Generative Adversarial Networks for Ensemble Projections of Future Urban Morphology

Melissa R. Allen-Dumas
allenmr@ornl.gov
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Abigail R. Wheelis
abby.wheelis@centre.edu
Centre College
Danville, Kentucky, USA

Levi T. Sweet-Breu
sweetlt@ornl.gov
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Joshua Anantharaj
jananth2@vols.utk.edu
University of Tennessee
Knoxville, Tennessee, USA

Kuldeep R. Kurte
kurtekr@ornl.gov
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Figure 1: Los Angeles, California Skyline (David McNew/Getty Images North America/Getty Images).

ABSTRACT

As city planners design and adapt cities for future resilience and intelligence, interactions among neighborhood morphological development with respect to changes in population and resultant built infrastructure's impact on the natural environment must be considered. For deep understanding of these interactions, explicit representation of future neighborhoods is necessary for future city modeling. Generative Adversarial Networks (GANs) have been shown to produce spatially accurate urban forms at scales representing entire cities to those at neighborhood and single building scale. Here we demonstrate a GAN method for generating an ensemble of possible new neighborhoods given land use characteristics and designated neighborhood type.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

urban morphology, generative adversarial networks, machine learning

ACM Reference Format:
Melissa R. Allen-Dumas, Abigail R. Wheelis, Levi T. Sweet-Breu, Joshua Anantharaj, and Kuldeep R. Kurte. 2022. Generative Adversarial Networks for Ensemble Projections of Future Urban Morphology. In *The 5th ACM SIGSPATIAL Workshop on Advances on Resilient and Intelligent Cities (ARIC '22) (ARIC '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3557916.3567819

ARIC '22, November 1, 2022, Seattle, WA, USA

1 INTRODUCTION

As populations grow and shift, they will demand new built infrastructure within and at the edges of cities. Plans for these changes must consider architecture, society, culture, environment and technology so that neighborhoods develop in resilient and intelligent ways [5]. Designs for new neighborhood morphologies have been approached historically using the concept of *urban tissues* [12], which requires identifying, subsetting and combining the physical elements (common size, shape and material) from existing neighborhoods into new ones [9]. Because a large amount of data on the built environment are available in different formats and on different scales, machine learning techniques, including neural networks, have been employed to find patterns among these various data [2]. In particular, Generative Adversarial Networks (GANs), a type of deep neural network, have been applied to geospatial urban morphological image data to calculate new and representative land-use configurations quickly saving human planners time and potentially offering effective and novel suggestions for optimizing priorities for new neighborhoods [18, 20, 21]. One type of GAN used for generating neighborhood morphologies examines the types of neighborhoods characteristic of different cities using different models with exemplary training data for each city [4]. Another approach uses one type of geospatially located image to project a new type of image associated with the same location [7]. However, results of these studies are not necessarily published in formats that urban planners can use. Here, we combine these two methodologies to generate neighborhood morphological images by neighborhood type (residential, commercial or mixed) using land cover characteristic images and target building footprint and height images. We then outline a process for turning the results into GIS-readable shapefiles for use by prospective decision makers.

2 METHODOLOGY

GAN models are a set of deep neural networks comprising a generator and a discriminator in which the generator attempts to create synthetic output that cannot be distinguished by the adversarially-trained discriminator from real samples (Figure 2). The conditional GAN that we employ for this work learns to map land cover characteristics represented as an image paired with a concomitantly geolocated set of target building footprints, heights and placement to a synthetically generated image of building footprints, heights and placement. The model implemented here was developed in PyTorch, and is based in part on [16]. The full model and approach are described in the subsections following.

2.1 Model and Implementation

The generator for the GAN used here is a deep neural network constructed with skips as a Unet [7, 15]. The Unet structure allows for low-level features (in this case, roads and waterways) to flow through the generator from layer i to layer $n-i$, where n is the number of layers. Allowance for these skips speed up the encoding and decoding processes in the generator by limiting the bottle-necking that can take place at the central layer. The encoder and decoder of the generator include standardized blocks of convolutional, batch normalization, dropout, and activation layers (Tanh for the output layer and ReLU otherwise).

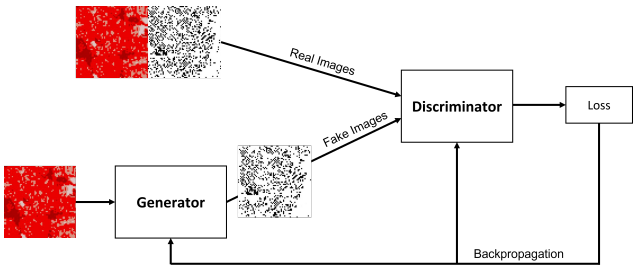

Figure 2: Overview of GAN architecture, including (left side) land cover inputs to the generator and land cover paired with generated or real buildings to the discriminator. The right side of the diagram shows how backpropagation of error (loss) influences training.

The discriminator is a PatchGAN classifier [7, 10] that determines whether sections of a generated image are real or fake–not just the image as a whole. This discriminator is run convolutionally across the image to provide decision output.

The objective function for the model is the same as that of [7] and is described in Equation (3). The generator for this GAN uses conditional adversarial loss for the discriminator model:

$$\mathcal{L}_{cGAN}(G,D) = E_{x,y}[logD(x,y)] + E_{x,z}[log(1-D(x,G(x,z))] \quad (1)$$

with G as the generator and D as the discriminator. Here x is the input image, y is the target image, and z is a random noise vector. L1 loss, or mean absolute pixel difference between the generated translation of the source image and the expected target image, is used for the generator:

$$\lambda\mathcal{L}_{L1}(G) = E_{x,y,z}[||y - G(x,z)||_1] \quad (2)$$

Again, x is the input image, y is the target image, z is the random noise, and G(x,z) is the generated image. λ in the equation is set equal to 100, which increases the importance of the L1 loss to 100 times that of the adversarial loss during generator training. These two losses are combined (weighted sum) into a composite loss function, which is used to update the generator:

$$G* = arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G) \quad (3)$$

While the objective function described here includes random noise as an input into the generator, it was ultimately not included by [7] and was not used here because this model structure tended to ignore the random input.

2.2 Data and Data Preparation

Three models were trained for this study so that characteristics of three different types of neighborhoods (1.92km x 1.92km tiles) could be captured: residential, commercial and mixed. Two types of images associated with the Los Angeles, California area were used: land cover images and images that included building footprints and locations each with a height attribute. Using the image-to-image approach, the models were trained to read in the land cover data and output a set of buildings to fit the land cover description. The

land cover images are drawn from the National Land Cover Database (NLCD) [3], which classifies 30-meter resolution patches of land according to their cover. The land cover classifications include woodland types (greens), scrublands (tan), and different densities of developed land (reds). With developed land in particular, darker shades of red represent more dense urban areas, but does not indicate the placement of individual buildings. To create the building imagery, building polygons obtained from Model America [11] were rasterized, aggregated to a 30-meter resolution to match the NLCD data, and given height as the raster value encoded in grayscale. In this case, the darker the grayscale in a certain area, the taller the building at that location. After rasterization, the datasets were split into square neighborhood tiles. A simple Python script was written to ensure the naming conventions from the two data sets matched and that the tile pairs were accurately linked geospatially. A sample pair is shown in Figure 3, with land cover on the left and building representations on the right.

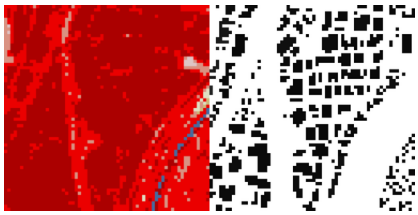

Figure 3: Example input and target pair of land cover and building images.

Using the PyTorch deep learning library function ImageFolder, we labeled corresponding 1.92km x 1.92km neighborhood tiles in the Model America polygon data set as residential, commercial or mixed by calculating the percentage by volume of each building type (included as attributes in Model America) in each tile. Images were labeled as residential if they included a total residential building volume of at least 80%. To be labeled as commercial, the total commercial building volume threshold value was 75%. Otherwise, the tiles were labelled as mixed. We used these labels to identify and sort the respective images used in training the models.

2.3 Hyperparameter Tuning

As in [4], hyperparameters for both generator and discriminator in the models include an Adam optimizer initialized with a learning rate of 0.0002 and betas (exponential decay rates) of 0.5 and 0.999 respectively. Batch size for the models was 18, and each model was run for 500 epochs.

2.4 Model Training

Using mini-batch stochastic descent optimization [7], the three image-to-image translation models were trained, one each on residential, commercial and mixed neighborhoods. The residential data set contained 514 image pairs; the commercial data set contained 224 image pairs and the mixed neighborhood data set contained 748 image pairs. For each neighborhood type, 80% of the image pairs were used for training and 20% of the image pairs were used for testing the models. For each iteration of training, the discriminator was first tested on a batch of real image pairs. Next, fake images were simulated by the generator. Afterwards the discriminator was tested and updated based on its evaluation of the fake images and the error calculated between the fake and the real images. Finally, the error found between the real and fake images was used to improve the generator. This procedure continually updated both the generator and the discriminator so that the learning of each network proceeded in parallel with the other.

2.5 Model Evaluation

Several types of model evaluation were used to evaluate the GAN results. First, visual examination of the output samples was completed for each iteration [1] along with the loss calculation [7]. Further evaluation of the model quality was conducted using Peak Signal to Noise Ratio (PSNR) [13, 17], which measures the quality of reconstruction in image compression [8]; and Structural Similarity Index Measure (SSIM) [14, 19], which is used to measure the similarity between two images. Table 1 in Subsection 3.4 shows the results of these metrics.

3 RESULTS AND DISCUSSION

Here we present the results of the image-to-image translation models on each of residential, commercial and mixed neighborhoods. One characteristic learned during model training for all of the neighborhoods was the black and white pixelated appearance of the target inputs. During training, model feedback rewarded grayscale pixel production. Models tended to simulate the appearance of the varying grayscale pixels convincingly after training for 500 epochs. Examples of these results are shown in Figures 4 through 9. This selection of outputs was captured from the validation stage of training on the final iteration of each model. Each figure displays a row with three images: the land cover representation, the building footprints (with grayscale encoded height) on the land and their placement, and the GAN-generated building footprints, grayscale height representation and placement for each neighborhood. Variation in size between the data sets of each neighborhood type resulted in differences in the total number of iterations required for each training set. However, batch size, number of epochs and number of graphical processing units (GPUs) across model training was kept constant to maintain the uniformity of the experiment (See Subsection 2.3).

3.1 Residential Neighborhood Generation

Figures 4 and 5 show simulated image results from the residential model. The characteristics of the outputs correspond to the patterns learned by the generator from the input and target images and assessment by the discriminator. One such pattern is the correspondence of the density of the land cover to the size and spacing of the buildings. As shown in Figure 4, in the high-density (red-colored land use) areas, buildings are small and tightly-packed. In lower-density areas, such as the pink areas shown in the land cover image in Figure 5, buildings are small and spread out. Results shown in Figure 5 also indicate that the generator learns that no buildings are located in areas of scrub (yellow) or woodlands (green).

Figure 4: Land cover image, target image and generated image for a high-density urban residential area in Los Angeles.

Figure 5: Land cover image, target image and generated image for an urban residential area with woodland and scrub in Los Angeles.

3.2 Commercial Neighborhood Generation

Figures 6 and 7 show that commercial buildings tend to be much larger, and sometimes closer together, than residential buildings. The high-density urban area shown in Figure 6 highlights two particularly large square buildings and many long rectangular structures (many clustered together) that may be strip malls or shopping centers with roads between. The low-density commercial tile shown in Figure 7 contains several patches with scrub (yellow) and woodland (green) and very few buildings. This tile is classified as commercial because of the building usage types in the urban area. Within that commercial development, buildings are smaller but still placed in patterns similar to those in the higher-density commercial areas. Here again, the generator learns to place buildings only in the urban area and not within the highly vegetated terrain.

Figure 6: Land cover image, target image and generated image for a high-density urban commercial area in Los Angeles.

3.3 Mixed Neighborhood Generation

The tiles shown in Figures 8 and 9 contain characteristics of both residential and commercial neighborhoods. Both figures display a relatively even mix of larger, densely packed buildings and smaller, spread-out structures. In each figure, the presence of large roads emerges as buildings are placed away from these long linear spaces. Additionally, buildings are not placed in the scrubland (yellow) patches in the tiles. The generated image in Figure 9 shows an attempt by the generator to place short small buildings in the space defined by the lake (blue) in the land cover image. Because very few lakes were included in the training images, it is possible that the generator had not yet learned how to interpret the lake designation in the land cover image. Otherwise, in both Figures 8 and 9, building size and placement is learned and produced well in the generated images.

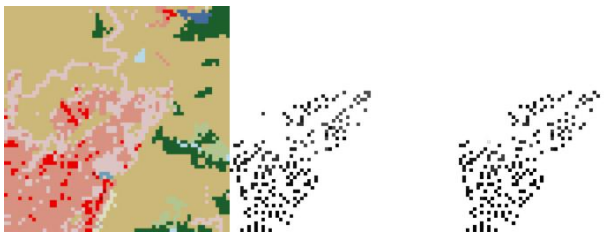

Figure 7: Land cover image, target image and generated image for a low-density urban commercial area with woodland and scrub in Los Angeles.

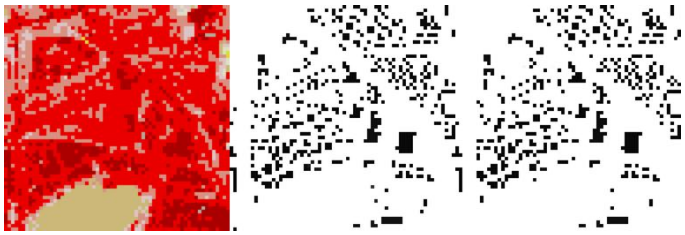

Figure 8: Land cover image, target image and generated image for a high-density urban mixed residential and commercial area in Los Angeles.

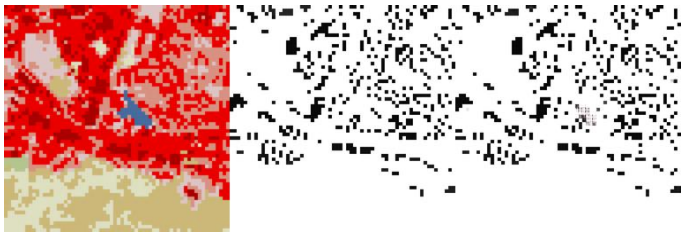

Figure 9: Land cover image, target image and generated image for an urban mixed residential and commercial area with scrub and lake in Los Angeles.

3.4 Model Performance

Three evaluation metrics were calculated for the generated images after four different numbers of epochs of model training: Loss

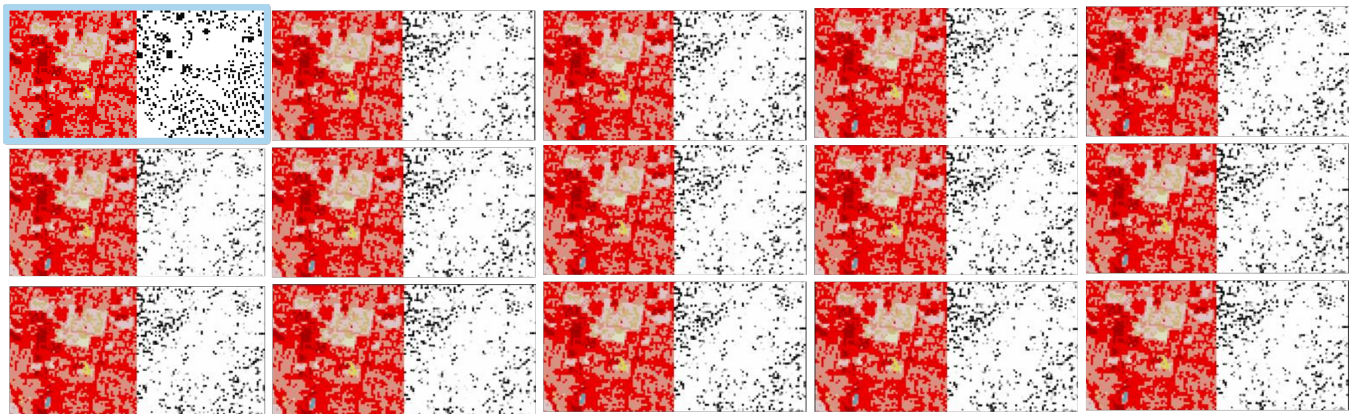

Figure 10: Land cover image, target image and ensemble of generated images for an urbanized residential area in Los Angeles.

(Subsection 2.1), PSNR (Subsection 2.5) and SSIM (Subsection 2.5). As shown in Table 1, the weighted sum of the loss values calculated from the objective function (Equation (3)) start high at 25 after one training epoch, then stabilize at 5 after 250 epochs–an 80% reduction in the loss value.

Results of an experiment by [17] to embed a watermark in the least significant bits of randomly selected pixels of a selected tif image showed that when these images were evaluated using PSNR, typical values were between 13 and 47 dB. Table 1 shows that to achieve PSNR values in that range for the tifs generated here, the minimum number of epochs required is around 50.

Assessment of image quality based on the degradation of structural information between the target image and the generated image was conducted using SSIM. Values for SSIM range between 0 and 1, with 1 reachable only in the case that the two images are identical [19]. Results here show that after 500 epochs, the target and generated images are highly structurally similar. However, as will be discussed in Subsection 3.5, our goal here is not to generate images identical to the target image, but to generate images with the basic properties of the target images so that a variety of potential neighborhoods may be tested for new developments under consideration by urban planners.

Table 1: Improvement in generator loss values, PSNR and SSIM over increasing numbers of epochs during training

NumEpochs	*Loss*	*PSNR(db)*	*SSIM*
1	25	1.27	0.009
50	15	16.7	0.468
250	5	22.9	0.946
500	5	25.0	0.980

3.5 Development of an Ensemble of Testable New Neighborhoods for a Planned Development

Once predicted new neighborhood developments are generated, these products can help with analyses of planned neighborhoods with respect to issues around architecture, society, culture, environment and technology (as mentioned in Section 1). For example, conversion of these products to polygon shapefiles can aid urban planners using geographical information science (GIS) tools to communicate plans to an existing or prospective community. Building energy modelers requiring characteristics of buildings in neighborhoods for the future to adapt buildings for more efficient energy use could also benefit from access to such files. These files can also provide input to models generating urban parameters for predicted neighborhoods that can be read by numerical weather models or high resolution climate models and integrated into the land and atmospheric process calculations of those models.

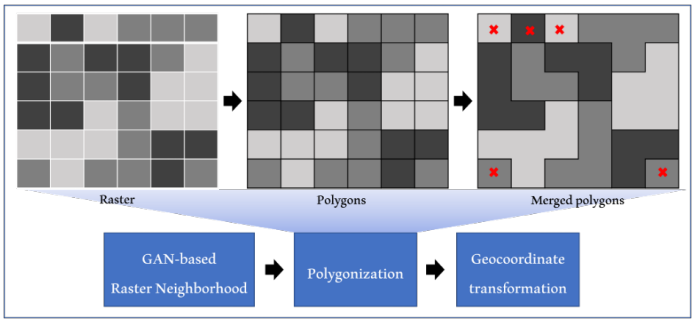

Figure 11: Post-processing pipeline to convert GAN-based raster urban neighborhood to vector form. Polygons marked with a red cross are spurious polygons and will be eliminated.

Because the future is hard to predict and different types of neighborhoods might satisfy different needs across new communities, it is important for a model used for these purposes to be able to generate multiple possibilities, that is, an ensemble of possibilities, for a new neighborhood development. We show in Figure 10, how the generation of an ensemble of neighborhood possibilities for a single residential tile for a Los Angeles neighborhood can be accomplished by capturing output every ten iterations over the last 150 iterations out of 6,000 model simulations. These outputs

have been successfully tuned to represent realistic building placements through many iterations of training and can be used for the development of future cities where environmental, geographical, and infrastructural impacts can be tested. While Figure 3.5 shows little difference among the outputs captured on different iterations, select patches of each generated tile show some variability. While this variability is small on the scale of 1.92 km^2, these differences in building size and density are significant at urban block scale.

4 CONCLUSIONS AND NEXT STEPS

We have shown that given urban land cover characteristics and associated target building heights, footprints and placements, an image-to-image GAN can produce an ensemble of potential neighborhoods compatible with the land cover characteristics.

Next steps for this research include the application of the method to future land cover projections and post-processing of the GAN-generated raster tif images of the urban neighborhoods to obtain vector layers of building footprints that collectively represent the synthetic neighborhoods. The output of the GAN is a raster image which is a collection of regions of pixels where each region shares a common pixel value. The pixel value shared by each region represents building height. A geospatial process called polygonization will be used to generate vector polygons for all connected regions of pixels sharing a common pixel value of building height (Figure 11). Each polygon in the output vector layer will be associated with an attribute indicating the height of the building. This polygonization process may generate a few spurious polygons which must be eliminated based on their area (polygons marked by red cross in Figure 11). Further, this vector layer will undergo a geospatial coordinate transformation to reproject polygons to the appropriate geospatial coordinate system. For the process of polygonization and geospatial coordinate transformation, we will use 'gdal_polygonize' and 'gdaltransform' utilities respectively, provided by a Geospatial Data Abstraction software Library [6]. The final building footprint layer can then be used for various geospatial analysis tasks such as urban planning, change analysis and street pattern design for optimal mobility.

However, the method, while useful, does pose a few limitations. For example, the pixel-level grayscale of generated images may ultimately not yield realistic building footprints, so some adjustment of the results may be necessary to produce viable morphological examples. Additionally, decisions about how to build neighborhoods may require different priorities now than those that were considered for the construction of existing neighborhoods, such as accommodation of larger population per square kilometer, the incorporation of electric vehicle charging stations or access to other modern infrastructure, changing cultural opportunities or other concerns; so an ensemble of potential future neighborhood morphologies based on current and past physical properties may not cover all of the considerations required for urban planning. The development of new neighborhoods must certainly be an iterative process, but this method for potential neighborhood generation provides a useful contribution to that process.

ACKNOWLEDGMENTS

This work is supported by the DOE Office of Science as a part of the research in Multi-Sector Dynamics within the Earth and Environmental System Modeling Program under the Integrated Multiscale Multisector Modeling (IM3) Scientific Focus Area.

REFERENCES

[1] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41–65.
[2] Vineet Chaturvedi and Walter T de Vries. 2021. Machine Learning Algorithms for Urban Land Use Planning: A Review. *Urban Science* 5, 3 (2021), 68.
[3] J Dewitz. 2021. *National Land Cover Database (NLCD) 2019 Products.* data release (ver. 2.0, June 2021). U.S. Geological Survey. https://doi.org/10.5066/P9KZCM54.
[4] Stanislava Fedorova. 2021. GANs for Urban Design. *CoRR* abs/2105.01727 (2021). arXiv:2105.01727 https://arxiv.org/abs/2105.01727
[5] Manuel Gausa. 2021. *Resiligence: Intelligent Cities/Resilient Landscapes.* Actar D, Inc.
[6] GDAL/OGR contributors. 2022. *GDAL/OGR Geospatial Data Abstraction software Library.* Open Source Geospatial Foundation. https://doi.org/10.5281/zenodo.5884351
[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1125–1134.
[8] Kamaldeep Joshi, Rajkumar Yadav, and Sachin Allwadhi. 2016. PSNR and MSE based investigation of LSB. In *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT).* IEEE, 280–285.
[9] Karl Kropf. 1996. Urban tissue and the character of towns. *Urban Design International* 1, 3 (1996), 247–263.
[10] Chuan Li and Michael Wand. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision.* Springer, 702–716.
[11] Joshua New, Mark Adams, Anne Berres, Brett Bass, and Nicholas Clinton. 2021. *Model America–data and models of every US building.* Technical Report. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States). 10.13139/ORNLNCCS/1774134.
[12] Vítor Oliveira. 2016. *Urban morphology: an introduction to the study of the physical form of cities.* Springer.
[13] PytorchIgnite. 2022. PSNR. Acessed from: https://pytorch.org/ignite/generated/ignite.metrics.PSNR.html.
[14] PytorchIgnite. 2022. SSIM. Acessed from: https://pytorch.org/ignite/generated/ignite.metrics.SSIM.html.
[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention.* Springer, 234–241.
[16] Aditya Sharma. 2021. Pix2Pix:image-to-image translation in PyTorch & Tensorflow. Acessed from: https://learnopencv.com/paired-image-to-image-translation-pix2pix/.
[17] Devendra Somwanshi, Indu Chhipa, Trapti Singhal, and Ashwani Yadav. 2018. Modified Least significant bit algorithm of digital watermarking for information security. In *Soft Computing: Theories and Applications.* Springer, 473–484.
[18] Dongjie Wang, Yanjie Fu, Pengyang Wang, Bo Huang, and Chang-Tien Lu. 2020. Reimagining city configuration: Automated urban planning via adversarial learning. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems.* 497–506.
[19] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
[20] Abraham Noah Wu, Rudi Stouffs, and Filip Biljecki. 2022. Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales. *Building and Environment* (2022), 109477.
[21] Chunxue Xu and Bo Zhao. 2018. Satellite Image Spoofing: Creating Remote Sensing Dataset with Generative Adversarial Networks (Short Paper). In *10th International Conference on Geographic Information Science (GIScience 2018) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 114)*, Stephan Winter, Amy Griffin, and Monika Sester (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 67:1–67:6. https://doi.org/10.4230/LIPIcs.GISCIENCE.2018.67