2022-06-10

OSM-GAN: Using Generative Adversarial Networks for Detecting Change in High-Resolution Spatial Images

Lasith Niroshan
Technological University Dublin, d19126805@mytudublin.ie

James Carswell
Technological University Dublin, james.carswell@tudublin.ie

Recommended Citation

Niroshan, L., Carswell, J.D. (2022). OSM-GAN: Using Generative Adversarial Networks for Detecting Change in High-Resolution Spatial Images. In: Bourennane, S., Kubicek, P. (eds) Geoinformatics and Data Analysis. ICGDA 2022. Lecture Notes on Data Engineering and Communications Technologies, vol 143. Springer, Cham. https://doi.org/10.1007/978-3-031-08017-3_9

Funder: Technological University Dublin College of Arts and Tourism

OSM-GAN: Using Generative Adversarial Networks for Detecting Change in High-Resolution Spatial Images

Lasith Niroshan and James D. Carswell

Technological University Dublin, Ireland
d19126805@mytudublin.ie ; james.carswell@tudublin.ie

Abstract. Detecting changes to built environment objects such as buildings/roads/etc. in aerial/satellite (spatial) imagery is necessary to keep online maps and various value-added LBS applications up-to-date. However, recognising such changes automatically is not a trivial task, and there are many different approaches to this problem in the literature. This paper proposes an automated end-to-end workflow to address this problem by combining OpenStreetMap (OSM) vectors of building footprints with a machine learning Generative Adversarial Network (GAN) model - where two neural networks compete to become more accurate at predicting changes to building objects in spatial imagery. Notably, our proposed OSM-GAN architecture achieved over 88% accuracy predicting/detecting building object changes in high-resolution spatial imagery of Dublin city centre.

Keywords: Change Detection, Remote Sensing, OpenStreetMap, Generative Adversarial Networks, GIS

1 Introduction

Geospatial change detection functionality is an acknowledged component of many practical GIS applications, for example, urban planning, natural disaster prediction, agricultural monitoring, etc. As such, various approaches have been explored and employed to automatically recognise changes in spatial imagery over time. These range from basic statistical implementations to traditional image processing techniques to more complex Deep Learning approaches. In urban cases, successfully obtaining accurate change detection results depends highly on imagery resolution, as lower resolution images can obfuscate a significant amount of important ground object detail – e.g., the precise edges and intersections of buildings in a crowded urban setting.

However, detecting feature/object changes in aerial/satellite (spatial) imagery is a challenging task due to many factors – e.g., a general lack of easily attainable/freely available high-resolution spatial imagery, automating the complex object (e.g., building) extraction process, and the comparatively modest accuracy (+/-80%) of current change detection algorithms applied to this domain. In support of this study, a customised spatial image *crawler* was developed to search for freely available *Google Earth* and *Bing Maps* satellite images from various sources at different spatial

resolutions. Once obtained, this raster data is merged with *OpenStreetMap* (OSM) vectors to train the novel OSM-GAN change detection mechanism described in this paper.

At present, Convolutional Neural Network (CNN) models are used a great deal in Artificial Intelligence (AI) applications for resolving general image processing and classification tasks. Among the various *Deep Learning* (a subfield of Machine Learning) methods, Generative Adversarial Networks (GAN) have recently been developed to learn (train) a function (model) that maps (translates) an *input* image to an *output* image. Over the past five years, the task of translating one possible representation of data into another, such as image-to-image translation, has become a common application for GANs. As an example, Isola et al. proposed a general-purpose adversarial network solution in 2017 for image-to-image translation named *Pix2Pix* [1].

Our approach applies the *Pix2Pix* image translation technique to predict/identify possible changes to building objects in high-resolution (30cm/pixel) satellite images. To begin, we must first convert OSM building footprint data (vector) to raster format for use as an output image - since *Pix2Pix* image translation expects both input and output images in raster format for training purposes. Figure 1 shows a real-world (Dublin) example of one *Pix2Pix* training set used to train our OSM-GAN model.

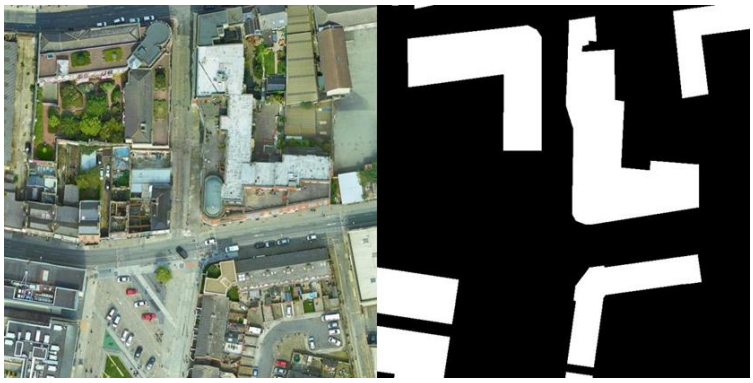

Fig. 1. A joined Pix2Pix raster training sample used for learning the OSM-GAN model. The left side is the input Google satellite image – the right side is the current OSM building footprint output image (feature-map) of the same Dublin area.

This paper describes how online crowdsourced spatial data can be utilised successfully in state-of-the-art Machine Learning applications. In it, we propose an automated change detection framework for OSM buildings that exploits GAN image-to-image translation techniques. The paper is organised as follows: Section 2 covers some background and related work on this topic; Section 3 explains our proposed OSM-GAN methodology in some detail; Section 4 reports on experimental results followed by some Conclusions with a brief discussion on plans for future work.

2 Background and Related Work

This section reviews some related background work relevant to our approach, including an overview of GANs, Conditional GANs, and other noteworthy change detection mechanisms in the literature. For example, applications and improvements to the GAN methodology have increased significantly over time, with different types of GAN frameworks developed for various purposes:

- CapsGAN to generate 3D images with various geometric transformations [3]
- GANSynth to produce audio streams [4]
- GauGAN to transform doodles into highly realistic landscapes [5]
- StyleGAN to generate more realistic images (e.g., human faces, cars, and rooms) [6]
- ChemGAN for drug discovery [7]

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GAN) were proposed by Goodfellow et al. in 2014 as a new class of Machine Learning models where two separate models compete against each other as if in a game, e.g., chess/backgammon/etc. [2]. The basic GAN works like a *minimax* recursive algorithm to find the optimal move for a player. One model is called the *Generator* (G), and the other is called the *Discriminator* (D). Briefly, the Generator trains a generative model to generate fake data similar to the real *feature-map* (right side of the training set image) from a random noise vector (array of 0s and 1s) as input. Conversely, the adversary Discriminator is trained to classify/distinguish between the generated (fake) feature-map and the ground truth (real) feature-map.

2.2 Conditional Generative Adversarial Networks

Mirza and Osindero introduced *Conditional Generative Adversarial Networks* (CGAN) also in 2014 [8]. The significant improvement of CGAN over GAN is the addition of a conditional state to the output generation, as usually there is no control over generating output in a GAN. CGAN includes a *condition* (uses both left and right sides of the training sample simultaneously) as input to the Generator and Discriminator to help resolve the issue of an image being real or fake. As it happens, including a condition (feature-map) in the training sample input to the Discriminator function results in a more accurate method for identifying real images – if there is a building in the satellite image, there should also be a predicted building in the resulting feature-map.

Image-to-Image Translation

The main idea behind *image-to-image translation* is that a given input image (e.g., a sketch/outline of an object) translates or transforms into another higher-level representation (e.g., a photo-realistic image) of the set of input information. Therefore, many computer vision and image processing problems (e.g., edge detection, object localisation, sketch-to-photo translation, etc.) can be interpreted as a form of image-to-image translation.

Isola et al. [1] presented several generalised uses of Conditional GAN based image-to-image translation such as labels-to-street scenes, black&white images-to-colour images, sketches-to-photos, style transfer applications, and especially aerial images-to-maps, the main focus of this study. *Pix2Pix* is their implementation of image-to-image translation, which is freely available for use in *GitHub*[1]. We use an updated version of *Pix2Pix* in the OSM-GAN experiments carried out in this work.

2.3 Detecting Spatial Changes in Spatial Images

An accurate change detection mechanism can initiate many other advanced geo-analytic research applications in the GIS domain – where the challenge of change detection has been investigated many ways over the years to address various mapping problems. For example, a considerable number of image processing and computer vision approaches have been introduced for temporal change detection in spatial images, such as Markov random fields [9], Principal Component Analysis [10], CNN based difference image approach [11], and Recurrent neural network-based U-Net models [12].

Recently (2018), a GAN investigation was conducted by the China University of Geosciences to enhance *Pix2Pix* (called *ePix2Pix*) classifications of remote sensing images [14]. They claim improvements to *Pix2Pix* that provide higher classification accuracy when compared to traditional methods. Previous studies reveal that traditional *Pix2Pix* has a limited ability to learn complex image features, such as complicated patterns, thus leading to low classification accuracy among other prediction complications [14] [15]. *ePix2Pix* proposed adding a *Controller* to the model – which now consists of three parts; Generator, Discriminator, and Controller. The Controller allows a relationship between classification and reconstruction, an additional step to improve classification accuracy. Experimental results report that *ePix2Pix* scored higher compared to *Support Vector Machines* (SVM), *Artificial Neural Networks* (ANN), *CycleGAN*, and *Pix2Pix*.

Also, in 2018, Lebedev et al. carried out an experiment on conditional adversarial networks to detect changes in season-varying remote sensing images [16]. Their approach presented three types of tests on synthetic and actual images [16]. In their approach, the Discriminator required three input images to perform the classification (two images for comparison and one for the difference map) - otherwise, the network structure is the same as the *Pix2Pix* structure. Even though the proposed methodology delivered accurate results, changes to *mutable* objects (e.g., vehicles) were also identified as a change in the map (Figure 6). Although mutable objects should not be considered as a "change in a map," this idea can be utilised to detect changes to immutable objects as well (e.g., buildings).

More recently, Albrecht et al. (2020) presented a method to programmatically identify outdated map regions from current OSM data [13]. This work introduced the use of GAN's, namely *Feature-Weighted CycleGAN* (fw-CycleGAN), to identify any changes in a given geographic area. Although this approach focused on finding changed areas to produce a heat map, it does not explicitly identify/obtain the changed objects themselves, such as a specific newly built building, along with its related ground

[1] https://github.com/phillipi/pix2pix

coordinates. Instead, the objective was to train *fw-CycleGAN* to recognise styles (colour/texture/etc.) in a given set of images and generate OSM-like maps with the same *look* (style/pattern) as output.

3 OSM-GAN Methodology

Our novel OSM-GAN approach integrates many incremental improvements noted in the above strategies for detecting changes in high-resolution spatial images. For example, it targets changes to immutable objects only (i.e., buildings) within a user-defined geographic area digitised on an aerial/satellite image. The designated polygonal Area of Interest (AoI) is then reduced to its minimum bounding rectangle (MBR) coordinates to facilitate further processing operations. We consider the appearance of a new building object or the disappearance of an old object as a change when compared to the current state of the OSM database. For example, if a new building appears in a raster satellite image but is not evident in the OSM vector database, that building is considered a potential changed object within the given AoI. The following system architecture diagram (Figure 2) illustrates the overall workflow of our automated OSM-GAN approach, and the following sections describe each step in more detail.

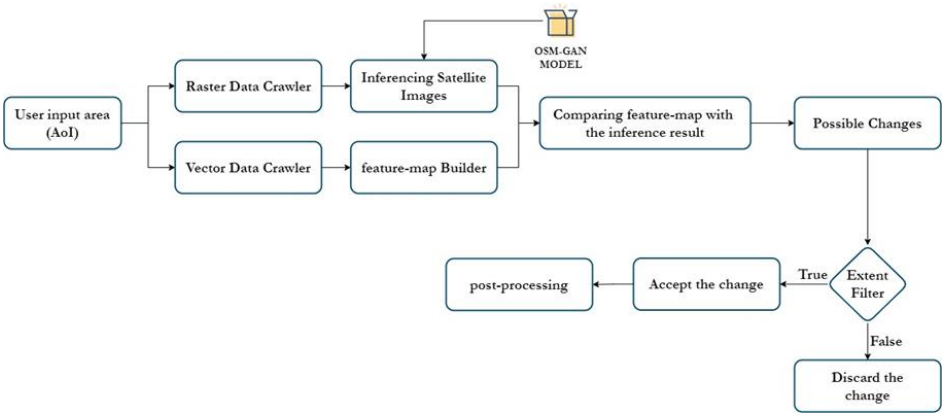

Fig. 2. System architecture diagram of proposed OSM-GAN approach. First, an Area of Interest (AoI) is manually digitised to begin the change detection process. Then, raster and vector data crawling processes launch automatically to download relevant satellite/OSM data. The feature-map prediction phase then starts by using a pre-trained OSM-GAN model. Simultaneously, an OSM feature-map containing predicted buildings is generated. Once both processes complete, feature-map comparisons (OSM to OSM-GAN), extent filtering, and post-processing is applied to each predicted feature-map.

3.1 Spatial Data Crawling and Processing

Data is the most valuable resource for any Machine Learning task, and our AI application requires several spatial data sources for input. As a result, specialised data-mining programs (raster/vector data *crawlers*) were developed to address our spatial data needs. The below sections describe how our necessary data requirements are fulfilled, with further data processing steps described thereafter.

Crawling Vector Data

OpenStreetMap [17] is the primary vector data source for this work since OSM vector data (e.g., building footprints) in *GeoJSON* format (*key*:*value* pairs) is relatively easy to handle by post-processing programs [18]. The vector data crawler starts by parsing an OSM *Overpass* query - "a read-only API that serves up custom selected parts of the OSM map data"[2] - into the program. One practical advantage of using the Overpass query is an unlimited freedom to change the OSM feature type (e.g., building, road, river, etc.), the geographic area, and many other OSM attributes by just updating the query without updating the code. The OSM feature extractor and mining program downloads and saves the vector data (within the AoI) into GeoJSON formatted files.

The main contribution of this OSM data is to create building object *mask* images (called *feature-maps*) for use in the training process. Reducing the effect of shadows is another partial benefit of using this object-mask method - since shadows of buildings can affect Mask-RCNN and other traditional image processing techniques. *Pix2Pix* predictions trained using object masks show that the effect of shadows does not significantly affect the accuracy of the OSM-GAN change detection mechanism.

Initially, all OSM crawled data is stored in one large GeoJSON file containing all extracted objects. The main disadvantage of this single file setup is the difficulty of extracting relevant building objects afterwards. Therefore, this large file gets automatically separated into a "one-object-one-file" format and subsequently stored as many individual building object files. This modification results in a significant acceleration of subsequent processes.

To use GeoJSON objects effectively, a translating program first converts them into binary images and stores them in separate directories based on their ground coordinates. Once this process completes, a merging process overlays each of these masks into a single 256x256 pixel sized *feature-map* containing all buildings and used for generating the training images. Figure 3 illustrates an example of separated building objects and the result of the merging process. Note that the white *blobs* in the figure indicate the relevant building objects converted to raster from OSM vector data.

[2] https://wiki.openstreetmap.org/wiki/Overpass_API

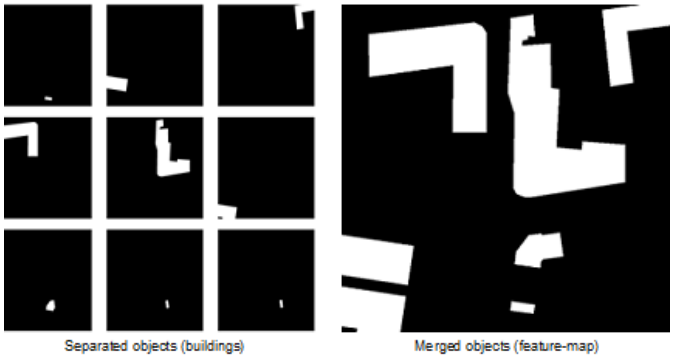

Fig. 3. An example of separated building objects and merged objects. The separated objects were created from the *GeoJSON* data crawled from OSM using their *Overpass* API.

Crawling Raster Data

A raster data crawler was developed for downloading the most up-to-date, freely available satellite imagery related to the same *Overpass* query AoI. Once the vector crawling process completes, raster data mining initiates. There are two main raster mining sub-programs: 1) *Bing* imagery crawler; 2) *Google* imagery crawler – both instructed to apply the relevant crawler at the resolution required for a given task. In this experiment, we chose *Google Earth* images due to their higher quality 30cm pixel resolution. Also, the downloaded images get automatically cropped (into 256x256 pixel tiles), stored, and indexed in a quadtree-based directory structure to make them easier to process in subsequent phases.

Combining and Filtering Input Data

Once the data crawling programs complete successfully, a data processing phase begins to assemble the acquired data according to the requirements of the Deep Learning algorithm. The OSM-GAN training process requires two input images - a satellite image and its conjugate OSM generated *feature-map* image. As the *Pix2Pix* program is pre-configured to use 600x300 pixel input images, both the satellite image and its conjugate feature-map are re-scaled (using *OpenCV*[3]) into 300x300 pixel tiles and joined together - resulting in the overall 600x300 pixel training image sample shown in Figure 1. However, it was found that low object-dense training images can increase the number of false positives predicted by the model. Therefore, feature-map images that do not contain objects present in their conjugate satellite image are eliminated from the training phase to achieve a higher *Pix2Pix* prediction accuracy

The Python *NumPy*[4] module is used to determine the number of white pixels, and a ratio is calculated from both the total number of pixels and the number of white (object) pixels. Then a threshold phase determines if the given feature-mask is eligible for use in the training process. Currently, predefined (trial/error) threshold values of 0.25

[3] https://docs.opencv.org/4.5.2/
[4] https://numpy.org/

(lower) and 0.75 (upper) are used, but adaptive thresholds are planned for in the next phase of development.

3.2 Training the OSM-GAN Model

After the above raster/vector data crawling and processing steps, the filtered data is forwarded to the training stage. The *PyTorch* version of the *Pix2Pix* implementation [1] is used to build the OSM-GAN model. The filtered data samples get split into a 3:2 ratio (train:validation), and fed into the *Pix2Pix* algorithm. The OSM-GAN model was trained using an NVIDIA Ge-Force RTX 2060 GPU with CUDA. With this configuration, the training process required 6 hrs to complete 400 epochs. Figure 4 illustrates the data flow diagram up to this stage.

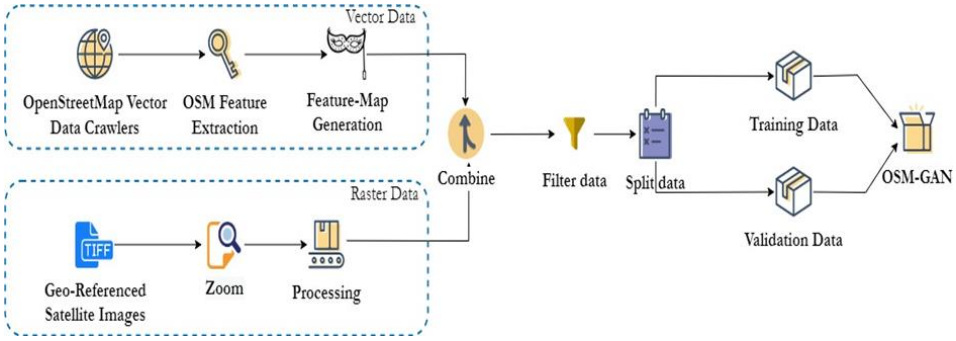

Fig. 4. The fully automated input/output data pipeline for the OSM-GAN framework.

It was observed that the accuracy of predictions for OSM-GAN models increased with higher resolution imagery – but only up to a certain zoom level, after which prediction accuracy starts to decrease. A series of experiments to discover the best image resolution for training OSM-GAN models revealed that prediction accuracy begins to decrease after 0.14m/pixel (zoom level 20) resolution. For example, it was found that OSM-GAN predictions on a 10cm resolution dataset classified water bodies and fields as buildings. As such, 30cm resolution images were committed to the qualitative phase of OSM-GAN model predictions. Figure 5 below illustrates some intermediate outputs from the training phase.

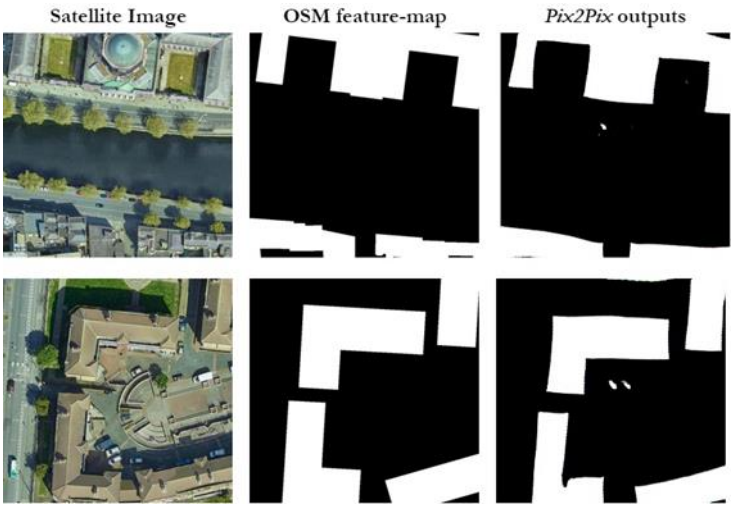

Fig. 5. Some intermediate outputs of the model training phase using satellite images with 30cm resolution. Note that shadows, water, and vehicles are not classified as buildings by this OSM-GAN approach

3.3 Detecting Changes

The process of detecting actual object changes in the resulting images is relatively straightforward when compared to previous steps. Once the *Pix2Pix* prediction is performed on a given satellite image, the current raster view of the predicted image is reconstructed using the current raster data converted from OSM. Figure 6 compares the satellite image, current OSM data, and the prediction image.

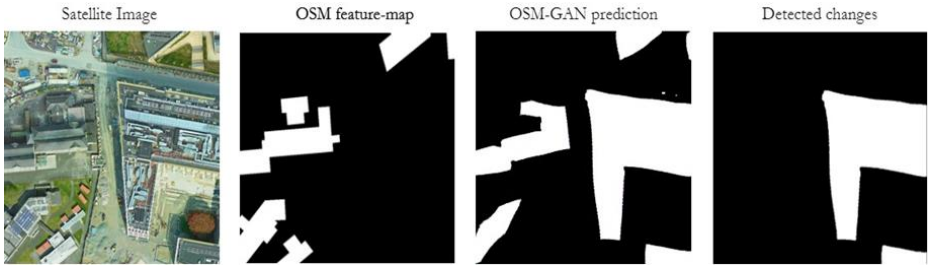

Fig. 6. The OSM-GAN prediction detects building object changes in a spatial image.

Next, the separate building objects are extracted from the prediction results using a conventional contour finding algorithm to perform this operation. After filtering and aggregating the changed object references, Figure 6 gets extended to include the last image showing detected building changes.

4 Experimental Results

After generating the final image of detected building changes, a post-processing mechanism activates to enhance the overall shape of the changed object(s). First, a *regularisation* operation to reduce the number of vertices is applied to individual building objects to smooth/straighten building outlines. The regularisation phase is a critical step because the OSM code of conduct [19] requires that any objects input to the OSM database must contain a minimal number of nodes. A *perpendicular distance* algorithm was found to work best with our OSM-GAN predictions. For example, Figure 7 compares a non-regularised polygon to a regularised polygon using perpendicular distance simplification.

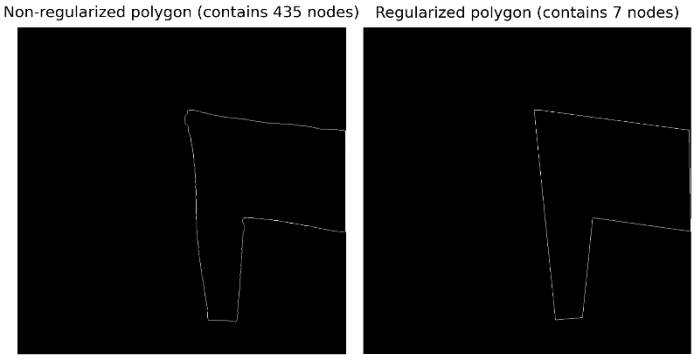

Fig. 7. A comparison of non-regularised and regularised polygons (building outlines).

As described in Section 3, OSM building footprint data was utilised to create training datasets for OSM-GAN. One advantage of using an OSM data-based approach is that it can also be used to calculate a confusion matrix (true/false positives/negatives) to evaluate the accuracy of predicted images quantitatively - based on the Object Overlap Matrix and corresponding OSM object labels (used as ground truth). Table 1 shows the experimental results of our OSM-GAN approach for detecting changes to buildings in 30cm spatial images are at least 88% accurate. However, how to quantitatively evaluate a generated (synthesised) image is still an open and complex problem. In future analyses, we propose to incorporate the "Inception Score" [20] and "Frechet Inception Distance" [21] measures into this metric.

Table 1. Analysis of OSM-GAN Prediction Accuracy

Accuracy	**88.4%**
Recall	62.0%
Precision	80.5%
F1 score	76.6%

5 Conclusions

This paper proposes an improved solution for detecting changes in spatial images based on combining the incremental developments found in previous work. The overall OSM-GAN system integrates raster/vector data crawling functionality with several other image processing operations, such as image-to-image translation, image difference calculation, and vector to raster conversions into a unified end-to-end workflow. The next phase of research plans to compare change detection results of various experimental combinations of 24-bit satellite images vs 8-bit (grayscale) imagery when applied to OSi[5] and OSM building footprints. Additionally, future experiments will be conducted using Kay[6] - Ireland's national supercomputer for academic researchers. Kay consists of a cluster of 336 nodes, each node having 2x20-core 2.4 GHz Intel Xeon Gold 6148 processors; an enormous advantage for minimising model training times.

Acknowledgments

The authors wish to thank all contributors involved with the OpenStreetMap project. This research is funded by Technological University Dublin College of Arts and Tourism, SEED FUNDING INITIATIVE 2019-2020.

References

1. Isola, P., Zhu, J.Y., Zhou, T. and 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).
2. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661.
3. Saqur, R. and Vivona, S., 2019, April. Capsgan: Using dynamic routing for generative adversarial networks. In Science and Information Conference (pp. 511-525). Springer, Cham.
4. Engel, J., Agrawal, K.K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A., 2019. Gansynth: Adversarial neural audio synthesis. arXiv preprint arXiv:1902.08710.
5. Park, T., Liu, M.Y., Wang, T.C. and Zhu, J.Y., 2019. Semantic image synthesis with spatially-adaptive normalisation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2337-2346).
6. Karras, T., Laine, S. and Aila, T., 2019. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).
7. Benhenda, M., 2017. ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity?. arXiv preprint arXiv:1708.08227.

[5] https://www.osi.ie/about/open-data/
[6] https://www.ichec.ie/about/infrastructure/kay

8. Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
9. Gong, M., Su, L., Jia, M., & Chen, W. (2013). Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images. IEEE Transactions on Fuzzy Systems, 22(1), 98-109.
10. Yousif, O., & Ban, Y. (2013). Improving urban change detection from multitemporal SAR images using PCA-NLM. IEEE Transactions on Geoscience and Remote Sensing, 51(4), 2032-2041.
11. de Jong, K. L., & Bosman, A. S. (2019, July). Unsupervised change detection in satellite images using convolutional neural networks. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
12. Papadomanolaki, M., Verma, S., Vakalopoulou, M., Gupta, S., & Karantzalos, K. (2019, July). Detecting urban changes with recurrent neural networks from multitemporal Sentinel-2 data. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (pp. 214-217). IEEE.
13. Albrecht, C.M., Zhang, R., Cui, X., Freitag, M., Hamann, H.F., Klein, L.J., Finkler, U., Marianno, F., Schmude, J., Bobroff, N. and Zhang, W., 2020. Change Detection from Remote Sensing to Guide OpenStreetMap Labeling. ISPRS International Journal of Geo-Information, 9(7), p.427.
14. Wang, X., Yan, H., Huo, C., Yu, J. and Pant, C., 2018, August. Enhancing Pix2Pix for remote sensing image classification. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 2332-2336). IEEE.
15. M. Lee and J. Seok, "Controllable generative adversarial network," arXiv preprint arXiv:1708.00598, 2017
16. Lebedev, M.A., Vizilter, Y.V., Vygolov, O.V., Knyaz, V.A. and Rubis, A.Y., 2018. CHANGE DETECTION IN REMOTE SENSING IMAGES USING CONDITIONAL ADVERSARIAL NETWORKS. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42(2).
17. OpenStreetMap. 2021. OpenStreetMap. [online] Available at: <https://openstreetmap.org/> [Accessed 22 April 2021].
18. Overpass-turbo.eu. 2021. overpass turbo. [online] Available at: <https://overpass-turbo.eu/> [Accessed 22 April 2021].
19. Code of conduct. 2021. OpenStreetMap. [online] Available at: <https://wiki.openstreetmap.org/wiki/Code_of_conduct/> [Accessed 22 April 2021].
20. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X., 2016. Improved techniques for training gans. Advances in neural information processing systems, 29, pp.2234-2242.
21. Nunn, E.J., Khadivi, P. and Samavi, S., 2021. Compound Frechet Inception Distance for Quality Assessment of GAN Created Images. arXiv preprint arXiv:2106.08575.