CS63 Spring 2022
Generating Labelled Satellite Imagery using a Deep Convolutional Generative Adversarial Net

Lu Min Lwin and Eduardo Aguilar

May 9, 2022

1 Introduction

In this project, we train a deep convolutional generative adversarial network (DCGAN) on the EuroSAT dataset to generate synthetic 64×64 pixel RGB satellite images conditioned on 1 of 10 land cover labels such as "permanent crop", "residential", and "sealake". Our generator model achieved an average Fréchet Inception Distance (FID) score of 116.97 across all 10 classes, performing best on the "forest" class with an FID score of 8.97 and performing worst on the "industrial" class with an FID score of 312.99. While there is an imbalance of FID scores between classes, the FID scores of the best-performing classes come close to existing FID benchmarks for DCGANs applied to image generation [12].

Large volumes of labelled satellite images are needed for training data-hungry deep learning systems for remote sensing tasks such as detecting tree species for forest management [14] or mapping land use for urban planning [5] using satellite imagery. However, ground truth labels for satellite imagery are costly to obtain and may not be available for certain under-resourced regions of the world. Thus, data augmentation techniques to increase the number and completeness of training samples stand to improve the performance of these deep learning systems. While early approaches like [16] create new training examples by applying image transforms, the use of generative adversarial networks (GANs) have been shown to yield more realistic and higher quality synthetic satellite images [9] [1].

The GAN is a framework for finding generative models through the adversarial training of two component models called the discriminator and the generator [2]. The generator upsamples noise from a latent space into a data point resembling one drawn from the real data distribution whereas the discriminator classifies whether a given data point comes from the real data distribution or is generated by the generator. In general, the discriminator will maximize correct classifications while the generator will minimize correct classifications, hence the "adversarial" minimax nature of the training process.

We implement our generator and discriminator as deep convolutional neural networks as convolutional networks have been shown to perform well on image data. For data augmentation, we want to generate satellite images conditioned on a given class label. To achieve this, we implement a conditional GAN where the discriminator and generator additionally receive one-hot label vectors in their input layers. We use FID as a metric to measure the quality of our generative model.

FID is calculated over a set of real images and a set of generated images in which a lower score corresponds to a better performing GAN.

2 Methods

2.1 Data

The EuroSAT dataset [3] is a multi-spectral image dataset spanning 13 spectral bands, consisting of 27,000 64x64 pixel labeled images of 10 classes of different types of land use and land cover, examples of class labels include 'Forest', 'Industrial', and 'AnnualCrop'. For the purposes of our project, we used a EuroSAT dataset that only consisted of only the RGB spectrums. The EuroSAT dataset was created by cropping 64x64 image patches from both the European Urban Atlas 2012 and from image data provided by the Sentinel-2A satellite which captures the surface of Earth in multi-spectral bands.

We feed EuroSAT (RGB) images into the model, but first we need to normalize the inputs such that values are in the range [-1.0, 1.0] with the intention of reaching convergence faster, as the average of training set is close to 0 [7]. Next, we create a TensorSliceDataset to store tensors such that is contains image, label tensor pairs. To finalize the dataset we need to shuffle it. This subtle yet important step guarantees that every item has the same chance to occur at any point in time training, as such, it allows for more efficient leanring [7].

2.2 Generative Adversarial Nets

GANs are a framework for finding generative models through the adversarial training of two models, a generator model G that maps a vector from a latent space to the data space (e.g. an image tensor) and a discriminator model D that classifies whether a given sample came from G or from the training data. G and D can be any differentiable function that has parameters learnable by gradient descent. They are usually implemented as deep neural networks.

We use the minimax loss function for D as originally proposed in [2] where x comes from the real data distribution and z is a noise prior sampled from the latent space. The output of D is fed through a sigmoid function to turn it into a probability because the output layer of D in our implementation uses the Leaky Rectified Linear Unit (LeakyReLU) activation [1]:

$$-\log(\sigma(D(x))) - \log(1 - \sigma(D(G(z)))) \tag{1}$$

The loss function for G is as follows[2]:

$$-\log(\sigma(D(G(z)))) \tag{2}$$

From the loss functions, we can glean the adversarial nature of training D and G. If D is a perfect classifier, $\sigma(D(x))$ evaluates to 1 and $\sigma(D(G(z)))$ evaluates to 0. In this case, the discriminator loss is 0 and the generator loss is ∞. G, on the other hand, is trying to learn (by minimizing its loss function) how to generate "good" fake data. When classifying a perfect generator's output, D would judge both real and fake samples as equally likely.

[1] We use the official TensorFlow implementation of the "modified minimax loss":

[2] Here too, we use the official TensorFlow implementation of the "modified discriminator loss"

2.3 GAN Extensions

We implement our GAN as a deep convolutional GAN (DCGAN) [12], drawing upon the strengths of convolutional architectures in classifying images. Our generator takes as input a 128-dimensional vector and upsamples it through a series of 2-dimensional transpose convolutions to output a 64×64 RGB image. Our discriminator is a traditional convolutional neural network that takes as input a real or fake 64×64 pixel RGB image and outputs a scalar value indicating the discriminator's judgement of whether the input image is a real or a generated image.

We further modify our GAN as a conditional GAN [10] by appending a one-hot label vector to the generator and discriminator inputs. Since the EuroSAT data comprises 10 classes, the generator after modification takes as input a vector in 138 dimensions rather than 128. For the discriminator image inputs, we append the one hot vector to each pixel of the image, resulting in input images with 13 instead of 3 channels. In this way, we can condition image generation by class to produced synthetic satellite imagery labelled by land cover class.

2.4 Hyperparameters and Model Architecture

For both the discriminator and the generator, we use Adam optimization with a learning rate of 0.0002 and a `beta_1` value of 0.5 as well as batch normalization for every layer (except the output layer) based on experimental results from [12]. We use batch normalization to make sure the output of each layer has a mean and standard deviation close to 0 and 1 respectively.We use the rectified linear unit (ReLU) as the activation function for the generator's hidden layers and a tanh activation for the output layer. We use LeakyReLU activation with a leak of 0.2 for all discriminator layers.

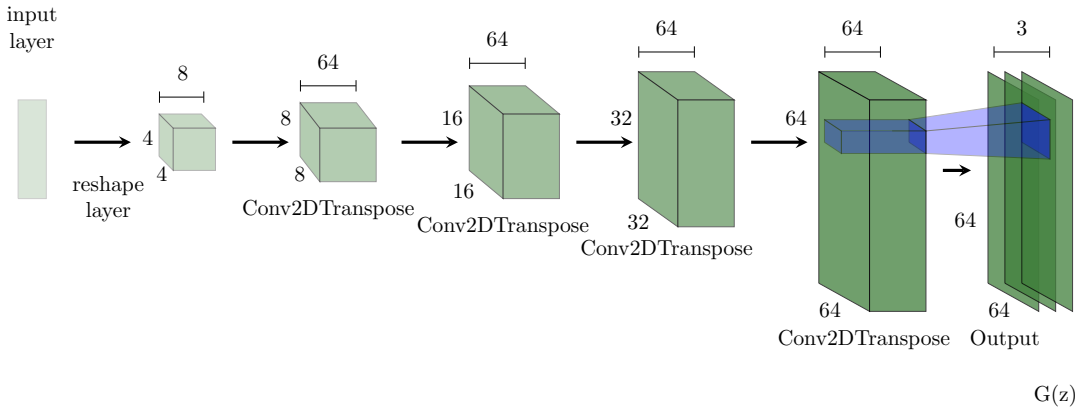

Figure 1: (Generator Architecture) We first reshape the input vector and upsample it through a series of Conv2DTranspose layers. The output layer is a Conv2D layer that outputs a 64×64 image in 3 channels.

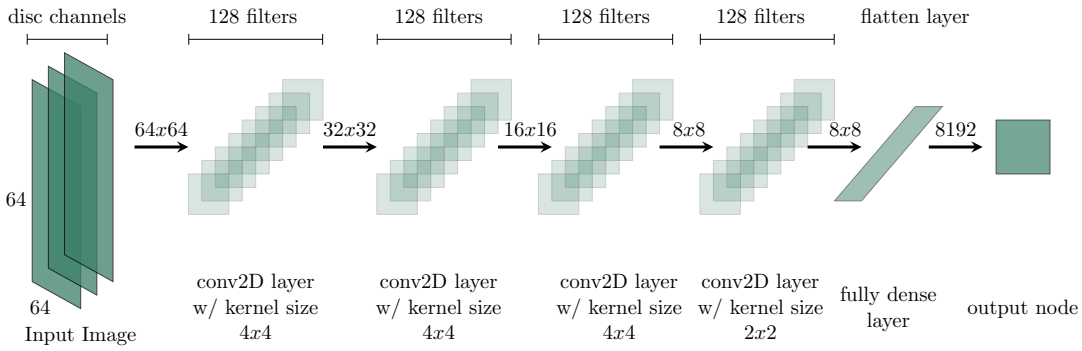

Figure 2: (Discriminator Architecture) The discriminator takes as input 64×64 images in 3 channels and outputs a scalar value indicating whether the given sample is fake or real. Not pictured here are the 10 extra channels resulting from extending the GAN into a conditional GAN.

2.5 Training Procedure

The training procedure we implemented for D and G are as follows. For each batch of training data, we train D and G one after the other in lock step. For each training image, we first generate a fake image of the same class by sampling a noise prior from a normal distribution and propagating it through G. The target labels for fake images are set to 0 and real images are set to 1. We propagate the real images through D and note the output. Similarly, we propagate the fake images through D and note the output as well. We then calculate the modified minimax loss (1) based on D's output for fake and real images. Finally, we calculate the gradient of this loss with respect to each of D's weights and update D's weights accordingly.

After training D, we train G by propagating the set of fake images through D, noting the loss based on D's output, and calculating the gradient of D's loss with respect to G's weights. We then update G's weights based on these gradients. Note here that we are not updating D's weights. This completes the training step for one batch of training data[3].

3 Results

3.1 FID Scores

Our model achieved an average FID score of 116.97 across all 10 classes, performing best on the "forest" class with an FID score of 8.97 and performing worst on the "industrial" class with an FID score of 312.99. The Fréchet Inception Distance (FID) [4] is a standard method for intrinsic evaluation of GANs. It measures the distance between a distribution of fake images and a distribution of real images. Instead of comparing images by their pixels, FID propagates images through the InceptionV3 [13] network pre-trained on ImageNet data and compares the distribution of activations in one of the deeper layers of the network.

[3]Since our DCGAN is implemented as a Keras model, we override the model's "train_step" method to implement this lock step training regimen based on and modifying code from the official Keras conditional GAN tutorial

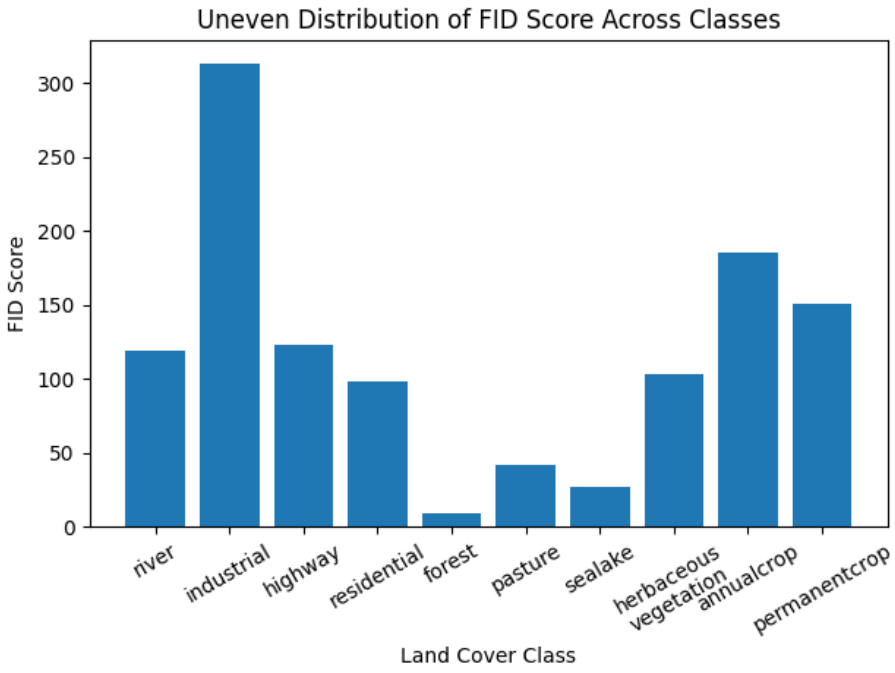

Figure 3: FID Score by Class

In figure 3, we see an imbalance of FID scores across different classes. This can be attributed to the differing complexities of images in different classes. For instance, the training images of the "industrial" class (Figure 4) consist of a mixture of buildings, green space, and concrete whereas training images of the "forest" class (Figure 6) are mostly uniformly green.

The FID scores from our top-performing classes ("forest" 8.97, "pasture" 41.51, and "sealake" 26.73) are comparable to benchmark FID values for image generation using DCGANs in [4] where their DCGAN implementation achieved a FID score of 12.5 on CelebA [8], 36.9 on CIFAR-10 [6], 12.5 on SVHN [11], and 57.5 on LSUN [15].

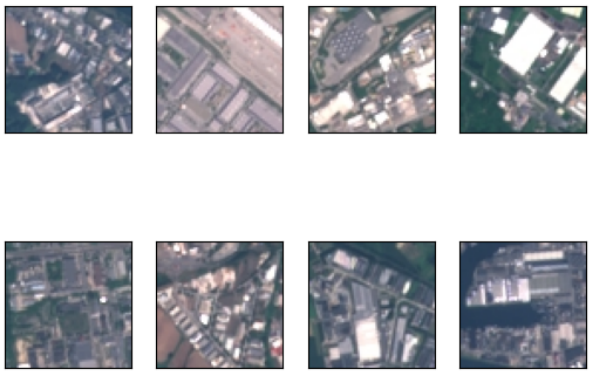

Figure 4: Real images from the “industrial” class

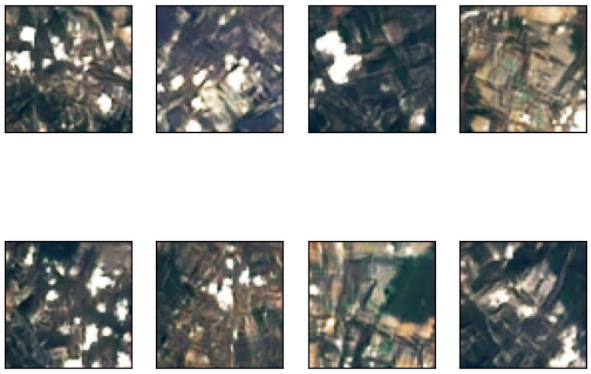

Figure 5: Generated images from the “industrial” class

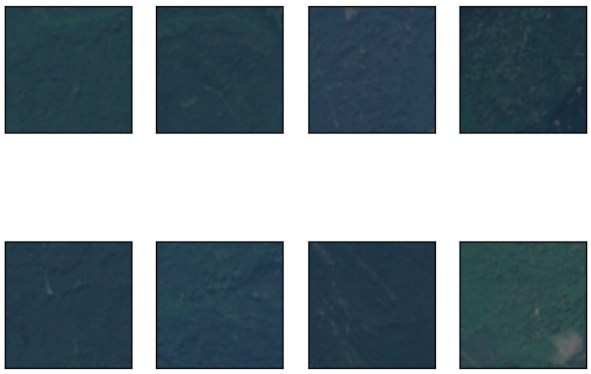

Figure 6: Real images from the “forest” class

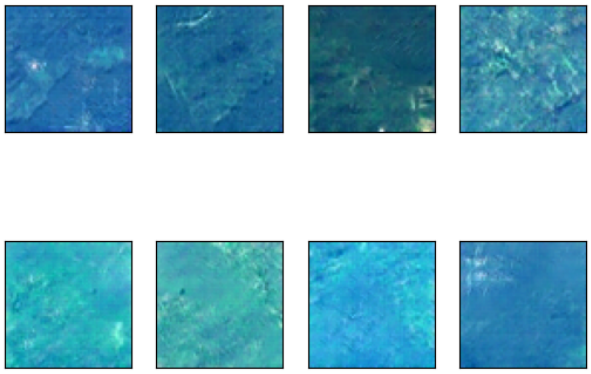

Figure 7: Generated images from the “forest” class

3.2 Qualitative Analysis

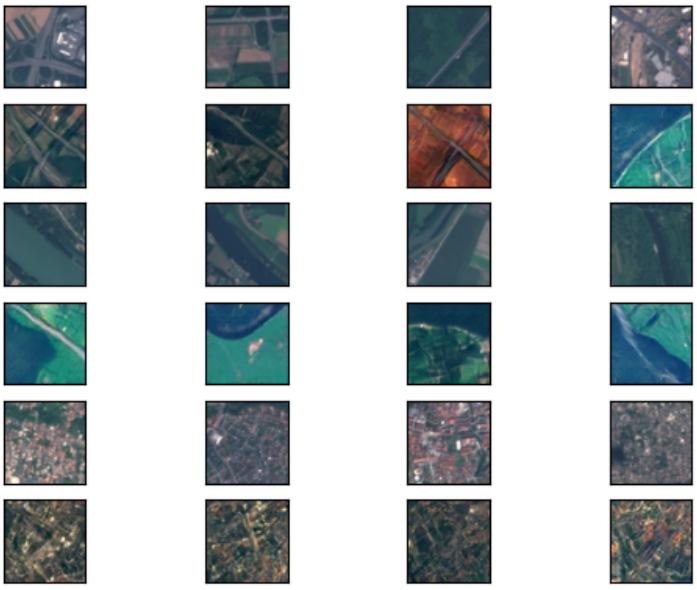

Figure 8: (Row number increases from top to bottom) Row 1: Real "highway" images, Row 2: Generated "highway" images, Row 3: Real "river" images, Row 4: Generated "river" images, Row 5: Real "residential" images, Row 6: Generated "residential" images

On visual inspection, we can see that our generator has learned some meaningful features even for the mid-performance classes ("highway", "river", and "residential" as show in Figure 8). For instance, for the "highway" class, our generator has learned that satellite images showing highways contain a line cross through them. You may also notice some cross-contamination across classes in Rows 2 and 4 where generated images for "highway" and "river" bear some resemblance with each other.

3.3 Training Convergence

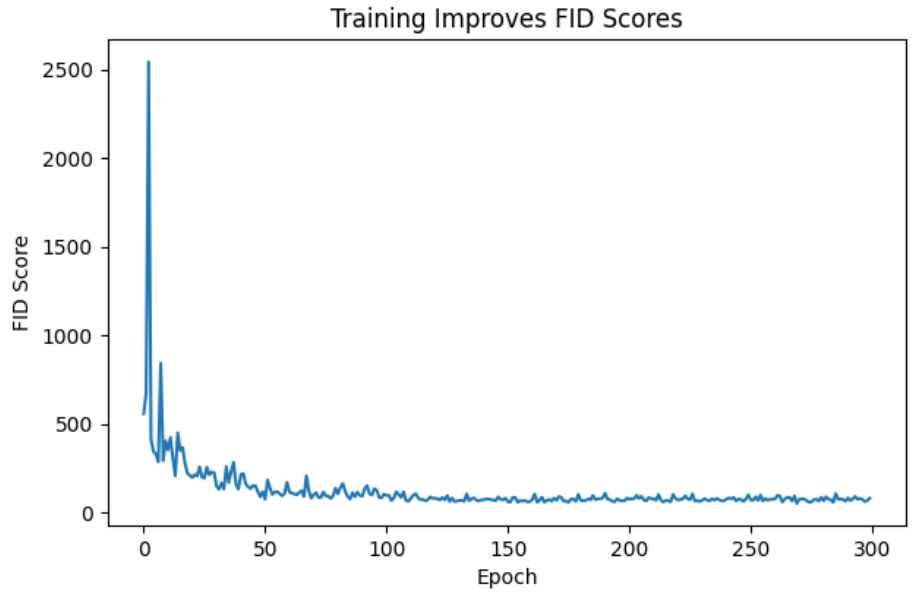

Figure 9: FID Score Throughout Training

Throughout 300 epochs of training, our model quickly achieves a low FID score after epoch 50, but quickly approaches a floor after that with little to no drops in FID score through epoch 300. Though our model produced satisfactory images, we believe that our model did not converge to an optimal equilibrium between the discriminator and generator. In Figure 10, we see the discriminator and generator losses wildly diverging starting from around epoch 80. Likewise in Figure 11, we see that the binary accuracy of the discriminator is actually going down with each successive epoch. We do not know the exact reason for this peculiar behavior, but we interpret this as evidence of non-convergence to an optimal equilibrium.

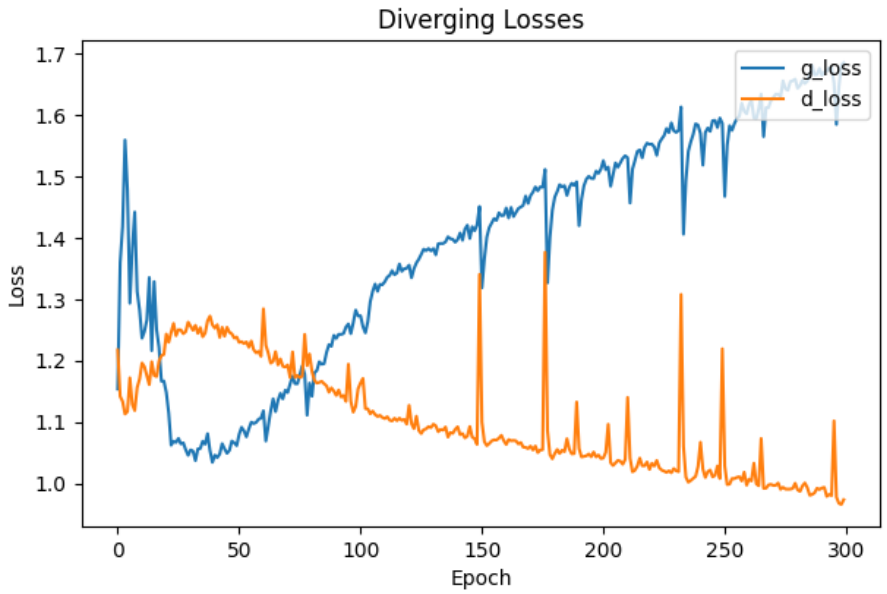

Figure 10: Loss Throughout Training

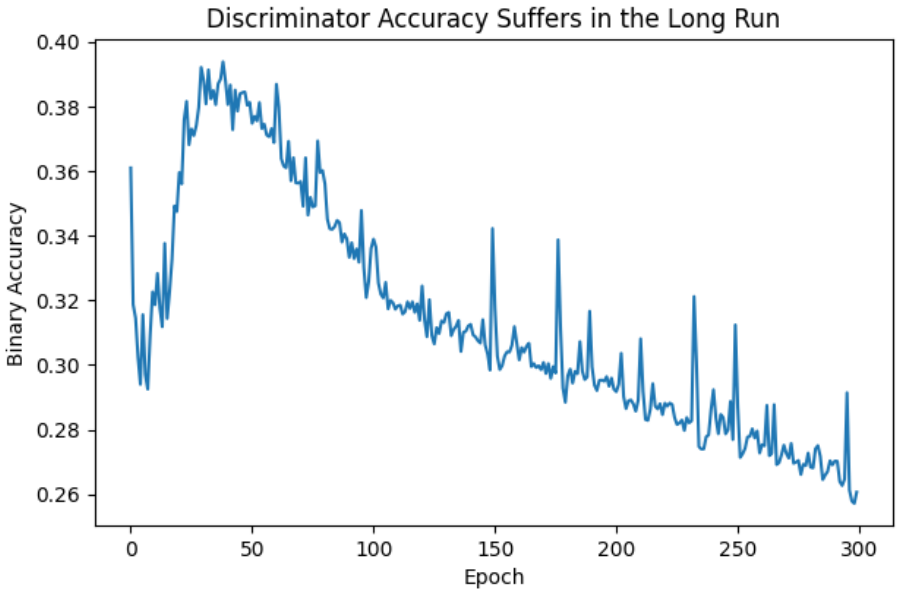

Figure 11: Binary accuracy of the discriminator throughout training

3.4 Limitations

3.4.1 Convergence and Early Stopping

The foremost limitation of our DCGAN implementation is its non-convergence. We also did not manage to implement early stopping due to time constraints and confusion around which metric to base early stopping on.

3.4.2 FID Score Calculation

Instead of calculating the FID score over the entire real image distribution and the entire fake image distribution, we had to randomly sample 500 real images and 500 fake images for FID calculation due to GPU memory constraints. This random sampling may have added noise and stochasticness to the FID score figures. Furthermore, FID uses the InceptionV3 network pretrained on ImageNet data, which is a far cry from satellite images. If we had the time, we would have liked to train the InceptionV3 architecture on EuroSAT and use the network to calculate more accurate FID scores.

4 Conclusions

In an effort to yield more realistic and higher quality synthetic satellite images, we apply the EuroSAT (RGB) dataset to our Deep Convolutional Generative Adversarial Net model and found that the model can produce satisfactory labelled satellite images of land use and land cover areas. Some image classes achieved comparable FID scores to existing DCGAN benchmarks while some classes scored poorly. In general, we observed an imbalance of FID scores across our 10 image classes.

That being said, we believe there is more future work to be done in areas such as model convergence, model architecture of both our generator and discriminator, and hyperparameter tuning. Further active investigation into said variables can further progress our DCGAN model and enhance the quality of the generated images.

We think that extending our model into other domains such as images involving depth, modern art, human faces and animals will

References

[1] L Abady, M Barni, A Garzelli, and B Tondi. Gan generation of synthetic multispectral satellite images. In *Image and Signal Processing for Remote Sensing XXVI*, volume 11533, pages 122–133. SPIE, 2020.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[3] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[5] Bo Huang, Bei Zhao, and Yimeng Song. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sensing of Environment*, 214:73–86, 2018.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[7] Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. *Efficient BackProp*, pages 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[9] Dongao Ma, Ping Tang, and Lijun Zhao. Siftinggan: Generating and sifting labeled samples to improve the remote sensing image scene classification baseline in vitro. *IEEE Geoscience and Remote Sensing Letters*, 16(7):1046–1050, 2019.

[10] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[14] Zhuli Xie, Yaoliang Chen, Dengsheng Lu, Guiying Li, and Erxue Chen. Classification of land cover, forest, and tree species classes with ziyuan-3 multispectral and stereo data. *Remote Sensing*, 11(2):164, 2019.

[15] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[16] Xingrui Yu, Xiaomin Wu, Chunbo Luo, and Peng Ren. Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. *GIScience & Remote Sensing*, 54(5):741–758, 2017.