

---

# Summer2Monsoon: Using CycleGAN for Image-to-Image Translation

---

**Arnav Garg**

Department of Computer Science  
UCLA  
arnavgarg@cs.ucla.edu

**Simranjit Singh**

Department of Computer Science  
UCLA  
simranjit@cs.ucla.edu

**Tanmay Sardesai**

Department of Computer Science  
UCLA  
tanmays@cs.ucla.edu

## Abstract

Image-to-image translation involves learning a mapping between an input and an output image. Here we consider a particular case of converting images taken in a summer setting to monsoon setting and vice versa. This specific translation finds applications like training dataset augmentation for autonomous driving systems, video conversion in film industry where waiting all year long for a different season can be cumbersome process. We employ CycleGAN for this purpose, where two GANs learn the required mapping by achieving cyclic consistency. We conduct a perceptual study to quantify the translations obtained. Our method also provides a solution to the problem of single image de-raining and we compare our results to a recent work that attempts to do the same.

## 1 Introduction

In the past couple of years, there has been a lot of research done on cross domain image to image translation, where an image is taken from one domain and then transformed into an image of the target domain. This is particularly important as a large number of Computer Vision and Machine Learning problems can be viewed as an image-to-image translation problem. For example, noise reduction can be considered a mapping between a noisy image to a corresponding noise free image. For many tasks, it is either very difficult or impossible to find paired data, and so in-order to find a mapping between different domains of images, an un-supervised setting is required. The goal of unsupervised image to image translation is to learn the mapping of special characteristics of one image collection and finding how these characteristics could be translated into the other image collection, all in the absence of any paired training examples.

In this paper, we perform image-to-image translation using cycle-consistent adversarial network, also known as CycleGAN, to translate a summer scene into a rainy one and visa versa. We also perform single image de-raining. Some of the applications of this include, helping the film industry shoot movies irrespective of the seasonal cycle and helping self-driving car researchers in data collection by transforming summer images to rainy images, which would allow them to train their models on both these environments simultaneously thereby decreasing the time needed for data collection and training.

Some of the other applications of our research include solving the problem of de-raining. There has been various research done on de-raining, as discussed in Section 2 and we believe that our proposed model would be able to solve the problem by training it to transform rainy images to summer.

## 2 Related Work

**Generative Adversarial Networks (GANs)** [2] were introduced by Ian Goodfellow in 2014 and have been very successful in image-to-image translations. GANs have one neural network that generates data while the other discriminates between real and fake (generated) data. Over time both networks get better at their tasks and the generator is now able to mimic the distribution of the input domain. GAN was used for a variety of applications and it has paved a way for series of GAN-family implementations for different applications.

**Image-to-Image translation** is the core of this project. The work that we will be exploring in this paper is CycleGAN [1]. Given two domains,  $X$  and  $Y$ , and two generators  $G : X \rightarrow Y$  and  $F : Y \rightarrow X$  then we try to achieve a cycle-consistency such that  $F(G(X)) \approx X$  and  $G(F(Y)) \approx Y$ . There is also other work done in the domain of image-to-image translation. The authors of CycleGAN previously proposed pix2pix [5] which used Conditional GANs. pix2pix also has multiple applications but one of the constraints is that pix2pix needs paired data for training. This is sometimes impossible or a really difficult task. For example in our case we would need images taken from the same exact location in 2 different seasons for our dataset. Some related work on image-to-image translation is also done by Nvidia using coupled GAN [3]. Concurrent to the work done in CycleGAN, Yi et al [6], published a paper on dual-GAN.

**Single Image de-raining** is a difficult problem to solve due to its ill-posed nature and unlike video based methods, images do not provide any temporal information. Traditionally it has been approached as considering an image  $y$  to be the sum of rain streak  $r$  and background image  $x$ , i.e.  $y = x + r$ . This is why most of the research approaches try to decompose the image into a background image and rain streak image. One of the earliest methods is sparse coding based [11] where image is decomposed using learned dictionary atoms that can sparsely represent two components clearly. Gaussian mixture models based priors [12] have also been used in image decomposition frameworks that can capture different orientations and scales of rain streaks. CNNs have also been employed to directly learn non linear mapping between synthesized images and ground truths successfully for image de-raining [10] [14] [15]. Conditional GANs are also used to achieve de-raining [4]. The work in this, and other papers on de-raining, doesn't completely solve the problem as they only remove rain from the image but some aspects of the image still look the same as the sky will still look cloudy and the roads will look wet even after removing the rain. These models only work on monsoon to summer translations but not vice versa.

## 3 Implementation

### 3.1 CycleGAN and training setup

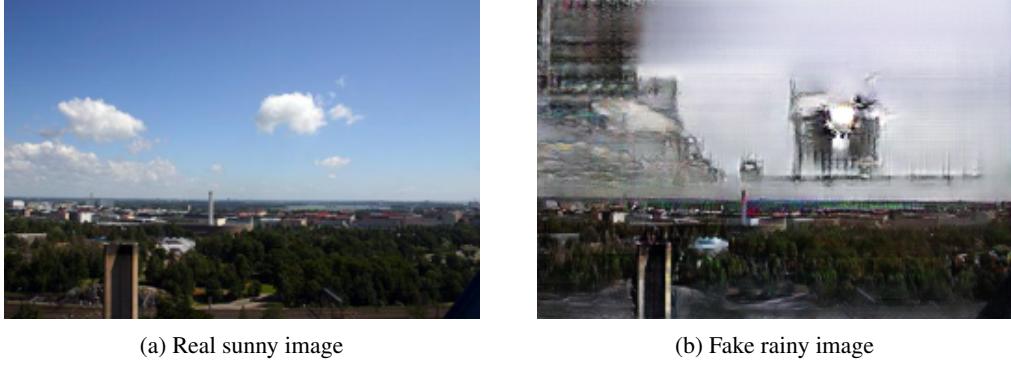
As mentioned before we use CycleGAN, which performs image to image translation, by learning the mapping between input image and output image using a training set of unpaired image pairs. CycleGAN captures the special characteristics of one domain and finds out how these characteristics could be translated into the other domain, all in the absence of any paired training examples.

For this project we will train the PyTorch implementation of cycleGAN that is publicly available on GitHub. Our dataset consists of 2 sets of images: summer images and rainy images. There is no one-to-one correspondence between the two sets. Since cycleGAN assumes some underlying relationship between the images from two sets, our first attempt will be to obtain images of the same dataset for two seasons and divide them into train and test datasets. We will show in the next few subsections how we approached this problem.

For training our models we used n1-highmem-2 (2 vCPUS, 13 GB memory) google compute engine. We also had one NVIDIA Tesla K80.

### 3.2 Attempt 1

For our first attempt, we used sunny and rainy images from *Camera as Weather Sensor: Estimating Weather Information from Single Images* [7]. This model didn't work as well as expected. On further inspection we saw some evidence of overfitting. For example as shown in 5 one can see 2 people in sky of the fake image. We attributed this to bad training data as the data didn't truly represent the real world. One of the reasons why we saw this was because the dataset is classified as sunny or rain based on an estimation that is computed using geotags and by crawling the web for weather properties.



(a) Real sunny image

(b) Fake rainy image

Figure 1: Example of overfitting during the first training

### 3.3 Attempt 2

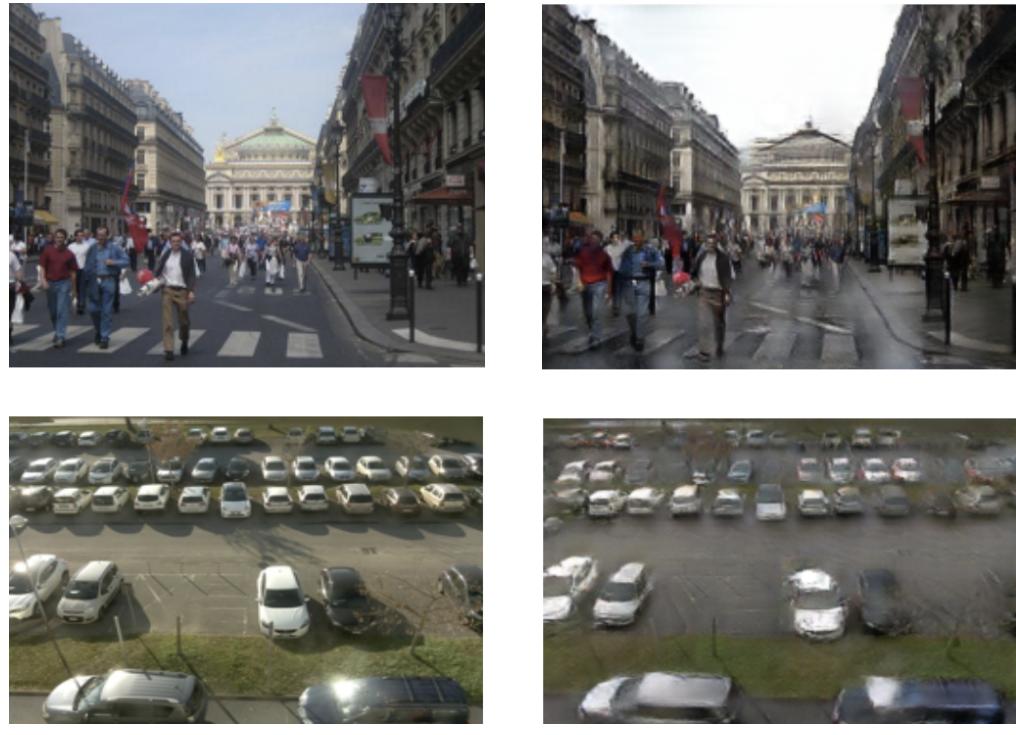
Due to the failures from our first attempt we decided to focus on a dataset which had similar images while still depicting the real world distribution. This is why we used images from *Oxford Robotcar Dataset* [8]. This led to some really good results due in most cases. One of the limitations of this dataset was that due to images taken from a camera on a car there was always a glare on the camera so sky was white in most images in both dataset. This is why while running the model on images not in the dataset we sometimes saw no change made by the Generator.

### 3.4 Attempt 3

Finally in this attempt we combined the datasets from above two attempts and *CNRPark+EXT A Dataset for Visual Occupancy Detection of Parking Lots*[9]. We then manually refined this dataset removing images that may have lead to failures in the previous attempts. This iteration of training gave us some really good results. As shown in Figure 2 and Figure 3. There were still some failures and limitations that we saw in this approach that are discussed in Section 5

### 3.5 De-raining

[10]



(a) Real sunny images

(b) Fake rainy images

Figure 2: Results from attempt 3 for sunny to monsoon



(a) Real rainy images

(b) Fake sunny images

Figure 3: Results from attempt 3 for monsoon to sunny

## 4 Results



Figure 4: Summer to Rainy Transformation



Figure 5: Rainy to Summer Transformation

We performed two different types of studies to help evaluate our results. For our image to image translation, we performed a perceptual study where we created a survey and asked participants to classify if images were real or fake and for the Single Image De-raining, as we had ground truth for these images, we computed the Structural Similarity Index Scores (SSIM) between to de-rained images and the ground-truth to evaluate how well our model performed. We have discussed both of these studies on in the subsection below.

### 4.1 Perceptual Study

Type	Total	Real	Fake
Survey 1 (Sunny)	0.591	0.825	0.543
Survey 2 (Rainy)	0.597	0.716	0.578

Table 1: Survey Results

For evaluating our image to image translation model, we conducted two online surveys, one with only summer images and the other with rainy images, in which we showed multiple fake (our model generated) and real images to the participant. In both the surveys, the participants were shown each image for 2 seconds, and then were asked to identify if the image they saw was fake or real. The results of the surveys are shown in table 1. In Survey 1, containing only fake and real summer images, we collected data from 35 participants and for Survey 2, containing only fake and real rainy images, we collected data from 20 participants. The ratio of fake to real images in both these surveys was 24 to 5.

From Table 1 we can see that for the Survey 1, only 59.1% of the participants were able to correctly guess if the image was real or fake and for Survey 2, only 59.7% of the people were able to a guess correctly. It is interesting to note that out of all the fake images in our survey, only 54.3% and 57.8% of the participants in Survey 1 and Survey 2 respectively were able to correctly classify that our model generated image was fake, which is impressive compared to other researches done in this area. Although, as the number of participants in both the surveys are very low, the result from this empirical data may or may not be close to the true result. We aim to continue our evaluation and collect more data to have a more accurate result in the future.

## 4.2 Structural Similarity Index (SSIM)



Figure 6: Results- Comparison with Fu et. al and Ground-truth

To evaluate our image-to-image model for de-raining, we compared our model generated de-rained image to the ground truth value and the results presented by Fu et al, in their research paper "*Clearing the Skies: A deep network architecture for single-image rain removal*" as shown in Figure 6.

Image	Ground Truth	Paper (Fu et. al)	Ours	Rainy Image
Umbrella	1	0.88	0.83	0.80
Rabbit	1	0.85	0.765	0.767
Bird	1	0.82	0.702	0.55
100 images (avg)	1	0.89	0.80	0.62

Table 2: SSIM Scores

Table 2 shows the SSIM scores for the ground truth, Fu et. al model, our model and the noisy image. It can be seen that our model performed comparable to Fu et. al model however wasn't able to beat their accuracy. Our understanding is that our model fell short because our training dataset contained rain noise that were only vertical strokes of rain. When we tested our model on Fu et. al's testing dataset, which contained diagonal rain strokes, our model couldn't outperform their model. Due to time constraints we weren't able to generate our desired dataset and train our model again. For our future work, we plan on training our model again with a more diverse training dataset, to see if it helps improve our accuracy and outperform Fu et. al's mode.

It also surprising to see that the SSIM score for the rain image of the bird was lower than the SSIM score of our model generated de-rained image and we plan on looking further into why that might be the case as part of our future work.

## 5 Limitation & Discussions

TODo

## 6 Conclusion & Future Work

We have employed the CycleGAN model for the purpose of translating images from summer to monsoon and vice-versa. We were successfully able to train the model to minimize the proposed loss function on our final dataset collected from various sources. We conducted a survey to see if users can identify fake images as real and achieved results comparable to what is discussed in CycleGAN paper. We also employed CycleGAN to de-rain the images as well as add rain to images and evaluated the results based on SSIM scores and compared to some of the previous works.

In future, we want to explore the model used in [3] that involves shared latent space and believe that it would give better results than our current model. In our model the generator loss did not converge but overall loss did and that is something we are keen on analyzing. The loss proposed in CycleGAN is weighted sum of different losses and we believe that these weights can be tuned to achieve even better results.

## References

- [1] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In International Conference on Computer Vision (ICCV), to appear, 2017
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Ben-gio. Generative adversarial nets. In NIPS, 2014.
- [3] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. arXiv preprint arXiv:1703.00848, 2017
- [4] H. Zhang, V. A. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. arXiv preprint arXiv:1701.05957, 2017.
- [5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In CVPR, 2017
- [6] Z. Yi, H. Zhang, T. Gong, Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In ICCV, 2017
- [7] Wei-Ta Chu, Xiang-You Zheng, and Ding-Shiuan Ding, "Camera as Weather Sensor: Estimating Weather Information from Single Images," Journal of Visual Communications and Image Representation, vol. 46, pp. 233-249, 2017.
- [8] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", The International Journal of Robotics Research (IJRR), 2016.
- [9] Amato, Giuseppe and Carrara, Fabio and Falchi, Fabrizio and Gennaro, Claudio and Meghini, Carlo and Vairo, Claudio, "Deep learning for decentralized parking lot occupancy detection". Expert Systems with Applications, vol 72, pp. 327-334, 2017
- [10] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the Skies: A deep network architecture for single-image rain removal," ArXiv eprints, Sep. 2016.
- [11] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 1742–1755, 2012.
- [12] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2736–2744.
- [14] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 633–640.
- [15] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Joint rain detection and removal via iterative region dependent multi-task learning," Corr, vol. abs/1609.07769, 2016. [Online]. Available: <http://arxiv.org/abs/1609.07769>