Learning from Simulated and Unsupervised Image Data

M. Brian Curlee
ITCS-5156 Fall 2021 - Lee

December 16, 2021

Abstract

This report is presented as a survey of a previous work[SPT+17]. Any assertions made within are subjective and do not represent those of the original author.

It has been said that a picture is worth a thousand words. A thousand pictures, however, is becoming more and more common in the realm of data science. With the increase of size in modern data sets, labeling and classifying raw data by human hand is inefficient and even improbable. But how can the labeling of these large data sets be automated and what are scientist currently surveying via Generated Adversarial Networks (**GAN**)? This paper seeks to explore some of the current implementations and surveys, as well as discuss a related implementation based on these works.

1 Introduction

In 2017, researchers from Apple.inc began a study to produce simulated and unsupervised image data sets, that would make the task of classifying eye gaze data more efficient. Shrivastava et al.[SPT+17] propose using existing GAN structure and the UNITYEYES [SPT+] application to create just such an implementation that produces simulated and refined eye gaze images, that can be used in place of, and to classify real image data. Their implementation is known as SimGAN, and it proposes an automated, simulated pipeline for classifying image data.

1.1 Problems and Questions Concerning Simulated Image Data

Often in the creation of simulated data, images become unrealistic or over produced. The process is often very resource intensive and simulators are unable to lessen the gap between simulated and raw real image distributions. This leads to either unrealistic data that is difficult to generalize or over fit data that does poorly when analyzing the details of real data.

The solution proposed by Shrivastav et al. uses the **Simulator**, in this case UNITYEYES, a **Refiner**, that adjusts simulated images by comparison with unlabeled raw real images, and a **Discriminator** that is fooled and adjusted to classify simulated data as real(*see Figure 1*). Related works, the target implementation, and my own pedestrian implementation will be discussed in the sections that follow.

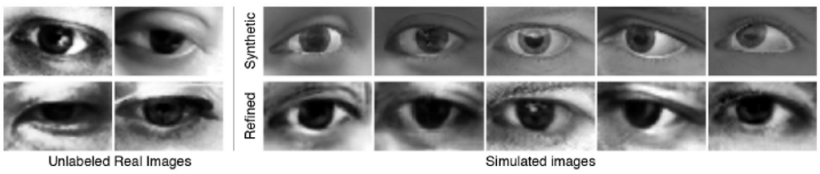

Figure 1: This image shows the comparison of real, simulated and refined images[SPT+17]

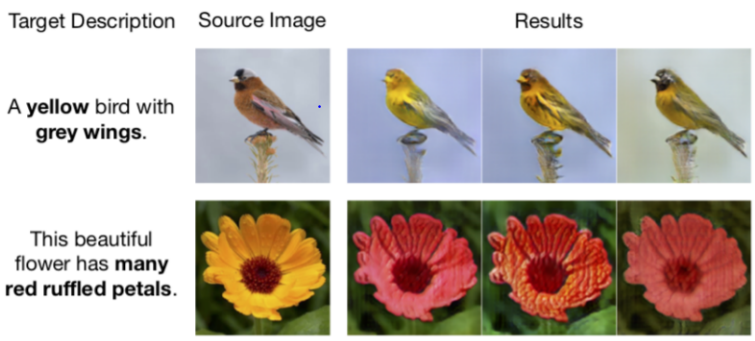

Figure 2: This image demonstrates the efforts of Yu et al. [YDL+19]

2 Related Works

2.1 SIMGAN: Photo-Realistic Semantic Image Manipulation Using Generative Adversarial Networks

In 2019, at the International Conference on Image Processing (ICIP), Yu et al[YDL+19]. presented their work, focused on manipulating raw images to match semantic descriptions of a proposed outcome(*Figure 2 above*). Their survey, known as Semantic Image Manipulation (SIM), creates a modified image from a textual description, that attempts to maintain the integrity of the image, and its labeling features, separately from the linguistic features of the text. The key challenge of SIM is to learn a mapping between these features.

Similar to the work by Shrivastava et al.[SPT+17], Yu and associates created a model that uses a descriminator and a generator(simulator). In this case, however, the refinement of the simulated images is handled by a three step process, in the generator.

In step one of this process, the raw image and description are passed through an encoder. The image data is extracted by the image portion of the encoder. This portion of step one is used to maintain the original distribution of the data. The target textual data is also encoded in this step, and concanted to the image data for step 2.

In step 2 of this process the encoded data is sent to a "residual block". The purpose of this step is to produce correlations between the two data descriptions that can be later used for the purpose of training. The output of step 2 becomes the input of step 3, the decoder, which actually produces the images as seen above.

Results of this survey produced high quality images that closely related to the supplied textual descriptions. The quantitative results showed that this implementation of SIMGAN performed better than previous implementations in categories such as, image sharpness, semantic accuracy, and consistency with raw image data.

Yu and associates propose future studies of this model to address semantic manipulation of other data sets, as well as, the production of manipulated images containing more complex objects or background.

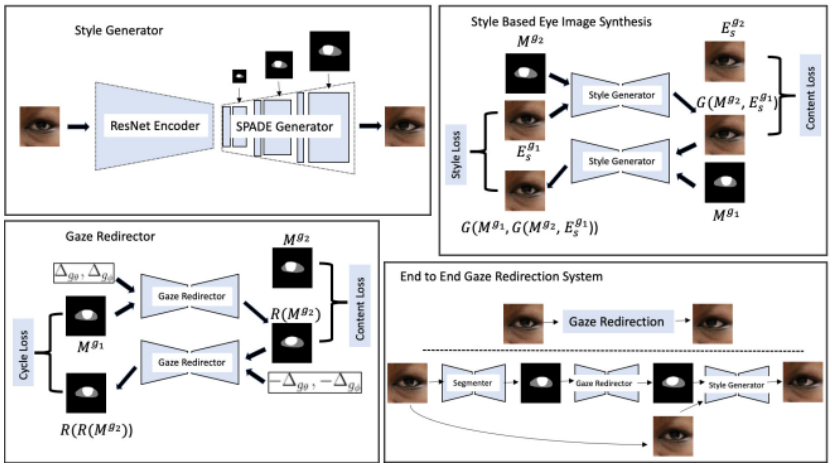

Figure 3: This image shows the work by Kaur et al. [KM21]

2.2 Subject Guided Eye Image Synthesis with Application to Gaze Redirection

In 2021, at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, researchers Kaur et al[KM21]. explored the possibilities of eye gaze manipulation, where real images were altered to produce different gaze format. The goal of this survey was to produce stylized or prescribed gaze images with a certain directional parameter. Researchers propose that this type of implementation could be used in virtual conferences, where participants appear to be looking away from viewers due to camera location. This process has been termed as Gaze Redirection by the authors.

In this model the network is trained using images from the same participant in four different "poses". The eye data is then sent through several generation, synthesis and redirection phases to produce images relevant to the required outcome.

As seen in Figure 3 above, the raw image is first fed through a ResNet encoder. This encoder captures the existing gaze data and style of the image. Next, the image is fed through a style generator, whose purpose is to calculate mask data at different image scales. Each of these images is then sent through the gaze redirector and trained on the masks captured in the previous step. The output of this generator is proposed to be an equivalent image focused in a different direction

Results of this survey found that, through this cyclic process, images can be manipulated or synthesized to meet a prescribed style and direction. Related work by Kaur et al. was able to produce outcome that removed eye glasses from participant images.

2.3 In Relation

Both of these works demonstrate the idea of simulating or manipulating raw data, through the use of GAN. Both implementations are modeled similarly to the target, in that data is generate/simulated and modified with loss functions, that correlate to those of real data. The purpose for these losses, in all cases, is to keep integrity of the data as it passes through the pipeline, as well as to avoid over fitting of simulated data by producing images that are "too good".

3 The Target of this Survey

The implementation of this work was found and slightly adapted from Kaggle, specifically the Eye Gaze repository[Sou19]. Several implementations were found and adapted for use on the hardware I had available. I should note that many of these test took several hours to complete, so incomplete or singular data is attributed to this time constraint. Project Repository.

$$\mathcal{L}_R(\boldsymbol{\theta}) = \sum_i \ell_{\text{real}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathcal{Y}) + \lambda \ell_{\text{reg}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathbf{x}_i),$$

Figure 4: This equation demonstrates the Refiner equation. [SPT+17]

$$\mathcal{L}_D(\boldsymbol{\phi}) = -\sum_i \log(D_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}_i)) - \sum_j \log(1 - D_{\boldsymbol{\phi}}(\mathbf{y}_j)).$$

Figure 5: This equation demonstrates the Discriminator equation. [SPT+17]

3.1 Pedestrian Explanation: In My Own Words

The purpose of my survey into this topic, and the work by Shrivastava et al.[SPT+17], was basic exploration and the enactment of concepts learned through out this course. There is no better teacher than the work of one's own hands.

First, data sets are growing at a rate matching, or even surpassing, the advancement of hardware resources. Cataloguing such data by hand is, at best, impractical and, more realistically, impossible. With this in mind a solution to these growing data sets has been proposed. Instead of labeling raw data, data should be simulated, created with applicable features and labeling contained.

With the increase in hardware capability, applications are available that can create and label such data, much more quickly than it could ever be labeled by hand. For this implementation UnityEyes is the simulator. Images simulated are refined by adjusting loss percentages between the real and simulated images. A discriminator, which is fed sets of both real and simulated images, is fooled in to classifying simulated images as real. The purpose of this is to make classification of real unlabeled images probable.

3.2 Simulating and Learning

As the goal of this survey is to create a real looking image, while keeping annotation data created in simulation, the first and most important step after simulation is refinement. In Figure 4, θ represents the parameters to be kept intact, $\tilde{x}$ represents the refined image and x a simulated unrefined image.

The first part of Figure 4 attempts to add realism to the simulated photos, while the second part preserves the annotations created in simulation. The culmination of these two losses smooths the differences between categorized real images and the simulated, while avoiding the artifacts that are normally found in simulated data.

In the Discriminator (Figure 5), updates the parameters by minimizing loss, in much the same way loss is handled with cross-entropy for two classes. In this equation D represents the probability of the image being simulated, while 1-D, that of the image being real. This is implemented via a ConvNet with the last layer predicting the the mentioned outcome. Of note, in the ConvNet ReLu activation is used at all layers.

In addition to smoothing losses in this fashion, the SimGAN model here sends a partial batch of the refined images to a buffer. The purpose of this is to keep a sequence of refinement over time. At the discriminator stage, when images are fed to the discriminator, a partial batch is fed from the buffer and another portion from the current refiner. This avoids spikes in image data and yields a very smooth output. In addition, images are taken on a pixel level in groups, as opposed to the whole image. Each group of pixels is then classified as either real or simulated, leading to an over all summation for the entire image. Resulting timeline example can be seen in Figure 1 above.

Shrivatava et al. find that this method is effective in creating images, with labeling intact, that can be used to classify real image data. Losses in their implementation dropped dramatically by 23 percent, while even human subjects could no longer tell the difference between the real and simulated images[SPT+17].

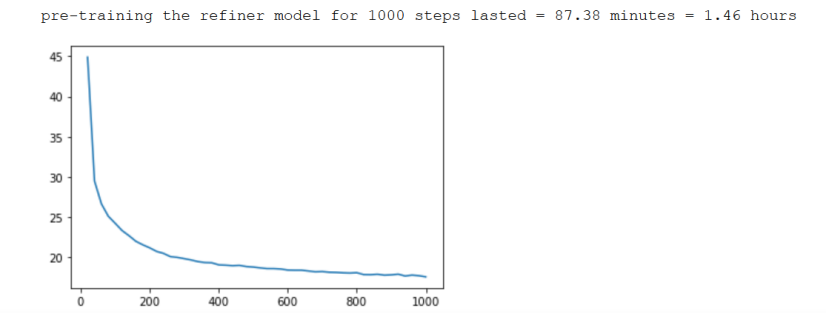

Figure 6: My refinement implementation results

4 My trial Implementation

After much tinkering, and many hours studying error code online, I was able to produce output at each level of the survey. In the Kaggle repository, both simulated and real data sets were provided for testing. This expedited my task, as I did not have the extra processing of simulating the data.

My run of the refinement pretraining, yielded results similar to those of Shrivastava et. al. After one thousand iterations I produced losses of below 20 percent, though I halved the number of iterations in the original implementation.

Similarly, in my pretraining of the discriminator model, I achieved similar results to those of Shrivastava and associates, in this case I lowered my iterations by 80 percent to just 200. My losses produced a similar graph to the previous, and quantitatively demonstrated 59 percent loss. This is to be expected as in the pretraining phase, especially since, I assume, that I lowered the number of iterations so dramatically.

In my full training phase, I was unable to get the same results as the target survey.I lowered the number of iterations by half and this still took several hours to complete. I assume that by lowering the number of iterations overall I am only eliminating the actual smoothing process found in this model. Researchers in the original, reported to have improved the loss by 23 percent, while mine remained closer to the raw simulated data. I saw an improvement of about 5 percent.

4.1 My Conclusions and Afterthoughts

I believe that given more time with this project I could have been more successful in my own implementation. I also believe that my hardware configuration and python/conda setup needs to be tweaked to actually run this type of implementation locally. I did not attempt to run this code on Google collab simply because I had already been timed out for usage due to our previous in class labs. Another concern was placing the actual databases on collab, so that they were accessible to the notebook.

Having said this, I am new to many of these concepts and happy that I was able to see some results. I believe that this method of data simulation is very important currently and for future data scientists. I even plan to give this project another shot over my holiday break to see if I can achieve better results.

I want to thank you for your organized and very systematic approach to these topics in this short semester. The labs and instruction have been crucial to my growing mastery of Machine Learning and its implementation. Thank you.

References

[KM21] Harsimran Kaur and Roberto Manduchi. Subject guided eye image synthesis with application to gaze redirection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 11–20, 2021.

[Sou19] Soundpoet. Simgan implementation using tensorflow/keras, Nov 2019.

[SPT+] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Supplementary material for:'learning from simulated and unsupervised images through adversarial training'.

[SPT+17] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[YDL+19] Simiao Yu, Hao Dong, Felix Liang, Yuanhan Mo, Chao Wu, and Yike Guo. Simgan: Photo-realistic semantic image manipulation using generative adversarial networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 734–738. IEEE, 2019.