Generation of synthetic satellite images using GANs

Vishisht Sharma 18315 Sarthak Mishra 18388 Shivan Bhatt 18249

Abstract—The field of remote sensing uses imagery captured from satellites, aircrafts, and UAVs in order to observe and analyze the Earth. Many remote sensing applications that are used today employ deep learning models that require large amounts of data or specific types of data. The lack of data can hinder model performance. A generative adversarial network (GAN) is a deep learning model that can generate synthetic data and can be used as a method for data augmentation to increase performance of data reliant deep learning models. GANs are also capable of image-to-image translation such as transforming a satellite image containing cloud coverage into one without clouds. These possibilities have led to many new and exciting GAN applications. This project aims to explore one such application of GANs in the area of satellite imagery, generation of synthetic images using data augmentation ability of GAN. The tasks carried out were accessing the data from an open access data source, preparing the data as per our requirements, using them to generate mask layers and finally translating the masks to output synthetic images.

Index Terms—Generative Adversarial Networks, conditional Generative Adversarial Networks, Data Generation, Data augmentation, Satellite imagery.

I. Introduction

With well over 7000 satellites orbiting the earth as of now, the whole planet has been extensively mapped and the data collected by these satellites has evolved to become an information gold mine. These satellites not only take RGB images of each corner of earth but also capture and stitch information in about 12-15 bands including near, far infrared and ultra-blue with wavelengths ranging from 443nm to 2190nm and resolution from 10m to 60m (for Sentinel 2 data) . Images from these bands can also be stitched together in all sorts of combinations to extract unique land cover information, e.g., combining visible and near infrared bands creates color infrared that emphasizes healthy and unhealthy

•Fig. 1, Color infrared made by combining B8, B4 and B3 channels, especially reflects chlorophyll, red area shows vegetation

Fig. 1. Infrared Image

vegetation, see fig.1. Thus, the aerial imagery data becomes very valuable especially in the field of remote sensing and finds immense use in academia along with urban planning, agriculture, government, transport and various other public plus private sectors.

Over the years many public access databases have been made available such as Landsat 8, Sentinel-2 A/B, and Pleiades,´ which provide excellent quality imagery data across multiple spectrums, still there is a high demand of land

cover data of higher resolution and multiple channels owing to the rise of use of ML algorithms in remote sensing like semantic segmentation of land cover. Remote sensing ML algorithms require a huge amount of multispectral data of every variety for

training which are often not easily available. Here comes the topic of data augmentation, i.e., supplementing the pre-existing dataset with an artificially generated one which helps in training and also testing the ML model. The conventional data collection procedures like geometric transformation, noise injection etc. become very hectic, time consuming and non-scalable in case of multispectral images and often do not provide the necessary diversity and heterogeneity. There's also a lack of higher quality image dataset that span across multiple spectrums with a higher resolution, these databases are expensive and not publicly available. Cloud cover and other weather conditions also determine the quality and usability of images. In most land cover detection models, clear images with little to no cloud cover are required

A. Research Gap and previous work

Ian Goodfellow et.al. invented the generative adversarial network (GAN) technology in 2014 in the breakthrough paper

, which proved to be game changer in the image processing and translation domain. Conditional GANs (cGAN) made targeted data generation possible. Most of the image generation work has been done on 8-bit greyscale or RGB images , expanding it to non-visible spectrum is a new front. Use of Deep Neural Networks (DNN) to generate multispectral satellite images becomes an extremely complex task due to the format and informative content of satellite images. Unlike normal RGB images, satellite camera usually captures images in a much higher resolution and pixel's reflectance is often represented by more than 8 bits. Also, each image possesses several spectral bands and specific inter-band dependencies, thus, DNNs need to have an excessively complex architecture to capture all these variances and need to be specifically trained and modelled for every kind of images which is a tedious task. Here, GANs prove to be a valuable asset.

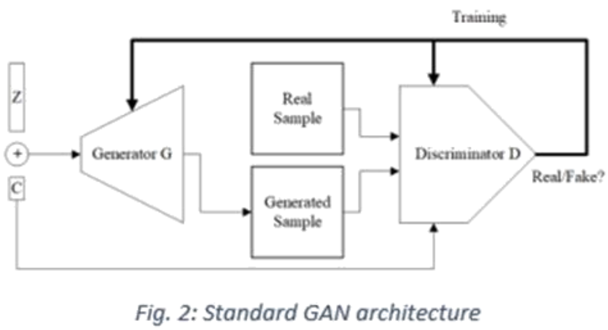

Fig. 2. Standard GAN Architecture.

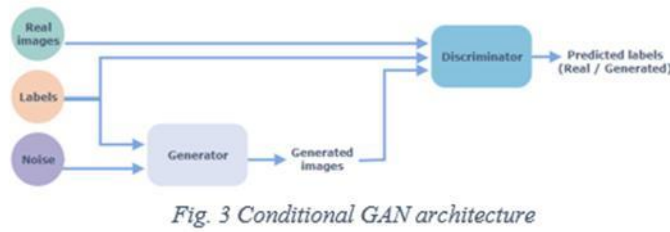

Fig. 3. cGAN Architecture

In the satellite imagery research area, the main focus has been on generating semantic labelled masks of the images and segment and classify the land cover, less attention has been paid to the concept of generating artificial synthetic images from the masks. Some work has been done in this respect; one such instance is the paper, in which the authors have generated synthetic RGB images using GANs and performed vegetation to desert style transference. The only other instance found is the work done by the radiant earth foundation, where they have applied GAN generation technology to 10 bands but with no satisfactory results in the earlier experiments due to the presence of numerous artefacts and loss of correlation between the bands. Later experiments with better band correlation produced good results albeit some artefacts. This shows that correlation between different bands plays a major role in maintaining quality and obtaining realistic results.

II. Contributions

- The project aims to a generate a indistinguishable multi-band satellite imagery dataset that can be used to train large scale ML models.
- The correlation between the bands is preserved using the ArcGIS software by mapping the coordinates of sentinel images with mask covers.
- The work is done on 256x256 size images in RGB along with near-infrared channels which can serve as a proof of concept and show validity of GANs data augmentation capabilities.

III. Background

The GAN technology and architecture has been used for several purposes such as data augmentation , image generation from scratch , style transfer , and classification among other purposes. The standard GAN architecture, illustrated in Fig.2, for image generation is made up of two convolutional neural networks: the generator is trained to produce images that are similar in distribution to the images used for training, the discriminator that classifies if the images are real (original images) or fake (generated). The two networks are trained together in a minimax fashion. The parameters of generator and discriminator are updated iteratively and alternatively. Specifically, first the discriminator is trained for one or more epochs, then the generator in trained for one or more epochs, and the process is repeated. The generator is kept constant during the discriminator training phase. Similarly, the discriminator is kept constant during the generator training phase. As the generator improves with training, the discriminator performance gets worse because the discriminator cannot easily tell the difference between real and fake. If the generator succeeds perfectly, then the discriminator has a 50 % accuracy (same as flipping a coin), reaching the convergence. The GAN model consists of a latent space vector onto which it maps the input images during training and produces results using this space during prediction. The result of the discriminator updates both the generator and the fixed dimension latent space with better model weights, increasing its accuracy with each epoch of training. While the standard GAN architecture can generate images similar to the input, it does no provide control over which specific image the generator will produce. This limitation was handled in the form of conditional GANs (cGAN) , which incorporate additional input layer of one-hot-encoded image labels. This additional layer guides the generator in terms of which image to produce. A standard cGAN structure is shown in fig. 3. Invention of Conditional GANs were a huge breakthrough in the field of image process-ing. The cGAN architecture specifically tailored for image-to-image translation is the Pix2Pix GAN. The Pix2Pix GAN changes the loss function so that the generated image is both plausible in the content of the target domain, and is a plausible translation of the input image. The architecture used in this project is also inspired by the Pix2Pix architecture. In general, the generator and discriminator models of the Pix2Pix use standard Convolution-BatchNormalization-ReLU blocks of layers as is common for deep convolutional neural networks.

IV. Materials and Methods

A. Data Preperation

The data was prepared from images taken by the Sentinel 2 database provided by Copernicus Open Access Hub. Images with little or no cloud cover were used for the project ArcGIS Pro software was used to prepare the data. In the first stage, using the Composite Bands tool Sentinel data bands: 4 (red), 3 (green), 2 (blue) and 8 (nir) all with a resolution of 10 meters, were combined and a four-band raster was created. Then the land cover vector data with the polygon topology were changed to raster data using the Polygon to Raster tool. In the second stage, it was crucial to match the coordinate system and extent with the data acquired from Sentinel. The extent value read from Sentinel data allowed us to process only a portion of a larger land cover dataset. It was also important to determine Snap Raster environment setting. In the third stage, by iterating through the grid items and using the Clip Raster tool, the Sentinel Image and Land cover raster data were cut into parts, overlapping with the consecutive mesh. The sample set was created. The same procedure was used for land cover data at both scales and to create samples of different sizes by changing the mesh parameter.

B. Model

The choice of the neural network architecture was determined by two factors. First one was the problem that we wanted to solve i.e. generating an artificial satellite image that meets precisely defined conditions in terms of land cover. Second one was the data sources that were at our disposal at the time of research. Such data comes in the form of single band TIFF masks where different pixel values represent land cover types. Also, Sentinel satellite, multi band data is publicly available. After defining our inputs (land cover masks) and outputs (artificial multiband satellite images) we were sure that Pix2Pix architecture will be a great starting point. An in-depth analysis of our GAN architecture is as follows:

C. Discriminator

The discriminator in a GAN is simply a classifier. It tries to distinguish real data from the data created by the generator. It could use any network architecture appropriate to the type of data it's classifying. The discriminator's training data comes from two sources: Real data instances, such as real pictures of people. The discriminator uses these instances as positive examples during training. Fake data instances created by the

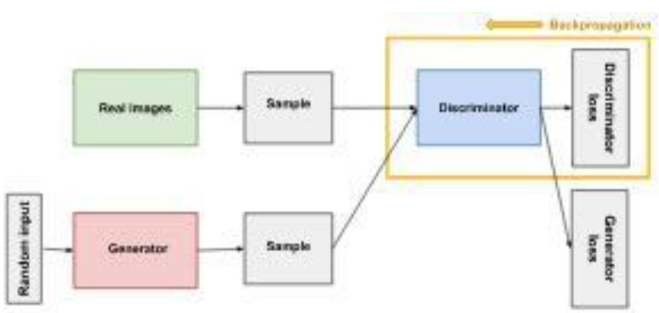

Fig. 4. Discriminator.

generator. The discriminator uses these instances as negative examples during training. In Figure 4, the two "Sample" boxes represent these two data sources feeding into the discriminator. During discriminator training the generator does not train. Its weights remain constant while it produces examples for the discriminator to train on. The discriminator connects to two loss functions. During discriminator training, the discriminator ignores the generator loss and just uses the discriminator loss. During discriminator training: The discriminator clas-sifies both real data and fake data from the generator. The discriminator loss penalizes the discriminator for misclassi-fying a real instance as fake or a fake instance as real. The discriminator updates its weights through backpropagation from the discriminator loss through the discriminator network.

Pix2Pix GAN (Discriminator) uses a PatchGAN rather than a DCNN which helps it classify patches of image as real or fake instead of an entire image. The discriminator is convolutionally run across the image where in we average all the responses to give the final output. The network outputs a single feature map of real and fake predictions that is averaged to give a single score. 70x70 patch size is considered to be effective across different image-to-image translation task.

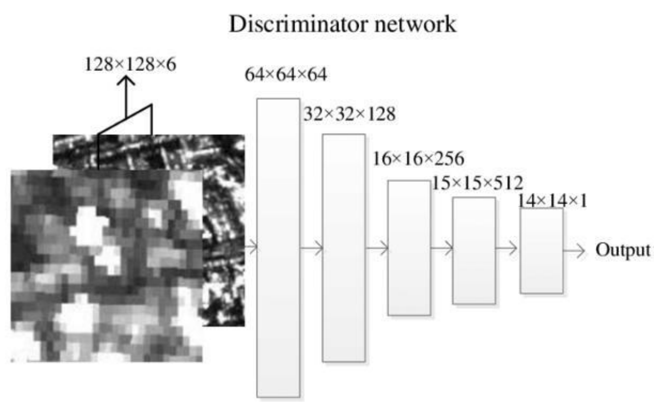

Fig. 5. Pix2Pix Architecture

D. Generator

The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real.

Generator training requires tighter integration between the generator and the discriminator than discriminator training requires.

Neural networks need some form of input. Normally we input data that we want to do something with, like an instance that we want to classify or make a prediction about. But what do we use as input for a network that outputs entirely new data instances? In its most basic form, a GAN takes random noise as its input. The generator then transforms this noise into a meaningful output. By introducing noise, we can get the GAN to produce a wide variety of data, sampling from different places in the target distribution.

To train a neural net, we alter the net's weights to reduce the error or loss of its output. In our GAN, however, the generator is not directly connected to the loss that we're trying to affect. The generator feeds into the discriminator net, and the discriminator produces the output we're trying to affect. The generator loss penalizes the generator for producing a sample that the discriminator network classifies as fake. This

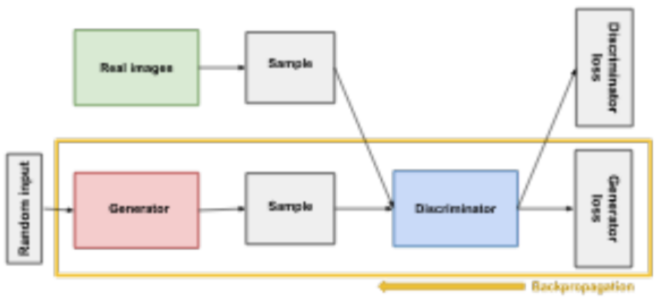

Fig. 6. Generator

extra chunk of network must be included in backpropagation. Backpropagation adjusts each weight in the right direction by calculating the weight's impact on the output — how the output would change if you changed the weight. But the impact of a generator weight depends on the impact of the discriminator weights it feeds into. So backpropagation starts at the output and flows back through the discriminator into the generator. At the same time, we don't want the discriminator to change during generator training. Trying to hit a moving target would make a hard problem even harder for the generator.

1) So we train the generator with the following procedure:
2) Sample random noise. 3) Produce generator output from sampled random noise. 4) Get discriminator "Real" or "Fake" classification for generator output. 5) Calculate loss from discriminator classification. 6) Backpropagate through both the discriminator and generator to obtain gradients. 7) Use gradients to change only the generator weights.

The U-Net model architecture is used for the generator model rather than the traditional encoder-decoder model which involves taking image as input and down-sampling it for a few layers until a layer where in the image is up-sampled for a few layers and a final image is outputted. The UNet architecture also down-samples the image and up-samples it again but would have skip-connections between layers of same size in encoder and decoder which would allow the information to be shared between input and output.

E. Training and L1/Adversarial Loss

First the discriminator model is trained in a standalone manner minimizing the negative log likelihood of identifying real and fake images, conditioned on a source image. The generator model is trained using both the adversarial loss for the discriminator model and the L1 or mean absolute pixel difference between the generated translation of the source image and the expected target image. The discriminator trains faster compared to the generator and since discriminator loss is required for training the generator, the discriminator loss is halved in order to slow down the training process. Discriminator Loss=1/2 Discriminator Loss

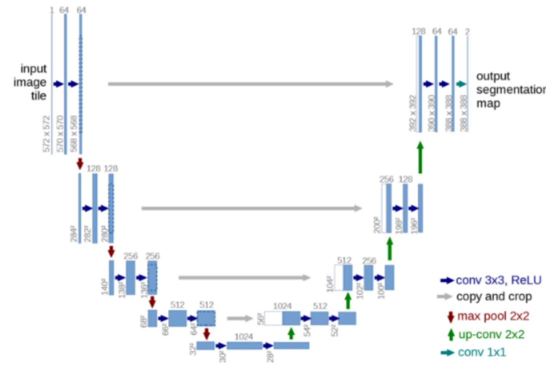

Fig. 7. UNet Architecture

The adversarial loss and the L1 loss are combined into a composite loss function, which is used to update the generator model. L2 loss was also evaluated and found to result in blurry images. The adversarial loss influences whether the generator model can output images that are plausible in the target domain, whereas the L1 loss regularizes the generator model to output images that are a plausible translation of the source image. As such, the combination of the L1 loss to the adversarial loss is controlled by a new hyperparameter lambda, which sets the importance given to L1 as opposed to Adversarial Loss.

$$\text{GeneratorLoss} = \text{AdversarialLoss} + \lambda * \text{L1Loss}$$

V. Results

We've decided that in the first iteration of the Sentinel cGAN project we will focus on verifying the usefulness of this technique for producing artificial data.

We have trained our model for 200 epochs on a total of around 2000 training samples. The following are the results obtained for artificially generated images.

Discriminator accuracy of the artificial images and the real images increases initially for a few epochs. After that it starts decreasing with increasing epochs.

Artificial images show a higher accuracy as compared to real images. The difference between the two has been displayed in Fig. 11.

Intermediate results from the 1st epoch of the training phase and the 200th epoch of the training phase. The intermediate prediction results were generated each epoch to enable visual assessment and based on masks that were not involved in the training process.

VI. Discussions

During this project we got a great insight into working with one of the computation extensive as well as data exhaustive technology i.e. Generative Adversarial Networks. During the

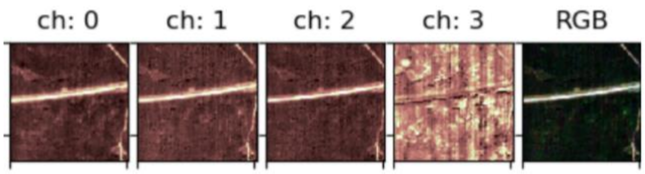

Fig. 8. Artificially generated images. We have identified that the quality directly depends on the quality of the input mask.

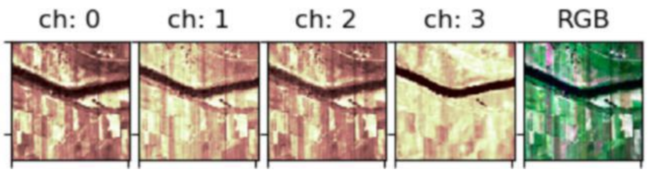

Fig. 9. Artificially generated images. 4 channel RGB output.

initial stages, we were getting introduced to issues pertaining to data preparation while we were unaware of the challenges that were yet to come mainly related to the training of our model. The model involving 200 epochs took us almost 28 to 30 hours to train completely, let alone the data preparation part, which itself was tedious.

Unlike vanilla GANs, cGANs add conditioning on the extra information. In our project those extra information are in the form of a mask image in which different color areas denote distinct classes. Adding conditioning is expected to allow better structuring of the latent space and its sampling and thus generate better results. This makes them perfectly suited for use beyond working with RGB images and opens the opportunity to apply them in satellite imagery where processing hyperspectral imagery is a standard use case.

A. Limitations

This whole study has been based on the possibility of creating a plausible real outcome, an image, by following a set of predefined rules. However, as is the general case there are certain limitations to our work which we were unable to address.

- We have trained our model only on 2000 samples which is very less for such a high scale supervised learning task that we planned to accomplish.

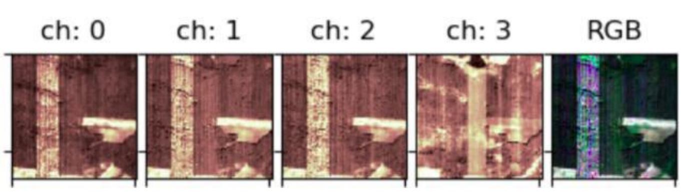

Fig. 10. Artificially generated images. We have identified that the quality directly depends on the quality of the input mask. This example used a mask full of small artifacts generated during the mask creation process thus the results are unsatisfactory.

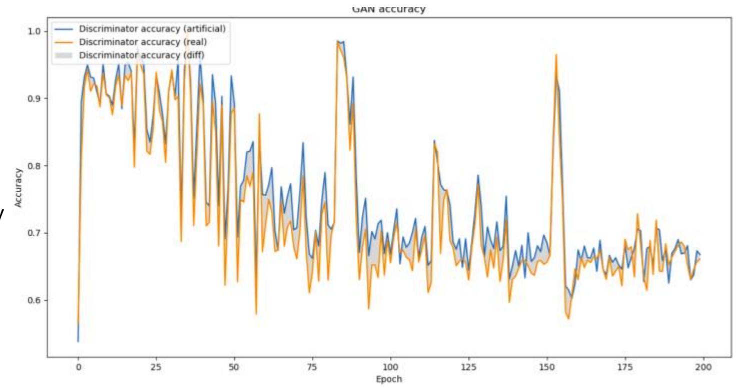

Fig. 11. Accuracy of the discriminator.

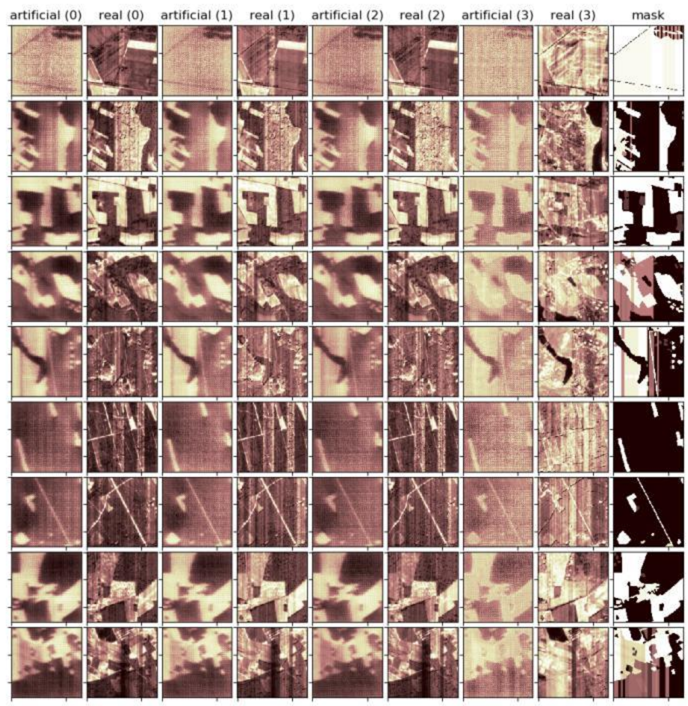

Fig. 12. Intermediate Result

- Our input and output images are of dimension 256 x 256.
- Given the long time it is taking for a single GAN model to train, we were able to explore only a couple of architectures. Out of them, based upon their accuracy outcome we finalized upon cGAN.

B. Future Recommendations

Having been worked on this project for quite some time, the following improvements could be made to the existing methodology -

- Work should be done with increased image resolution, which require greater computational and GPU power. See Fig 14.

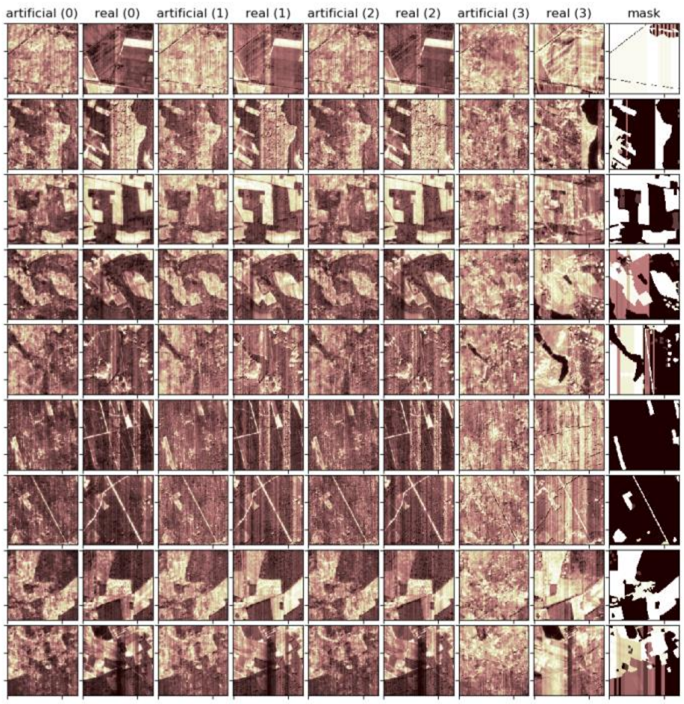

Fig. 13. Result after 200 epochs

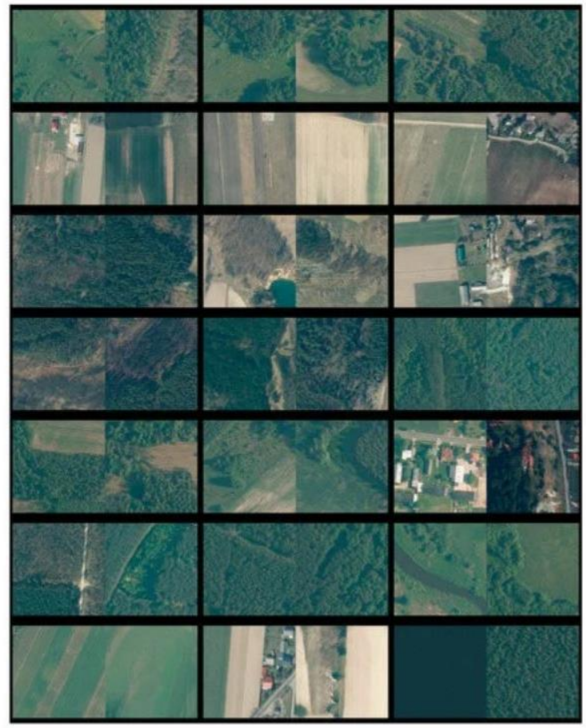

Fig. 14. Using higher resolution images with RGB bands visually striking artificial images can be generated by the model.

- Exploring other GAN architectures. In this project, we have done studies only about conditional GAN. However, other GAN architectures were there which can give better results. A suggestion would be to use proGAN architecture for the task.
- Surveying domain experts for verifying results would provide excellent evaluation metrics.

VII. Conclusion

Generative Adversarial Networks are a powerful tool that definitely found their place in both Geographical Information Systems (GIS) And Machine Learning toolboxes. In the case of satellite imagery they provide a data augmentation mechanism for creating decent quality artificial data samples.

On the other hand, GANs are really hard to train and prone to overfitting. To achieve a decent result you should be ready for a long run of trials and errors.

The results in this paper suggest that conditional generative adversarial networks are a promising approach for many image-to-image translation tasks. These networks learn a loss adapted to the task and data at hand, which makes them applicable in a wide variety of settings.

Acknowledgment

We would like to thank Dr. Vaibhav Kumar for his guidance all through the course of this project. He provided timely valuable insights on the progress of this project at regular intervals.

References

[1] Abady, Lydia Barni, Mauro Garzelli, Andrea Tondi, Benedetta. (2020). GAN generation of synthetic multispectral satellite images. 19. 10.1117/12.2575765.

[2] Goodfellow, I, Abadie, J. P., Mirza, M, Xu, B, Warde-Farley, D, Ozair, S, Courville, A, Bengio, Y., "Generative Adversarial Networks", NIPS, 2014.

[3] https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/

[4] Arjovsky, M., Chintala, S., Bottou, L., "Wasserstein generative adver-sarial networks", International conference on Machine Learning (2017), pp. 214-223

[5] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., "Improved techniques for training GANs", Advances in Neural Information Processing Systems (2016), pp. 2234-2242

[6] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial

[7] Landsat 8—Landsat Science. Available online: https://landsat.gsfc.nasa.gov/landsat-8/.

[8] Sentinel-2—ESA Operational EO Missions—Earth Online—ESA. Available online: https://earth.esa.int/web/guest/missions/ esa-operational-eo-missions/sentinel-2K. Elissa, "Title of paper if known," unpublished.

[9] Pleiades—eoPortal Directory—Satellite Missions. Available online:https://earth.esa.int/web/eoportal/satellite-missions/p/ pleiades.

[10] Tolosana, Ruben Vera-Rodriguez, Ruben Fierrez, Julian Morales, Aythami Ortega-Garcia, Javier. (2020). DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection.

[11] O'Hara, Chuck King, Jason Cartwright, John King, Roger. (2003). Multitemporal land use and land cover classification of urbanized areas within sensitive coastal environments. Geoscience and Remote Sensing, IEEE Transactions on. 41. 2005 - 2014. 10.1109/TGRS.2003.816573.

[12] Mohandoss, Tharun Kulkarni, Aditya Northrup, Daniel Mwebaze, Ernest Alemohammad, Hamed. (2020). Generating Synthetic Multi-spectral Satellite Imagery from Sentinel-2.
[13] Mirza, Mehdi Osindero, Simon. (2014). Conditional Generative Adver-sarial Nets.