VARIATIONAL AUTOENCODERS FOR SYNTHETIC IMAGE GENERATION

A Project Report Submitted to the
Graduate School of
The University of Arkansas at Little Rock

in partial fulfillment of requirements
for the degree of
MASTER OF SCIENCE
in Computer Science

in the Department of Computer Science

of the Donaghey College of Science, Technology, Engineering, and Mathematics

May 2021

Ojaswin Jain

This project, **“Variational Autoencoders For Synthetic Image Generation”** By Ojaswin Jain, is approved by:

Project Advisor:

- Dr. Mariofanna Milanova

Project Committee:

- Dr. Jan Springer
- Dr. Albert Baker

Abstract

This project solves the problem of insufficient x-ray image data available for deep learning models needed for development of threat detection models at airports. The data augmentation of the existing datasets is an urgent need in this field. The project uses Variational Autoencoder to fix this problem. Multiple models were studied to arrive at the final model being used to solve this problem. Existing x-ray image datasets were used to create an input data pipeline to train the deep learning model developed to solve this problem. Finally, the project describes the results obtained, i.e., sample images that were generated after the training was completed.

1. Introduction

The need for border security, especially at sensitive locations like airports is at an all time high. This has resulted in the development of multiple solutions to improve this situation over the years. These solutions range from technological investments to better personnel training and the continuous improvements of the overall security systems in place.

But security doesn't just include looking at the control of arms and ammunition across borders. There are other contrabands that are needed to be screened for at borders, not only for the safety of the travellers but also for security measures in place for the destinations. Other contrabands, such as certain food items can be part of the things that are not allowed to pass through. Other things that may be screened are bottles containing liquids and vials that may hold chemicals. These screenings may also be needed to identify possible organic material in travellers' luggage that may have a harmful effect on the environment.

A crucial tool that is being currently used at airports and other heightened security locations is the x-ray scanner. A common apparatus that most people have experienced is a long moving platform with a x-ray scanner attached to it. The security personnel watches the images generated by the equipment in real time to look for any contrabands that may be on the potential threat list.

Most of these systems at this time require a lot of manual input and require constant supervision by human operators. This has the potential to be circumvented due to either human error. By automating a lot of these processes, the potential for these contrabands to pass through can be reduced. One such improvement can be automatic detection of contrabands on the images generated by the system. Such a system allows for better monitoring of the luggage that passes through, and reduces potential human error.

To develop a system that can identify threats in real time requires applications that use deep learning models to identify these threats. By training models to identify these threats on x-ray data, applications can be developed that are able to identify the threats in real time. Training such deep learning models requires a lot of data. The data needs to be properly annotated and must be of enough variety such that most real world cases be covered. Creating such large datasets is an extremely time-consuming and expensive task. Domain experts need to look at thousands of images, and label each of them individually. In addition, these images need to be annotated to identify individual items being shown on these images.

This project tries to reduce the burden of data creation to certain extent. Data augmentation is the process of increasing the size of existing datasets. A deep learning dataset needs a large amount of data. In this case, x-ray images are used to train these models. Hence, the data that needs to be generated to be augmented is also x-ray images.

This project uses Variational Autoencoders to create synthetic x-ray images to augment the existing datasets. The process involves the use of existing x-ray image datasets to train a variational autoencoder-based, and ten generate more images that have similar features to the originals. These synthetic images can then be fed into the deep learning models to increase the size of the dataset being used in models that power the security systems.

2. Literature Survey

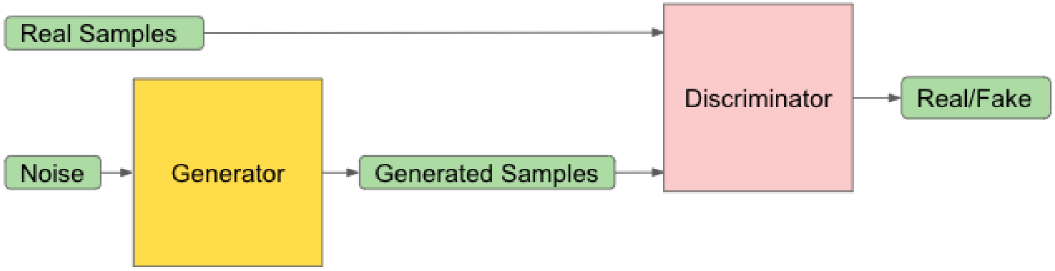
This section will include the descriptions of the various models that were researched to aid the development of the solution.

2.1. Generative Adversarial Networks(GANs)

Generative Adversarial Networks, or GANs for short, are an approach to generative modeling using deep learning methods, such as convolutional neural networks.

Generative modeling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset.

GANs are a clever way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or fake (generated). The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples.



A block diagram representing the GAN architecture

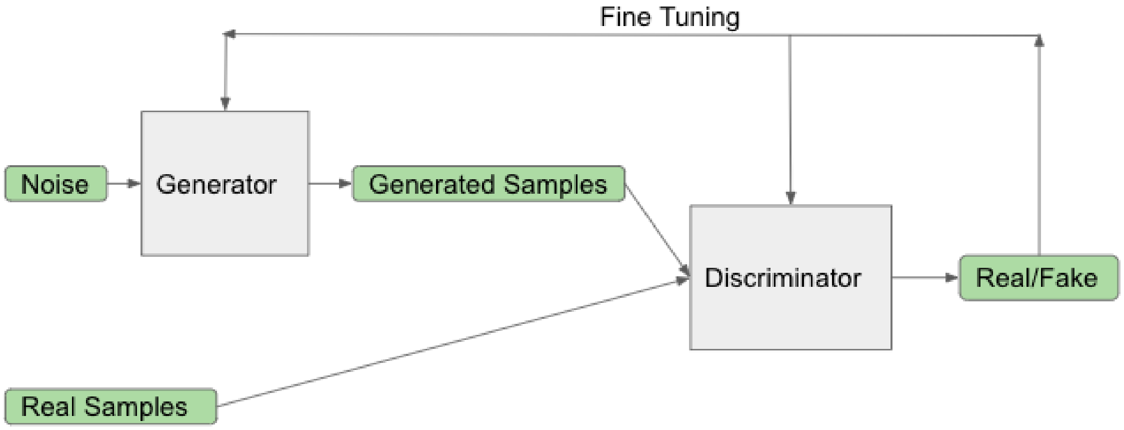
A generative adversarial network (GAN) has two parts:

- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake. As training progresses, the generator gets closer to producing output that can fool the discriminator. Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real, and its accuracy decreases. Both the generator and the discriminator are neural networks. The generator output is connected directly to the discriminator input. Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights.

2.2. Deep Convolutional GANs

A DCGAN is a direct extension of the GAN described above, except that it explicitly uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively. It was first described by Radford et. al. in the paper Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks. The discriminator is made up of strided convolution layers, batch norm layers, and LeakyReLU activations.The generator is composed of convolutional-transpose layers, batch norm layers, and ReLU activations. The strided conv-transpose layers allow the latent vector to be transformed into a volume with the same shape as an image.



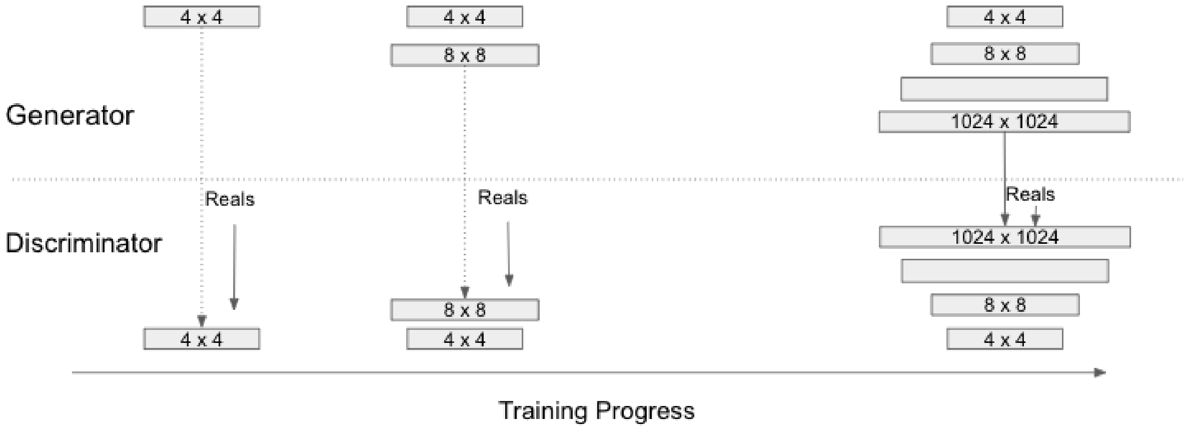
2.3. Progressive Growing Of GANs

A solution to the problem of training stable GAN models for larger images is to progressively increase the number of layers during the training process. This approach is called Progressive Growing GAN, Progressive GAN, or PGGAN for short.

The approach was proposed by Tero Karras, et al. from Nvidia in the 2017 paper titled “Progressive Growing of GANs for Improved Quality, Stability, and Variation” and presented at the 2018 ICLR conference.

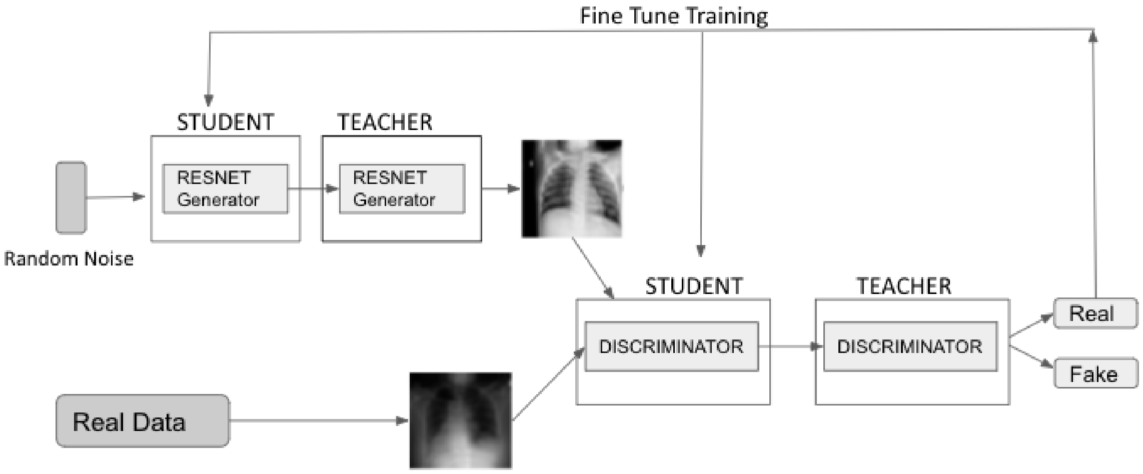
Progressive Growing GAN involves using a generator and discriminator model with the same general structure and starting with very small images, such as 4×4 pixels. During training, new blocks of convolutional layers are systematically added to both the generator model and the discriminator models.

The incremental addition of the layers allows the models to effectively learn coarse-level detail and later learn ever finer detail, both on the generator and discriminator side. This approach allows the generation of large high-quality images, such as 1024×1024 photorealistic faces of celebrities that do not exist.



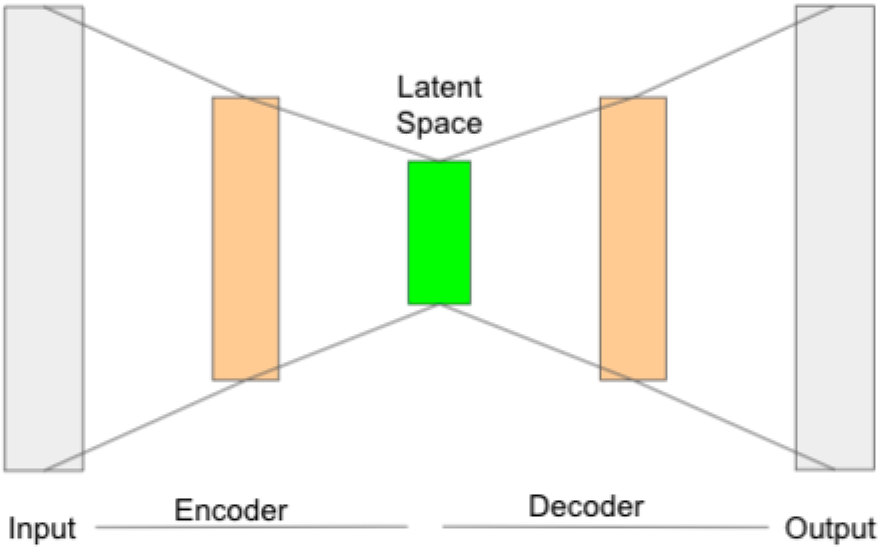
2.4. Mean Teacher + Transfer GAN

Mean Teacher + Transfer GAN (MTT-GAN) is a novel approach by a team at UMBC that is created specifically to generate COVID19 chest X-ray images of high quality. In order to create a more accurate GAN, they employ transfer learning from the Kaggle Pneumonia X-Ray dataset, a highly relevant data source orders of magnitude larger than public COVID19 datasets. Furthermore, they employ the Mean Teacher algorithm as a constraint to improve stability of training. Their qualitative analysis shows that the MTT-GAN generates X-ray images that are greatly superior to a baseline GAN and visually comparable to real X-rays.



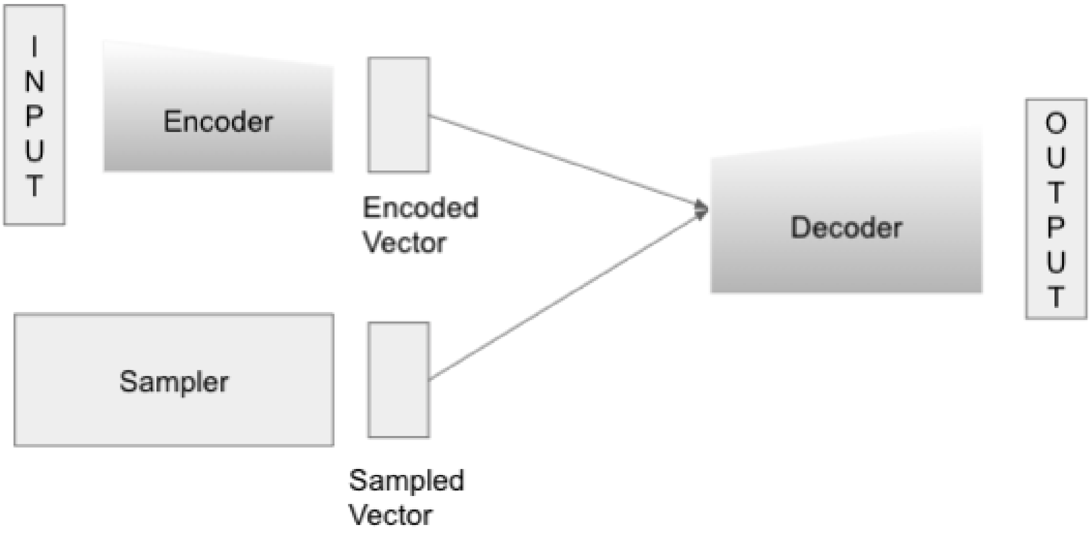
2.5. Autoencoders

An autoencoder is a special type of neural network that is trained to copy its input to its output. For example, given an image of a handwritten digit, an autoencoder first encodes the image into a lower dimensional latent representation, then decodes the latent representation back to an image. An autoencoder learns to compress the data while minimizing the reconstruction error.



2.5. Variational Autoencoders

A VAE is a probabilistic take on the autoencoder, a model which takes high dimensional input data and compresses it into a smaller representation. Unlike a traditional autoencoder, which maps the input onto a latent vector, a VAE maps the input data into the parameters of a probability distribution, such as the mean and variance of a Gaussian. This approach produces a continuous, structured latent space, which is useful for image generation.



3. Problem Statement and Proposed Solution

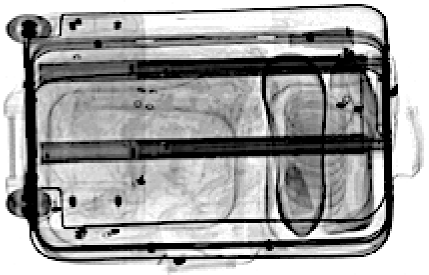
The goal is to generate x-ray images that can be used to train deep learning algorithms that can detect threat objects in baggage.

The solution proposed here is to use a Variational Autoencoder using an existing baggage x-ray dataset to create more images that share the same features to augment the said dataset. Once that is done, these images would be used to increase the total dataset size for the deep learning models.

4. Dataset

The dataset used here consists of 1877 grayscale x-ray images of baggage which contain threat objects. The database is further divided into a training set and a test set. The training set has 1313 images, while the test set contains 564 images. All these images had variations in their dimensions. To allow for the training to take place, all of these had to be transformed to a single dimension choice. Hence, the images were scaled to a dimension 224x224 and no cropping was done to retain the image features. Each image had a file size around 90 KB and the entire dataset

came out to be less than 100 MB. This reduction in file size allows for the dataset to work on cloud environments without worrying about RAM requirements.
Image samples:



5. Programming Environment

Google Colab was chosen as the environment. It allows for use of a cloud environment with support for most major deep learning libraries. This allows for easier collaboration between teams without worrying about configuration problems. I upgraded the account to the Pro version because of its support for GPU processing and better price/performance ratio compared to its competitors.

6. Model

After researching multiple models, Variational Autoencoders were chosen to solve the problem.

Encoder network

This defines the approximate posterior distribution , which takes as input an observation and outputs a set of parameters for specifying the conditional distribution of the latent representation . In this example, simply model the distribution as a diagonal Gaussian, and the network outputs the mean and log-variance parameters of a factorized Gaussian. Output log-variance instead of the variance directly for numerical stability.

Decoder network

This defines the conditional distribution of the observation p(x|z), which takes a latent sample z as input and outputs the parameters for a conditional distribution of the observation. Model the latent distribution prior p(z) as a unit Gaussian.

Network architecture

For the encoder network, we use two convolutional layers followed by a fully-connected layer. In the decoder network, mirror this architecture by using a fully-connected layer followed by three convolution transpose layers (a.k.a. deconvolutional layers in some contexts).

Training Steps

- Start by iterating over the dataset
- During each iteration, pass the image to the encoder to obtain a set of mean and log-variance parameters of the approximate posterior q(z|x)
- then apply the *reparameterization trick* to sample from q(z|x)
- Finally, pass the reparameterized samples to the decoder to obtain the logits of the generative distribution p(x|z)

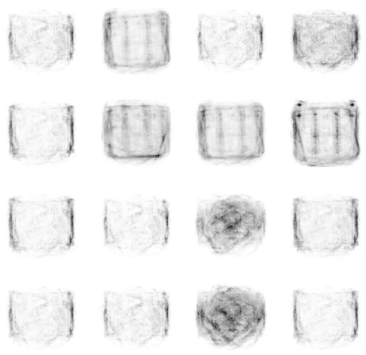
Generating images

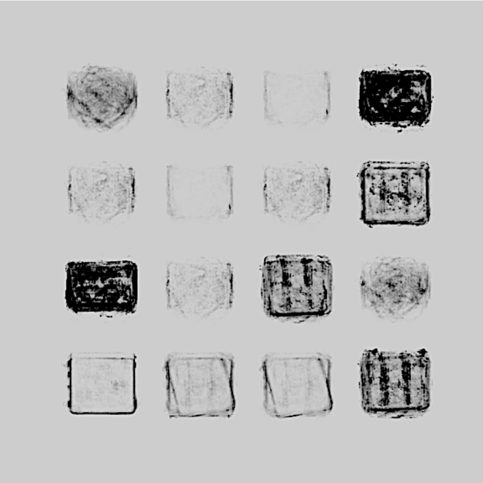
- After training, it is time to generate some images
- The process starts by sampling a set of latent vectors from the unit Gaussian prior distribution p(z)
- The generator will then convert the latent sample z to logits of the observation, giving a distribution p(x|z)

7. Experimental Results

The images generated are able to recreate the luggage enclosures, but are unable to recreate the objects at this moment. Some of the factors that are affecting the performance of the model are:

- Image sharpness is low
- Clutter in the image is high





8. Limitations and Challenges

Deep learning in general is a time and resource intensive process. This process only gets more complicated as the input dataset increases in size and the network architecture becomes more complicated. Another factor that affects the performance is the actual image size being used to train the model. In some instances, a larger image size may not work for the model, and may need to be modified to improve for the modified dataset to work.

9. Conclusion

Seeing the results, it is easy to determine that this problem is far from being solved. A lot needs to be done. Performance can be improved by improving upon the dataset input pipeline and experimenting with different configurations of the data. In addition, the model architecture can be modified to include more layers. This step would affect performance and training time will increase. Experiments can be done on the image size that is being used to create the datasets, as well doing color correction on the images to sharpen edges.

References

1. Kingma, Diederik P., and Max Welling. "An introduction to variational autoencoders." arXiv preprint arXiv:1906.02691 (2019).
2. Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).
3. Kovalev, Vassili, and Siarhei Kazlouski. "Examining the Capability of GANs to Replace Real Biomedical Images in Classification Models Training." International Conference on Pattern Recognition and Information Processing. Springer, Cham, 2019.
4. Menon, Sumeet, et al. "Generating Realistic COVID19 X-rays with a Mean Teacher + Transfer Learning GAN." arXiv preprint arXiv:2009.12478 (2020).
5. Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).
6. Creswell, Antonia, et al. "Generative adversarial networks: An overview." IEEE Signal Processing Magazine 35.1 (2018): 53-65.
7. Mery, Domingo, et al. "GDXray: The database of X-ray images for nondestructive testing." Journal of Nondestructive Evaluation 34.4 (2015): 1-12.

8. Zhao Z., Zhang H., Yang J. (2018) A GAN-Based Image Generation Method for X-Ray Security Prohibited Items. In: Lai JH. et al. (eds) Pattern Recognition and Computer Vision. PRCV 2018. Lecture Notes in Computer Science, vol 11256. Springer, Cham. https://doi.org/10.1007/978-3-030-03398-9_36

9. Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

10. https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

11. https://developers.google.com/machine-learning/gan/gan_structure