Video-to-Video Synthesis With Semantically Segmented Video

Jordan Hasty and Aydan Mufti

Abstract—Our project involves studying the usage of generative adversarial networks (GANs) to translate video of semantically segmented masks to photo-realistic video in a process known as video-to-video synthesis. In our study, we implement a model that is able to learn a mapping from semantically segmented masks to real-life images which depict the corresponding semantic labels. To achieve this, we employ a conditional GAN-based learning method derived from the architecture introduced by ***pix2pix***. The model produces output conditionally based on the source video to be translated. Our model is capable of synthesizing a translated video, given semantically labeled video, that resembles real video by accurately replicating low-frequency details from the source. Though the model does generate video that is subject to flickering due to not having information about previous frames, the overall setting of the scene roughly remains consistent as similar semantic maps result in similarly synthesized images, so adjacent frames in a video sequence still somewhat coherently flow into one another. More importantly, this study demonstrates how simplistic conditional GANs are capable of translating semantically labeled videos.

Index Terms—Machine Vision, Conditional GAN, Generative Adversarial Networks, pix2pix, Semantic Segmentation, vid2vid, Video-to-Video, Image-to-Image

1 Introduction

Image-to-image translation is the task of translating an image from one domain to another, using another image as the condition for the translation. In image-to-image translation, images are typically transformed so that they have the style, or characteristics, of another image. [1]–[5]

Similarly, video-to-video synthesis is the task of synthesizing video by translating a sequence of images from one domain to another [6], [7].

Video-to-video translation would have many useful applications, which makes it an important subject to be studied. Potential applications range from more efficient rendering techniques to being able to extract key features from videos of various scenes that could be useful for things such as autonomous vehicles.

Image-to-image translation has been a well-explored topic, and while video-to-video translation is simply an extension, it is not a trivial task as it presents the unique challenge of maintaining temporally coherent images in the sequence as failure to do so would potentially result in image sequences that do not appear to be related.

In our case, we apply video-to-video translation to transform images of semantically segmented video frames into photo-realistic image sequences that depict the corresponding semantic labels.

To achieve this, conditional generative adversarial networks are typically used as they allow the generation of images to be controlled by some additional input. In conditional GANs, the generator and discriminator are given some signal, such as a label, in addition to their normal inputs. This allows the networks to learn to conditionally map the random latent vector to an output. Using this method, the image to be translated can be inputted into the GAN so that an appropriate mapping from the source image can be generated. This makes conditional GANs the ideal method for this task. However, the GAN must also be given some information about previous images in the sequence for the outputted video to appear continuous and coherent, as failure to do so could result in videos that do not appear to smoothly transition or look realistic.

Our method, currently, is based on the architecture introduced by ***pix2pix*** [1]. Using this model, the GAN learns a mapping from images of semantic labels of urban street scenes to the original source images from which the labels are based. The model, as is, is able to successfully learn a mapping from the semantic labels to the source images, however, high frequency details are mostly absent in the generated images, leaving room for future improvements. Our future research will also aim to find methods to preserve temporal coherence between images in sequences so that video may be translated from semantic labels.

2 Related works

Generative adversarial networks Generative adversarial networks [6], [8]–[10] are capable of generating photo-realistic natural images by using an adversarial process. A discriminative model aims to distinguish between model distributions and data distributions. Generative models attempt to produce synthesized images that pass through the discriminative model undetected. Both models improve their methods until the generated images are indistinguishable from the real ones. GANs are used in many areas such as image generation [6], [11], video applications [12], and visual manipulation [13].

J. Hasty and A. Mufti are with the College of Computing and Software Engineering, Kennesaw State University, Marietta, GA, 30060 USA. Emails: jhasty11@students.kennesaw.edu and amufti1@students.kennesaw.edu

There are multiple types of generative image models: parametric and non-parametric [11]. Parametric models are used for generating natural images and have had recent success doing so. The study, [14], created a variation Bayesian sampling approach that would generate images.

Their results found that the estimator used for the lower bound could be optimized using a standard gradient method, however, this produced blurry images. Another study, [15], generated images using an iterative forward diffusion process that would systematically destroy structure in a data distribution and then restore it, thus creating a flexible, generative model. Using the output from the GAN study [8], which produced images that were very noisy, the study [16] found that using a cascade of convolutional networks within a Laplacian pyramid structure can generate higher quality images.

A Laplacian pyramid is a sequence of error images where each one represents differences between Gaussian layers. The linear invertible image representation is sampled at successively sparse densities [17]. This study however had difficulties with the noise created when chaining several models. Other studies such as [18], [19], which used a recurrent network and a deconvolution network approach respectively, were able to produce highly realistic natural images, however, they struggled with working with supervised tasks.

Non-parametric models are used to perform matching to a database of existing images. Usually these models will often use patches of images. Studies such as [20] performed texture synthesis using non-parametric sampling. Texture synthesis is a way to verify texture analysis methods and can be used for occlusion fill-in, lossy image and video compression, and foreground removal. [20] further describes the process of texture synthesis as growing a new image outward from an initial seed one pixel at a time. Another study, [21], used a non-parametric model to perform example-based super-resolution. Additionally non-parametric models can be used for in-painting. For example, [22] preformed image completion, filling a hole in an image, by compositing several image patches from the original image and paint out objects.

Image-to-Image translation Image-to-Image translation is the process of translating a particular image scene into another given similar training data [1]. Many attempts to perform image-to-image translations use adversarial networks [2]–[4], [23] as the network will adapt using a trainable loss function when discriminating between differences of synthesized and real images. Adversarial models are commonly applied when models are both multilayer perceptrons. Multilayer perceptron (MLP) [24] is part of a feed forward layered neural network. The three layers are an input layer, which takes in an input signal to begin, an output layer, which performs the desired prediction or classification, and a certain number or hidden layers that separate the input and output layers. Each neuron is trained using back propagation.

In a pix2pix study [1], the framework applied conditional GANs in for applications like the transformation of Google Maps images to satellite views and synthesizing handbags from drawn images. The pix2pix method [1] consists of a generator G whose goal is to take semantic maps as input and translate them to appear realistic. The discriminator D will attempt to distinguish the real images from the generated images. In a previous study [6], a U-Net was used as a generator and a patch-based CNN as the discriminator. Convolutional networks are commonly used on classification tasks where the output to an image is a single class label. U-net architecture is a type of convolutional network architecture that can be used for a fast segmentation of images. It contains multi-channel feature maps and a specified number of maps per channel. U-net learns segmentation in an end-to-end setting [25]. In addition the Cityscapes dataset [26], which provides annotated images of street view scenes, was used to train the framework but resulted in low-resolution images due to training instability, optimizations, and quality of the generation. A study by Chen and Koltun [5] proposes a solution that involves direct regression objective training which is dependent on perceptual loss as specified by [27]. The trained visual perception network contains activation layers representing increasing levels of abstraction. Abstractions can be edges, colors, or specified objects [5]. The activations are matched and applied to both the generated image and input image. This solution allowed Chen and Koltun to generate better results and are able to generate 2048 x 1024 images.

Video-to-Video Synthesis There are a variety of forms of video synthesis such as unconditional video synthesis, future video prediction [28]–[30], and video-to-video synthesis [6], [7], [31]–[33]. Unconditional video synthesis is when a generator takes in random variable inputs converts them to video [34], [35]. Video-to-video synthesis is the framework to take semantically segmented video and produce photorealistic video that mimics the original [6]. A learned synthesis model can have applications such as generating realistic videos without scene specification [7], [36], [37].

Video-to-Video translation models are often found to have several constrains such as the amount of data they require and its generalization capability. The models usually require a massive amount of images for training and, for example in the pose-to-human vid2vid, the learned model often can only synthesize poses for a single human in a training set instead of being able to generalize for any human. The study [7] attempted to achieve the ability to synthesize videos of unseen domains using their proposed few-shot vid2vid framework. The few-shot framework takes in two inputs for the generated video, an input semantic video just like vid2vid does and additionally a second input of a few example images of the eventual target domains. Using these example images they can dynamically configure the synthesizing mechanism through a novel network weight generation mechanism which is performed using a separate trained model. Few-shot resulted in successfully being able to transfer motion in any example images when synthesizing humans an successfully generating different images based on example street scenes and input semantics. It did however struggle with variations between some test domains and training domains being too different. For example, projecting poses on CG characters did not perform well.

Another study that used vid2vid, [37], focused on achieving long-term temporal consistency. Limitations that previous vid2vid models underwent was the ignorance of the 3D world being rendered and generated based on previous frames. By introducing a novel vid2vid, [37] achieved a framework that uses previously generated frames during rendering by condensing the 3D world into a guidance image. [37] defines world-consistency as a superset of temporal consistency that is temporally stables and consistent across the 3D world that the viewer is observing. Temporal consistency only looks at the consistency between frames in a video whereas world-consistency makes the output more realistic and can promote future developments where multiple viewers can observe the same scene from different perspectives. Guidance images are defined as physically-grounded estimates of what the next output frame should look like based on the current world generation. The guidance image is generated using the motion field which essentially details the "true motion" of each 3D point in the world. The guidance image also accumulates information from all the past frames creating a consistency in future generated images with the entire history of the world. Using a novel Multi-SPADE module, [37] were able to condition their model, bettering the quality of the output, and create a video generator that successfully performs video to video synthesis. Limitations their vid2vid framework encountered delt with the reliance on SfM, which caused possible failure in consistency, and possible change in time of day or lighting in the dataset. SfM is a Structure from Motion python library that performs feature detection, feature matching, and building a scalable reconstruction pipeline.

In the study conducted by Wang et al in [6], the application of conditional GANs for high-resolution image synthesis is explored. Conditional GANs have typically been limited to low-resolution and been unable to produce photo-realistic imagery. The model created within in this study is capable of generating photo-realistic 2048 x 1024 pixel images using their novel architecture.

The study improves upon the pix2pix framework introduced in [1] by using a coarse-to-fine generator, a multi-scale discriminator, and a novel adversarial learning objective function. The coarse-to-fine generator is decomposed into two sub-networks, with one being a global generator and the other being the local enhancer. The global generator produces images at a resolution of 1024x512 and the local enhancer network upscales the image by 4x, increasing the size of the image by 2x along each dimension.

The multi-scale discriminator addresses the issue of requiring a large receptive field to be effective by using 3 discriminators that share an identical network structure but operate at different image scales. The samples are downsampled by a factor of 2 and 4 to create an image pyramid of 3 different scales. Each of the three discriminators are then trained to differentiate real and synthesized images at the 3 different resolution scales. The combination of these generators allows the network to learn to maintain global consistency while also encouraging the generation of finer details.

This approach of multi-scale discriminators and coarse-to-fine generators allows for easier training since the network can be extended for higher resolution generation by adding an additional discriminator at the finest level, without having to retrain the network from scratch.

This study also improves the GAN loss function by incorporating a feature matching loss based on the discriminator, which forces the generator to produce natural statistics at multiple scales.

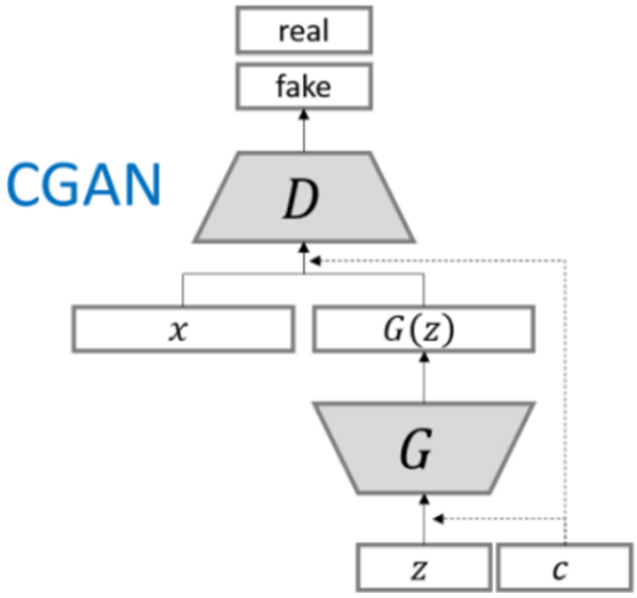

Fig. 1: Overview of the conditional GAN model [38]

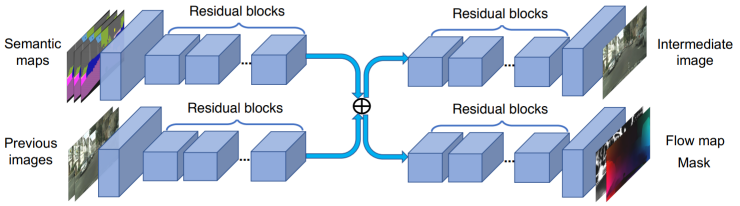

Fig. 2: Overview of the vid2vid model [6]

3 Proposed Method

Our current proposed method involves the use of conditional GANs to learn mappings from semantically labeled maps to photo-realistic images. In the conditional GAN, the generator takes as input a random vector in the latent space and the label/semantically segmented map, as shown in Figure 1.

3.1 Baseline Method

NVidia's vid2vid model [6] is built on a conditional GAN framework where G is a generator that maps a semantically segmented video input sources to an output frame sequence $x_1^T = G(s_1^T)$. In this equation, x_1^T is the "sequence of corresponding real video frames".

NVidia's vid2vid model [6] is often used as the baseline for evaluating the performance of video-to-video translation models. The model learns to map an input source sequence to a photo-realistic output frame sequence that is capable of maintaining temporal coherence throughout long sequences. The model introduced by [6] also uses a

conditional GAN framework for the task of conditionally matching video distributions. Because of the large amount of redundant information that exists between consecutive video frames, the vid2vid model proposes a model which analyzes optical flow to warp current frames into future frames.

As illustrated in 2, the model uses an optical flow prediction network to estimate the optical flow of the video frames. Optical flow is estimated using both input source images and images previously synthesized by the generator. The model also utilizes occlusion masking to handle warping over elements of the picture that will be occluded by nearer elements.

The vid2vid model utilizes a foreground and a background model to improve the synthesis performance. The model divides the image into foreground and background areas based on the semantic labels, where buildings and roads are background elements and cars and pedestrians are foreground elements. This allows for the background region to be generated through warping and so the background model only needs to synthesize occluded areas. Foreground elements have larger motions which makes estimating the optical flow a difficult task, so the foreground model has to synthesize most of the foreground content.

3.2 Proposed Model

The model used is a conditional GAN that uses a generator and discriminator network that both receive as input the semantically labeled images to conditionally affect the output of the generator, which allowed outputs to be conditioned on the given semantically segmented input labels.

The goal of the generator is to take in the input image and generate an output with the same dimensions. Additionally the basic structure of the input is the same as the output. Therefore, an encoder-decoder network is used. Within the generator is a series of layers that progressively downsamples the input data until reaching the bottleneck. The data is then progressively up-scaled to the original input dimensions in the same number of steps, as shown in Figure 3. Because most low-level information, such as edges, are shared between the input and output, skip connections are used so that this original information is preserved and is available to the rest of the network, bypassing intermediate layers. Additionally, the skip connections are unweighted, so they are effectively concatenated to the corresponding post-bottle neck layers.

The discriminator is designed to model high-frequency details by restricting attention to the structure within local image patches. The discriminator is given N x N pixel patches of the image which it determines as being real or fake. The overall discriminator output is calculated by running the discriminator convolutionally across the image and averaging the results. Using a patch-based discriminator also has the advantage of being more computationally efficient than giving the entire image to the discriminator since the patch-based discriminator has fewer parameters. It also has the benefit of being able to be applied to images of varying sizes.

While this patch-based discriminator is suitable for generating high-frequency details, such as fine edges and textures, low-frequency details are also important for the generated images to appear realistic. The low-frequency details of the image are learned through an additional L1 term that does not rely on the discriminator. This results in a higher loss for images that do not preserve low-frequency details.

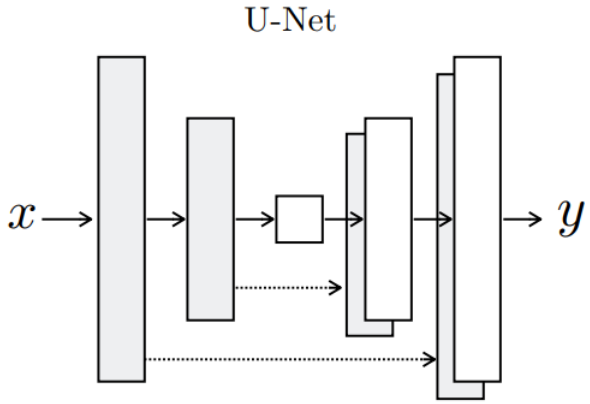

Fig. 3: Structure of the U-Net architecture [1]

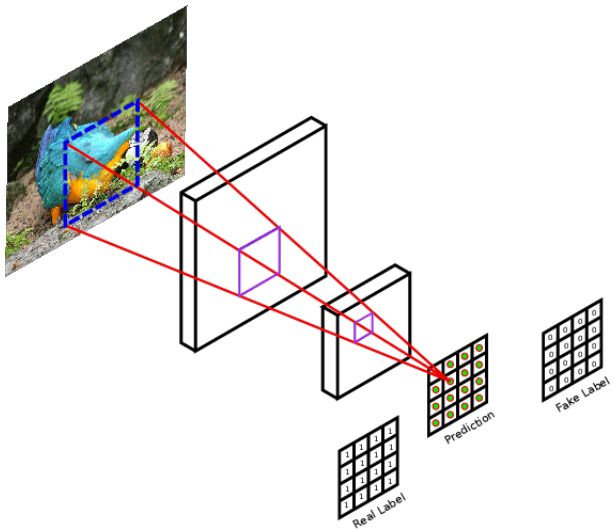

Fig. 4: Demonstration of the patch-based discriminator [9]

3.3 Implementation Details

The model is implemented in Python using TensorFlow and Keras. The model is implemented in a manner which separates the components of the conditional GAN into smaller sub-components. The discriminator and generator are defined separately, and during generator training, a composite, logical model is created by combining the two.

The discriminator is implemented as a convolutional neural network that analyzes both the source image (semantically labeled image) and target image (real or fake sample) simultaneously as they are fed in as a single concatenated image. The network then performs a series of convolutions, with layers of 64, 128, 256, 512, and 512 convolutional filters. Following each of the convolutional layers are batch

TABLE 1: Training hardware specifications

CPU Architecture	Zen 2
Model	Ryzen 7 3800X
# of Cores	8
Core Frequency	4.5 GHz
Main Memory	32 GiB
GPU	GeForce RTX 2080

normalization layers then followed by leaky-ReLU activations. Finally, the discriminator ends with a layer of a single convolutional filter with a sigmoid activation to act as the patch output for the patch-based classification that is applied convolutionally. The loss of the discriminator is calculated using binary cross-entropy.

The generator is considerably more complex, implemented using series of encoder and decoder blocks to create the U-Net architecture. The generator consists of a series of seven encoder blocks that progresses from 64 to 512 filters before reaching the bottleneck. Next, the network has another series of seven decoder blocks that regresses from 512 back to 64 filters. The generator then outputs pixel values ranging from -1 to 1 using a tanh activation.

Encoder and decoder blocks are defined within helper functions that are used in creating the generator. Encoder blocks down-sample the data and the decoder blocks upscale the data.

To create the GAN, the generator and discriminator are passed as parameters to the function which constructs a combined, logical model for updating the generator. When the GAN is created, the generator's output is fed as input into the discriminator. During back-propagation, the discriminator's weights are marked as non-trainable so that only the generator's parameters are updated.

4 Experimental Setup

The model was implemented and trained on the hardware shown in Table 1. The model was implemented using Anaconda for Python. Jupyter Notebook was used for the development and demonstration of the model.

4.1 Research Questions

The goal of our research is to answer the question of how to generate photo-realistic video given a semantically labeled source video. To find an answer to this question, we must first formulate sub-problems from which questions and answers can be found. One of these sub-questions is how to translate a semantically labeled image into a realistic image. We study this problem by analyzing works such as [1] where the authors create a conditional GAN for the purpose of image-to-image translation. Another question that then must be answered is how to maintain temporal coherence between generated images in a sequence. Following finding answers to these questions, we may then begin trying to solve our larger problem.

4.2 Dataset and Preprocessing Techniques

Our dataset used for the training process was obtained from the the Cityscapes Dataset [26]. To prepare data for training, image pairs of the semantic labels and original source images are scaled to a resolution of 256x256. The images of the semantic labels and original sources are then concatenated together, producing a single 512x256 image that contains the input and desired output.

To generate video, we had to create our own set of semantically labeled image sequences. Because the authors of [26] do not provide sequences of semantic masks, we resorted to finding videos from works which generate semantic masks given original source images. Using their video outputs, we generated sequences of images. These images are then scaled down to 256x256 and fed as input to the generator to generate synthesized image sequences. Because these semantic masks, which we used as the source for our semantically labeled video, were generated by neural networks, they are subject to some flickering, unlike the semantic masks of the Cityscapes Dataset which are labeled by hand.

4.3 Evaluation Metrics

The performance of the GAN is evaluated by analyzing the output of the loss functions used in training. For the discriminator, its loss is compared between loss from real samples and fake samples. The loss of the discriminator is measured using binary cross-entropy. For the generator, the loss is measured as a weighted average of the adversarial loss and the L1 loss. However, evaluating the performance of the entire GAN is a difficult task that is still an open problem.

In Figure **??**

Other studies involving image synthesis, such as [1] and [6], often use groups of human observers to judge the results of the synthesized images by judging whether samples are real or fake to evaluate their model's performance. Doing so can be an expensive process, so we rely on visually assessing our results.

In future studies, we would like to create a survey in which students and other peers at Kennesaw State University would be able to judge our model's performance and provide a measure of human preference of the synthesized images. The survey would likely be conducted similarly to the one used in [1] where human judges were given a number of pairs of real and fake samples which they had a limited time to view and then judge.

5 Results and Discussion

5.1 Model Evaluation

In Figure 5, we show the output of the generator after training for 10 epochs. The model is able to map specific labels to certain styles with well-defined edges. As shown in the output images, the generator properly reconstructs low-frequency detail with distinct edges and shapes, however, much of the high-frequency detail is lost. This issue is further demonstrated in Figure 6, which shows that the model is able to conditionally generate images by mapping the labels to styles, however, details within the regions of labels are not well-represented. We suspect that this could

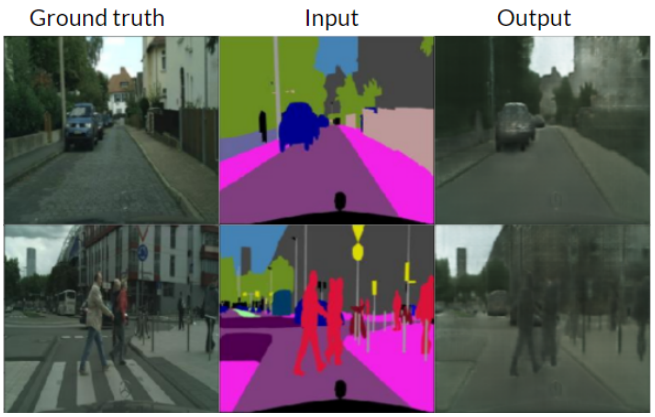

Fig. 5: Two random samples with their associated semantic labels and output created by the generator

Fig. 6: Comparison between an original source image and the corresponding generated image shows loss of high-frequency details

be due to a number of reasons, which we discuss. Shown in Figure 7 is the resulting output video given an input of a semantically segmented video at every 15th frame. Though the model does not have information about previous frames, the world appears mostly consistent throughout the video. In Figure 8, we also show the adversarial loss of the generator during training. The loss never stabilizes during training since the loss function is learned by the discriminator.

5.2 Discussion

We believe that there are a number of potential reasons why the images synthesized by our model lack some high-

Fig. 7: Screenshots from output video at every 15th frame

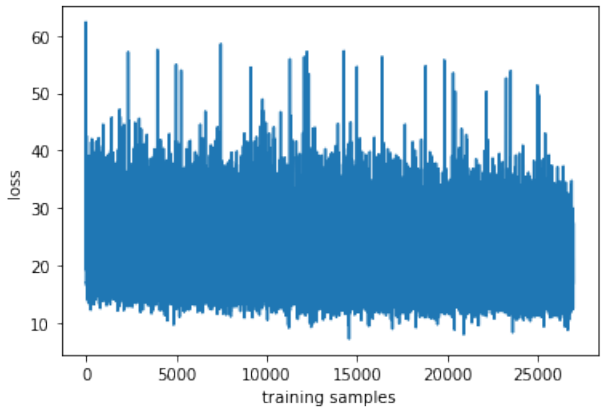

Fig. 8: Adversarial loss of the generator during training

frequency details. For one, our model may have not been trained for long enough, however, we believe that the performance of the model has plateaued after 10 epochs. We were unable to use the loss of the generator or discriminator to judge whether the model's performance was increasing since both the generator and discriminator continually improve, increasing the loss of the other network. This is demonstrated in 8 which shows that the adversarial loss of the generator does not stabilize during training. We also believed that the L1 loss term in our network's loss function could have been weighted too highly relative to the adversarial loss which could explain why the generator is capable of synthesizing images which replicate low-level details but lack high-frequency details that the patch-based discriminator detects, however, increasing the relative weighting to increase adversarial loss introduced visual artifacts into the synthesized images and resulted in less defined edges. We also attempted to improve the quality of the generated images by updating the weights of the discriminator with a higher factor since we believed that the discriminator was not learning quickly enough to be able to effectively discriminate against the synthesized outputs, however, this change did not seem to yield any better results.

6 Conclusion

Our model is able to generate low-frequency detail images given semantically labeled images of various street scenes. The current model leaves room for improvement as it currently fails to replicate high-frequency details that exist in the original source images. The synthesized images are still quite blurry. We expected that changing the weighting of the L1 loss term could potentially solve this issue as it may have been causing only low-frequency details to be replicated, however, increasing this loss term introduced visual artifacts into synthesized images and resulted in less defined edges. We also attempted to improve the quality of the generated images by updating the weights of the discriminator with a higher factor since we believed that the discriminator was not learning quickly enough to be able to effectively discriminate against the synthesized outputs, however, this change did not seem to yield any better results.

Another reason that the generated images lack detail is because the dataset used to train the model only consisted

of 256x256 images. The original datasets released by [26] do include higher quality images, however, the source images and semantic masks are not paired together, so they could not be used to train the model.

Our current model also has the issue of not maintaining temporal coherence between generated image frames given a sequence of semantically labeled images. The synthesized video is subject to some flickering as the model does not have information about previous frames, however, the overall setting of the scene mostly remains consistent since similar semantic maps result in similarly synthesizes images, so adjacent frames in the synthesized video sequences are still able to coherently flow into one another.

This issue could be resolved by creating a model with an architecture that takes as input previously synthesized frames, however, there were no available datasets that presented pairs of source images and semantic masks in sequential order, making training a model to be temporally coherent impossible. After contacting the authors of [26] to inquire about accessing their full dataset, the data we were given still did not include sequences of images or the semantic masks. After contacting the authors about this issue, they simply responded saying that they do not have the data that we needed. Though other video-to-video translation models reference [26] as the dataset that they used, we were unable to access the same dataset.

In future works, we would like to create our own dataset of sequences of pairs of source images and semantic masks. We would achieve this by utilizing game engine software so that we could generate video sequences using both the original textures and replacing the textures with solid colors that represent specific semantic labels.

Though the images synthesized by our model lack detail and are subject to some flickering, this study demonstrates how simplistic conditional GANs are capable of translating semantically labeled videos.

REFERENCES

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[2] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[3] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016.

[4] 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," 2017.

[5] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," 2017.

[6] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," 2018.

[7] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, B. Catanzaro, and J. Kautz, "Few-shot video-to-video synthesis," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

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

[9] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," 03 2018.

[10] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," 2016.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016.

[12] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2016.

[13] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," 2018.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.

[15] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," 2015.

[16] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," 2015.

[17] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[18] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," 2015.

[19] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," 2015.

[20] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *IEEE International Conference on Computer Vision*, Corfu, Greece, September 1999, pp. 1033–1038.

[21] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, pp. 56–65, 2002.

[22] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, 2007.

[23] T. Kaneko, K. Hiramatsu, and K. Kashino, "Generative attribute controller with conditional filtered generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] S. Abirami and P. Chitra, "Chapter fourteen - energy-efficient edge based real-time healthcare support system," in *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*, ser. Advances in Computers, P. Raj and P. Evangeline, Eds. Elsevier, 2020, vol. 117, no. 1, pp. 339–368.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[26] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[27] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] Z. Hao, X. Huang, and S. J. Belongie, "Controllable video generation with sparse trajectories," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7854–7863, 2018.

[29] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," 2017.

[30] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," 2016.

[31] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[32] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, "Mocycle-gan: Unpaired video-to-video translation," 2019.

[33] O. Gafni, L. Wolf, and Y. Taigman, "Vid2game: Controllable characters extracted from real-world videos," 2019.

[34] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[35] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint arXiv:1708.05866*, 2017.
[37] A. Mallya, T.-C. Wang, K. Sapra, and M.-Y. Liu, "World-consistent video-to-video synthesis," 2020.
[38] A. Mino and G. Spanakis, "Logan: Generating logos with a generative adversarial neural network conditioned on color," 10 2018.