# Project Report: Occupancy Networks

Yujun Lin Joong-Won Seo Yunan Li
Technical University of Munich

## Abstract

*In the area of 3D reconstruction, the rise of deep neural networks has sparked interest in learning-based approaches. Unlike 2D images, efficient representation of high-resolution geometry with arbitrary topology still remains many challenges. In this report, we present enhancements and changes to Vanilla Occupancy Networks. Our model employs ConvNeXt as the backbone, integrates a feature pyramid in front of the Encoder, and introduces a Pose-aware decoder. Through experiments, we had improved training performance in respect to IoU and number of parameters compared to the vanilla Occupancy Network.*

## 1. Introduction

Occupancy Network (O-Net) is a neural network designed for occupancy estimation tasks, leveraging ResNet-18 as its backbone architecture for feature extraction from single RGB images. Its output consists of volumetric occupancy probabilities, where each voxel in the 3D space is assigned a probability indicating the likelihood of occupancy. O-Net is aiding in 3D reconstruction and 3D Object understanding tasks. While O-Net outperforms many models, there are still many drawbacks due to it's backbones and architectures. O-Net utilizes ResNet-18, an older architecture with MLP as its backbone, which may lead too many weights, the drawbacks such as vanishing gradients in deeper networks is not avoidable. This has also been experimented by the original papers, where they compared the performance of ResNet18, ResNet32 and ResNet50 and had the conclusion that less weights lead to better performance. Additionally, O-Net's design does not inherently accommodate learning under varying scales or capturing intricate local features, potentially limiting its effectiveness in tasks requiring spatial understanding and local feature understanding under different scales.

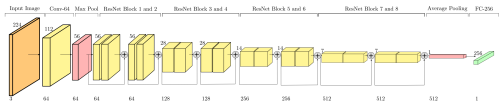

Figure 1. Encoder for single image 3D reconstruction

## 2. Related Work

### 2.1. Occupancy Networks

Occupancy Networks [2] represent a novel approach for learning-based 3D reconstruction task. Traditional methods often use discrete representations like point clouds or voxel grids to represent output 3D shape. On the other hand, O-Nets use a continuous and implicit representation for output. This is achieved by approximating a 3D function with a neural network $f_\theta$ that takes a 3D point location $p \in \mathbb{R}^3$ along with an observation $x \in \chi$ as input pair $(p, x)$ and outputs the corresponding occupancy probability:

$$f_\theta : \mathbb{R}^3 \times \chi \rightarrow [0, 1] \tag{1}$$

To handle various input types including single images, noisy point clouds and coarse voxel grids, Occupancy Networks employ a task-specific encoder to extract shape information and underlying 3D structure from the input and map the data into latent feature space. Encoder for single images is shown in Figure 1. The learned function implicitly represents the 3D surface as a continuous decision boundary which allows to extract 3D meshes at any resolution. The overview of the architecture is illustrated in Figure 2.

### 2.2. ConvNeXt

ConvNeXt based on convolutional neural network (CNN) architecture and it has been developed by researchers at Facebook AI Research (FAIR) and UC Berkeley [1]. It stands out for its model architecture design, incorporating interconnected convolutional blocks with multiple pathways for information flow. Unlike traditional CNNs, ConvNeXt diversifies information processing through parallel pathways, enhancing learning efficiency and representation of complex visual patterns. One of ConvNeXt's key strengths lies in its hierarchical structure, enabling the network to capture multi-scale information at various levels

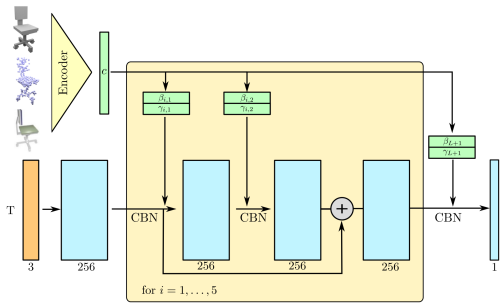


Figure 2. Overview of Occupancy Network architecture

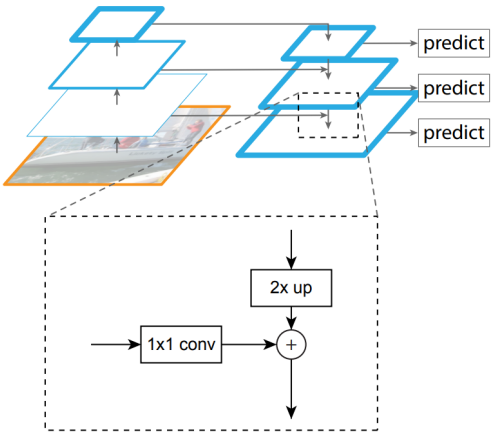


Figure 3. A top-down architecture with skip connections, with predictions made independently at all levels and the building block illustrating the lateral connection and the top-down pathway, merged by addition.

of abstraction. This feature enhances robustness to object size, orientation, and appearance variations commonly encountered in real-world scenarios. With impressive results across image classification, object detection, and semantic segmentation tasks, ConvNeXt demonstrates versatility and scalability for deployment in various applications, including autonomous driving, robotics, healthcare, and surveillance systems.

### 2.3. Feature Pyramids

Feature Pyramid Networks (FPNs) was proposed for object detection tasks in the field of computer vision. They address the challenge of recognizing objects at various scales within an image. As illustrated in Figure 3, the architecture computes a pyramidal hierarchy of features at multiple scales based on a single-scale image during the bottom-up pathway, and then upsamples coarser feature maps from higher pyramid levels in the top-down pathway with lateral connections which merge feature maps of the same size from both pathways. Each level of the pyramid can be used for detecting objects at a different scale while all levels are semantically strong.

## 3. Proposed Modifications

In this section, we will start by explaining the baseline architecture in detail. Then, we will walk through each modification we made, explaining what we changed and the reasons behind modifications. And the expectations on how these modifications could improve performance.

### 3.1. Modification 1: Encoder Backbone

The Backbone of the vanilla O-Net is based on ResNet, this is a older Network with MLP as its backbone. We propose to change the backbone to ConvNeXt. There are several advantages which can be anticipated, leveraging the strengths inherent to ConvNeXt. ConvNeXt offers enhanced performance over traditional convolutional neural networks by incorporating cross-scale connections and multi-level feature fusion. This design facilitates better information flow across different scales and resolutions, enabling the network to capture both local and global context effectively. Additionally, ConvNeXt's architecture promotes more efficient parameter utilization, leading to improved model capacity without significant increases in computational overhead. Moreover, ConvNeXt's adaptability to various input resolutions and its ability to learn hierarchical representations make it well-suited for occupancy estimation tasks, potentially resulting in more accurate and robust predictions compared to ResNet. Overall, the adoption of ConvNeXt as the backbone for Occupancy Network holds promise for enhancing performance by capitalizing on its advanced architectural features and capabilities. ConvNeXt is an upgraded Version of Convolutional Neural Networks (ConvNets), ConvNets dominate computer vision due to their inherent properties aligning with visual processing needs, like translation equivariance. Initially used for limited object categories, ConvNets evolved with region-based detectors in the 2010s. Meanwhile, in natural language processing, Transformers replaced recurrent networks. In 2020, Vision Transformers (ViT) emerged base on the idea of Transformer, which enhanced image classification performance further. However, ViT faces challenges by 3D objects classification and generation tasks, it lacks ConvNet's sliding window advantage and has quadratic complexity with higher resolutions. Hierarchical Transformers like Swin Transformer reintroduce ConvNet-like strategies, bridging the gap between ConvNets and Transformers. Figure 4 shows the Classification Result for ResNet, ViT, Swin Transformer and ConvNeXt, the prediction accuracy of ConvNeXt outperformed another architectures on ImageNet-1K Dataset.

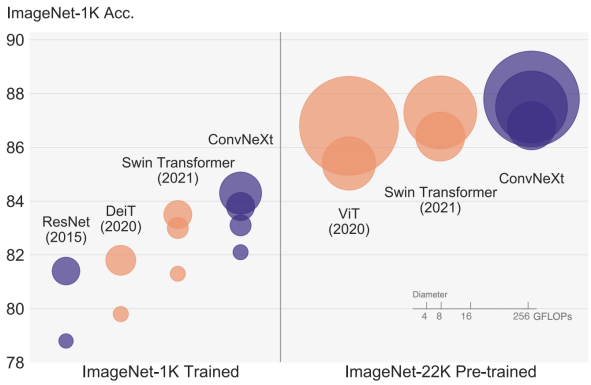


Figure 4. ImageNet-1K Classification Result for ResNet, ViT, DeiT, Swin Transformer and ConvNeXt

### 3.2. Modification 2: Multi-scale Input

We introduced feature pyramid to enable the multiscale images as input. By integrating the feature pyramid into the encoder of an Occupancy Network we try to make an enhancement aimed at improve the model's capacity to understand hierarchical representations of the input data under different scales. A feature pyramid architecture involves extracting features at multiple scales and this augmentation has several advantages can lead to better performance and expanded capabilities in understanding the local features of the images. The first advantage is that the feature pyramid architecture enables the model to capture multi-scale information present in the input data. By incorporating features from different scales, the network gains a more comprehensive understanding of the scene or object being represented. This can be particularly beneficial in tasks where objects or structures of interest vary significantly in size. Additionally, the feature pyramid architecture facilitates improved spatial reasoning and context-awareness within the network. Features extracted at different scales provide the decoder with contextual information about the surrounding environment, allowing for more informed decisions regarding occupancy predictions. This enhanced contextual understanding can lead to more accurate and coherent reconstructions of complex scenes or objects.

### 3.3. Modification 3: Pose Information

Finally, we experiment with a naive approach to incorporate camera pose information, along with the single image, with the intention of enhancing the Occupancy Network's understanding of spatial relationships. Specifically, we encode the camera's 4x4 pose matrix using a simple linear layer after flattening. This encoded pose is then concatenated with the output from the image encoder, on which the decoder conditions to output the occupancy predictions. This approach is motivated by the observation that, under very specific circumstances, such as distinguishing between elements such as walls, floors, and roofs can be challenging without understanding the camera's perspective. However, it's important to note that this method does not employ a continuous representation for rotation, due to the inherent discontinuities in such representations.

## 4. Experiments

### 4.1. Dataset

For our experiments, we use Pix3D [4] as our new data source which contains images, masks, meshes and camera positions. We split the data into a training set (70% of the whole dataset), a validation set (10% of the whole dataset) and a test set (20% of the whole dataset). We follow the procedure to extract the data from meshes that was used in Occupancy Networks [2] as following. First we use the code provided by Stutz et al. [3] to create watertight version of meshes. Second, we normalize the shape so that the bounding box is centered at the origin. By using an uniform random distribution, we sample 100k points from the new bounding box. Last, we subsample 2048 points and compute the corresponding occupancy during training.

### 4.2. Results

We conducted our experiments on the Pix3D chair dataset, training each model for 3000 steps using the Adam optimizer. Our model was compared against the baseline Occupancy Network equipped with a ResNet18 encoder for the single image reconstruction task. It's noteworthy that all models consistently began to overfit early in the training process, typically between 500 to 1000 steps.

**(1) Encoder Backbone**: When we replaced the encoder backbone with ConvNeXt, there was an observable improvement in Intersection over Union (IoU) metrics, albeit with an increased parameter count. To dissect the impact of parameter count versus architectural enhancements, comparisons were made with ResNet34, ResNet50, and ResNet101 variants. These comparisons revealed that varying ResNet architectures did not contribute to performance improvements, contrasting with the consistent gains achieved through ConvNeXt, as seen in Table 1.

**(2) Multi-scale Input**: The introduction of a multi-scale input, facilitated by a feature pyramid architecture, resulted in a substantial increase in the model's parameter count—nearly tripling due to the necessity for a separate encoder at each scale level. Contrary to expectations, this modification did not enhance model performance. Instead, it led to a decrease in performance, with the IoU metric dropping by approximately 50% (see 1).

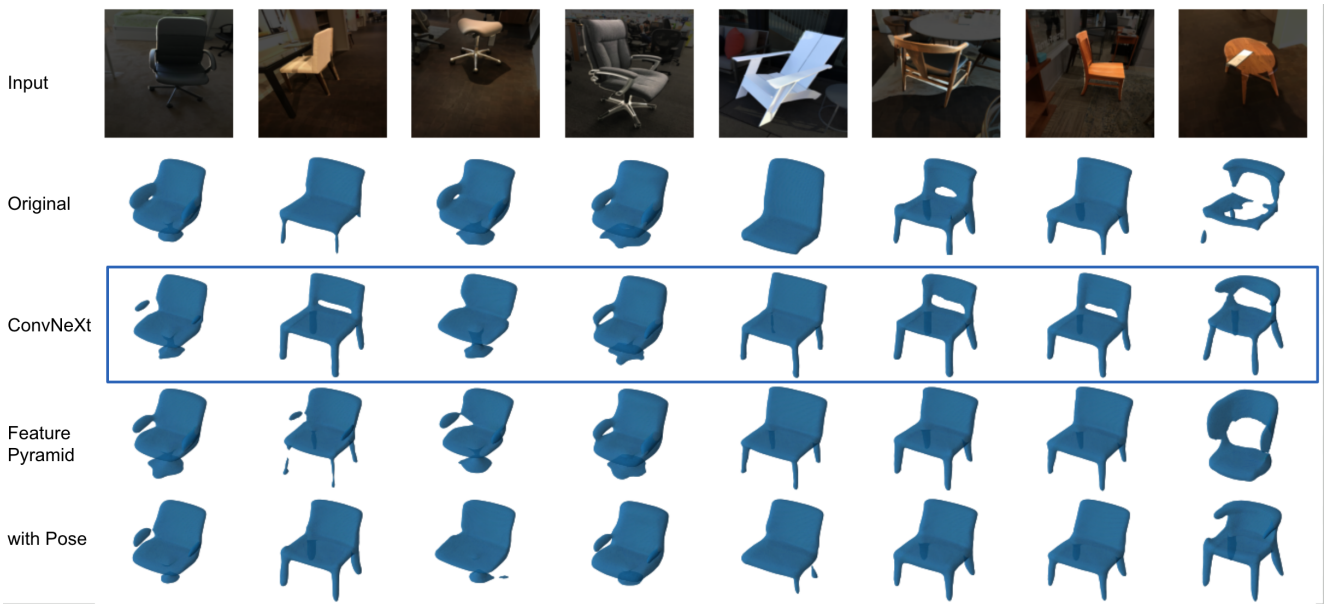


Figure 5. Single image reconstruction on 8 samples from the test dataset. Top row shows the input image with the mask at 0.3 alpha, but the model receives a fully masked input (background pixels are set to 1). The bottom 3 rows show our modifications (1), (2) and (3).

| Model | ResNet18 (base) | ResNet34 | ResNet50 | ResNet101 | ConvNeXt Tiny | + 3-scale FP | + Pose |
|---|---|---|---|---|---|---|---|
| # of Params | **13.4M** | 23.5M | 26.1M | 45.1M | 30.1M | 86.1M | 86.3M |
| IoU on Val | 0.4041 | 0.4128 | 0.3941 | 0.4061 | **0.4296** | 0.4166 | 0.3838 |
| Improvement | - | +2.2% | -2.5% | +0.5% | **+6.3%** | +3.1% | -5.0% |

Table 1. Comparison of various encoder architectures and their effect on the validation IoU, evaluated on Pix3D chairs dataset. Right three columns are each results of our modifications (1), (2) and (3)

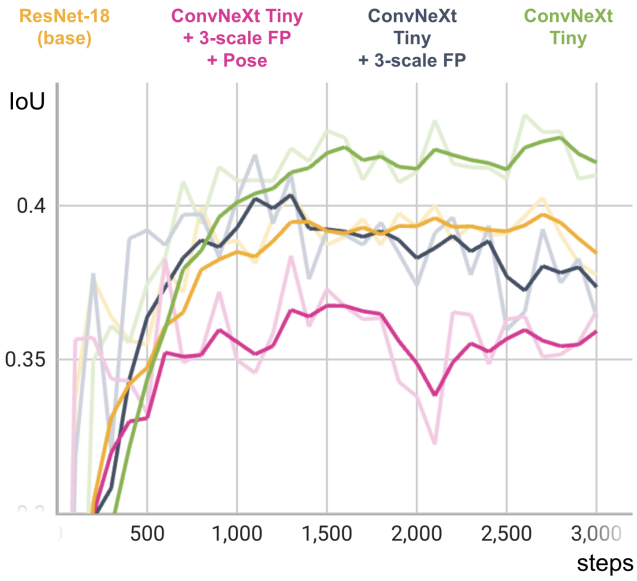


Figure 6. Validation IoU for each of our modifications.

**(3) Pose Information**: In our prepared dataset, each object is represented by a single image and one corresponding pose, a result of our dataset conversion implementation. Incorporating pose information through a simplistic pose encoder did not significantly affect the overall parameter count. However, this modification appeared to adversely affect the model's generalization capability, as indicated in Table 1.

## 5. Discussion

Our work encountered several challenges throughout the experimentation phase, each with its potential solution outlined below:

**Small Dataset**: The dataset size was notably small, with only 220 objects compared to larger datasets like ShapeNet, which contains 6778 objects. This limitation potentially restricts the model's learning capability and generalization. A viable solution is to pretrain the model on the extensive ShapeNet dataset and then fine-tune it on our smaller dataset. This approach could leverage the vast diversity of ShapeNet to improve feature learning and

adaptation to specific nuances in our dataset.

**Pose Encoding**: The challenge with pose encoding arises from the discontinuous nature of rotation representations. To address this, we propose adopting the 5D or 6D rotation representation introduced by [5]., which offers a more continuous and geometrically interpretable form of encoding rotations, potentially enhancing model performance in understanding and utilizing pose information.

**Few Image/Pose per Mesh**: Our dataset conversion resulted in a scenario where each object is associated with too few images and poses, limiting the model's ability to learn robust correlations between them. To mitigate this, generating synthetic data by rendering the mesh from various random camera poses could enrich the dataset. This enhancement would provide a broader spectrum of visual and pose data, aiding in the model's learning process.

**ONet and Larger Encoders/Data Augmentation**: Occupancy Networks (ONet) have shown issues when integrated with larger encoders and undergoing data augmentation, although the precise reasons behind these complications are not fully understood. This presents an intriguing area for further investigation. Understanding why ONet behaves unfavorably under these conditions could unlock new methodologies for improving its architecture and overall performance in 3D reconstruction tasks.

## 6. Conclusion

In this report, we updated the Occupancy Network by adding a ConvNeXt backbone, a feature pyramid, and a pose-aware decoder to better handle 3D reconstruction challenges. Our work shows that a better encoder like ConvNeXt can improve performance in terms of IoU, but there's still room to explore different decoder designs.

Adding a feature pyramid for multi-scale input showed potential but also highlighted the need for better regularization to manage the increased number of parameters without losing performance.

The introduction of pose information, though intended to enhance model understanding of spatial relationships, ended up not benefiting the model as expected. It made the model too dependent on this data, hurting its ability to generalize.

Overall, our adjustments to O-Net offer insights into enhancing 3D reconstruction models but also underscore the complexity of achieving significant performance gains. Future work should look into optimizing decoder architectures and finding effective ways to incorporate multi-scale inputs and pose information without compromising the model's generalization capabilities.

## References


[1] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 1

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

[3] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018. 3

[4] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[5] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. 5