UNVEILING SPACES
Architecturally meaningful semantic descriptions from images of interior spaces

Demircan Tas
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
tasd@mit.edu

Rohit Priyadarsh Sanatani
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
sanatani@mit.edu

Abstract

There has been a growing adoption of computer vision tools and technologies in architectural design workflows over the past decade. Notable use cases include point cloud generation, visual content analysis, and spatial awareness for robotic fabrication. Multiple image classification, object detection, and semantic pixel segmentation models have become popular for the extraction of high-level symbolic descriptions and semantic content from two-dimensional images and videos. However, a major challenge in this regard has been the extraction of high-level architectural structures (walls, floors, ceilings windows etc.) from diverse imagery where parts of these elements are occluded by furniture, people, or other non-architectural elements. This project aims to tackle this problem by proposing models that are capable of extracting architecturally meaningful semantic descriptions from two-dimensional scenes of populated interior spaces. 1000 virtual classrooms are parametrically generated, randomized along key spatial parameters such as length, width, height, and door/window positions. The positions of cameras, and non-architectural visual obstructions (furniture/objects) are also randomized. A Generative Adversarial Network (GAN) for image-to-image translation (Pix2Pix) is trained on synthetically generated rendered images of these enclosures, along with corresponding image abstractions representing high-level architectural structure. The model is then tested on unseen synthetic imagery of new enclosures, and outputs are compared to ground truth using pixel-wise comparison for evaluation. A similar model evaluation is also carried out on photographs of existing indoor enclosures, to measure its performance in real-world settings.

1. Collaboration and Logistics

Presented work is the result of a collaboration between Demircan Tas (DT) and Rohit Priyadarsh Sanatani (RS). RS is mostly responsible for implementing the the Pix2Pix model and domain adaptation, while DT implemented synthetic data generation and data augmentation. However, authors' efforts overlap substantially in terms of developing a holistic model for defining architecturally significant features and eliminating others. System design and research, data collection, documentation, as well as troubleshooting were conducted with equal effort from both authors[1].

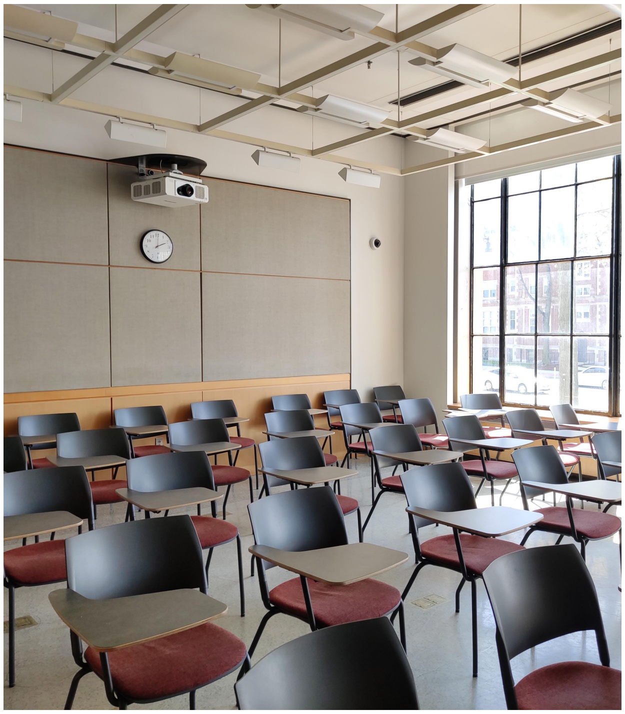


Figure 1. A classroom from MIT, with a variety of objects (clock, projector, wall panels, chairs, etc.) that are irrelevant for an architecture model

[1] Sections with individual efforts from specific authors are specified with footnotes. Unless explicitly stated, presented work, findings, and comments result from joint efforts.

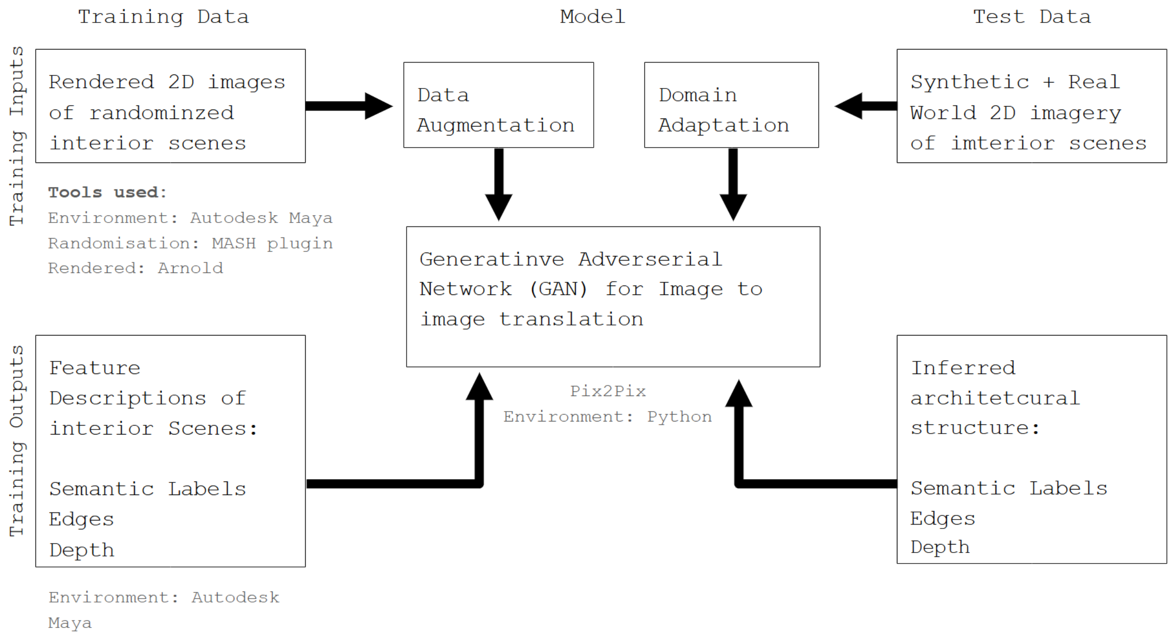

Training Data
Model
Test Data
Training Inputs
Rendered 2D images of randominzed interior scenes
Tools used:
Environment: Autodesk Maya
Randomisation: MASH plugin
Rendered: Arnold
Data Augmentation
Domain Adaptation
Synthetic + Real World 2D imagery of imterior scenes
Generatinve Adverserial Network (GAN) for Image to image translation
Pix2Pix
Environment: Python
Training Outputs
Feature Descriptions of interior Scenes:
Semantic Labels
Edges
Depth
Environment: Autodesk Maya
Inferred architetcural structure:
Semantic Labels
Edges
Depth

Figure 2. System diagram

2. Introduction and Background

The Architecture, Engineering and Construction (AEC) industry has been increasingly relying on recent advances in computer vision for a variety of workflows. While photogrammetric three-dimensional point cloud generation from multiple shots of interior or exterior scenes has been around for quite a few years within the domain of heritage/as-built documentation, generating point clouds from single scenes using monocular depth prediction is becoming increasingly popular [11], [8]. Object detection models play important roles in the analysis of behavioral patterns within the built environment [9]. There have been notable developments within the field of robotic construction, fabrication and assembly, aiding efforts to provide robotic agents with spatial awareness [1].

A central theme in computer vision workflows is the extraction of high-level semantic content and symbolic descriptions from low-level descriptions (pixels) in images and videos. Multiple image classification, object detection, and semantic segmentation models have been used in the recent past for this purpose. Datasets such as ImageNet [4] and Places [12] allow for training models to classify scenes into symbolically meaningful categories. Moreover, datasets such as ADE20K [13] and CityScapes [2] allow for the segmentation of each pixel of an indoor or outdoor image, into semantic categories.

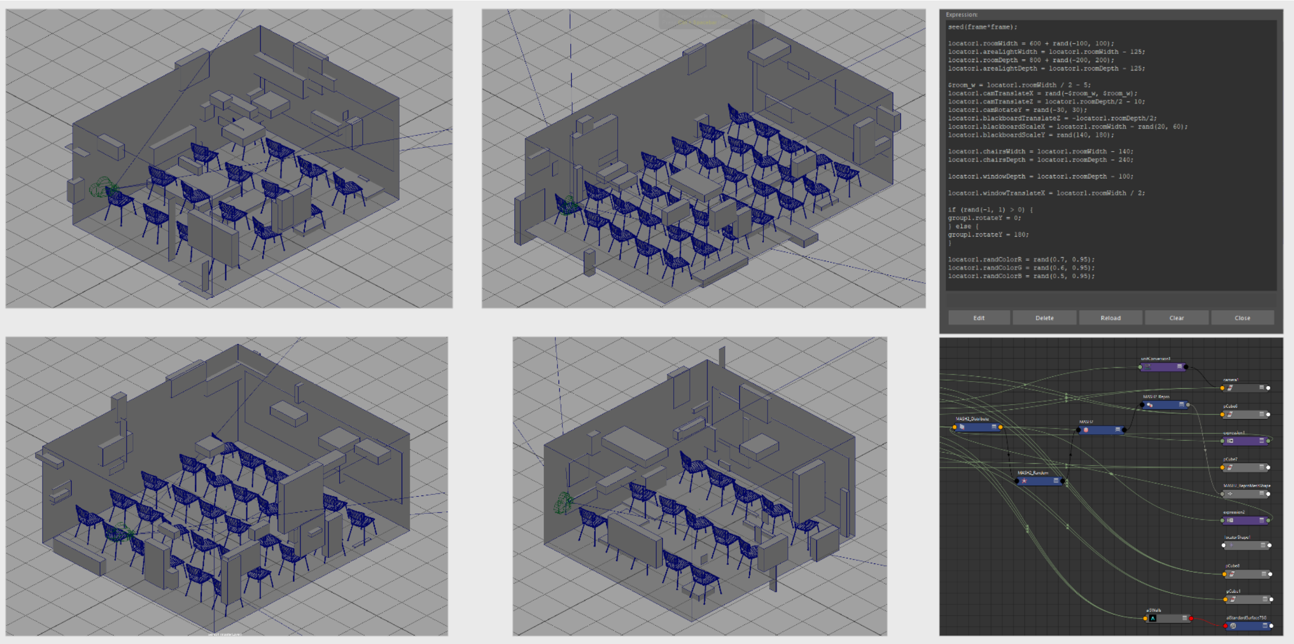
While these models are suitable for a variety of use cases, they are not adequately equipped to extract high level architectural structure from indoor images. The Building Information Modeling (BIM) paradigm has over the years developed a hierarchical framework of architectural descriptions. These descriptions consist of basic concepts such as walls, floors, ceilings and roofs, along with nested classes of objects such as doors, windows and skylights that are hosted within these units. While conventional object detection/segmentation models are able to classify each pixel as a ceiling or as a wall, they are not able to extract the high-level structure of these units, or the relationships among them. This is largely due to the fact the descriptions encoded within the training datasets of these models do not contain adequate information with regards to the hierarchical structuring of architectural elements. As a result, these models lack the ability to infer the structure of architectural elements which are occluded by visual obstructions that are natural in everyday interior scenes.

2.1. Related Work Generative Adversarial Networks

Recent advances in the development of robust Generative Adversarial Networks (GAN's) [3] show a lot of promise with regards to the task of inferring missing/occluded portions of images. Training of GAN's in-

Package/Tool/Library	Where is it available? (eg url)	What does it do?	Use Case
TensorFlow (2.8.0)	https://www.tensorflow.org/	Open source machine learning library for Python	Implementing Pix2Pix as a predictive model.
OpenCV (4.5.5.64)	https://opencv.org/	Provides computer vision tools and functions in Python.	As a general toolkit for image processing and computer vision.
PyTorch (1.11)	https://pytorch.org/	Open source machine learning framework for Python	Implementing style transfer for data augmentation, and domain adaptation.
Maya (2021) with plug-ins: MASH, Arnold (MtoA)	https://www.autodesk.com/products/maya/features	Production level 3d animation software	Three-dimensional virtual scene generation, parametric asset scattering, physically based rendering. We used the integrated Python implementation and Maya Embedded Language (MEL) to create a parametric classroom model.

Figure 3. Software and libraries used for implementation.



```
seed(frame*frame);

locator1.roomWidth = 600 + rand(-100, 100);
locator1.areaLightWidth = locator1.roomWidth - 125;
locator1.roomDepth = 800 + rand(-200, 200);
locator1.areaLightDepth = locator1.roomDepth - 125;

$room_w = locator1.roomWidth / 2 - 5;
locator1.camTranslateX = rand(-$room_w, $room_w);
locator1.camTranslateZ = locator1.roomDepth/2 - 10;
locator1.camRotateY = rand(-30, 30);
locator1.blackboardTranslateZ = -locator1.roomDepth/2;
locator1.blackboardScaleX = locator1.roomWidth - rand(20, 60);
locator1.blackboardScaleY = rand(140, 180);

locator1.chairsWidth = locator1.roomWidth - 140;
locator1.chairsDepth = locator1.roomDepth - 240;

locator1.windowDepth = locator1.roomDepth - 100;

locator1.windowTranslateX = locator1.roomWidth / 2;

if (rand(-1, 1) > 0) {
group1.rotateY = 0;
} else {
group1.rotateY = 180;
}

locator1.randColorR = rand(0.7, 0.95);
locator1.randColorG = rand(0.6, 0.95);
locator1.randColorB = rand(0.5, 0.95);
```

Figure 4. Parametric Maya scene, MEL expressions for randomization, and node editor connections

volve the training of two networks pitted against each other. One of the networks - the generator - is successively trained to generate images that match the distribution of training samples. The other network - the discriminator - is successively trained to discriminate between samples generated by the generator, and those drawn from the training data distribution. Through iterative training involving these two adversarial networks, GAN's allow for the learning of complex representations of real world topologies even in the absence of large amount of labeled training data.

GAN's have gained popularity in the recent past for a variety of generative tasks. Notable examples include the generation of non-existent human faces, pets, or even cities. Models such as StyleGAN [7] have become important in these regard. The use of GAN's for image to image translations can be useful for inferring high-level topological structure from low-level representations. Models such as Pix2Pix [6], shorthand for 'pixel to pixel' have proven to be powerful and flexible tools for numerous research methodologies.

This project draws upon the potential of GAN's to propose models that are capable of extracting architecturally meaningful semantic descriptions from two-dimensional scenes of populated interior spaces. It uses image-to-image translation through a GAN trained on a dataset comprising of architecturally meaningful semantic abstractions, in order to extract high level architectural structure from low level pixel representations of interior scenes.

3. Methodology

3.1. Generating synthetic data for training

[2]1000 parametrically defined interior scene geometries were generated in Maya, based on photographs of classrooms from MIT. Classrooms were chosen as a context due to the relatively low variety of furniture and ease of access. While we used a generic model for chairs, all remaining items were represented by cube geometries. A black surface was explicitly defined for each room as a blackboard. The remaining cubes were scattered randomly.

Box geometries with randomized *width*, *height*, and *depth* values were scattered on walls, floors and ceilings via randomized u, v coordinates using *MASH* plug-in for Maya. Moreover, diffuse colors of box geometries were randomized in *hue*, *saturation*, and *value*.

Each of these scenes are rendered from a camera with random rotations in the y axis and random translations in the x (lateral). Camera yaw was constrained within ± 30 degrees, while pitch was strictly constrained to the horizontal axis, and no camera roll was allowed. Lateral movement was limited between the walls of the room with a 50cm offset. Each camera angle was used to generate a 512x512 pixel photo realistic image, with physically-based material and lighting information.

For each shot as generated above, output features containing semantic information of architectural structure were generated. These were encoded as 3 output images per scene, namely: Walls, ceilings and floors (defined as red, green, and blue), and depth maps. A total of 1000 images and corresponding feature sets were generated for training.

Given our render resolution, boxes were visually convincing placeholders for a variety of objects from the real classrooms. However, this approach is limited to low resolutions and objects that are not close to the camera. For a higher fidelity model, a library of objects will be necessary with matching physically based materials.

3.1.1 Data Augmentation

[3]To improve the robustness of our model, we applied a style transfer algorithm based on [5], stating that style and content representations can be separated in a convolutional neural network (CNN). Given a target and input image, style transfer migrates low level features of the target image into the input image while preserving high level features or *semantic* qualities of the input image. The output is an image where objects' shapes (or semantics) are preserved, but color, texture, and border qualities are taken from classroom photos 5.

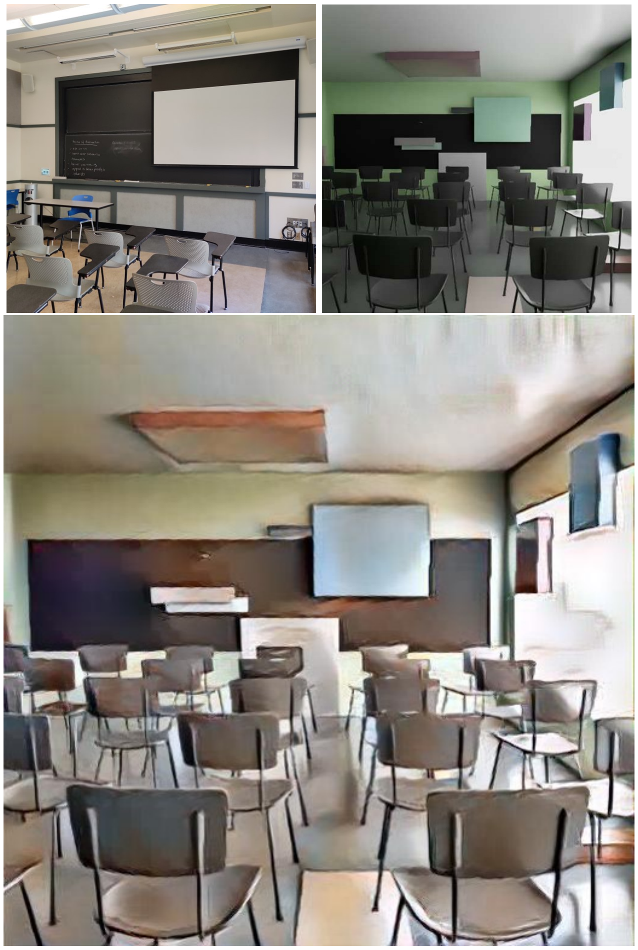


Figure 5. Classroom photo (top left), rendered image (top right), augmented synthetic image (bottom)

Using style transfer, we augment synthetic datasets to better represent the target context. Moreover, we use it as a means of domain randomization to account for the differences between synthetic objects (random cubes) and real ones (panels, projectors, key switches, screens, etc.).

3.2. Supervised training on synthetic data

[4]A Pix2Pix image-to-image translation model [6] was trained on the image pairs. Pix2Pix has been used in the past for a variety of image-to-image translation tasks, and has proven successful and robust in generating visual content from high level topological representations. Notable examples include the generation of photo-realistic images from conceptual sketches, or the generation of street level urban imagery from semantically segmented images. More importantly, the model has also been successful in generating/extracting high level representations from noisy two dimensional imagery. Examples in this regard include the

[2] This section presents efforts of Demircan Tas

[3] This section presents efforts of Demircan Tas

[4] This section presents efforts of Rohit Priyadarsh Sanatani

extraction of building footprint and street network information from satellite imagery of urban environments.

As discussed earlier, our proposal involves the extraction of high level, architecturally meaningful semantic representations from two-dimensional imagery of interior scenes. We note that both the synthetic interior scenes, as well as their architectural representations (labels and depth) are raster pixel-based images. However, since the output pixels embody architecturally meaningful information (the class of architectural elements and spatial relations between them) and missing parts are filled by Pix2Pix, the challenge posed by occlusion in the reconstruction of three-dimensional architectural structure from two-dimensional populated imagery is overcome.

For training, input-output pairs of images were preprocessed and down-scaled to 256x512 pairs. They were then split up into training (750), validation (200) and testing (50) sets. In addition to the test samples in the dataset, 30 real images of MIT classrooms were collected and used as an additional test set to evaluate the performance of the model in real world scenarios. Training parameters for Pix2Pix were taken into consideration, and were finalised as `batch_size = 1` and `lambda = 100`.

In total, three models were trained as part of our methodology. The first model (M_1) was trained to predict class labels from synthetic imagery without any data augmentation. The second model (M_2) was trained on synthetic imagery which was augmented using style transfer to match the distribution of the real world imagery as discussed earlier. Finally, the third model (M_3) was trained to predict depth from synthetic imagery without style transfer.

Training for *(M_1)* was first carried out over 1000 steps to gauge the performance of the model with current settings, and then continued successively over 10,000, 20,000 and finally 30,000 steps. Predictions on the test set were saved for each of the three training benchmarks. Training time was around 1 hour. For *(M_2)*, training was carried out over 10,000 steps. *(M_3)* was trained on the depth labels also over 10,000 steps.

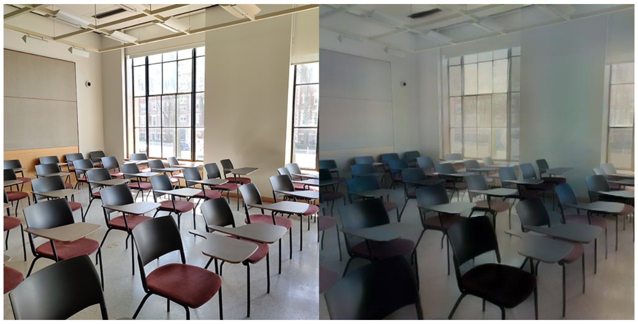


Figure 6. Real photograph (left) adapted styled to match synthetic imagery (right)

3.2.1 Domain Adaptation

[5]While data augmentation provides visually consistent images that yield marginally better results with the test data, predictions on photographs was not satisfactory. As an alternative approach to data augmentation, we also used style transfer using synthetic images for style, and test photos for content, prior to running the prediction model. This technique was used to match the the domain (real world pictures) to the data (synthetic images).

Using this approach, we sacrifice a minimal amount of accuracy on test data from synthetic images, but achieve superior accuracy in predictions on real world images.

4. Model Performance and Evaluation

4.1. Evaluation Metrics

For evaluation of the model, a pixel-wise comparison of predicted imagery and ground truth was undertaken. Two different methods were tested out in this regard. The first method involved involved iterating over each pixel of the predicted image, and checking equality with the corresponding pixel in the ground truth. The evaluation metric is defined as the percentage of pixels in the predicted image that match the ground truth. This method yields extremely low evaluation scores. This is because, while the two images may appear to be very similar upon visual comparison, and even have pixel values that are very similar to each other, the values have to be exactly the same to register as correct[6].

To work around this, the second evaluation metric considers a threshold value to check for equality between the pixels of the two images. If the squared error between the values lay within the threshold, they are considered equal. Choosing an appropriate threshold is critical in this regard. A threshold that is too large would fail to detect incorrect label predictions. A very low threshold on the other hand would penalize predictions that are very close to the ground truth and lead to low accuracy scores. After iterative testing, a threshold value of 10 was decided upon ($10 \times 10 = 100$ for comparison of squared error).

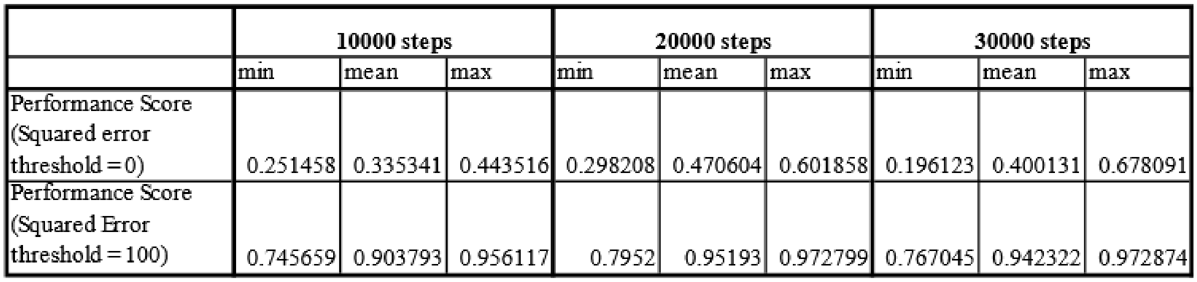
4.2. Model Performance

4.2.1 Model M_1 Trained on Synthetic Data without Augmentation

Model M_1 yielded impressive results in predicting architectural labels from synthetic interior scenes. Specifically, the structure of edges, and window boundaries were accurately reconstructed. Performance scores varied between 0.792

[5] This section presents efforts of Rohit Priyadarsh Sanatani

[6] Pix2Pix yields subtle differences in predicted colors. A *red* in the prediction, while perfectly red to the observer, is usually off by a slight margin in terms of pixel values.



	10000 steps			20000 steps			30000 steps		
	min	mean	max	min	mean	max	min	mean	max
Performance Score (Squared error threshold=0)	0.251458	0.335341	0.443516	0.298208	0.470604	0.601858	0.196123	0.400131	0.678091
Performance Score (Squared Error threshold=100)	0.745659	0.903793	0.956117	0.7952	0.95193	0.972799	0.767045	0.942322	0.972874

Figure 7. Model performance

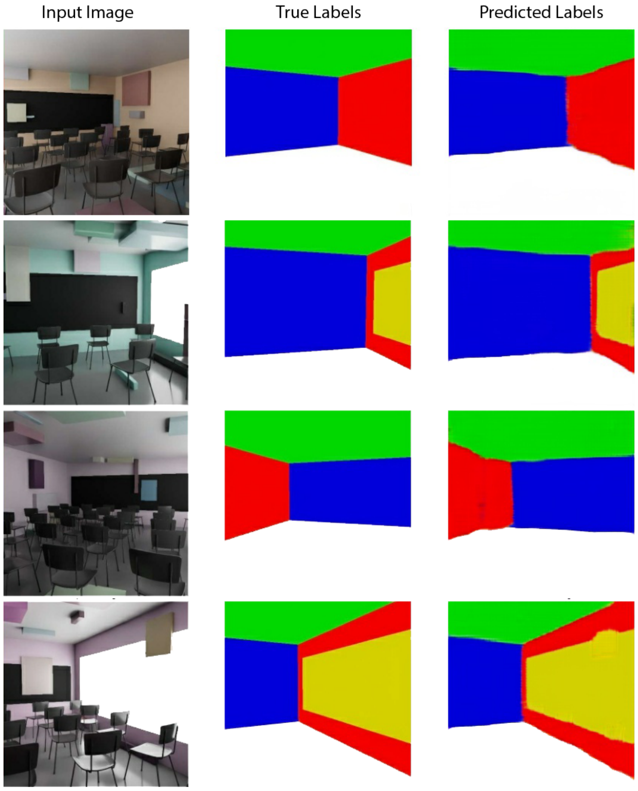


Figure 8. Sample predictions for Model MI on synthetic test data

and 0.956 over the 50 test samples, with a score of 0.904 (when trained through 20,000 steps). When trained for 30,000 steps however, the prediction accuracy was lower, indicating that the model over-fitted. Table 7 below depicts the performance scores at different stages of training. Fig. 8 shows representative predictions, as well as notable failure cases.

4.2.2 Model M_2 Trained on Synthetic Data with Data Augmentation

For model M_2, training was carried out on augmented data using style transfer as discussed earlier. This was done to improve the robustness of the model, and evaluate its performance not only on synthetic data, but also on real world imagery of actual classrooms. This was also done to provide a comparative evaluation of the system's performance when trained on augmented data and tested on real world imagery, as opposed to being trained on synthetic data and tested in adapted test imagery. Figure 9 depicts examples of the model's performance when tested on augmented synthetic data. The mean performance score (with squared error threshold = 100) was 0.940, with minimum and maximum scores of 0.879 and 0.959 respectively.

When tested on real world imagery, however, the performance of the model was far worse than expected. There were multiple failure cases, with many of the failures exhibiting systematic and consistent patterns. Figure 10 shows some representative predictions. When the same real world images were adapted to match the distribution of synthetic data and predicted through model M_1, the results turned out to be far better. These are discussed further in the proceeding section.

4.2.3 Model M_3: Trained on Synthetic Data and Depth Maps

Model M_3 was trained on synthetic data without augmentation, and the corresponding depth-maps of the enclosures without any non-architectural elements. Predicting depth of architectural elements at each point pixel of an enclosure irrespective of occlusions is critical, as it allows for the three-dimensional reconstruction of architectural features, when coupled with label predictions. The model performance seemed impressive visually, but the evaluation metrics were worse than that of models M_1 and M_2. The performance scores ranged between 0.30 and 0.97 with an average per-

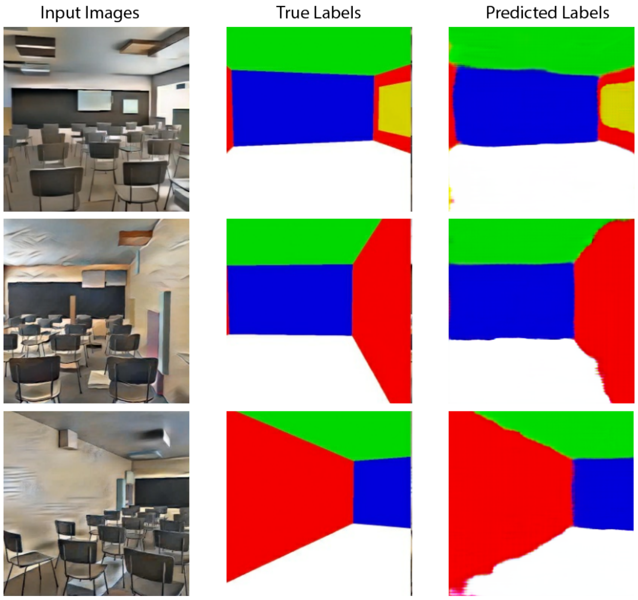


Figure 9. Sample predictions for Model M_2 on synthetic and augmented test data

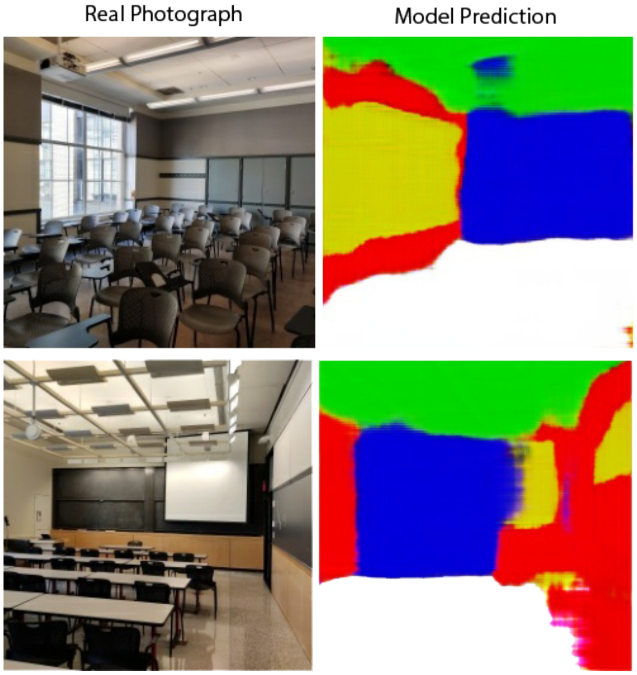


Figure 10. Sample predictions for Model M_2 on real world imagery

formance score of 0.79. Depth images are single channel, and unlike the label images, each pixel denotes a continuous depth variable rather than a discrete class. As a result, minor variations in the value of the pixels correspond to significant errors in depth prediction. Figure 11 shows representative examples of depth prediction.

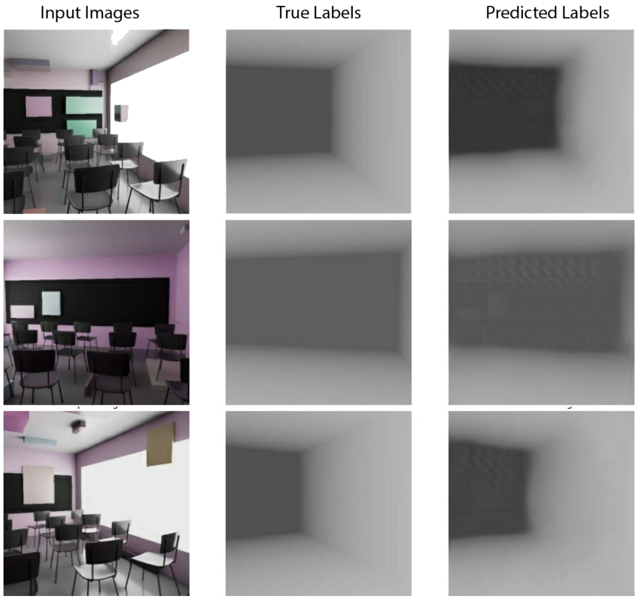


Figure 11. Sample predictions for Model M_3 on synthetic imagery

5. Discussion

5.1. Notable failure cases and patterns

While both models performed well on the synthetic test data which matched the training distribution, there were nevertheless notable cases of failure. In general, two major patterns of failure consistently occured across most failure samples. The first pattern involved failure to accurately detect window boundaries, or detecting windows where there were none. This was because, the windows in the synthetic images were characterized by very high pixel intensities. This resulted in parts of walls which also exhibited high intensities to be mis-labeled as windows (Fig 12 (top)). We see this pattern repeating in many cases, with very bright patches of walls being classified as windows.

The second failure pattern involved the detection of wall edges. In many cases, especially with augmented training data, edges of obstructions were misinterpreted by the model to be corners between two walls. This resulted in single walls with obstructions being labeled as two or more walls separated by the obstruction edges.

Fig. 10 also betrays notable failure patterns in predictions from real world imagery. In the upper image, window prediction (yellow) *bleeds* into the side walls (red) due to the continuity of horizontal line features, with transoms[7]. An artifact of this effect can be observed in the wavy vertical transition between window and side wall predictions.

[7] Horizontal crossbars in windows, over doors, or between windows, doors, or fanlights [10]

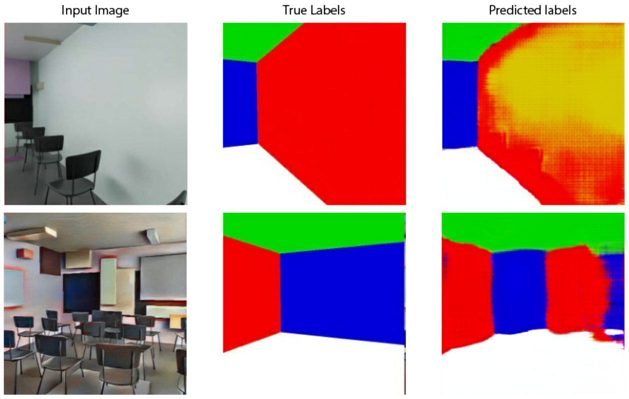


Figure 12. Notable failure patterns

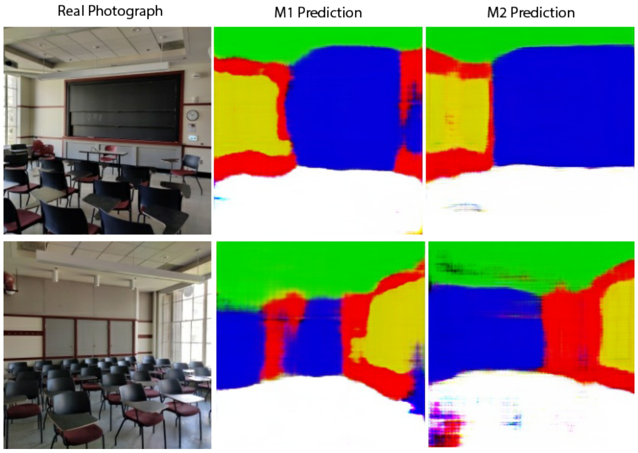


Figure 13. Performance comparison of data augmentation (M_1) and domain adaptation (M_2) on real world imagery

The bottom image presents an error case where high intensity areas of the right wall are mis-classified as windows. This issue is caused by window features in our test images consisting of over-exposed areas.

5.2. Data augmentation vs Domain adaptation

As we see from the results, model M_1 trained on augmented data yields better results on the synthetic augmented test data (mean performance score = 0.95), but fails to perform well when tested on real world imagery. On the other hand, model M_2 performs equally well on synthetic unaugmented test images (mean performance score = 0.94), but drastically better results with real-world images with domain transfer. Fig 13 illustrates this point, and compares the performance of both models on real world imagery. We thus conclude that training a model on un-augmented images and testing them on real world imagery adapted to match the training distribution shows promise, and can be taken forward for further development of this methodology.

5.3. Limitations

The results, though impressive, highlight a couple of important limitations of this methodology. Firstly, despite the use of data augmentation and domain adaptation to improve the model performances to real world data, a qualitative assessment of the outcomes reveal that they perform far worse on real world imagery as compared to the synthetic imagery that they were trained on. This however also opens up potentials for improving upon this work through the training of image-to-image GAN's on labeled real world data, rather than synthetic data. Secondly, the scope of this work was restricted to a certain architectural typology, namely classrooms. Applying this model to other typologies will result in lower prediction accuracies. A randomised generative algorithm similar to the one used in this work can be created for other typologies as well, to train the model to be more robust against a wider distribution of spatial typologies. Finally, while our work aimed at extracting both semantic labels as well as depth, the methodology can be extended to generate three-dimensional point clouds and meshes directly from two-dimensional imagery.

6. Conclusion

The body of research described in this paper is one of the first lines of inquiry that aim to extract architectural structure from noisy two-dimensional imagery where non-architectural obstructions occlude significant parts of the underlying architectural elements. Our results show that an image-to-image translation methodology is effective in this regard, capable of extracting high level structure from both synthetic as well as real world imagery.

As discussed earlier, a major challenge in contemporary architectural photogrammetry tasks is the semantic enrichment of photogrammetric meshes to reflect true architectural labels. This is valuable form a Building Information Modeling (BIM) point of view, as models generated through such a method will reflect the vocabulary of BIM rather than that of geometric modeling. Moreover, current photogrammetric models are singular meshes, where architectural elements such as walls and floor remain fused with non architectural elements such as furniture. A methodology for generating purely architectural models using photogrammetry is valuable for the AEC industry, with the potential to greatly speed up the generation of BIM models of existing structures.

We anticipate that the methodology described in this paper will prove to be a valuable foray into a variety of use cases within the world of architectural photogrammetry, building information modeling, and heritage documentation.

References

[1] Jonas Buchli, Markus Giftthaler, Nitish Kumar, Manuel Lussi, Timothy Sandy, Kathrin Dörfler, and Norman Hack. Digital in situ fabrication-challenges and opportunities for robotic in situ fabrication in architecture, construction, and beyond. *Cement and Concrete Research*, 112:66–75, 2018. 2

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[3] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 3, 4

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[8] Logambal Madhuanand, Francesco Nex, and Michael Ying Yang. Self-supervised monocular depth estimation from oblique uav videos. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176:1–14, 2021. 2

[9] G Sreenu and MA Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):1–27, 2019. 2

[10] Merriam Webster. Merriam-webster online dictionary. *Retrieved June*, 20:2013, 2006. 7

[11] Anny Yuniarti and Nanik Suciati. A review of deep learning techniques for 3d reconstruction of 2d images. In *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, pages 327–331. IEEE, 2019. 2

[12] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 2

[13] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2