International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 1 • January-March 2021

Satellite Imagery Noising With Generative Adversarial Networks

Mohamed Akram Zaytar, Abdelmalek Essaadi University, Morocco
Chaker El Amrani, Abdelmalek Essaadi University, Morocco

ABSTRACT

Using satellite imagery and remote sensing data for supervised and self-supervised learning problems can be quite challenging when parts of the underlying datasets are missing due to natural phenomena (clouds, fog, haze, mist, etc.). Solving this problem will improve remote sensing data augmentation and make use of it in a world where satellite imagery represents a great resource to exploit in any big data pipeline setup. In this paper, the authors present a generative adversarial network (GANs) model that can generate natural atmospheric noise that serves as a data augmentation preprocessing tool to produce input to supervised machine learning algorithms.

KEYWORDS

Artificial Neural Networks, Data Augmentation, EUMETSAT, Generative Adversarial Networks, MDEO, MetOp, Remote Sensing, Satellite Imagery

INTRODUCTION

Remote sensing data is the cornerstone of modern environmental monitoring. Both rule-based and AI-powered systems heavily rely on high-resolution satellite imagery in domains such as agriculture, forestry, disaster management, geology and many more.

In recent years, many deep learning architectures have been used to tackle some of the most challenging remote sensing-related problems, new state-of-the-art results are established in far-apart domains such as building footprints (Bischke, B, 2019), land use classification (Zhang, C, 2018), iceberg detection (Zhang, X, 2018), deforestation (Shah, U, 2017), weather forecasting (Lin, S, 2018), Poverty estimation (Perez, A, 2019), and more. This surprising success is linked to the massive amounts of daily imagery collected from satellites, and, in many cases, to the high spectral resolution that comes with the data, including up to dozens of visual bands and allowing for rich data mining using deep neural networks.

A common issue when pre-processing satellite imagery for self-supervised tasks is the lack of adequate input (or data features), we only have real (ground-data) images and we're responsible for creating input data features to learn a certain task, a prime example is when we want to create an image-to-image interpolator where the input image has some missing pixels and the output image is complete, this problem is the main motivator behind the proposed approach.

DOI: 10.4018/IJCINI.2021010102

This article, published as an Open Access article on November 6, 2020 in the gold Open Access journal, International Journal of Cognitive Informatics and Natural Intelligence (converted to gold Open Access January 1, 2021), is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Generally speaking, when collecting sensor information, we are dealing with data that is incomplete, and we want to learn the distribution of that missing information so we can reproduce it while engineering training input data. In our case, we are interested in augmenting synthetically generated noise to satellite images so that downstream models are trained for real-world cases. For a problem like image interpolation, the pre-processed input data should simulate the incompleteness of the remote sensing data we receive.

The goal of this paper is to synthetically generate remote sensing noise that simulates natural noise found while directly collecting satellite imagery, the main contribution of this work consists of a model architecture based on generative adversarial networks (or GANs). We trained a model that can learn the underlying noise structure from pre-processed 50×50 pixel patches containing missing/damaged pixels represented by 0, and healthy pixels represented by 1. After training, the generator produces noise samples that are indistinguishable from the real distribution of satellite noise images.

The proposed generative adversarial network is comprised of the following elements described below.

Real Data

A collected and pre-processed data set consisting of 1M 50x50 pixel images with noise associated with natural phenomena such as atmospheric and weather conditions, sensor quality, and satellite position. Each image pixel is either a valid measurement represented by 1 or an invalid, missing, or damaged pixel, represented by 0 the original imagery is directly collected from MDEO's (El Amrani, C, 2013) data pipeline, we discarded the actual measurements to focus on learning the 2D noise distribution instead of the distribution of the underlying values. Following are the data pre-processing steps to get the final patches:

1. Download image tiles without time nor location filters from MDEO's storage servers;
2. Extract $N \times N$ patches from all image tiles.
3. Filter to keep patches with noise pixel (represented as 0s) percentage ranging from 40% to 60%.
4. Assign a random value ranging from 0 to 0.3 to each missing pixel, and a random value ranging from 0.7 to 1. to each healthy pixel.

Random Input

Uniformly random vectors fed to the generator, sometimes called input entropy, represent the Input to the Generator Network. In Our case, these random vectors are comprised of 100 uniform random values $\in [0,1]$.

The Generator

Responsible for producing vector representations of 50×50 pixel patches with noise ranging from 40% to 60% zeroed pixels. The generator is trained using the feedback loop coming from the discriminator's decisions, in other words, the error is back-propagated using predictions coming from the discriminator.

The Discriminator

Represents a feed-forward neural network responsible for deciding whether a noised satellite patch is real or not. Simply put, it's a binary classifier trained on both real patches coming from the pre-processed data distribution and fake image patches produced by the Generator.

Training Loop

Initializes pre-training functions and hyper-parameters such as the cost functions for the generator and discriminator networks, the optimizers, the batch size, the number of epochs, parameters that control the training balance between $D\left(.\right)$ and $G\left(.\right)$ and the actual training loop.

RELATED WORK

Generative adversarial networks have been successfully applied to many problems in computer vision and intersecting areas of interest, while none was specifically targeted at generating synthetic satellite imagery noise, the following are some of the most prominent use cases in generating visual attributes, transforming or compressing visual data, and harnessing image quality.

One of the most prominent applications using GANs with visual data is what's called "Domain transfer". It's used in adaptation problems where we are interested in scenarios in which a model trained on a source distribution is used in the context of a different but related target data distribution. The work of (Isola, P, 2017) uses conditional adversarial networks as a general-purpose solution to learn a mapping (or a translation) from an input image to an output image. Another method (Xian, W, 2018) makes use of texture to control image synthesis, allowing for complete control of the synthesized visual objects.

Additionally, problems relating to image inpainting and quality enhancement have found interesting solutions using GANs. The work of (Xian, W, 2018) targets context-based pixel predictions, it uses contextual visual information to predict missing parts or regions of interest of an image, the sharpness of the results was further enhanced by introducing an adversarial component to the suggested model. A related but more popular problem is super-resolution using GANs, the work of (Ledig, C., 2017) for example, proposes an adversarial approach with two cost functions (adversarial and content losses) that is capable of inferring photo-realistic natural images at $4\times$ upscaling factors.

Lastly, generating high-resolution synthetic or interactive imagery has seen many efforts using GANs, (Karras, T, 2017) outlines best practices around training, architecting, and evaluating GANs-based networks for generating high-quality images. (Nam, S., 2018) gives a glimpse into how we can manipulate images using natural text, by learning the relationship between the semantic visual attributes of an image and the latent attributes of the suggested natural description text. Finally, (Shrivastava, A., 2017) suggests a model to enhance the quality of synthetically generated or simulated data to serve as input, by learning, in an unsupervised manner, from a real data distribution. An interesting approach was taken by (Ganguli, S., 2019) to automatically generate standard map layers from raw satellite imagery using conditional GANs with reconstruction and style loss, learning semantic relations between imagery and concepts such as "land", "sea", "road".

PROBLEM

To properly state the problem, we define the following entities:

- x : Represents satellite imagery (real or fake), a single data point is a matrix of shape 50×50 pixels, each pixel holding a numerical value between 0 and 1 (after normalization);
- z : The input noise to the generator, represents a vector comprised of 100 random values between 0 and 1 ;
- $G\left(z,\theta_g\right)$: The generator function, or the Generator neural network;
- θ_g : The parameters, or the weights of the generator network $G\left(.\right)$;
- $D\left(x,\theta_d\right)$: The discriminator function, or the discriminator neural network;

- θ_d : The parameters, or the weights of the discriminator network $D(.)$.

To optimize for a smooth loss function, ground-data pixel values were transformed so that values between 0.0 and 0.3 represent a missing pixel (instead of simply 0 , and values between 0.7 and 1.0 represent an available pixel measurement (instead of 1).

$D(X)$ outputs the probability that X came from the real data distribution (pre-processed satellite patches) rather than p_z (the generator's learned distribution). We train the discriminator to maximize the probability of correctly assigning the true label to x and the false label to samples from p_z . We also simultaneously train the generator to minimize $log(1-D(G(z)))$.

$D(.)$ and $G(.)$ play the following two-player minimax game with value function $V(G,D)$ (Goodfellow, I., 2014):

$$min_G\, max_D\, V = E_{x\sim p_d(x)}\left[log\left(D(x)\right)\right] + E_{z-p_z(z)}\left[log\left(1-D\left(G(x)\right)\right)\right]$$

In other words, the problem of generating a new flattened 50x50 noised satellite image is equivalent to sampling from the "Remote Sensing Natural Noise Distribution" coming from MDEO's data store. We aim to solve the problem of generating a random variable for a specific probability distribution.

APPROACH

We present the following architectures for the generator and discriminator neural networks.

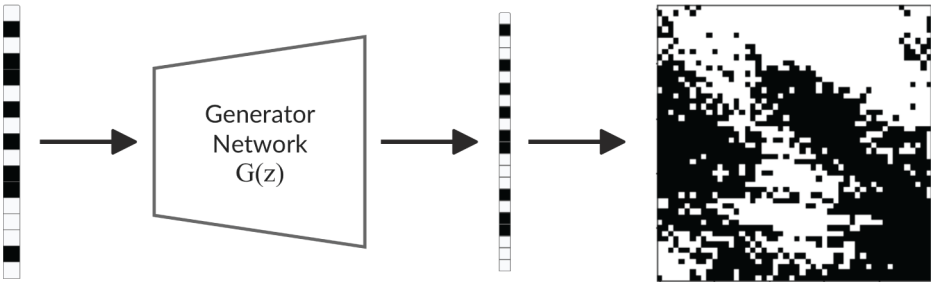
Generator

The generator is modeled as a feed-forward neural network that takes as input a uniform random variable and returns a random variable that follows the target distribution (real noised images):

- Input: tensors of shape $(batch\,size,100)$, consisting of initialized random values between 0 and 1 :
- Output: tensors of shape $(batch\,size,1,50,50)$. 1-channel 50×50 .generated noise images.
- $1st$ (input) layer: 256 output features.
- $2nd$ layer: 512 output features.
- $3rd$ layer: 1024 output features.
- $4th$ layer: 2048 output features.
- $5th$ (output) layer: 2500 (50×50) output features.
- LeakyReLU (Xu, B., 2015) ($f_\alpha(x) = max(0,x) - \alpha\, max(0,-x)$): used as an activation function for all layers except the output layer.
- Sigmoid ($S(x) = \dfrac{1}{1+e^{-x}}$): used as an activation function for the output layer (since we want values between 0 and 1).
- 1-D batch normalization (Ioffe, S., 2015): used for the middle 3 layers to improve the performance and stability of the generator network.

The network reshapes the final output vector to an image of 50×50 pixels and 1-channel.

Figure 1. The generator learns the distribution of real noise, output vector is reshaped to a matrix to produce the noise mask

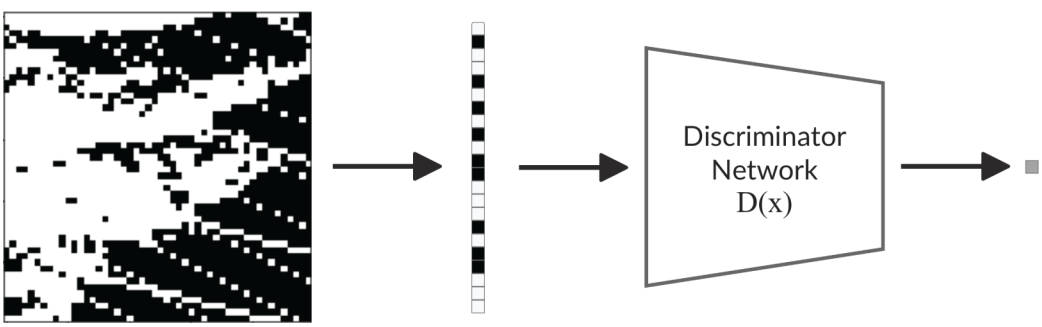


The Discriminator

The network first flattens images to vectors of size $2,500$ before feeding them into the first layer:

- Input: tensors of shape $\left(batch\,size,1,50,50\right)$, consisting of real/fake image patches 0 and 1.
- Output: tensors of shape $\left(batch\,size,1\right)$, represent the probabilities that each image patch is real.
- $1st$ (Input) layer: 512 output features.
- $2nd$ layer: 256 output features.
- $3rd$ (output) layer: 1 output feature (a probability).
- LeakyReLU: used as an activation function for the first two layers.
- Sigmoid: used as an activation function for the last output layer (to produce a probability).
- 1-D batch normalization: used for the middle 2 layers to improve the performance and stability of the discriminator network.

Figure 2. Example: Takes in an image and outputs a single value that represents the probability that the image is real



Training

Optimizers and Loss Function

For the networks' loss functions, we chose binary cross-entropy, which measures the distance between the predictions vector and the target labels vector, the loss can be described as:

$$l\left(x,y\right)=\{l_1,...,l_N\}^T,l_n=-\left[y_n\ .logx_n+\left(1-y_n\right)\ .log\left(1-x_n\right)\right]$$

where x represents the prediction tensor, y is the target tensor, and N is the batch size.

As for the optimizers, we chose the Adam optimizer (Kingma, D. P., 2014), adaptive moment estimation (Adam) computes adaptive learning rates for each parameter. It stores an exponentially decaying average of past squared gradients v_t and an exponentially decaying average of past gradients m_t where:

$$m_t = \beta_1 m_{t-1} + \left(1-\beta_1\right) g_t$$
$$v_t = \beta_2 v_{t-1} + \left(1-\beta_2\right) g_t^{\ 2}$$

with parameters β_1, β_2. Adam fights zero-related biases in the parameters by computing bias-corrected 1st and 2nd-moment estimates, $\widehat{m_t}$ and $\widehat{v_t}$ (Ruder, S., 2016), the optimization algorithm is as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon}$$

The following algorithm showcases a high-level overview of the training loop:

Algorithm1: Training Loop

```
Data: G(θg0), D(θd0), DataLoader, Epochs, gOptimizer, dOptimizer,
loss, S [Entropy].
Result: G(θg ), D(θd ).
begin
   for e ∈ epochs do
      for batch ∈ DataLoader do
            /* Calculate Discriminator Losses */
            dLossReal ← loss(D(batch),1s)
            dLossFake ← loss(D(G(S)),0s)
            dLoss ← dLossReal + dLossFake
            /* Optimize with Back-propagation */
            dLoss.backward()
            dOptimizer.step()
            /* Control Generator Training */
            if (step % n_critic = 0) then
                  dDecisions ← D(G(S))
                  gLoss ← loss(dDecisions,1s)
                  /* Optimize with Back-propagation */
                  gLoss.backward()
                  gOptimizer.step()
            step ← step + 1
```

RESULTS

The training was conducted on a cloud GPU instance with the following specifications:

- GPUs: A cloud-based Tesla P100 environment
- CPUs: 4
- Memory: 24 GB RAM
- Data Storage: 250 GB SSD

Training specifications are outlined below:

- Batch size: 256 images per batch;
- Epochs: we set a 100 epochs for the entirety of the training loop. after training, both discriminator/generator losses had converged;
- η, the learning rate of both generator/discriminator optimizers is set to 0.0002;
- $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

Training balance between the generator and discriminator was reached by continuously training the discriminator while training the generator in intervals controlled by two parameters: $step$ and n_{critic}, where n_{critic} was set to 10, meaning, the generator network was trained every 10 consecutive batches.

Training Loop

After training, we minimized the loss of both the discriminator and generator networks. In generative adversarial networks, there are no ideal loss values that must be reached by the generator/discriminator for the networks to reach an equilibrium, as long as both losses decrease in a controlled manner. After training, the discriminator couldn't distinguish between real noise and generated remote sensing noise.

We improved on the model's inception scores (Salimans, T., 2016) to stabilize the performance of our unconditional GANs, the Kullback-Leibler formula was used to calculate the score, KL divergence measures how similar and different two probability distributions are, to calculate the inception score, we averaged the exponential KL divergence score overall images. Table 1 showcases the final loss values and the accuracy of the discriminator after training.

After training the generator to produce 50x50 patches of noise, we project the model's noise to mask healthy patches with zeros and produce naturally noised satellite imagery that can be used for several self-supervised learning problems such as 2-D Interpolation and super-resolution.

FURTHER WORK

This model can be considered the first step towards a highly sophisticated synthetic noise generator for remote sensing imagery and sensor measurements in general, improvements to the model may

Table 1. Overall loss and the discriminator final accuracy

Metric	Score
Generator Loss	0.53
Discriminator Loss	0.77
Discriminator Accuracy	56%

include parametric noise control, e.g. specifying the amount of noise (the proposed model generates 0.4 to 0.6 noise) and generating images of dynamic resolutions.

We can achieve parametric noising by using a conditional generative adversarial network (CGANs) (Mirza, M., 2014), in which we add a conditioning layer as an input to both the generator and discriminator to produce noise in specific percentages, by changing the following terms in the optimization minimax problem:

$$log\left(D\left(x\right)\right) \Rightarrow log(D(x \mid y)) \; log\left(1 - D\left(G\left(z\right)\right)\right) \Rightarrow log(1 - D(G(z \mid y)))$$

where the discriminator $D\left(.\right)$ and generator $G\left(.\right)$ are optimized on conditional parameters.

Another interesting research direction is to learn representations between noise parameters and the distribution of the noise using InfoGAN (Chen, X., 2016) by adding more channels to the image containing information about measurement conditions, this can lead to minimizing the ratio of damaged pixels in the future by figuring out the most contributing factors in generating the noise.

A parametric generator will need more data and hyper-parameters tuning to solve the issue of instability and non-convergence of the loss function in GANs, this can also be addressed in the future by collecting more data, moving from a fully-connected architecture to a convolutional neural network and training on bigger patches.

CONCLUSION

Deep learning opens the door to limitless applications in domains such as environmental science, agriculture, pollution, disaster monitoring, and many others (Pouyanfar, S, 2019). Remote sensing data providers (Klaes, K, 2007) also play a central role in the development of the field and in advancing environmental research using machine learning and data science, without big data resources, deep learning methods fall short. Fortunately, we have a wealth of satellite imagery and remote sensing data to train and make use of our models in live production systems.

We believe that artificial intelligence will revolutionize climate science and lay the ground to build effective solutions that address the environmental problems we are facing today. By using high-resolution imagery, remote sensing data, and satellite sensor-based imagery in general, we can gain a deeper understanding of the atmospheric processes and build systems that learn complex dependencies and correlations and can contribute to many domains such as energy, pollution, agriculture, oceans, and climate science.

In this paper, we proposed a model architecture based on generative adversarial networks to solve the problem of synthetic generation of remote sensing noise for self-supervised machine learning tasks. Our model learns the distribution of remote sensing noise from a real source data set comprised of 1M patches of satellite images and can generate new noise masks that come from the same source distribution. Experimental results show the stable convergence of both generator and discriminator networks and showcase the indistinguishability of the generator's exported patches from the real dataset.

REFERENCES

Bischke, B., Helber, P., Folz, J., Borth, D., & Dengel, A. (2019, September). Multi-task learning for segmentation of building footprints with deep neural networks. In *2019 IEEE International Conference on Image Processing (ICIP)* (pp. 1480-1484). IEEE. doi:10.1109/ICIP.2019.8803050

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems (pp. 2172-2180). Academic Press.

El Amrani, C., Rochon, G. L., El-Ghazawi, T., Altay, G., & Rachidi, T. (2013, July). System architecture of the Mediterranean Dialogue Earth Observatory. In *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS* (pp. 600-603). IEEE. doi:10.1109/IGARSS.2013.6721228

Ganguli, S., Garzon, P., & Glaser, N. (2019). *Geogan: A conditional gan with reconstruction and style loss to generate standard layer of maps from satellite images.* arXiv preprint arXiv:1902.05611

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680). Academic Press.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* arXiv preprint arXiv:1502.03167

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134). IEEE.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). *Progressive growing of gans for improved quality, stability, and variation*. arXiv preprint arXiv:1710.10196

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980

Klaes, K. D., Cohen, M., Buhler, Y., Schlüssel, P., Munro, R., Luntama, J. P., von Engeln, A., Clérigh, E. Ó., Bonekamp, H., Ackermann, J., & Schmetz, J. (2007). An introduction to the EUMETSAT polar system. *Bulletin of the American Meteorological Society*, *88*(7), 1085–1096. doi:10.1175/BAMS-88-7-1085

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., & Shi, W. et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690). doi:10.1109/CVPR.2017.19

Lin, S. Y., Chiang, C. C., Li, J. B., Hung, Z. S., & Chao, K. M. (2018). Dynamic fine-tuning stacked auto-encoder neural network for weather forecast. *Future Generation Computer Systems*, *89*, 446–454. doi:10.1016/j.future.2018.06.052

Mirza, M., & Osindero, S. (2014). *Conditional generative adversarial nets*. arXiv preprint arXiv:1411.1784

Nam, S., Kim, Y., & Kim, S. J. (2018). Text-adaptive generative adversarial networks: manipulating images with natural language. In Advances in Neural Information Processing Systems (pp. 42-51). Academic Press.

Perez, A., Ganguli, S., Ermon, S., Azzari, G., Burke, M., & Lobell, D. (2019). *Semi-supervised multitask learning on multispectral satellite images using Wasserstein generative adversarial networks (gans) for predicting poverty.* arXiv preprint arXiv:1902.11110

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2019). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5), 92. doi:10.1145/3234150

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. arXiv preprint arXiv:1609.04747

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In Advances in neural information processing systems (pp. 2234-2242). Academic Press.

Shah, U., Khawad, R., & Krishna, K. M. (2017, May). Detecting, localizing, and recognizing trees with a monocular MAV: Towards preventing deforestation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1982-1987). IEEE. doi:10.1109/ICRA.2017.7989229

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., & Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2107-2116). doi:10.1109/CVPR.2017.241

Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., & Hays, J. et al. (2018). Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8456-8465). doi:10.1109/CVPR.2018.00882

Xu, B., Wang, N., Chen, T., & Li, M. (2015). *Empirical evaluation of rectified activations in convolutional network*. arXiv preprint arXiv:1505.00853

Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., & Atkinson, P. M. (2018). An object-based convolutional neural network (OCNN) for urban land use classification. *Remote Sensing of Environment*, *216*, 57–70. doi:10.1016/j.rse.2018.06.034

Zhang, X., Liu, Y., Zheng, Y., Zhao, Z., Li, J., & Liu, Y. (2018, December). Distinction Between Ships and Icebergs in SAR Images Using Ensemble Loss Trained Convolutional Neural Networks. In *Australasian Joint Conference on Artificial Intelligence* (pp. 216-223). Springer. doi:10.1007/978-3-030-03991-2_22

Mohamed Akram Zaytar obtained his Bachelor of Applied Mathematics and Computer Science degree from The Faculty of Science and Technology, Tangier in 2013. He received his master, in Computer Systems and Networking from the FSTT, in 2016. Currently, a PhD student at the FSTT. His primary research interest are in Data Science, Artificial Intelligence, Machine Learning, Cloud Computing, and Big Data.

Chaker El Amrani is Doctor in Mathematical Modelling and Numerical Simulation from the University of Liege, Belgium (2001). He joined Abdelmalek Essaadi University, Morocco in 2003. He is currently Chair of the Computer Engineering Department at the Faculty of Science and Technology, Tangier. He lectures distributed systems and is promoting High Performance Computing education in the University. His research interests include Cloud Computing, Big Data Mining and Environmental Information Systems. Dr. El Amrani's research has been supported by national and international organisms. Dr El Amrani has served as an active volunteer in IEEE Morocco Section. He is currently Vice Chair of IEEE Communication and Computer Societies Morocco Chapter, and advisor of the IEEE Computer Society Student Branch Chapter at Abdelmalek Essaadi University. He is the NATO Partner Country Project Director of a real-time remote sensing initiative for early warning and mitigation of disasters and epidemics in Morocco.