Although fine-tuning a model based on a modified objective function has precedent, the results show that, the output of the network was still far from an actual dashcam video despite decreases in our loss. Quantitatively, our results show some overfitting when using SSIM over MSE.

Table 1: Results on Fewshot

Model	Loss Function	Training Err. *(per img)*	Test Err. *(per img)*
Pretrained Baseline	MSE	N/A	0.4425
Pretrained Baseline	SSIM	N/A	-0.1000
Pretrained + 10 epochs	MSE	0.0212	0.0048
Pretrained + 50 epochs	MSE	0.0128	0.0012
Pretrained + 10 epochs	SSIM	-0.2680	0.0030
Pretrained + 50 epochs	SSIM	-0.4258	-0.0001

Our choices of loss function may have also led to non-realistic results. The difficulty with training for SSIM was that loss did not monotonically decrease (Fig. 9); however, MSE has the problem that non-realistic (warped) images can achieve high MSE without looking realistic. [3]

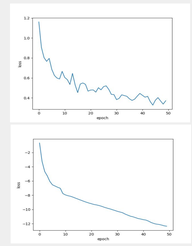

Fig 9. Loss curve over 50 epochs showing non-monotonicity; MSE (above) vs. SSIM (below).

Future

Next steps for this study would be to further investigate the creation of a custom loss function with a generated image. As seen in our results, the loss functions provided by Pytorch are not suitable for this purpose. The qualitative improvement of the output when training with the SSIM loss function for more iterations indicates that further studies can be conducted on an increase in training time.

Finding a loss function that is both convex with respect to the parameters of optimization and captures the qualitative criterion of "realistic-ness" is a formidable mathematical challenge, but could potentially yield significant performance improvements.

References

[1] Wang, T.C., Liu, M.C., Zhu, M.C. et al. "Video-to-Video Synthesis." arXiv:1808.06601 [cs.CV]. 3 Dec 2018.

[2] Wang, T.C., Liu, M.C., Tao, A. et al. "Few-shot Video-to-Video Synthesis." arXiv:1910.12713 [cs.CV]. 28 Oct 2019

[3] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[4] 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 Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] A. Geiger, P. Lenz, C. Stiller and R. Urtasun,"Vision meets Robotics: The KITTI Dataset," International Journal of Robotics Research (IJRR), 2013.

[6] Yi Zhu*, Karan Sapra*, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, Bryan Catanzaro, "Improving Semantic Segmentation via Video Propagation and Label Relaxation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019, https://nv-adlr.github.io/publication/2018-Segmentation