ARCHITECT: EXTRACTING BUILDING RELATED INFORMATIONS AND CHANGING ARCHITECTURAL STYLE IN AR

Giorgiana Violeta Vlăsceanu[1], Alin Drăguț[2], Gabriel Sandu[3], Nicolae Tarbă[4], Mihai-Lucian Voncilă[5], Costin-Anton Boiangiu[6], and Nicolae Goga[7]

With recent advancements in mobile computing, augmented reality (AR) applications can now achieve real-time performance. AR provides a novel way to engage with our surroundings. Platforms such as ARKit or ARCore make AR app development easy. ARchitect aims to allow users to experience different architectural styles and cultural heritages by altering the appearance of buildings in their environment. The presented solution utilizes deep learning and computer vision techniques to classify architectural styles, which is a challenging task due to the inter-class relationships between styles. The approach presented in the paper alters the style of a building detected from the phone's camera and retrieves information such as its current style, age, and distance from the user. To train the deep learning models, a publicly available dataset of photos of buildings with 25 different styles is used. The obtained results for architectural style classification surpass state-of-the-art methods, demonstrating the effectiveness of deep learning techniques. Additionally, the presented approach for architectural style transformation is also promising and will improve as new research is conducted in image generation, monocular depth estimation, and surface reconstruction methods.

Keywords: AR, architectural style change, architectural style classification, image generation, monocular depth estimation, surface reconstruction, object saliency detection

[1]Teaching Assistant, PhD Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: giorgiana.vlasceanu@cs.pub.ro

[2]Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: alin.dragut@stud.acs.upb.ro

[3]Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail:gabriel.sandu1709@stud.acs.upb.ro

[4]PhD Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: nicolae.tarba@upb.ro

[5]PhD Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: mihai_lucian.voncila@stud.acs.upb.ro

[6] Professor, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: costin.boiangiu@cs.pub.ro

[7] Professor, Faculty of Engineering in Foreign Languages, University "Politehnica" of Bucharest, Romania, e-mail: nicu.goga@upb.ro

1. Introduction

An architectural style comprises a distinct set of features and characteristics that enable identification of buildings or structures with a particular historical period. Over time, architectural styles evolve, reflecting the cultural development of a region.

The challenge at hand, however, lies in the classification of these architectural styles. This task becomes increasingly complex due to the gradual process of changes in style and the cultural variation within a given style. Added to this, the architectural vision of the creator often bridges styles, blurring boundaries and adding another layer of complexity to their relationships, leading to classification issues [1].

In the current era of digitization, generating realistic building facades in a different architectural style has emerged as a field of significant interest. However, this process is laden with challenges, particularly regarding the interpretability and controllability of the resulting 3D model. Current research into generating novel 3D models that retain the architectural features of the input model, similarly to style transfer for images, is rather limited.

Addressing this gap, the central focus of this paper is to present a novel method that leverages deep learning techniques to transform architectural styles. The proposed approach involves capturing a picture of a facade, generating a facade image in a different architectural style, and then estimating depth information per pixel to successfully reconstruct a 3D mesh of a facade in another architectural style. The significance of this work lies in its potential applications across various domains, such as real estate or automated architectural design, particularly in augmented reality where changing the architectural style of buildings can have impactful use-cases. This marks an exciting direction for future architectural visualization and could revolutionize the way we perceive and interact with buildings in a digital landscape.

2. Related Work

Deep learning has revolutionized the field of object detection [2], finding applications across diverse domains such as healthcare, autonomous driving, and anomaly detection. With the advent of high-performance GPUs and extensive research into deep learning, the performance of object detectors has been dramatically enhanced.

Traditionally, image processing algorithms have been utilized for object detection. While these methods do not require training and thus are beneficial in certain scenarios, they are highly susceptible to changes in conditions such as illumination and occlusion. Conversely, deep learning models, despite their need for large amounts of data, usually achieve higher accuracy rates and have become the preferred method for many researchers. This shift in preference is largely due to the availability of large, publicly accessible datasets like ImageNet [3] and MS COCO [4].

Deep learning approaches for object detection can be broadly classified into two categories: two-stage detectors and one-stage detectors. The former first leverages deep learning networks such as AlexNet, ResNet, or others to extract deep features for approximating object regions, which are then used for classifying the detected object and computing the bounding box. One-stage detectors, on the other hand, eliminate the region proposal step, which increases speed at the expense of accuracy. Despite this trade-off, recent advancements in deep learning-based object detection methods have led to significant improvements in both detection accuracy and speed, solidifying their popularity and importance across various applications.

Image classification serves as the fundamental problem in computer vision, with other problems often built upon it. Convolutional Neural Networks (CNNs) [5] are currently the state of the art for image classification, with several high-performing CNNs such as AlexNet [6], GoogLeNet [7], and ResNet [8] emerging from intensive research in this field.

In the specific domain of building architectural style classification, both CNN and non-CNN-based approaches have been proposed. Zhang et al. [9] presented an approach based on Deformable Part-based Models (DPM) and Multinomial Latent Logistic Regression (MLLR), which abstracts architectural components and covers probabilistic analysis and multi-class issues in latent variable models. However, CNN-based approaches like the ones presented by Wang et al. [10] and Zhang [11] have shown promising results. These methods were designed to recognize regional differences within the same architectural style and classify buildings across various architectural styles, respectively. However, the method by Miao et al. [12], which uses DPM in preprocessing and an Improved Ensemble Projection for better image grouping with SVM for classification, is currently considered state of the art.

Generative Adversarial Networks (GANs) have been extensively investigated and implemented in computer vision applications such as feasible imagine synthesis, image-to-image translation, and face feature alterations. GANs consist of a generator, which creates images, and a discriminator that discerns between real and generated images. Numerous variations of GAN architectures have been proposed to achieve different purposes [13]. For image generation, GANs can be based on either supervised or unsupervised learning.

Supervised GANs, such as pix2pix [14] and cycleGAN [15], generally require smaller datasets but necessitate conditional input both in training and inference, which can limit practical use cases. Unsupervised GANs, like bigGAN [16] and styleGAN2 [17], require larger datasets and more training time but typically perform better in practice.

In the domain of building facade generation, various approaches based on both supervised and unsupervised GANs have been proposed. Zhang et al. [18] suggested decomposing a 3D model into multiple 2D levels and using pix2pix and cycleGAN to generate new image sequences that are then recomposed into a 3D model. GAN Loci [19], which uses pix2pix but requires depth map input, also proposed a similar method. Although GAN Loci has a variant based on styleGAN

[20], its application is limited to 512x512 images due to hardware constraints. Other methods based on unsupervised GANs include City-GAN [21], which incorporates additional label information to guide the architectural style of the buildings in the generated images, and Shengyu Meng's approach [22], which is based on style-GAN2 [17] and introduces techniques for visualizing the high-dimensional latent space of the generated images.

3. Materials and Methods

The text on architectural style transfer examines the present-day state of the field, which either uses 2D images of building facades as both input and output or 3D meshes of a building as input and output. The proposed solution, however, will use a 2D image of a building facade as input and output a 3D mesh of the facade in a different architectural style that will be applied in AR over the original building to replace the old facade. The following section presents an overview of the approach to performing architectural style classification and transformation.

3.1. Architectural Classification

The style categorization method used in this paper is based on a transfer learning method that employs EfficientNet [23]. Transfer learning is a prominent machine learning method in which a model generated for one job is utilized as the foundation for a model for another task. Due to the large computation and time resources necessary to create deep neural network models for these challenges, pre-trained models are widely utilized as a starting point for computer vision and natural language processing tasks in deep learning.

EfficientNet-B0, which has 4M parameters and demonstrates strong capabilities for transfer learning, is used as the base model in this work. The dataset used for training [9] contains around 5000 images in 25 different architectural styles, and has been used in related works regarding architectural style classification to enable fair comparison against other methods.

To increase the likelihood of a correct classification, a data augmentation step using BING [24] is added during training to propose regions of interest. The output of this step is cropped images containing these regions of interest that are used to expand the training dataset. It has been observed that this approach increases the accuracy by a few percent.

Several layers have been defined for the initial network. In the first layer, the input is resized to 224x224 for RGB channels. After the resize, a data augmentations layer is defined, which applies random horizontal flip based on a preset random probability. Data augmentation is a useful strategy for increasing the performance and results of machine learning models by adding new data for training. The flip was chosen because it makes sense for buildings. To reduce model overfitting, the EfficientNet output is sent via a batch normalization layer, a global average

pooling layer, and a dropout layer. As the final activation function for class distribution, the softmax function is used to normalize the network's result to a probability distribution among expected output classes.

To prevent the model from overfitting and to evaluate it effectively, the input data is separated into train and validation subsets. The dataset is divided into 80% train and 20% validation to avoid overlearning from the training data.

3.2. Architectural Style Transformation

To modify the architectural styles of the images, a GAN based on styleGAN is employed to have control over the style of the building output image. The input image is projected into W latent space, which is the origin of generated images. Methods like linear interpolation can be applied on 2 latent space vectors to blend the style of 2 different images, which can be used to transform the architectural style of the input image.

FIGURE 1. Images projected in W (center) and W+ latent space (right)

Common techniques for projecting the input image into latent space (also known as image inversion) are latent optimization [25] or an encoder [26]. To achieve low reconstruction error (high image similarities between the source and output images), the latent space `W` can be extended by creating a separate `W` for each layer of the generator, which is typically referred to as `W+`. Both a latent optimization technique with W and an encoder approach with `W+` were tested. Figure 1 shows the results, and the latter approach was chosen because of its greater reconstruction quality.

The dataset used for training [27] contains 10113 images of buildings in 25 different architectural styles. The dataset is balanced, containing about 400 images for each category. Following the transformation of the input image, the subsequent step involves 3D reconstruction of the generated building. The process of 3D reconstruction from a single 2D image involves retrieving the internal and external

parameters of the camera, and computing the distance between the pixels from the image to the camera, commonly known as the depth map. Once the depth map is obtained, a point cloud can be reconstructed with the 3D positions of the image pixels. The depth map can be obtained using either classical methods or deep learning methods. The classical methods usually consist of multiple steps:

- Line segment detection
- Vanishing points estimation
- Image rectification

A classical method [28] was initially considered for depth map retrieval. However, in order to detect the necessary vanishing points for 3D reconstruction, the input image needs to display two building facades, which is not feasible with our current pipeline. Instead, we opted for a deep learning method that can directly retrieve the depth map from the RGB image. These methods learn the correspondence between pixels and depth by training on large and diverse datasets. The LeReS method [29] was chosen for our project, which performs monocular depth estimation from a single image. An example of the resulting depth map for an outdoor image containing a building can be found in Figure 2.

FIGURE 2. Original image and depth map

Once LeReS is applied to the generated image, a point cloud representation is generated for scene reconstruction, using the Open3D library[30]. This library is specifically designed for software development involving 3D data and provides optimized data structures for point clouds, meshes, and RGB-D images. Open3D supports basic processing algorithms such as I/O, sampling, visualization, and conversion. The resulting point cloud for the image and depth map from Figure 2 is illustrated in Figure 3. To create a mesh from the point cloud and RGB information, the Poisson surface reconstruction technique [31] is employed, along with a crop method and vertex removal based on density method to clean up the resulting mesh.

3.3. Application Architecture

The proposed approach adopts a client-server architecture, where the server comprises two pipelines, namely, an information extractor pipeline and a facade transformation pipeline. The client application captures an image of a building

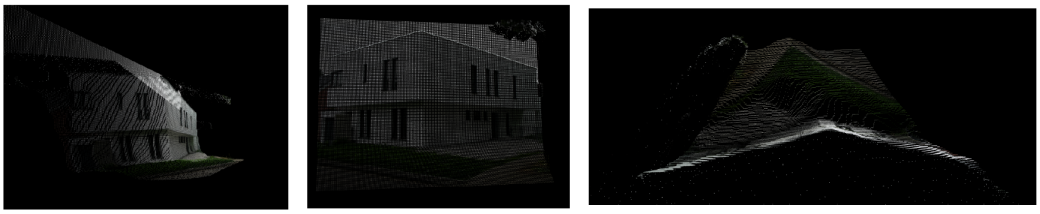

FIGURE 3. Point clouds

facade and presents information about the facade along with its transformation in the selected architectural style to the user.

In the first pipeline, the server receives an image of a building facade from the client. This image is passed through a neural network classifier to determine the facade architectural style. This information is sent back to the client and displayed to the user.

In the second pipeline, the server takes an image of a facade and generates a new facade image in the gothic architectural style using GAN. This new image is used to extract the depth map, which is subsequently utilized to generate a point cloud. The resulting point cloud is converted to a mesh using Poisson surface reconstruction after applying a crop method and vertex removal based on density method for mesh cleaning. Finally, the resulting mesh is sent back to the client for user viewing.

The client application is developed using *Unity*[1], *AR Foundation*[2], and *Vuforia*[3]. It consists of three screens. The first screen visualizes the area using the phone camera and lets the user photograph the selected building facade. The facade appears on the second screen, together with architectural style information acquired from the server. Additionally, AR Foundation is used to display the distance from the user to the facade. By pressing the ”transform” button, the client sends the image of the facade along with the desired architectural style to the server. The settings menu allows the user to specify offsets for a bounding box that can be used to crop the resulting mesh and remove artifacts, sky, etc. Another setting, minimum density, is used to remove points from the point cloud that do not have enough neighboring points in proximity.

4. **Results**

In the following measurements, the validation accuracy for the classifier and the Fréchet inception distance (FID) score for the generated images were utilized. The FID score is a metric used to evaluate the quality of images produced by generative models. The architectural style classifier exhibits an approximate accuracy of 81%. The training process of the presented classifier involves two steps. In the first

[1]https://unity.com

[2]https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html

[3]https://developer.vuforia.com

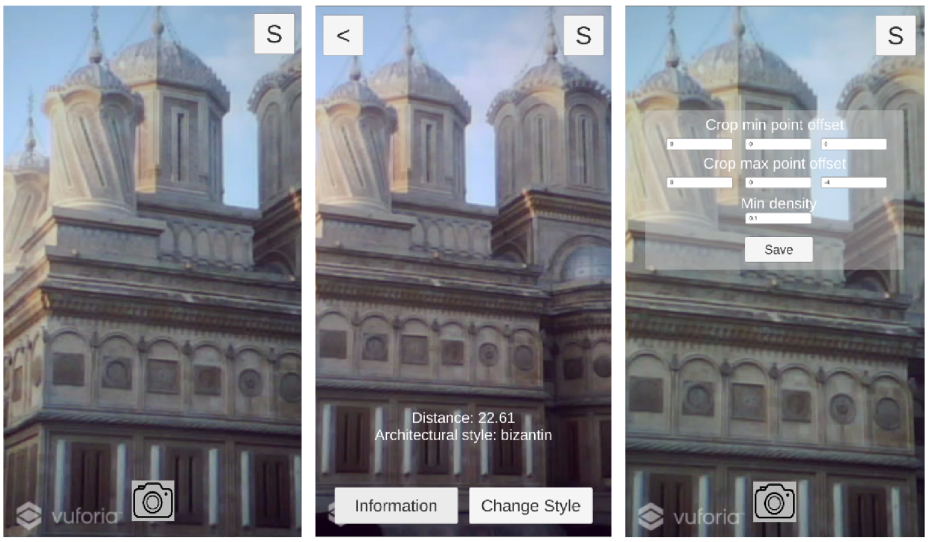

FIGURE 4. Screenshots from the client application

TABLE 1. Validation accuracy comparison

DPM+LSVM	DPM+MLLR	MLLR+SP	DPM+IEP+LR	DPM+IEP+SVM	Proposed solution
37.69%	42.55%	46.21%	53.52%	55.35%	81.00%

step, a classifier was trained using the base model (EfficientNet-B0) with frozen weights initialized to ones based on a complete training of the model on the ImageNet dataset. In the second step, fine-tuning was performed by unfreezing the last 20 layers of the base model (except batch normalization layers) with a lower initial learning rate. The first step runs for 25 epochs, and the second one runs for 20 epochs. The graphics illustrating the model accuracy and loss evolution are presented in Figure 5. Additionally, a comparison of accuracy to the state of the art is offered in Table 1.

After 950 thousand images, the GAN obtains a FID score of 23.70. The FID score is unlikely to rise above this level, because the dataset's dimension. Four examples of building facades generated with the proposed GAN, with a resolution of 256x256, are shown in Figure 6.

5. Conclusions

The projected images using the proposed trained GAN on the gothic dataset (approximately 300 images) didn't have enough quality to use them in the facade generation pipeline. Results can be seen in Figure 7.

To improve the quality of our projected images, we used the pretrained StyleGan2 model style-gan2-church-config-f in our application. A larger dataset would be helpful in improving the projected image quality to match those from the StyleGan2 pretrained model. One potential option for expanding the dataset would be to

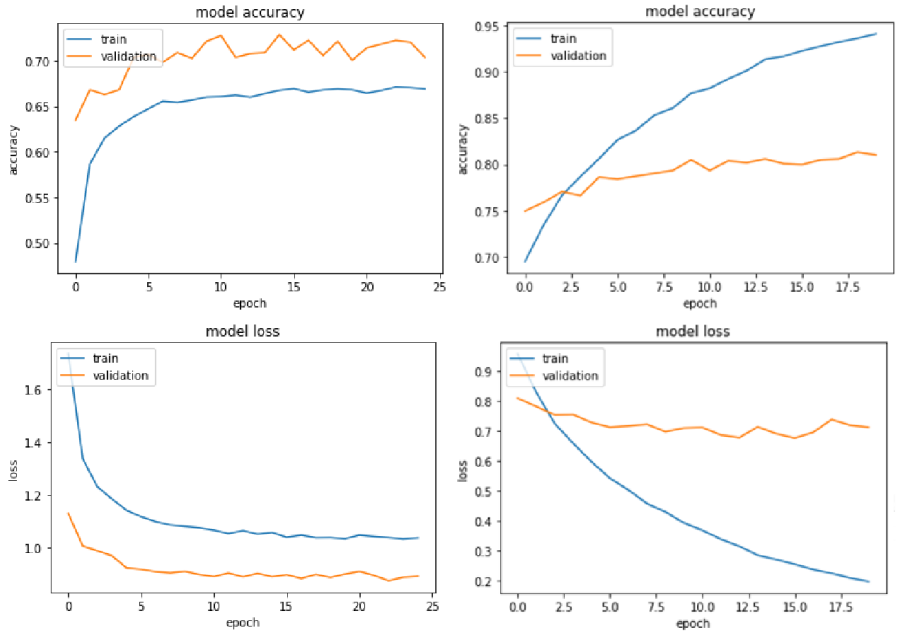

FIGURE 5. Train versus validation accuracy and loss in the base training(left) step and in the fine-tuning step(right)

FIGURE 6. Facades generated by the proposed GAN solution

use images from Google Street View along with a facade detection algorithm. This approach would work well for the GAN's training dataset as it typically doesn't require labeled data.

A related issue encountered in this project was with the placement of the reconstructed mesh. The solution relied on the AR Foundation framework's plane detection and placement, but the results were not satisfactory. However, it is a possibility that better results can be obtained with improved mesh placement methods. It is possible that the AR Core framework with the newest Apple devices can already accomplish this task better. The achieved results for the proposed solution is referring to Figure 8.

FIGURE 7. Outputs from the proposed GAN solution

FIGURE 8. Mesh placement

A potential area for further investigation involves hyperparameter tuning, which was not fully explored due to hardware limitations. Additionally, exploring other models for depth estimation could improve the quality of reconstructed meshes, particularly for facades that are distant from the image projection. Investigating 3D GANs may also be a promising research direction, as they could streamline the pipeline by eliminating multiple steps, such as monocular depth estimation and surface reconstruction.

Aknowledgement

The results presented in this article has been funded by the Ministry of Investments and European Projects through the Human Capital Sectoral Operational Program 2014-2020, Contract no. 62461/03.06.2022, SMIS code 153735.

REFERENCES

[1] Peipei Zhao, Qiguang Miao, Jianfeng Song, Yutao Qi, Ruyi Liu, and Daohui Ge. Architectural style classification based on feature extraction module. *IEEE Access*, 6:52598–52606, 2018.

[2] Sankar K. Pal, Anima Pramanik, J. Maiti, and Pabitra Mitra. Deep learning in multi-object detection and tracking: State of the art. *Applied Intelligence*, 51(9):6400–6429, sep 2021.

[3] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[5] Farhana Sultana, Abu Sufian, and Paramartha Dutta. Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 122–129, 2018.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[8] Asifullah Khan and Noorul Wahab. Deep residual learning, 08 2016. arXiv:1512.03385 [cs] 2015.

[9] Zhe Xu, Dacheng Tao, Ya Zhang, Junjie Wu, and Ah Chung Tsoi. Architectural style classification using multinomial latent logistic regression. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 600–615, Cham, 2014. Springer International Publishing.

[10] Rui Wang, Donghao Gu, Zhaojing Wen, Kai Yang, Shaohui Liu, and Feng Jiang. Intra-class classification of architectural styles using visualization of cnn. In Xingming Sun, Zhaoqing Pan, and Elisa Bertino, editors, *Artificial Intelligence and Security*, pages 205–216, Cham, 2019. Springer International Publishing.

[11] Yun Kyu Yi, Yahan Zhang, and Junyoung Myung. House style recognition using deep convolutional neural network. *Automation in Construction*, 118:103307, 2020.

[12] Qiguang Miao, Ruyi Liu, Peipei Zhao, Yunan Li, and Erqiang Sun. A semi-supervised image classification model based on improved ensemble projection algorithm. *IEEE Access*, 6:1372–1379, 2018.

[13] Zhengwei Wang, Qi She, and Tomás E. Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Comput. Surv.*, 54(2), feb 2021.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[15] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[16] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020.

[18] Hang Zhang and Ezio Blasetti. 3d architectural form style transfer through machine learning (full version), 08 2020.

[19] Kyle Steinfeld. *Imaging Place Using Generative Adversarial Networks (GAN Loci)*, pages 513–516. John Wiley and Sons, Ltd, 2022.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.

[21] Maximilian Bachl and Daniel C. Ferreira. City-gan: Learning architectural styles using a custom conditional gan architecture, 2019.

[22] Shengyu Meng. Exploring in the latent space of design: A method of plausible building facades images generation, properties control and model explanation base on stylegan2. In Philip F. Yuan, Hua Chai, Chao Yan, and Neil Leach, editors, *Proceedings of the 2021 DigitalFUTURES*, pages 55–68, Singapore, 2022. Springer Singapore.

[23] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning, 2019*, 2019.

[24] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, 2014.

[25] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4431–4440, 2019.

[26] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), jul 2021.

[27] Zhe Xu. Architectural styles. https://www.kaggle.com/datasets/dumitrux/architectural-styles-dataset, Accessed: 2022-05-14.

[28] Georgios Vouzounaras, Juan Diego Perez-Moneo Agapito, Petros Daras, and Michael G. Strintzis. 3d reconstruction of indoor and outdoor building scenes from a single image. In *Proceedings of the 2010 ACM Workshop on Surreal Media and Virtual Cloning*, SMVC '10, page 63–66, New York, NY, USA, 2010. Association for Computing Machinery.

[29] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image, 2020.

[30] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing, 2018.

[31] Hoppe Hugues Kazhdan Michael, Bolitho Matthew. Poisson Surface Reconstruction. In Alla Sheffer and Konrad Polthier, editors, *Symposium on Geometry Processing*. The Eurographics Association, 2006.