# Applying Multi-Agent RL to SLAM with Graph Pose for Sampled-Data MPC and CPN of Autonomous Drone Swarms

Rohith Gandhi G rgg296 | Elmon Toraman et1811 | Kunru Lu kl3743 | Yue Jin yj1672

July 13, 2020

## Abstract

Developing methods that allow drones to autonomously navigate in different environments has been a topic of extensive research in recent years. This is mostly due to the ability of drones to navigate to places where it is dangerous for humans. Drones have been used to investigate penstocks and in search and rescue operations. Using a swarm of drones that autonomously navigates in a post-catastrophe scenario to optimally map the disaster zone, i.e map the scenario as fast as possible has been a problem less explored. Mapping the post-catastrophe environment allows better mobilization of rescue operations which in turn can lead to saving lives. In this work, we would like to propose an approach which utilizes Reinforcement Learning to enable the swarm of drones to navigate and optimally map the entire post-catastrophe environment. We demonstrate this in a simulation[1]. The optimal path planning of the drones will happen offline, i.e on the drones and it has to adapt dynamically to the changes in the environment. We take up the 2010 Haiti earthquake as an example and demonstrate methods to identify structural changes from satellite images, generate synthetic data and construct fragility equations which can be used for assessing damage state at a location.

***Advisors:*** *David K A Mordecai, PhD and Giuseppe Loianno, PhD.*

***Organization:*** *RiskEcon® Lab for Decision Metrics @ Courant Institute of Mathematical Sciences NYU, in coordination with Agile Robotics and Perception Lab (ARPL)* @ Tandon School of Engineering

## 1 Introduction

Natural disasters such as earthquakes and hurricanes can cause catastrophic damages to urban environments. Apart from damages to properties, there is a huge loss of lives. The total number of deaths due to a disaster can be decreased if we locate and quantify the areas of extensive damage which can be done through mapping the post-catastrophe environment. This will also help to better mobilize and allocate resources. Drones are an ideal candidate for this scenario as they are small and agile. Autonomous drones have

[1]Due to COVID restrictions, we show the output only in simulation and not on the drones as we do not have access to the lab

been previously used for mapping penstocks [17] and carrying out search and rescue operations [24]. The process of mapping the post-catastrophe environment becomes much faster when we use a swarm of drones and the operation becomes more robust to failures and noise inherent in measurements. This allows coverage of every nook and corner which in turn reduces the emergency response time. The drones cannot rely on GPS signals for navigation as it can get obfuscated in post-catastrophe urban environments. Using a swarm of drones that navigate autonomously in a GPS deprived scenario is a problem that has not been extensively studied.

There are classical methods available that allow you to achieve the same objective, however, these methods are highly susceptible to noise in sensors and are not dynamic to adapt to different scenarios. This can be overcome if we use a learning-based approach to solve the objective. The advantage of using Reinforcement Learning over classical methods is that they are pliable to various changes in the environment and on average can provide optimal solutions than classical counterparts, i.e they can generalize better. However, setting up a right incentive structure in the simulation that emulates the *real-life* and training the agents (drone swarms) to achieve the objective requires rigorous theoretical and empirical evaluations.

## 2 Literature Survey

Multi-Agent Reinforcement Learning methods have been previously employed to a swarm of drones to optimally map a coverage area [18]. Graph Neural Networks have also been employed to devise an optimal policy for robots such that each robot can achieve their goal state as fast as possible without conflicts in path [11] [14]. However, these methods rely on GPS signals or assume no constraints on power sources. Unlike the above approaches, we try to optimize our method such that it works under constraints on power source, GPS and adaptable to a real-world scenario.

## 3 Data Collection

One of the initial steps necessary for the drones to devise a path would be to identify how much the environment has changed after the catastrophe. To experiment with various change detection methods, we selected the 2010 Haiti earthquake and obtained pre and post-disaster images of Port-au-Prince collected by Google maps and GeoEye [6]. The pre-disaster images are satellite images while the post-disaster images are aerial images. Therefore, the observed post-disaster images were of higher spatial resolution than the pre-disaster images. We made use of geotags present in the pre and post-catastrophe images and cropped them into multiple paired small-sized images which totalled to around 300 images. Examples of a paired pre and post disaster image is shown in Figure 1.

(a) Pre-Catastrophe image (b) Post-Catastrophe image

Figure 1: A sample from collected Images

# 4 Methods - Including Results

There are multiple facets to this project which are elucidated in detail below.

## 4.1 Change detection

Quantifying how much the environment has changed due to the disaster can help in assessing the total damage in a location which can be help prioritize the locations (cells/rasters) during navigation of drones. To quantify the changes, we extracted SIFT [15], SURF [2] and ORB [19] features from the pre and post-catastrophe environment and filtered out the high confidence values. These high-confidence features are visualized in the Figure 2. The pre-catastrophe images were observed to have low spatial resolution relative to post-catastrophe images. Therefore, we were able to identify more features from the post-catastrophe images. This discrepancy in the number of features caused a difficulty in identifying which features in the pre-catastrophe image were actually changed. This shows that working with the images in pixel space to identify the structural changes is difficult due to the observed discrepancy in spatial resolution.

## 4.2 Generating synthetic data

The amount of paired pre and post-catastrophe images in our sample is relatively low. Therefore, we wanted to generate synthetic post-disaster images given pre-disaster images for a specific location that would emulate the structural changes due to the disaster. Training the agents (drones) on pixels that emulate the structural changes can lead to better generalization. A solution to this problem is to use Generative Adversarial Networks [5] to produce more synthetic data as they are shown to be successful in generating images with high spatial resolution [10, 9]. The collected images were pre-processed to change brightness and contrast as the pre and post-catastrophe images are taken during different timelines, these images were used to train deep learning models to generate more synthetic data. We trained a Pix2Pix [8] network that uses paired images to convert from one do-

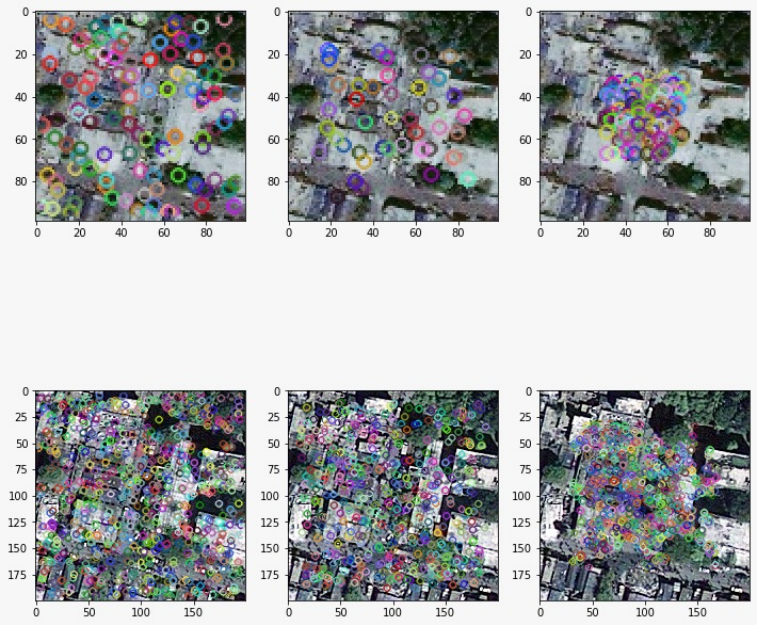
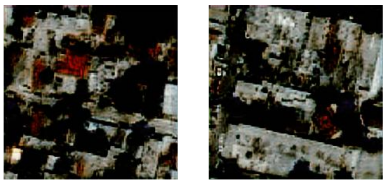


Figure 2: The top row represents pre-catastrophe images and the bottom row represents its post-catastrophe images. The images are visualization of SIFT, SURF and ORB features from left to right

main to another. We also trained a CycleGAN [29] network that performs unpaired image translation from one domain to another. We also modified the existing architecture of the generator module in CycleGAN. We replaced the classic UNet which is used as generator and replaced the backbone network from VGGNet to EfficientNet b3 [23] and EfficientNet b4 [23]. The generated samples are shown in Figure 3. CycleGAN model with different backbone networks are able to generate better looking samples.

(a) Generated samples from Pix2Pix

(b) Generated samples from classic CycleGAN



(c) Generated samples from CycleGAN with EfficientNet b3 as backbone for generator module

(d) Generated samples from CycleGAN with EfficientNet b4 as backbone for generator module

Figure 3: Generated post-catastrophe samples

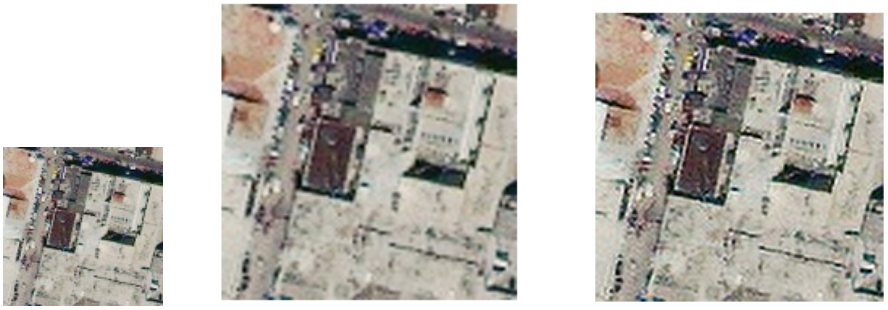
Evaluating the quality of the generated images by just looking at them is an inefficient

and subjective way to decide if the images are good. In order to evaluate the generative model objectively, we used the inception score [20] metric. Even though the images generated by EfficientNet b3 and b4 as backbone *look* relatively better, they achieve a low inception score and can observe that some pixels are blacked out. Therefore, the problem as mentioned in the previous subsection, could lie in the discrepancy of the spatial resolution between the pre and post-disaster images.

### 4.3 Enhancing pre-disaster images

The previous subsections noted that the observed discrepancy in the spatial resolution of the images could be a bottleneck. To overcome this, we used methods to upsample the spatial resolution of the images in pixel space. These methods are listed down below and upsampled images are shown in Figure 4

- ***Interpolation methods*** - we interpolated the pixels of the image to increase the spatial resolution. The different interpolation methods we used are Nearest Neighbor, Bilinear and Bicubic interpolation. These methods, however, are not specified and trained for satellite images and hence suffer from problems such as bleeding-edge pixels and also since these images are taken from altitudes in excess of 200 miles, the adjacent pixel values have a huge difference in them and this leads to high degrees of spatial uncertainty.
- ***Neural network based upsampling*** - There is a plethora of literature available [28, 4, 27, 1, 13] for image super-resolution, i.e improving the resolution of the images. This literature, however, does not concentrate on satellite images and the pre-trained models were trained on images from ground-based photodetection platforms. For the sake of completeness, we tried out a state of the art model, RDNet [28] to upsample these images.



(e) Orignal image (f) Bicubic interpolation (g) RDNet upsampling

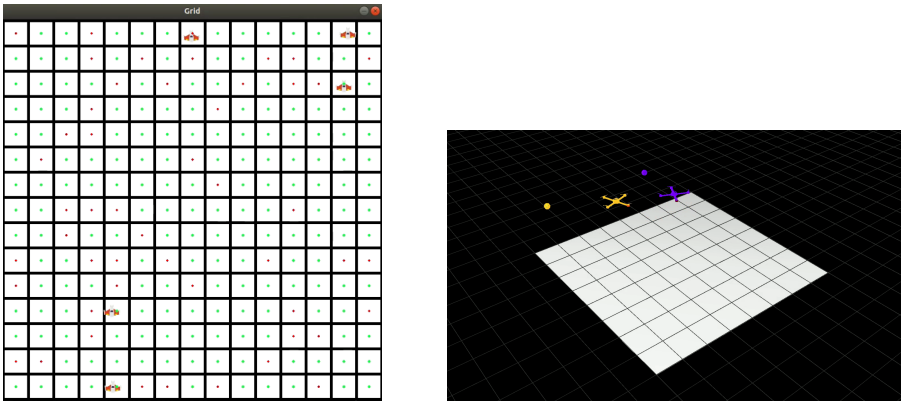
Figure 4: Original and upsampled images

As seen from Figure 4, the upsampled images suffer from the bleeding edge pixel problem. A solution to this bottleneck is to work in the spectral space of the images instead of the pixels and seek to improve the spectral-resolution of the images instead of

the spatial resolution. To achieve this, we must convert the pixels to spectral wavelengths and this can be done by converting the RGB pixels to HSV format which is a piece-wise linear transformation and the Saturation and Value can be discarded as they contain only brightness/darkness of colours, the Hue values should be taken which represents the colors in the image and there is a one-to-one mapping between hue values and wavelengths [22]. We also obtained different spectral images of the same location (Port au Prince) from Landsat [25] and Sentinel [21] satellites. These spectral images contain information from different frequency bands. By conflating the information from different frequency bands, we can upsample the images in spectral space [3, 26]. We leave the task of upsampling the image in spectral space for future work.

### 4.4 Simulations and RL

To train and evaluate the learning algorithm, we have built a custom simulation that replicates the task we are trying to achieve, i.e optimally map the entire environment. While traversing the environment, the drones might have to map certain locations (cells/rasters) for a longer period of time due to the increased damage or noise in measurements, we call these locations with this type of uncertainty as anomalous. We used open-source tools to build a 2D and 3D simulation where the drones make decisions to cover a grid optimally. The major difference between the 2D and 3D simulation is the number of degrees of freedom the drones have. In the 3D simulation, the drones have 6 degrees of freedom whereas in the 2D simulation the drones only have 3 degrees of freedom. The difference in degrees of freedom increases the complexity of decisions that the drones have to make at each time step. Screenshots from the simulation are shown in Figure 5. In order to start from a less complex task (lesser degrees of freedom) for the reinforcement learning algorithm, we train and evaluate our agents on the 2D simulation to begin with.




(a) 2D Simulation (b) 3D Simulation

Figure 5: Screenshots from Simulation

The cells to be mapped by the drones are represented as grids. In the 2D simulation, the red dots represent anomalous cells and hence they requires multiple passes by the drones to reduce its uncertainty. Once all the cells are mapped by the drone swarm, the episode

terminates. The agent gets a reward every time it successfully maps a cell and a relatively higher reward when it maps an anomalous cell. There is a negative reward, i.e a punishment that exponentially increases per time step. This punishment forces the agent to terminate the episode as soon as possible, hence, covering the cells as fast as possible.

We used Deep Q-Learning algorithm [16] for training the agent which controls the drones. The agent has a Multi-Layer Perceptron (MLP) as a policy for each of the drone. These MLPs map a sparse representation of the input state to an action for each of the drone at a time-step. During training, the agent randomly explores the environment and collects information and stores them in a buffer which is then later sampled from at intervals and used to train the MLP. The Deep Q-learning algorithm iteratively approximates the Q function for each of the state and is established as a baseline for the learning based methods. The algorithm struggles to generalize and the drone moves to one side of the environment, i.e it always picks the same action every time. We describe this as a form of action collapse. To overcome this problem, we decided to use actor-critic methods with policy gradients to update the actor and critic network. The agent is trained using Advantage Actor-Critic (A2C) [12] algorithm. The actor estimates an action given the state, the critic scales the reward function encouraging action with high rewards, the critic is subtracted by a baseline value which is the Value function, this reduces the noisiness in the signal and it is called advantage. Both Actor and Critic are Multi-Layer Perceptron network with memory state implemented using Gated Recurrent Unit (GRU). The agent provides a sample of probabilities over the set of discrete actions and we sample from that set for the agent to act. We observe that when we maintain the number of rollout steps and decrease the number of total steps the agent is trained, we are able to avoid the problem of action collapse. We hypothesize that this could be because as the number of steps increases, the drones memorize the most frequent action and repeats the same during the test time. This is similar to the problem of overfitting in machine learning

### 4.5 Damage Equations

To evaluate the seismic effects on the structures, we decided to build damage equations that model this property using certain factors pertinent to the catastrophe. Damage equations can be used to make a likelihood estimation of damage in a cell/raster. We modelled fragility equations using [7] which is used to quantify the extent of damage at a given location given the spectral displacement due to the quake at that location. This estimation can be used as an additional parameter to the path planning algorithm which can prioritize cells based on this estimation and also prune the generated synthetic samples. The damage state is estimated is classified according to European Macroseismic Scale (EMS-98) from DG1 to DG5. DG1 indicating negligible damage for a structure whereas DG5 indicating destruction. We were able to quantify the damage states for different zones based on structural density which is show in Figure 6, we can infer that shanty zones are at a higher risk relative to others. This part of our work is a supplement and can be considered

an experimental addition.

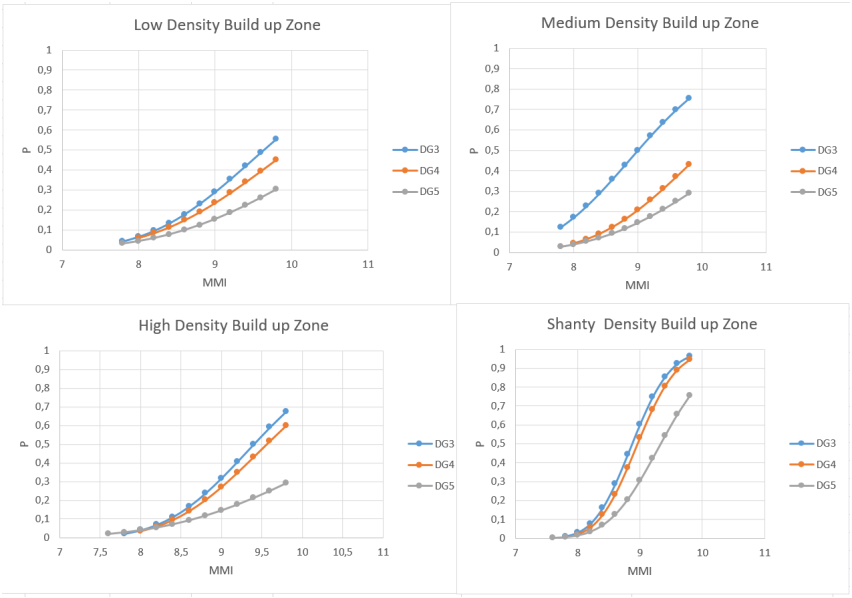



Figure 6: Damage states based on structural density. X-axis represents Modified Mercalli Intensity scale (MMI) and Y-axis represents probability. Shanty zones represent crudely built structures.

## 5 Conclusions

In this work, we have conducted experiments on ways to identify changes and generate synthetic data. We showcased that enhancing satellite images in pixel space is a complex task and it can be conducted in spectral space. We also showcase 2D and 3D simulation environments we have built and also trained agents in these environments using Q-learning and policy gradient methods. As an addition, we showcase how fragility curves can be constructed to model damage state at a location.

We intend to train the agents in a 3D simulation environment that better replicates the real-life and generate synthetic images in spectral space and prune the generated images based on damage equations and also detect changes from the images. We leave these for future work.

## 6 Acknowledgement

We would like to sincerely thank Dr. Joseph Bullock (AI Researcher at UN), Dr. Debra Laefer (New York University) and Dr. Ufuk Hancilar (Bogazici University) for their valuable feedback and guidance in this project. Any errors in the manuscript are our responsibility.

# References


[1] Saeed Anwar and Nick Barnes. "Densely Residual Laplacian Super-Resolution." en. In: *arXiv:1906.12021 [cs, eess]* (July 2019). arXiv: 1906.12021. URL: http://arxiv.org/abs/1906.12021 (visited on 07/13/2020).

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features." In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8.

[3] V. G. Bochkareva et al. "Methods for improving image quality using spatial spectral analysis." In: *Journal of Computer and Systems Sciences International* 54.6 (Nov. 2015), pp. 897–904. ISSN: 1555-6530. DOI: 10.1134/S1064230715060027. URL: https://doi.org/10.1134/S1064230715060027.

[4] Tao Dai et al. "Second-Order Attention Network for Single Image Super-Resolution." en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 11057–11066. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.01132. URL: https://ieeexplore.ieee.org/document/8954252/ (visited on 07/13/2020).

[5] Ian J. Goodfellow et al. "Generative Adversarial Networks." en. In: *arXiv:1406.2661 [cs, stat]* (June 2014). arXiv: 1406.2661. URL: http://arxiv.org/abs/1406.2661 (visited on 03/28/2020).

[6] *Haiti imagery layer now available*. en. Library Catalog: maps.googleblog.com. URL: https://maps.googleblog.com/2010/01/haiti-imagery-layer-now-available.html.

[7] Ufuk Hancilar, Fabio Taucer, and Christina Corbane. "Empirical Fragility Functions based on Remote Sensing and Field Data after the 12 January 2010 Haiti Earthquake." en. In: *Earthquake Spectra* 29.4 (Nov. 2013), pp. 1275–1310. ISSN: 8755-2930, 1944-8201. DOI: 10.1193/121711EQS308M. URL: http://journals.sagepub.com/doi/10.1193/121711EQS308M (visited on 04/30/2020).

[8] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks." In: *CVPR* (2017).

[9] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks." en. In: *arXiv:1812.04948 [cs, stat]* (Mar. 2019). arXiv: 1812.04948. URL: http://arxiv.org/abs/1812.04948 (visited on 07/09/2020).

[10] Tero Karras et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation." en. In: *arXiv:1710.10196 [cs, stat]* (Feb. 2018). arXiv: 1710.10196. URL: http://arxiv.org/abs/1710.10196 (visited on 07/09/2020).

[11] Arbaaz Khan, Vijay Kumar, and Alejandro Ribeiro. "Graph Policy Gradients for Large Scale Unlabeled Motion Planning with Constraints." en. In: *arXiv:1909.10704 [cs, math]* (Sept. 2019). arXiv: 1909.10704. URL: http://arxiv.org/abs/1909.10704.

[12] Vijay R Konda and John N Tsitsiklis. "Actor-Critic Algorithms." en. In: (), p. 7.

[13] Shu Kong and Charless Fowlkes. "Image Reconstruction with Predictive Filter Flow." en. In: *arXiv:1811.11482 [cs, eess]* (Nov. 2018). arXiv: 1811.11482. URL: http://arxiv.org/abs/1811.11482 (visited on 07/13/2020).

[14] Qingbiao Li et al. "Graph Neural Networks for Decentralized Multi-Robot Path Planning." en. In: *arXiv:1912.06095 [cs]* (Dec. 2019). arXiv: 1912.06095. URL: http://arxiv.org/abs/1912.06095.

[15] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints." en. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: http://link.springer.com/10.1023/B:VISI.0000029664.99615.94 (visited on 06/21/2020).

[16] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning." en. In: (), p. 9.

[17] Tolga Özaslan et al. "Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs." In: *IEEE Robotics and Automation Letters* 2.3 (July 2017). Conference Name: IEEE Robotics and Automation Letters, pp. 1740–1747. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2699790.

[18] Huy Xuan Pham et al. "Cooperative and Distributed Reinforcement Learning of Drones for Field Coverage." en. In: *arXiv:1803.07250 [cs]* (Sept. 2018). arXiv: 1803.07250. URL: http://arxiv.org/abs/1803.07250.

[19] E. Rublee et al. "ORB: An efficient alternative to SIFT or SURF." In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571.

[20] Tim Salimans et al. "Improved Techniques for Training GANs." en. In: *arXiv:1606.03498 [cs]* (June 2016). arXiv: 1606.03498. URL: http://arxiv.org/abs/1606.03498 (visited on 06/21/2020).

[21] *Sentinel-2 MSI: MultiSpectral Instrument, Level-2A*. en. Library Catalog: developers.google.com. URL: https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2_SR (visited on 07/13/2020).

[22] Brian Smits. "An RGB-to-Spectrum Conversion for Reflectances." In: *J. Graph. Tools* 4.4 (Dec. 1999), pp. 11–22. ISSN: 1086-7651. DOI: 10.1080/10867651.1999.10487511. URL: https://doi.org/10.1080/10867651.1999.10487511.

[23] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." en. In: *arXiv:1905.11946 [cs, stat]* (Nov. 2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946 (visited on 06/21/2020).

[24] Teodor Tomic et al. "Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue." In: *IEEE Robotics Automation Magazine* 19.3 (Sept. 2012). Conference Name: IEEE Robotics Automation Magazine, pp. 46–56. ISSN: 1558-223X. DOI: 10.1109/MRA.2012.2206473.

[25] *USGS Landsat 8 Collection 1 Tier 1 Raw Scenes*. en. Library Catalog: developers.google.com. URL: https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C01_T1 (visited on 07/13/2020).

[26] A. Villa et al. "Super-resolution: an efficient method to improve spatial resolution of hyperspectral images." In: *2010 IEEE International Geoscience and Remote Sensing Symposium*. 2010, pp. 2003–2006.

[27] Xintao Wang et al. "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks." en. In: *arXiv:1809.00219 [cs]* (Sept. 2018). arXiv: 1809.00219. URL: http://arxiv.org/abs/1809.00219 (visited on 07/13/2020).

[28] Yulun Zhang et al. "Residual Dense Network for Image Super-Resolution." en. In: *arXiv:1802.08797 [cs]* (Mar. 2018). arXiv: 1802.08797. URL: http://arxiv.org/abs/1802.08797 (visited on 07/13/2020).

[29] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss." In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.