3D Reconstruction from Single Image Using Occupancy Network with Vision Transformer Architecture

Van Cuong Dam ge83mus@mytum.de
Volkan Özer volkan.oezer@tum.de
Furkan Yakal furkan.yakal@tum.de
Kaan Özgen kaan.oezgen@tum.de

Abstract

In this paper, we present a modified version of Occupancy Networks tailored for the single image 3D reconstruction task. Empirical evidence demonstrates that Occupancy Networks outperform the majority of state-of-the-art baselines.

With the aim to achieve enhance overall performance, we initially propose a Detection Transformer-based architecture. Regrettably, this model encountered memory-related issues, prompting us to seek an alternative approach. Subsequently, we propose a Vision Transformer-based architecture, which yielded favorable results. We evaluated these methods on the ShapeNet dataset w.r.t three metrics IoU, Chamfer L_1 distance and Normal Consistency.

1. Introduction

In single image 3D reconstruction, the goal is to infer the 3D structure, shape, and spatial layout of the scene from the information available in the 2D image. There can be multiple plausible 3D interpretations for a single 2D image. As a result, the quality and accuracy of the reconstructed 3D model can vary depending on the complexity of the shape or scene, the available information in the image, and the sophistication of the reconstruction approach chosen.

In this paper, we propose a new deep learning framework, which is an adaptation of the Occupation Networks—a prominent approach in 3D reconstruction from images [4]. By integrating the Vision Transformer [3], which is a cutting-edge image recognition model, into the Occupancy Network, we aim to harness the potent capabilities of self-attention and enhance the performance of 3D reconstruction tasks. The model takes a RGB image of an object along with its corresponding point clouds from ShapeNet [2] as input and outputs the occupancy probabilities.

Our contributions to the Occupancy Networks [4] are outlined as follows: Firstly, we replace the ResNet Backbone, used for feature extraction in the Occupancy Network, with the Vision Transformer [3]. Secondly, we incorporate standard CNNs instead of multiple fully-connected ResNet-blocks as in Occupancy Networks to output occupancy probabilities. Lastly, we replace the Conditional Batch Normalization with Conditional Group Normalization. For detailed information about our work please refer to our GitHub repository as well.[1].

2. Related Work

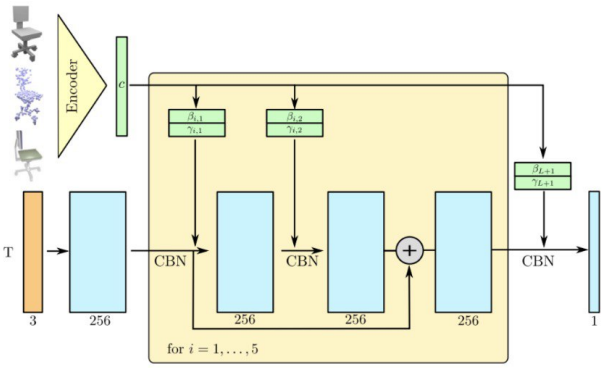

Figure 1. **Occupancy Network** architecture. Firstly, an embedding C for the images is computed using ResNet-18 architecture. Then the input points are fed through multiple fully-connected ResNet-blocks. In these ResNet-blocks, Conditional Batch Normalization is used to condition the network on C. Finally, the output is projected to one dimension using a fully-connected layer and the sigmoid function is applied to obtain occupancy probabilities [4].

The utilization of 3D representations across diverse applications has seen significant advancements. Various methods, such as voxels and point clouds, have been employed for 3D representations. However, it is unfortunate that each of these representations exhibits its own weaknesses [5]. Voxels, for instance, are suffered from computational and memory demands, particularly when aiming for high-resolution reconstructions, as the associated costs grow cubically with the resolution [4, 5]. On the other hand, point

[1]Our project repostitory: https://github.com/kaanozgen12/ML43D-PROJECT

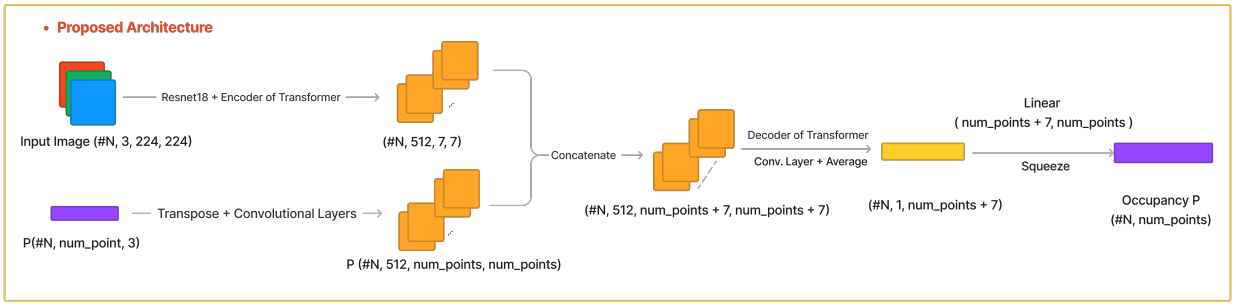

Figure 2. Initially proposed **DeTrOcNet** architecture. The ResNet-18 encoder architecture of the Occupancy Networks is combined with the encoder of the Detection Transformer [1]. Points clouds are transposed and feed through the convolutional layers. Encodings and point clouds are concatenated, and then decoded using the decoder of Detection Transformers [1]. Subsequently, several layers are used to ensure the output with the same size of (batch size, num point) as in the Occupancy Network [4].

cloud representations often face limitations concerning the number of points they can handle, and requires additional non-trivial post-processing steps to generate the final 3D mesh [4, 5]. The aforementioned representations are discrete in nature, lacking continuity [5]. In contrast, implicit representations offer a continuous framework for representing 3D shapes and scenes, addressing the challenges posed by discrete representations [5].

Occupancy Networks introduce a new perspective to address the representational problems in 3D by providing a learning-based approach to implicit function representation [4]. They output the occupancy probability of 3D space points, which allows for accurate representations of complex and detailed 3D shapes without the need for dense sampling [4]. Furthermore, Occupancy Networks have demonstrated better metrics over some of the prominent works in the field such as 3D-R2N2, PSGN, Pix2Mesh, and AtlasNet [4]. Therefore, Occupancy Networks present a promising direction for further development in the 3D representation domain.

3. Method

In this section, we begin by introducing our initially proposed model, DeTrOcNet. Subsequently, we present our final model, ViTOcNet.

3.1. DeTrOcNet

This proposed architecture's fundamental inspiration stems from the similarity between Occupancy Networks [4] and Detection Transformer [1] in their preprocessing approaches for 2D images. Building upon this observation, we decided to replace the encoder part of Occupancy Networks [4] with the Transformer architecture, mirroring the design of DeTrOcNet. To combine the extracted features from 2D images with the point cloud data, we opted for a straightforward method: concatenating them together instead of adopting Conditional Batch Normalization, as done in Occupancy Networks [4]. Notably, the concatenation takes place after the encoder of the Transformer, in contrast to Occupancy Networks [4], where it occurs immediately after the pretrained ResNet.

The concatenated latent code is then forwarded through the decoder of the Transformer. To match the output tensor size of $(N, numpoint)$, similar to Occupancy Networks [4], we employed multiple layers and operations.

During the model building and training process, we encountered an obstacle when attempting to visualize the results. The recurrent error of "running out of memory" persisted despite our attempts to fix the issue, such as reducing the batch size to 1 or decreasing the number of parameters. After exhaustive investigations, we identified the root cause to be the varying number of point clouds used in different processes. During training, only 2048 point clouds were utilized, while 32^3 point clouds were required for visualization. Consequently, in the DeTrOcnet architecture, certain layers contained parameters that were dependent on the number of point clouds. A notable example is the linear layers at the end of the architecture, where the number of parameters equaled $numpoint \times (numpoint + 7)$. In the visualization phase, this resulted in number of parameters $32^3 \times (32^3 + 7)$, which leads to a substantial number, thus triggering the memory limitation issue.

Despite our efforts to address this challenge, the complexity of the problem surpassed our expectations. Additionally, the constrained timeframe, with approximately 2 weeks remaining, imposed further limitations. Consequently, we had no alternative but to adopt a more straightforward approach, aiming to demonstrate at least some preliminary results for our project.

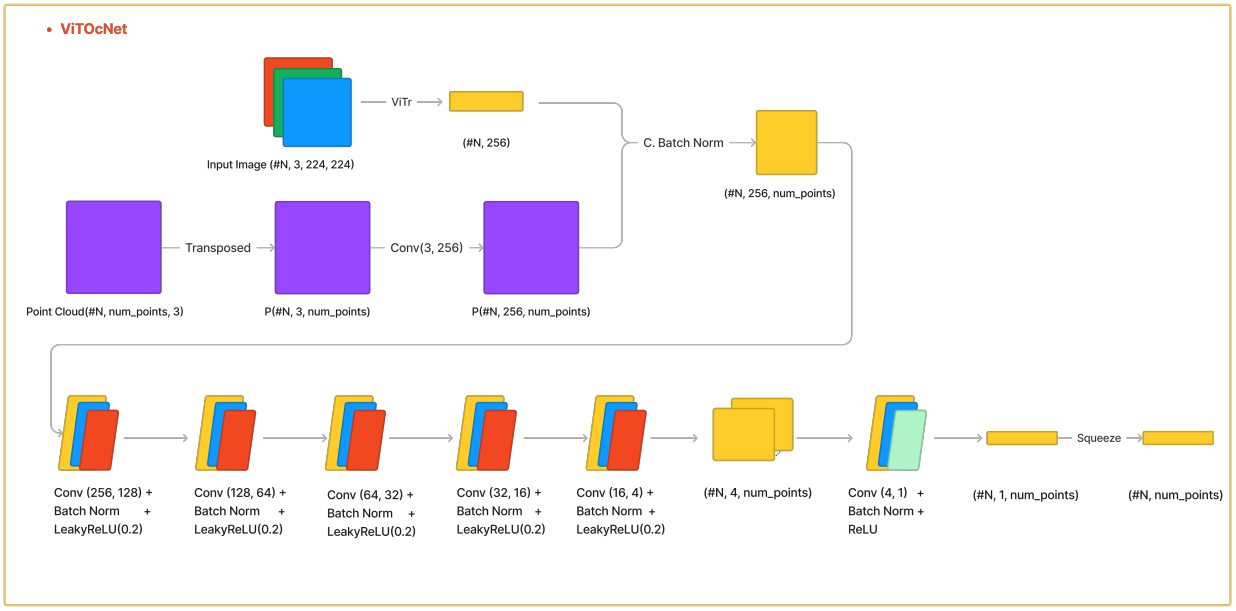

Figure 3. Our final architecture **ViTOcNet.** The ResNet-18 encoding block of the Occupancy Network architecture is replaced with the Vision Transformer [3]. Then point clouds and embeddings are combined with single layer of Conditional Group Normalization. Followed by convolutional layers for obtaining the probabilities.

3.2. ViTOcNet

In our research, we aimed to enhance the encoder component of our model by investigating alternative architectures. Initially, our intention was to replace the ResNet architecture with a Detection Transformer [1] architecture. Regrettably, despite our efforts to fine-tune the hyperparameters, the encoder of the Detection Transformer failed to converge. Consequently, we had to resort to using the Vision Transformer [3] as the encoder.

Following the encoder step, we obtained the embedding vector "c", which shared the same size $(N, 256)$ as in the Occupancy Networks [4] architecture. Integrating this embedding vector with the point cloud posed challenges, primarily due to the substantial number of point clouds involved in the visualization process. Given this constraint, we had to identify an operation capable of handling the size of the data without incurring excessive computational costs.

After careful consideration, we determined that the most suitable operation was Conditional Batch Normalization. This choice was influenced by the previous findings of Occupancy Networks [4], where the operation's calculations were implemented based on broadcasting, resulting in significantly faster matrix calculations. Thus, we were compelled to utilize Conditional Batch Normalization to merge the point cloud and embedding efficiently.

It is worth noting that in contrast to Occupancy Networks [4], where multiple layers of Conditional Batch Normalization were employed, we opted to use only one layer. This decision was motivated by the input's two-dimensional size $(N, 256, numpoint)$, which we treated as a 1D image. Accordingly, we utilized a 1D CNN to process the data and generate occupancy probabilities as outputs.

During our experimentation, we also explored the possibility of replacing the batch normalization layer in Conditional Batch Normalization with a Conditional Group Normalization layer. Surprisingly, both modifications yielded similar performance outcomes.

To summarize our modifications, the encoder was replaced with Vision Transformer [3], which was built from scratch rather than using pretrained weights. Moreover, standard CNN was employed instead of fully-connected ResNet-blocks to generate occupancy probabilities. Lastly, Conditional Group Normalization was utilized to integrate the point cloud and embedding data.

4. Experiments

4.1. Dataset

In accordance with the project requirements, we have opted for the OmniObject3D [7] dataset as an alternative data source to train our model. The challenge, however, lies in the need to prepare our dataset in a manner akin to

Category	IoU						Chamfer-L_1						Normal Consistency					
	3D-R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	Ours	3D-R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	Ours	3D-R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	Ours
airplane	0.426	-	0.420	-	**0.571**	0.310	0.227	0.137	0.187	**0.104**	0.147	0.410	0.629	-	0.759	0.836	**0.840**	0.708
bench	0.373	-	0.323	-	**0.485**	0.117	0.194	0.181	0.201	**0.138**	0.155	0.687	0.678	-	0.732	0.779	**0.813**	0.603
cabinet	0.667	-	0.664	-	**0.733**	0.549	0.217	0.215	0.196	0.175	**0.167**	0.344	0.782	-	0.834	0.850	**0.879**	0.752
car	0.661	-	0.552	-	**0.737**	0.582	0.213	0.169	0.180	**0.141**	0.159	0.232	0.714	-	0.756	0.836	**0.852**	0.766
chair	0.439	-	0.396	-	**0.501**	0.311	0.270	0.247	0.265	**0.209**	0.228	0.529	0.663	-	0.746	0.791	**0.823**	0.706
display	0.440	-	**0.490**	-	0.471	0.296	0.314	0.284	0.239	**0.198**	0.278	0.598	0.720	-	0.830	**0.858**	0.854	0.670
lamp	0.281	-	0.323	-	**0.371**	0.189	0.778	0.314	0.308	**0.305**	0.479	0.809	0.560	-	0.666	0.694	**0.731**	0.520
loudspeaker	0.611	-	0.599	-	**0.647**	0.511	0.318	0.316	0.285	**0.245**	0.300	0.504	0.711	-	0.782	0.825	**0.832**	0.712
rifle	0.375	-	0.402	-	**0.474**	0.255	0.183	0.134	0.164	**0.115**	0.141	0.321	0.670	-	0.718	0.725	**0.766**	0.656
sofa	0.626	-	0.613	-	**0.680**	0.401	0.229	0.224	0.212	**0.177**	0.194	0.430	0.731	-	0.820	0.840	**0.863**	0.666
table	0.420	-	0.395	-	**0.506**	0.222	0.239	0.222	0.218	0.190	**0.189**	0.508	0.732	-	0.784	0.832	**0.858**	0.715
telephone	0.611	-	0.661	-	**0.720**	0.515	0.195	0.161	0.149	**0.128**	0.140	0.304	0.817	-	0.907	0.923	**0.935**	0.828
vessel	0.482	-	0.397	-	**0.530**	0.289	0.238	0.188	0.212	**0.151**	0.218	0.394	0.629	-	0.699	0.756	**0.794**	0.617
mean	0.493	-	0.480	-	**0.571**	0.350	0.278	0.215	0.216	**0.175**	0.215	0.467	0.695	-	0.772	0.811	**0.834**	0.686

Table 1. Numerical comparison of our approach and the baselines for single image 3D reconstruction on ShapeNet. We measure the IoU, Chamfer-L_1 distance and Normal Consistency with respect to ground truth mesh [4].

the Occupancy Network's methodology. The paper requires watertight meshes as input to determine if a point lies in the interior of a mesh (e.g., for measuring IoU) which necessitates the generation of "pointcloud.npz" and "points.npz" files from the "obj" files of objects [4]. To achieve this task and preprocess our dataset, we initially adapted the script developed by Stutz et al. [6] for creating watertight meshes. Regrettably, this process involves dependencies on C/C++ and requires the installation of OpenGL libraries, which, unfortunately, is not permissible on the cluster. Moreover, conducting the preprocessing locally is not a viable option due to the lack of computational power and substantial size of the processed dataset, which, akin to the original paper, amounts to approximately 70GB to 80GB. As a result of these challenges, we are compelled to train and test our model with the preprocessed data, ShapeNet, provided in the original paper [2, 4].

We subdivide the data into a training and a validation set for tracking the loss of our model to determine when to conclude training.

4.2. Metrics

For evaluation of the performance of our model, we utilize various metrics used in the Occupancy Networks such as volumetric IoU, Chamfer-L_1 distance and normal consistency score. Volumetric Intersection over Union (IoU) is calculated as the ratio of the union volume to the intersection volume of two meshes [4]. The Chamfer-L_1 distance is a composite metric, comprising of an accuracy and completeness component. The accuracy metric calculates the mean distance between points on the output mesh and their nearest neighbors on the ground truth mesh, while the completeness metric does the opposite [4]. Finally, the introduced normal consistency score measures the mean absolute dot product of the normals in one mesh and the normals at the corresponding nearest neighbors in the other mesh [4].

4.3. Results

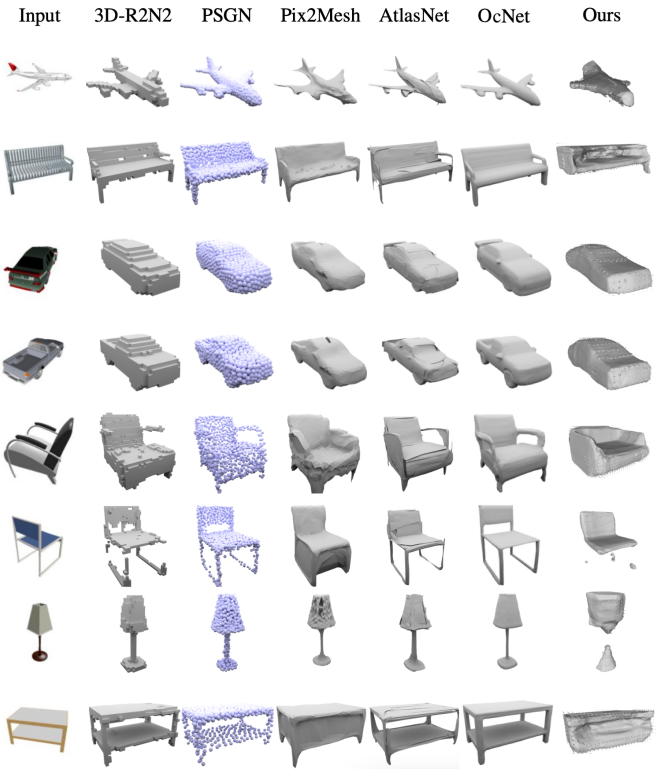

Figure 4. Qualitative results for single image 3D reconstruction on ShapeNet. The input image is shown in the first column, the other columns show the results for our method compared to various baselines [4].

Employing the identical dataset as the original paper allowed us to conduct a comprehensive comparative analysis of our model against several state-of-the-art baselines, including 3D-R2N2, PSGN, Pix2Mesh, and AtlasNet, as exemplified in the paper [4]. The comparison encompassed both qualitative and quantitative evaluations, enabling a thorough assessment of our model's performance in relation

to these cutting-edge approaches.

The qualitative comparison between our model and the baselines is depicted in the Fig. 4. It is evident that our model faces limitations in representing object details. While some shapes are discernible, certain objects, such as chairs and lamps, appear to pose significant challenges, because our method exhibits difficulties in capturing connectivity, particularly on thin surfaces.

The quantitative results are presented in the Table 1. It is apparent that our method does not yield any improvement over occupancy networks in either of the metrics. Nevertheless, we strongly believe that the performance of our network could be enhanced with a longer training time.

5. Conclusion

In this paper, we aimed to propose a new approach for the single image 3D reconstruction task. Our motivation was based on the similarity of the preprocessing step between the Occupancy Networks and the Detection Transformer Network. Consequently, we hypothesized that integrating a transformer-based architecture could enhance overall performance owing to the higher complexity of Detection Transformers or Vision Transformer in comparison with standard Convolutional Neural Network. However, our initial attempt using a Detection Transformer-based architecture encountered memory-related challenges, and the Vision Transformer-based architecture failed to outperform the original results.

Despite not achieving significant improvements over the occupancy networks, this endeavor provided valuable insights and experience. Through this process, we gained a deeper understanding of transformer architecture, design, and how to approach scientific problems.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 2, 3

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. 1, 4

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1, 3

[4] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space, 2019. 1, 2, 3, 4

[5] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks, 2020. 1, 2

[6] David Stutz and Andreas Geiger. Learning 3d shape completion under weak supervision. *International Journal of Computer Vision*, 128(5):1162–1181, oct 2018. 4

[7] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation, 2023. 3