Cycle-GANs for Cloudy-Sunny Image Translations

Ayush Baid

abaid@gatech.edu

Shreya Varshini shreyavarshini@gatech.edu Vijay Upadhya vupadhya6@gatech.edu

Abstract

We studied the task of image-to-image translation, where we learn a mapping function from an input source image to an output photo realistic image that appears to be sampled from the target distribution space, while maintaining similar content to the source image. In this project, we propose image-to-image translation between cloudy and sunny images by employing the cycle consistent generative adversarial learning framework. We start with the baseline architecture and make suitable modifications to the generator and discriminator architectures, coupled with an additional new content similarity objective. We achieve photo realistic image results on a diverse set of input formats. Furthermore, the quantitative evaluation of our generative model gave a reasonable per class accuracy of 68%.

1. Introduction

Image translation is a classical problem in computer graphics and vision. The key challenges are to capture the complex features of images in a concise, learnable model, and to find efficient algorithms for learning such models and synthesizing new image data. Thus the task of image-toimage translation can be applied to learn these mappings between an input image and an output image. In this paper we are concentrating on one such application of this task where we transform cloudy images to sunny images. One of the biggest hurdles faced by self-driving cars is their poor performance measures in bad weather conditions. Overcast weather not only decreases on-road visibility, but also opens up vulnerabilities to potential accidents. In addition to this we can also take into consideration the woes of a photographer during bad weather conditions and the countless vacation pictures that have been spoiled due to the same.

In this work we have side-tracked the basic image enhancement techniques such as contrast stretching, and instead explored the possibility of using deep machine learning models to learn the underlying content of these images and later apply content-specific enhancements (e.g. sky should appear blue in sunny images, outdoor images should have the yellow hue). Specifically, we have leveraged the convolutional net based generative adversarial networks (GANs) to learn the content structure in images and their probability distributions respectively. Furthermore, we have also incorporated an automatic quantitative measure for assessing the performance of our models using a fullyconvolutional network.

2. Related Work

Our translation problem appears related to the dehazing problem, which has been an active area of research since a long time. Cai et al. used CNN based architecture predict the haze transmission map and perform end-to-end dehazing [1]. AOD-Net internalize the transmission model in the net and directly predict the enhanced image [5]. However, out current task differs from the dehazing problem as cloudy images are more general and do not always have haze. Because of haze, we have partial occlusion of the image contents, whereas as in cloudy images we have washed out and grayed colors even without occlusion.

The literature on image color perturbation is very diverse. Deshpande et al. use variational auto-encoders (VAEs) to learn a latent space of color images, and map the gray-scale images using conditional distribution [2]. Zhang et al. take user-input as cues and use CNN architecture to generate the colored image. A major challenge in using standard re-coloration techniques is the lack of paired image datasets for cloudy-sunny images and the infeasibility to generate one.

Several deep generative models have contributed immensely to the field of image generation including GANs, VAEs, and PixelCNN. GANs in particular have recently achieved impressive results in image generation, image editing, and representation learning. Since the seminal work by Goodfellow et al. [3] in 2014, a series of GAN-family methods have been proposed for a wide variety of problems. The original GAN can learn a generator to capture the distribution of real data by introducing an adversarial loss that forces the generated images to be indistinguishable from real photos. Since then, various conditional GANs have been proposed to condition the image generation on class labels [9], attributes [10], texts [11], and images [4, 6].

In Cycle-GAN, a recent work by Zhu et al.[13], an effec-

tive model architecture for unpaired image-to-image translation has been proposed. This work introduces a cyclic consistency loss on the cascase of the two generators. By combining this loss with adversarial losses a full objective for unpaired image-to-image translation is realised.

We use GANs as a probalistic approach which can learn to sample from the desired distribution makes more sense for this problem. As cycle-GANs have been used to perform day-night translations, we will try it out in our image translation task.

3. Approach

We use the Cycle-GAN architecture introduced by Zhu et al. [13] as a reference point and perform small modifications to the architecture. In the cycle-GAN setup, there are two generators (G_A and G_B) and two discriminators (D_A and D_B) for the respective image translation tasks. The task of the generator G_A is to take an image from class B as an input and generate an image which appears to be from class A. The task of the discriminator D_A is to distinguish images of class B from the input dataset and the images generated by the generator. The model setup is to have a minimax game between the discriminators and generators.

In addition to the usual GAN loss, there is a cycle loss which enforces cycle-consistency: $G_B(G_A(B)) \sim B$ and $G_A(G_B(A)) \sim A$. [13] observed and argued that this cycle loss constraints the generated images to be similar to the input images. There is also an optional identity loss between input and output of generators to prevent the generator from introducing wild color changes and artifacts.

3.1. Generator and Discriminator Architecture

We removed the downsampling and upsampling layers in the generator defined in the cycle-GAN paper due to the following reasons:

- We wish to preserve the structure of the images and only perturb the colors. The residual blocks thus allow a short pathway for the input image information to propagate through to the generated image. This enables the convolutional layers to focus on generating the color perturbation and rely on the residual pathway for the structure.
- On addition of the downsampling and upsampling layers we observed checkerboard pattern of artifacts which we believe was happening due to the overlap in conv-transpose layers in upsampling block. This behaviour has been described in detail in [12].

We also reduced the number of parameters in the discriminator by decreasing the number of channels. Our intuition behind the reduction is that checking for sunny/cloudy images does not require a lot of unique features and 128



Figure 1. *Model components and loss functions*. In this figure we just show components and pathways arising from input image from class B. The same things also exists for class A and have been omitted for the sake of simplicity.

channels should be sufficient. Also discriminators with high model complexity might overpower the generator and might not let the generator approximate the target distribution.

3.2. Content Similarity loss

In this image-to-image translation task, we want the objects in the scene and the general structure to be the same in input as well as generates images. The original paper has an identity loss on the generated images and their inputs [13]. We extend that idea to add a loss function which penalizes the difference in the structure of the image. For this loss, we pass the generated image and its corresponding input to the first layer of a pre-trained image classification network. The first layer of image classification networks generally focuses on simple textures and its essential that our generators do not change them. We apply a MSE loss on the output of the pair of input and generated images. This loss component helps combat the appearance of random artifacts in the generated images, which are a common problem in GANs.

3.3. Final Model

The final model and the loss functions are in Figure 1. For the GAN loss, we tried the binary cross entropy loss as the discriminator is a binary classifier which tries to label the inputs as real or fake. We noticed poor training performance as indicated in [13], [8] and moved to a MSE loss. For the identity loss and cycle consistency loss, we follow the cycle-GAN paper and use L1 loss between images. For our novel content similarity block, we use L2 loss.

The generator architecture is 6 residual blocks sandwiched between two Convolution-InstanceNorm-ReLU layer. The discriminator has 4 Convolution-InstanceNorm-



Figure 2. Sample results from the baseline model. The left column are cloudy input images and right column contains the corresponding generated sunny images.

LeakyReLU layers. The content similarity block has a single Convolution-InstanceNorm-ReLU layer where the convolution weights are from a pretrained VGG-11 network and were not optimized for during the training.

4. Experiments and Results

As an initial baseline, we tried adapting the original Cycle-GAN model [13] to this task. Rather than training it from scratch, we took the winter2summer model that was presented in the Cycle-GAN paper and trained it on our sunny/cloudy images. The reasoning behind this was that the tasks are similar in that they both are changing conditions of a given image. The content of the images is largely the same, but the colors and lighting represent a majority of the change. Training this for a couple epochs already posed some problems. Sample results are presented in Figure 2. The models primarily focuses on making the sky blue, but it appears overdone fake. We can also notice the checkerboard artifacts, specially in the sky.



Figure 3. Results without the content similarity loss. Notice the blacked out regions and appearance of green artifacts

These artifacts and higher computation time for training prompted us to make the changes in the generator and discriminator architecture. Sample images are presented in Figure 3. The change solved the checkerboard pattern artifact but introduced weird artifacts. The grassy or very dark parts of the images on the left become blocked out in their sunny versions on the right. Entire parts of the brush are blocked off in this solid dark green color. We also see a bright green outline appearing on a number of objects.

To address this, the Content Similarity Loss was introduced. We tried Alexnet first and then moved to VGG11 as it generated better results. Figures 4 and 5 show some of the results from the final model.

The final model was trained using Adam over 100 epochs, starting with random weight initialization. This was based on [13], where the authors trained for 200 epochs for most of their tasks. This appears to a be a simpler problem than something like style transfer, so we chose 100. The learning rate started at 2e-4 and linearly decayed to 0 after 50 epochs. Training using the dataset took 8.5 hours in total, using a T4 GPU through Google Cloud. Figure 6 shows the loss plots, with all the different losses broken down.

The discriminator loss drops rapidly in comparison with the generator losses, effectively slowing down training. This is a reasonably common issue within GAN architectures, where the discriminator loss disappears, in a sense overpowering the generator. The discriminator is very good at telling the generated images from the real, so the generator continues to do poorly. Ideally, the loss should balance each other out in some fashion so learning can continue.



Figure 4. Final model results of cloudy to sunny translation. Column 1 and 3 are cloudy input images. Column 2 and 4 are corresponding output images images



Figure 5. Final model results of sunny to cloudy translation. Column 1 and 3 are sunny input images. Column 2 and 4 are corresponding output images



Figure 6. The loss plots during training over 100 epochs. The discriminator loss drops very quickly and is much smaller than the generator loss.

When the discriminator loss goes to zero, generator stops getting better at sampling the images from the desired target distribution.

4.1. Evaluation

Densely connected convolutional network (DenseNet) was employed in comparing the performance of our cycle-GAN against a quantitative baseline. We chose the transfer learning approach in order to initialize the parameters from a network pre-trained on ImageNet data and modify the final fully connected layer of the pre-trained network to a new fully-connected layer producing 2 responses indicative of the predicted probabilities of the two classes(cloudy or sunny). We initially trained this image classification model on our source data-set and achieved a per class accuracy of 86%. This model was later tested on an equal sized generated image dataset which resulted in a per class accuracy of 68%. Figure 7 gives us a summary of results obtained in evaluation of the two test datasets.

	DenseNet	
Metric	Source Images	Generated Images
Accuracy	83.04	68.23
Precision	79.78	61.05
Recall	90.87	75.55
F1-score	89.23	72.91

Figure 7. Summary of results obtained in the supervised binary classification task when tested on source images and generated images.

5. Implementation details

The dataset of cloudy and sunny images was downloaded from [7]. It has 5000 images for each class. We divided the dataset into train, val, and test set using 70-15-15 split, resulting in 3500 images for training. The preprocessing applied was to center the dataset.

We implemented the model in Pytorch. The authors of cycle-GAN have a public repository in Pytorch which we initially tried using their pre-trained models, which was trained on a different task. We then developed code for this project from scratch, using their code as a reference in some places. We had to redo the code as we introduced a new content similarity block, and changed the architecture to reduce the number of parameters to allow faster training.

The code is publicly available on Github at this link.

6. Conclusion

The cycle-GAN architecture with our modifications learns the color distribution of different components like sky, reflective surfaces, and the general contrast and brightness of the scene. However, the training time is too high. Also, the discriminator learns very quickly and does not generate gradients of sufficient magnitude for the generator, which stagnates the learning process. Different discriminator designs can be evaluated for this task which help overcome this problem.

References

- B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. 1
- [2] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6837–6845, 2017. 1
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information* processing systems, pages 2672–2680, 2014. 1
- [4] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 1
- [5] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. Aod-net: Allin-one dehazing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4770–4778, 2017. 1
- [6] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016. 1

- [7] C. Lu, D. Lin, J. Jia, and C.-K. Tang. Two-class weather classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3718–3725, 2014. 5
- [8] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017. 2
- [9] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1
- [10] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez. Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355, 2016. 1
- [11] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv* preprint arXiv:1605.05396, 2016. 1
- [12] X. Zhang, S. Karaman, and S.-F. Chang. Detecting and simulating artifacts in gan fake images. arXiv preprint arXiv:1907.06515, 2019. 2
- [13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired imageto-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1, 2, 3