IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 10, October, 2025, pp. 2563-2570

Journal homepage: http://iieta.org/journals/isi

Pruning-Based Efficient Point Generation Network for 3D Reconstruction of Single Images

Anny Yuniarti*, Agus Zainal Arifin, Nanik Suciati

Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Corresponding Author Email: anny@its.ac.id

https://doi.org/10.18280/isi.301004

Received: 15 July 2025
Revised: 24 September 2025
Accepted: 3 October 2025
Available online: 31 October 2025

Keywords:
point cloud generation, 3D reconstruction, pruning, model compression, efficiency

ABSTRACT

Single view reconstruction is a problem of 3D reconstruction given only a single 2D RGB image. Recently, an end-to-end learning framework has been implemented, resulting in a 3D point generation network. Despite the effectiveness of a 3D point generation network, there are needs for high storage and high computational cost during reconstruction. This paper proposes a new method of single view reconstruction using pruning and template-based point generation network (PGN) given only a single RGB image as the input. The template, which is the encoded structure of the input image, used to guide the point generation process and helps maintain spatial consistency during reconstruction. We propose a 3D template-based PGN followed by network pruning that can reduce a significant number of resources while preserving the reconstruction performance. Experiments on the ShapeNet dataset achieved a 45% reduction of network parameters without sacrificing much Chamfer distance increment, i.e., 0.001238. This study shows that weight pruning on the image encoder layers can improve efficiency without reducing the effectiveness of a 3D point generation network.

1. INTRODUCTION

Single view reconstruction aims to reconstruct 3D structure inferences such as 3D points, meshes, given only a single 2D RGB image. The single view reconstruction covers a wide range of applications, including architectural surveying [1], cultural heritage preservations [2, 3], robotics [4], creation of digital content [5], and both virtual reality (VR) and augmented reality (AR) [3].

In general, approaches to the 3D reconstruction of single images are called classical approaches, which have limitation in the number of required input images. These methods forecast 3D changes in the input images by employing image registration techniques. However, these conventional techniques can lead to mistakes both among different observers and within the same observer [6] because they require certain manual pre-processing steps, such as the manual alignment of landmarks.

In recent years, deep learning-based methods for the 3D reconstruction of single-images problem became more popular largely driven by the accessibility of large datasets like ShapeNet [7] and ModelNet [8]. There is a growing body of literature that proposes end-to-end learning for the 3D reconstruction of single images. Based on the representation of the 3D reconstructed objects, methods that implemented voxel representation are widely used in research. Research in this area comprises 3D-R2N2 [9] and 3D-VAE-GAN [10]. The 3D-R2N2 model consists of a 3D convolutional LSTM network followed by a 3D deconvolutional neural network, capable of producing a 32 × 32 × 32 grid of voxels. At the same time, 3D-VAE-GAN consists of several elements: an image encoder, a decoder that employs a generator from 3D-GAN, and a discriminator responsible for reconstructing a voxel grid measuring 64 × 64 × 64. Recently, a depth fusion approach that combines GAN-based coarse generation with depth-guided diffusion refinement was proposed [11]. The depth fusion approach needs depth map estimation to guide the 3D model refinement.

Additionally, point cloud representation characterizes 3D objects by an unordered collection of points on their surfaces. This approach is more adaptable than using voxel representation. Frontier works that employ point cloud representation for 3D reconstruction include PointNet [12], PointNet++ [13], and AtlasNet [14]. Furthermore, point cloud data have been useful for analyzing the existing environment during the architectural design process. Study by Alkadri et al. [15] investigated the use of point cloud data in constructing the solar envelope during an architectural design process.

Despite its effectiveness in reconstructing 3D points given only a single 2D RGB image, the storage and computational costs of a 3D point reconstruction network are high. Therefore, the applicability of the network, e.g., in embedded systems, autonomous agents, or mobile devices, is limited. The deep architectures are composed of millions of parameters to be trained. Hence, it leads to model over-parameterization, meaning having more parameters than training samples. For example, the AtlasNet model for the task of single view reconstruction (SVR) has approximately 12.8 million model parameters, taking up more than 150MB in storage space to reconstruct 3D points from a single image. Overparameterization plays a crucial role in the effective training of neural networks. However, once a network structure that generalizes well is achieved, pruning becomes essential to minimize redundancy while preserving robust performance [16]. Although much research has been carried out on pruning deep convolutional networks for image classification [17-24], little if any empirical work has been done to investigate network pruning for 3D reconstruction from single images. Recent works that explore pruning in deep learning for 3D tasks, including the studies [25, 26], has demonstrated the benefits of pruning for model efficiency and generalization in 3D tasks, such as 3D ultrasound localization microscopy [25] and 3D point cloud registration [26].

This paper proposes a novel application of pruning methods to reduce the computational cost of well-trained 3D point reconstruction networks. The number of pruned neurons implies the network acceleration due to a reduction in matrix multiplications. The proposed method introduces a global unstructured weight pruning on the image encoder layers and reconstructs 3D point clouds more efficiently.

In the following sections, we outline the structure of this paper. Section 2 delves into various studies that are pertinent to the method we propose. Then, the details of our proposed method are described in Section 3. Section 4 presents the experimental findings using our proposed approach. In Section 5, we provide the conclusion.

2. RELATED WORKS

Neural network pruning is the task of reducing the size of a network by removing either nodes or weight parameters. Following the pruning framework as proposed in the study by Han et al. [27], the pruning technique consists of a three-step training pipeline: (1) train connectivity to convergence, (2) prune connections, and (3) fine-tuning for weight training. Steps (2) and (3) are iteratively performed in N iterations. Step (2) is the most crucial in the pruning framework. The criteria used for pruning should be stable and significantly reduce the computational complexity of deep neural networks [16].

Generally, pruning methods varied primarily in the pruning structure (structured or unstructured), the pruning scoring/criteria, the pruning scheduling (all at once, fixed fraction or according to a more complex function) and the fine-tuning (whether involving fine-tuning or not, if involving fine-tuning, then whether continue or reinitialize training) [28]. For example, a pruning process may serve as a model accelerator applied to an algorithm based on a validation set [29]. A pruning step may also be used with transferability remaining in domain adaptation [30]. Another approach applied the model pruning on the server and further carry out fine-tuning on the clients [31].

Additionally, Hawks et al. [32] introduced a mix of network pruning and network quantization during the training. Also performed during the training, Park et al. [33] proposed hypothesis pruning for selecting the best output in order to maintain the quality of the output. Zhu et al. [21] employed a Squeeze-Excitation-Pruning (SEP) block at the end of their hybrid CNN models for the task of breast cancer image classification. In the study by Shan et al. [34], reinforcement learning (RL) was used to predict pruning strategies based on feedback from the hardware condition.

Recent development for person re-identification exploited pruning and demonstrated that pruning can significantly decrease model complexity besides preserving the accuracy [35]. Moreover, a recent systematic literature review also concluded that pruning can considerably reduce model sizes with little or no degradation in the network's performance. In addition, it was reported that almost all 81 surveyed recent papers employed Top-1 or Top-5 image classification accuracy changes to measure pruning quality. Therefore, we conclude that the application of pruning methods is currently mainly for image classification problems.

3. METHODOLOGY

3.1 Dataset preparation

The single-view-reconstruction framework in our experiment employs an end-to-end learning method. Therefore, we need a dataset with a huge amount of 3D models and the corresponding 2D images of the 3D models. The ShapeNet dataset [7] consists of 3D objects categorized in several classes. We use the subset of the ShapeNet dataset consisting of thirteen categories, divided into two subsets: the train set, and the validation set. Table 1 shows the number of the 3D objects for each category used in the experiment. Note that each category has more than 1,000 unique 3D objects. Then, we use the rendering images of each 3D object as in the study by Choy et al. [9]. Each 3D model will have 24 images rendered from different views. Each image has the resolution of 224 × 224 pixels, whereas each 3D model is represented as 1024 points per object.

3.2 Network architecture and evaluation design

This part outlines the network architecture introduced in this study, specifically the single-view-reconstruction framework, which includes an image encoder and a 3D point decoder, as illustrated in Figure 1.

The performance evaluation of the network is an objective function comparing the 3D result with the 3D ground-truth, utilizing the Chamfer distance as in Eq. (1). Consider a scenario where we define a collection of 3D ground-truth points as S_1 and a collection of 3D reconstructed points as S_2. The Chamfer distance (d_{CD}) between these two sets, S_1 and S_2, is calculated in the following manner: For each point in S_1, determine the smallest distance to any point in S_2, then sum the squares of these distances. Similarly, for each point in S_2, find the smallest distance to any point in S_1 and add up the squares of these distances.

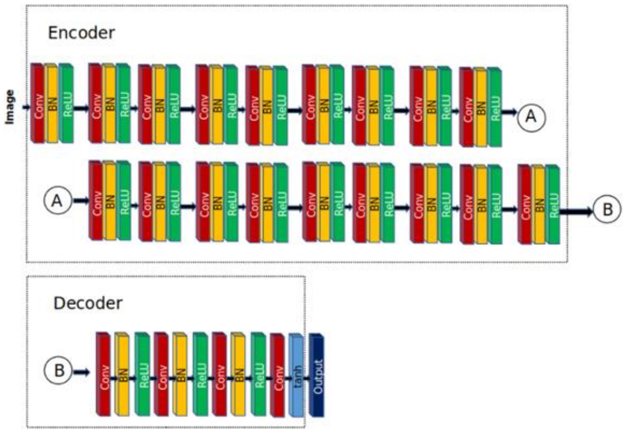


Figure 1. The single-view-reconstruction framework that consists of an image encoder and a 3D point decoder

The Chamfer distance between S_1 and S_2 is obtained by adding the outcomes of the two summations, as shown in Eq. (1):

$$d_{CD}\left(S_1, S_2\right) = \sum_{x \in S_1} \min_{y \in S_2} \| x - y \|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \| x - y \|_2^2 \quad (1)$$

where, x represents points in the point set S_1, and y represents points in the point set S_2.

In addition, after our pruning method is applied, the performance evaluation is characterized by two metrics, i.e., the network quality measured by the Chamfer distance on our validation dataset and the network efficiency measured by the number of parameter reductions.

Table 1. The number of 3D objects for each category within the dataset

Category	Train Set	Validation Set	Total
airplane	3,326	809	4,045
bench	1,452	364	1,816
cabinet	1,257	315	1,572
car	5,996	1,500	7,496
chair	5,422	1,356	6,778
display	876	219	1,095
lamp	1,854	464	2,318
loudspeaker	1,294	324	1,618
rifle	1,897	475	2,372
sofa	2,538	635	3,173
table	6,807	1,702	8,509
telephone	841	211	1,052
vessel	1,551	388	1,939
Total	35,021	8,762	43,783

3.3 The pruning method

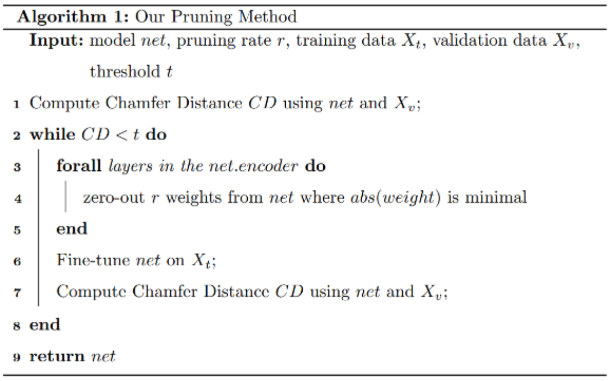
In this part, we introduce a pruning technique for reconstructing 3D models from individual images. The proposed method performs a one-shot channel pruning using the L1-norm of weights for selecting filters because the filters with smaller weights always produce weaker activations. L1-norm was chosen due to its effectiveness in inducing sparsity and simplicity in implementation. We illustrate the proposed method in Algorithm 1 (see Figure 2).

In general, a feedforward neural network comprises neurons that are arranged in a series of layers, with each neuron receiving input from one or more previous layers and propagating its output to every neuron in subsequent layers via a potentially nonlinear mapping. Suppose that we represent neurons in the neural network using weight (W_1, W_2, ...) and bias (b_1, b_2, ...) parameters, then after the neural network is trained using training data, the weight and bias parameters are determined.

The proposed pruning method is as follows. Given the weight parameters of a trained model net, find the unimportant synapse connections (the weight parameters) and set the weights to zero. To find the unimportant weights, we use the magnitude of the weights. This method to prune a network for 3D point reconstruction is simple yet effective. In addition, there is no need for additional data samples after training.

L1-norm pruning is based on the magnitude of individual weights, where weights with smaller absolute values are considered less important and are set to zero. This method is simple and effective, meaning that it does not require additional training data or complex computations. L1-norm naturally encourages sparsity in the network, which is beneficial for reducing model size and computational cost. Since it relies only on the trained weights, it can be applied directly after training without retraining or fine-tuning.

In pruning scenarios, where the goal is to identify and remove unimportant synapse connections in a trained model for 3D reconstruction, L1-norm provides a straightforward way to reduce redundancy while preserving performance. L2-norm, while useful for regularization during training, tends to retain small weights rather than eliminate them, which is less effective for pruning.



```
Algorithm 1: Our Pruning Method
  Input: model $net$, pruning rate $r$, training data $X_t$, validation data $X_v$,
         threshold $t$
Compute Chamfer Distance $CD$ using $net$ and $X_v$;
while $CD < t$ do
    forall layers in the net.encoder do
        zero-out $r$ weights from $net$ where $abs(weight)$ is minimal
    end
    Fine-tune $net$ on $X_t$;
    Compute Chamfer Distance $CD$ using $net$ and $X_v$;
end
return $net$
```

Figure 2. The proposed pruning method

4. RESULTS

This section reports results of the pruning experiment using our proposed method. The encoder of the trained model used in our experiment implements the ResNet-18 architecture. Therefore, the encoder consists of 18 convolutional layers. Firstly, we report the performance of our proposed method using pruning rates of {0%, 5%, 10%, ..., 90%}. Second, we report the visualization of the reconstructed 3D points using a sample input image to assess the performance of our proposed method.

Table 2. Pruning performance shown as the Chamfer distance (CD) multiplied by 1,000 and the number of parameter reduction

Pruning Rate (%)	Chamfer Distance	Parameter Reduction
0	3.762	0
5	3.762	575,958
10	3.758	1,151,917
15	3.760	1,727,875
20	3.770	2,303,834
25	3.808	2,879,792
30	3.847	3,455,750
35	3.975	4,031,709
40	4.210	4,607,667
45	4.784	5,183,626
50	5.286	5,759,584
55	6.475	6,335,542
60	10.294	6,911,501
65	21.603	7,487,459
70	41.328	8,063,418
75	62.070	8,639,376
80	73.955	9,215,334
85	84.396	9,791,293
90	98.978	10,367,251

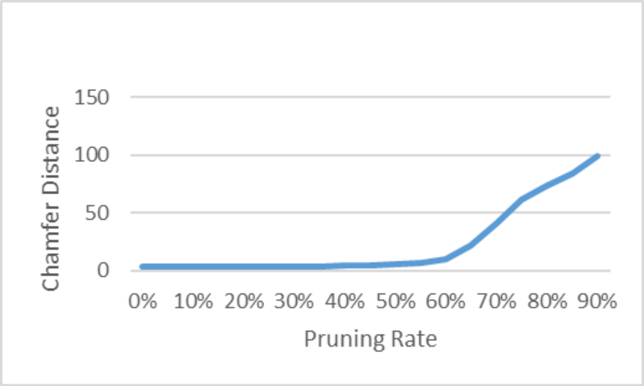


Figure 3. The curve of Chamfer distance versus pruning rate of a sample input image using pruning rates of 10% through 90%

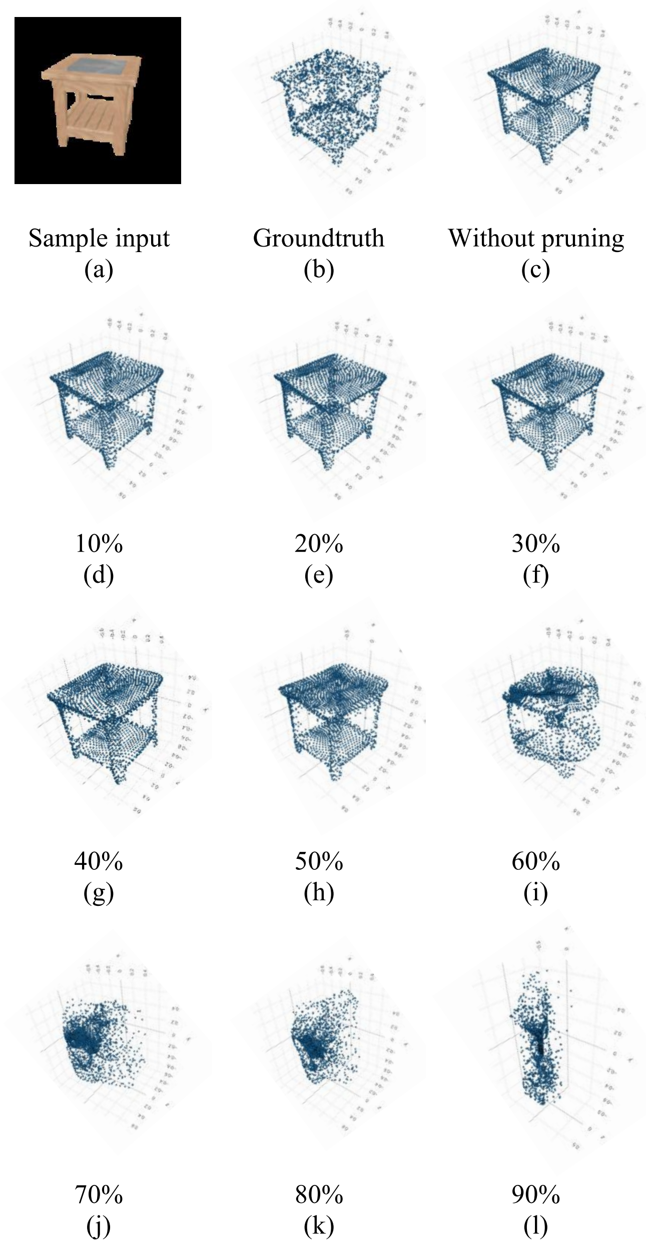

Sample input
(a)
Groundtruth
(b)
Without pruning
(c)
10%
(d)
20%
(e)
30%
(f)
40%
(g)
50%
(h)
60%
(i)
70%
(j)
80%
(k)
90%
(l)

Figure 4. The visualization of a sample input image, the ground truth 3D points, and the reconstructed points without pruning followed by those using pruning rates of 10% through 90%

Table 2 shows the performance evaluation characterized by two metrics, namely the Chamfer distance (CD) on the validation dataset and the number of parameter reductions. The parameter reduction value obtained by our pruning method almost reaches 5.2 million, 45% of the total network parameters while maintaining the Chamfer distance to be below 0.005. The total network parameters are 12.8 million.

Visually, the quality of 3D reconstruction deteriorates noticeably when the Chamfer distance exceeds 0.005, indicating a significant deviation from the ground truth. As shown in Figure 3, we evaluated the reconstruction performance across a range of pruning rates: {0%, 5%, 10%, ..., 90%}. The Chamfer distance between the reconstructed point cloud and the ground truth remains relatively stable up to a pruning rate of 45%. However, starting from 50%, the curve begins to rise sharply. This trend highlights the sensitivity of the model to aggressive pruning and underscores the importance of maintaining a balance between model compression and reconstruction accuracy.

The visualization of the reconstructed 3D points using a sample input image is as shown in Figure 4. Figure 4(a) shows a sample input image, that is, a table. The ground truth points are shown in Figure 4(b). Without any pruning applied to the network, the reconstructed points are shown in Figure 4(c). The corresponding Chamfer distance between points in Figure 4(b) and points in Figure 4(c) was 0.003762. Lastly, the corresponding reconstructed points using the proposed method at pruning rates of {10%, 20%, ..., 90%} are shown in Figure 4(d)-4(l). The quality of the reconstruction degraded significantly after the pruning rate increased by more than 50%.

4.1 Ablation study

To determine whether decoder layers are more crucial to the reconstruction performance, we perform decoder pruning as follows. The decoder of the trained model used in our experiment consists of five convolutional layers, namely: conv1, conv2, conv_list[0], conv_list[1], and last_conv.

Each pruning strategy is applied using pruning rates of (0%, 5%, 10%, ..., 90%) on each layer of the decoder. Figure 5 shows the scatter plot of parameter reduction values and its Chamfer distance using the l1 unstructured method on each decoder layer. From Figure 5, it can be shown that pruning on the conv2 layer (shown as orange dots) of the decoder leads to the best pruning performance.

Furthermore, when we choose a threshold value for the 3D reconstruction performance as 0.005, the highest pruning performance was the reduction of the 367,002 parameters, which was achieved by pruning the conv2 layer with 70% of the layer-wise pruning rate. Then, it was followed by the conv_list[1] layer with 75% layer-wise pruning rate that achieved 196,608 parameter reduction. The subsequent highest reduction was achieved by the conv_list[0] layer with 70% layerwise pruning rate that reduced 183,501 parameters. The first and last convolutional layers have fewer parameters than the other layers. Hence, a high pruning rate will much affect the 3D reconstruction performance. We can observe this from the blue dots in Figure 5 that appear very close to the left side, i.e., the number of parameter reduction is very small.

Next, we perform the same layer-wise pruning method on each encoder layer as follows. Using the unstructured method l1, Figure 6 shows the scatter plot of the number of parameter reductions and its Chamfer distance on individual encoder layers. Several encoder layers have very high parameter reduction while maintaining the Chamfer distance minimum. The least significant reduction is obtained from the first convolutional layer. We may infer this observation as that the first convolutional layer is an important layer for the final reconstruction network. This observation agrees with Voita et al. [36].

Early layers tend to capture low-level features critical for spatial structure, making them more sensitive to pruning. Later layers, which capture higher-level abstractions, show more robustness. This is supported by empirical trends observed in our experiments.

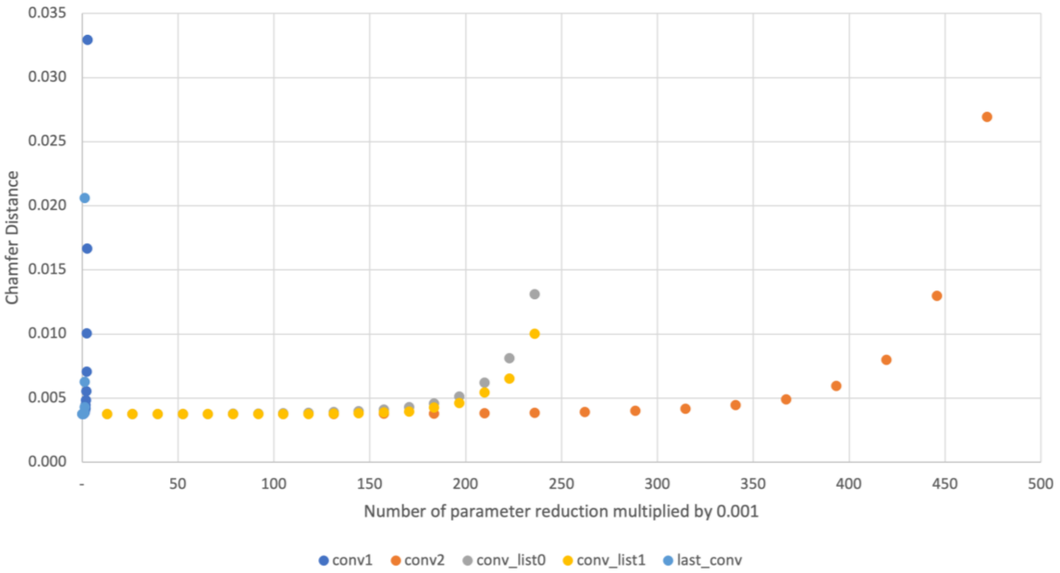


Figure 5. The visualization of the number of parameter reduction and the Chamfer distance for each decoder layer

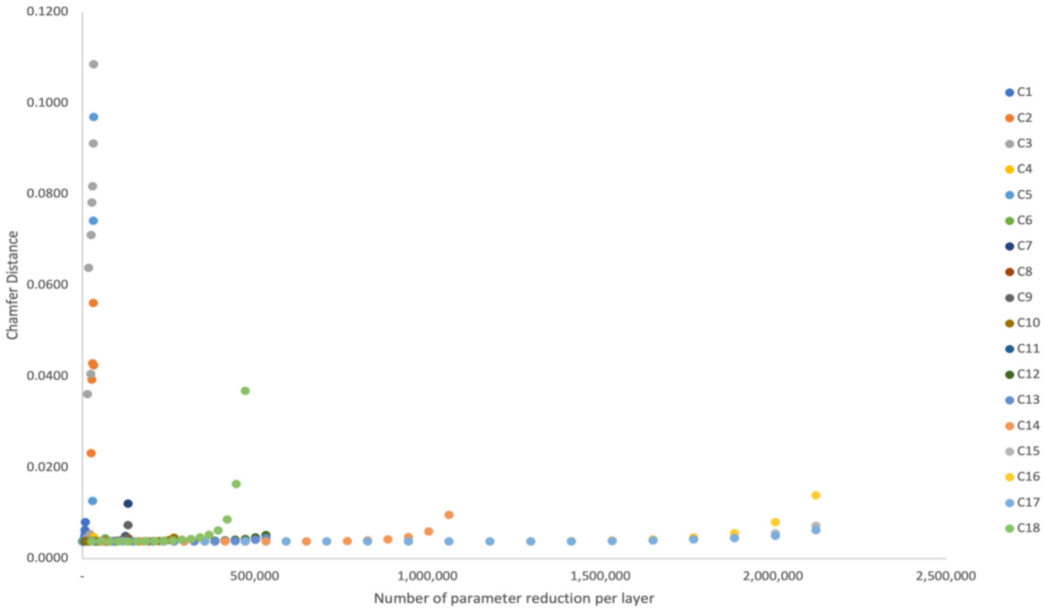


Figure 6. The visualization of the number of parameter reduction and the Chamfer distance for each encoder layer

Moreover, suppose that we set the maximum Chamfer distance allowed for the network pruning performance to be 0.005, then the maximum parameter reduction obtained from each encoder layer is as described in Table 3. For each encoder layer, with reconstruction performance 0.004-0.005, Table 3 shows the number of parameter reductions and the corresponding layer-wise pruning ratio. The convolutional layers C15, C16, and C17 obtain more than 1.5 million parameter reductions. Thus, the tail encoder layers, except the last, are the least important layers in the reconstruction network.

The next experiment scenario is to implement the global pruning rather than the layerwise one. First, we prune globally only on the decoder layers. Table 4 shows the results of global pruning on the decoder layers, without and with the batch-normalization layers. Secondly, we prune globally on both the encoder and decoder layers and the result is as shown in Table 5. From Table 4 and Table 5, suppose that we set the maximum Chamfer distance allowed for the network pruning performance to be 0.005, then the maximum parameter reduction obtained by pruning decoder layers globally is limited, i.e., 263,936. A higher pruning performance was obtained by globally prune both decoder and encoder layers, i.e., 1,885,853 of parameter reduction value. The best pruning performance was reached by our pruning method as in Algorithm 1. Therefore, the decoder layers were not pruned.

This approach can reduce a significant number of parameters, that is, 5,183,626, as shown in Table 2.

Table 3. The number of parameter reductions obtained by pruning an encoder layer using a layer-wise pruning rate

Encoder Layer	Chamfer Distance	Parameter Reduction	Pruning Rate
C1	0.0047	7,056	75%
C2	0.0047	23,962	65%
C3	0.0038	20,275	55%
C4	0.0048	33,178	90%
C5	0.0040	27,648	75%
C6	0.0044	66,355	90%
C7	0.0050	125,338	85%
C8	0.0045	132,710	90%
C9	0.0047	132,710	85%
C10	0.0046	265,421	90%
C11	0.0046	530,842	90%
C12	0.0046	501,350	85%
C13	0.0045	530,842	90%
C14	0.0047	943,718	80%
C15	0.0047	1,887,437	80%
C16	0.0046	1,769,472	75%
C17	0.0045	1,887,437	80%
C18	0.0047	340,787	65%

Table 4. Pruning performance shown as the Chamfer distance (CD) multiplied by 1,000 and the number of parameter reduction that was applied on the five decoder layers, excluding and including the batch normalization (BN) layers

	Without BN Layers		With BN Layers	
Pruning Rate (%)	Chamfer Distance	Parameter Reduction	Chamfer Distance	Parameter Reduction
0	3.762	0	3.762	0
5	3.805	52,659	3.805	52,787
10	3.850	105,318	3.850	105,574
15	3.854	157,978	3.854	158,362
20	4.100	210,637	4.100	211,149
25	4.399	263,296	4.401	263,936
30	5.686	315,955	6.092	316,723
35	7.581	368,614	7.880	369,510
40	20.355	421,274	21.164	422,298
45	67.206	473,933	68.643	475,085
50	98.520	526,592	100.457	527,872
55	148.674	579,251	154.332	580,659
60	221.365	631,910	222.362	633,446
65	251.538	684,570	251.268	686,234
70	268.826	737,229	268.821	739,021
75	287.101	789,888	288.732	791,808
80	301.928	842,547	304.969	844,595
85	326.120	895,206	328.181	897,382
90	348.817	947,866	343.447	950,170

4.2 Comparison

This section reports the comparison between the proposed pruning method and two other pruning methods, namely by randomly choosing the weight parameters in the encoder and choosing the minimum weight parameters across the encoder and decoder. Network pruning that is applied to both encoder and decoder networks has been used in a depth estimation design [37].

Figure 7 shows the quality performance measured in Chamfer distance at pruning rates of (0, 0.005, 0.01, …, 0.9) obtained by the proposed (blue), random (orange) and alternative (gray) methods. Our proposed method successfully maintains the reconstruction quality shown by almost flat blue lines up to a pruning rate of 55%. The two other methods only reach a 15%-20% pruning rate before they drop in the quality of the reconstruction.

Table 5. Pruning performance shown as the Chamfer distance (CD) multiplied by 1,000 and the number of parameter reduction that was applied on the five decoder layers, excluding and including the batch normalization (BN) layers

Pruning Rate (%)	Chamfer Distance	Parameter Reduction
0	3.762	0
5	3.851	628,618
10	3.863	1,257,235
15	4.403	1,885,853
20	6.136	2,514,470
25	9.526	3,143,088
30	28.159	3,771,706
35	79.057	4,400,323
40	101.994	5,028,941
45	178.876	5,657,558
50	203.963	6,286,176
55	235.446	6,914,794
60	262.055	7,543,411
65	292.197	8,172,029
70	310.637	8,800,646
75	324.054	9,429,264
80	334.895	10,057,882
85	360.974	10,686,499
90	360.974	11,315,117

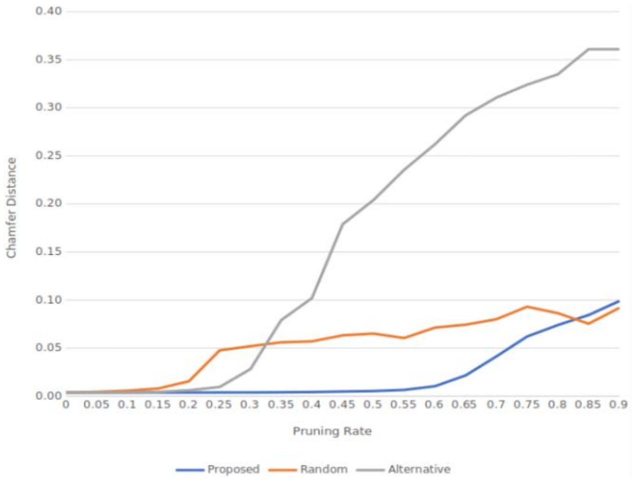


Figure 7. The reconstruction performance shown by Chamfer distance (lower better) after network pruning using several pruning methods at various pruning rate

5. CONCLUSIONS

We have presented a new application of neural network pruning for a single view reconstruction problem. The proposed method achieved a 45% reduction in network parameters without sacrificing much Chamfer distance increment, that is, 0.001238. This study shows that weight pruning on the image encoder layers can improve the efficiency of a 3D point reconstruction network without reducing the effectiveness of the network.

REFERENCES

[1] Wang, Q., Kim, M.K. (2019). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. Advanced engineering informatics, 39: 306-319. https://doi.org/10.1016/j.aei.2019.02.007

[2] Kaldeli, E., Menis-Mastromichalakis, O., Bekiaris, S., Ralli, M., Tzouvaras, V., Stamou, G. (2021). CrowdHeritage: Crowdsourcing for improving the quality of cultural heritage metadata. Information, 12(02): 64. https://doi.org/10.3390/info12020064

[3] Van Nguyen, S., Le, S.T., Tran, M.K., Tran, H.M. (2022). Reconstruction of 3D digital heritage objects for VR and AR applications. Journal of Information and Telecommunication, 6(3): 254-269. https://doi.org/10.1080/24751839.2021.2008133

[4] Kumra, S., Joshi, S., Sahin, F. (2020). Antipodal robotic grasping using generative residual convolutional neural network. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vegas, NV, USA, pp. 9626-9633. https://doi.org/10.1109/IROS45743.2020.9340777

[5] Chen, C.W., Hu, M.C., Chu, W.T., Chen, J.C. (2021). A real-time sculpting and terrain generation system for interactive content creation. IEEE Access, 9: 114914-114928. https://doi.org/10.1109/ACCESS.2021.3105417

[6] Hou, B., Khanal, B., Alansary, A., McDonagh, S., et al. (2018). 3-D reconstruction in canonical co-ordinate space from arbitrarily oriented 2-D images. IEEE transactions on medical imaging, 37(8): 1737-1750. https://doi.org/10.1109/TMI.2018.2798801

[7] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., et al. (2015). Shapenet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012. https://doi.org/10.48550/arXiv.1512.03012

[8] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J. (2015). 3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, USA, pp. 1912-1920. https://doi.org/10.1109/CVPR.2015.7298801

[9] Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S. (2016). 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In European conference on computer vision, pp. 628-644. https://doi.org/10.1007/978-3-319-46484-8_38

[10] Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. Advances in Neural Information Processing Systems, 29. https://doi.org/10.48550/arXiv.1610.07584

[11] Saadi, S., Nini, B., Kada, B. (2025). DepthFusion: A depth-guided framework combining GAN and diffusion for high-fidelity 3D reconstruction from single images. Ingénierie des Systèmes d'Information, 30(8): 2157-2163. https://doi.org/10.18280/isi.300821

[12] Qi, C.R., Su, H., Mo, K., Guibas, L.J. (2017). Pointnet: Deep learning on point sets for 3D classification and segmentation. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, pp. 652-660. https://doi.org/10.1109/CVPR.2017.16

[13] Qi, C.R., Yi, L., Su, H., Guibas, L.J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, California, USA, pp. 5105-5114. http://dl.acm.org/citation.cfm?id=3295222.3295263.

[14] Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M. (2018). A papier-mâché approach to learning 3D surface generation. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, pp. 216-224. https://doi.org/10.1109/CVPR.2018.00030

[15] Alkadri, M.F., De Luca, F., Turrin, M., Sariyildiz, S. (2020). An integrated approach to subtractive solar envelopes based on attribute information from point cloud data. Renewable and Sustainable Energy Reviews, 123: 109742. https://doi.org/10.1016/j.rser.2020.109742

[16] Yeom, S.K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.R., Samek, W. (2021). Pruning by explaining: A novel criterion for deep neural network pruning. Pattern Recognition, 115: 107899. https://doi.org/10.1016/j.patcog.2021.107899

[17] Li, G., Xu, G. (2021). Providing clear pruning threshold: A novel CNN pruning method via L0 regularisation. IET Image Processing, 15(2): 405-418. https://doi.org/10.1049/ipr2.12030

[18] Wang, J., Li, G., Zhang, W. (2021). Combine-net: An improved filter pruning algorithm. Information, 12(7)L: 264. https://doi.org/10.3390/info12070264

[19] Galchonkov, O., Nevrev, A., Glava, M., Babych, M. (2020). Exploring the efficiency of the combined application of connection pruning and source data preprocessing when training a multilayer perceptron. Eastern-European Journal of Enterprise Technologies, 2(9): 104. https://doi.org/10.15587/1729-4061.2020.200819

[20] Zhang, S., Wu, G., Gu, J., Han, J. (2020). Pruning convolutional neural networks with an attention mechanism for remote sensing image classification. Electronics, 9(8): 1209. https://doi.org/10.3390/electronics9081209

[21] Zhu, C., Song, F., Wang, Y., Dong, H., Guo, Y., Liu, J. (2019). Breast cancer histopathology image classification through assembling multiple compact CNNs. BMC Medical Informatics and Decision Making, 19(1): 198. https://doi.org/10.1186/s12911-019-0913-x

[22] Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J. (2017). Pruning convolutional neural networks for resource efficient inference. In 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings.

[23] Sun, X., Ren, X., Ma, S., Wang, H. (2017). meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In International Conference on Machine Learning, pp. 3299-3308.

[24] Li, H., Samet, H., Kadav, A., Durdanovic, I., Graf, H.P. (2017). Pruning filters for efficient convnets. In 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings.

[25] Rauby, B., Xing, P., Porée, J., Gasse, M., Provost, J. (2025). Pruning sparse tensor neural networks enables deep learning for 3D ultrasound localization microscopy. IEEE Transactions on Image Processing, 34: 2367-2378. https://doi.org/10.1109/TIP.2025.3552198

[26] Wang, J., Li, Z. (2024). 3DPCP-Net: A lightweight progressive 3D correspondence pruning network for accurate and efficient point cloud registration. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, VIC, Australia, pp. 1885-1894. https://doi.org/10.1145/3664647.3681320

[27] Han, S., Pool, J., Tran, J., Dally, W. (2015). Learning both weights and connections for efficient neural network. Advances in neural information processing systems, 28.

[28] Blalock, D., Ortiz, J.J.G., Frankle, J., Guttag, J. (2020). What is the state of neural network pruning? arXiv preprint arXiv:2003.03033. https://doi.org/10.48550/arXiv.2003.03033

[29] Shao, R., He, H., Chen, Z., Liu, H., Liu, D. (2020). Stochastic channel-based federated learning with neural network pruning for medical data privacy preservation: Model development and experimental validation. JMIR Formative Research, 4(12): e17265. https://doi.org/10.2196/17265

[30] Zhang, X., Huang, W., Gao, J., Wang, D., Bai, C., Chen, Z. (2021). Deep sparse transfer learning for remote smart tongue diagnosis. Mathematical Biosciences and Engineering, 18(2): 1169-1186. https://doi.org/10.3934/mbe.2021063

[31] Imteaj, A., Amini, M.H. (2021). FedPARL: Client activity and resource-oriented lightweight federated learning model for resource-constrained heterogeneous IoT environment. Frontiers in Communications and Networks, 2: 657653. https://doi.org/10.3389/frcmn.2021.657653

[32] Hawks, B., Duarte, J., Fraser, N.J., Pappalardo, A., Tran, N., Umuroglu, Y. (2021). Ps and Qs: Quantization-aware pruning for efficient low latency neural network inference. Frontiers in Artificial Intelligence, 4: 676564. https://doi.org/10.3389/frai.2021.676564

[33] Park, Y., Park, W.S., Kim, Y.B. (2021). Anomaly detection in particulate matter sensor using hypothesis pruning generative adversarial network. ETRI Journal, 43(3): 511-523. https://doi.org/10.4218/etrij.2020-0052

[34] Shan, N., Ye, Z., Cui, X. (2020). Collaborative intelligence: Accelerating deep neural network inference via device-edge synergy. Security and Communication Networks, 2020(1): 8831341. https://doi.org/10.1155/2020/8831341

[35] Masson, H., Bhuiyan, A., Nguyen-Meidine, L.T., Javan, M., Siva, P., Ayed, I.B., Granger, E. (2021). Exploiting prunability for person re-identification. EURASIP Journal on Image and Video Processing, 2021(1): 22. https://doi.org/10.1186/s13640-021-00562-6

[36] Voita, E., Talbot, D., Moiseev, F., Sennrich, R., Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 5797-5808. https://doi.org/10.18653/v1/p19-1580

[37] Wofk, D., Ma, F., Yang, T.J., Karaman, S., Sze, V. (2019). Fastdepth: Fast monocular depth estimation on embedded systems. In 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, pp. 6101-6108. https://doi.org/10.1109/ICRA.2019.8794182