İSTANBUL UNIVERSITY CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2025; 25: 1-14

Three-Dimensional Model Generation from Two-Dimensional Image Sequences Using Machine Learning

Mustafa Dağtekin, Batuhan Aşıroğlu, Özgür Can Turna

Department of Computer Engineering, İstanbul University-Cerrahpaşa, Faculty of Engineering, İstanbul, Türkiye

Cite this article as: M. Dağtekin, B. Aşıroğlu and Ö. C. Turna, "Three-dimensional model generation from two-dimensional image sequences using machine learning," *Electrica*, 25, 0014, 2025. doi: 10.5152/electrica.2025.25014.

WHAT IS ALREADY KNOWN ON THIS TOPIC?

- *3D models are used in various fields, including digital twins, game development, simulation, the Metaverse, industrial automation, robotic systems, medical imaging, and cartography.*
- *Deep learning methods are effective for pattern-recognition tasks.*
- *3D reconstruction uses 2D images to create 3D models.*
- *Both single and multiple 2D images are used for 3D reconstruction.*
- *Common 3D model representations include point clouds and voxels.*

WHAT THIS STUDY ADDS ON THIS TOPIC?

- *A deep learning model generates voxel-represented 3D models from 2D silhouette images.*

Corresponding Author:
Özgür Can Turna

E-mail:
ozgurcan.turna@iuc.edu.tr

Received: January 24, 2025
Revision requested: January 27, 2025
Last revision received: January 28, 2025
Accepted: February 4, 2025
Publication Date: April 25, 2025

DOI: 10.5152/electrica.2025.25014

ABSTRACT

In this study, a deep learning model was utilized to generate voxel-represented three-dimensional models of some objects using silhouette images of size 128 × 128 captured from four different angles. The proposed model is trained using the ShapeNet dataset. The deep learning model, along with the proposed error function, has been favored to reduce the number of parameters and capture features of different dimensions. A total of 34 691 different data were obtained in seven categories. The performance metrics of the proposed model have been compared with other studies in the literature using the Intersection over Union (IoU) metric. The comparison reveals that the proposed method achieves an IoU score of 0.5283, which outperforms both the 1 image and 5 image input versions of both McRecon and (Perspective Transformer Nets) PTN methods in categories other than the chair category.

Index Terms— Deep learning, machine learning, three-dimensional (3D) reconstruction

I. INTRODUCTION

Currently, three-dimensional (3D) models are used in areas including but not limited to digital twins, game development, simulation, and the Metaverse [1, 2]. Beyond entertainment and virtual realms, these models play a crucial role in the development of industrial automation and robotic systems [2, 3], in medical imaging for better diagnoses and planning surgical procedures [4], and in the cartography industry to generate more realistic and accurate maps [5]. Considering all these applications, it has become beneficial to create 3D representations of real-world objects.

With the current advancements in computational power, using artificial intelligence algorithms is becoming increasingly practical. In particular, deep learning methods have proven to be highly effective in handling intricate tasks involving patterns, such as language translation, object recognition, and object detection. The advancement in deep learning is directly proportional to the availability of big data. This is because the structures of deep learning models inherently involve millions and, in some cases, billions of parameters. The process of fine-tuning these extensive numbers of parameters is typically carried out using a substantial amount of data. (Visual Geometry Group) VGG [6] and residual network (ResNet) [7] are among the most widely used deep learning models developed for object recognition problems, and they have been shown to be capable of addressing different problems, such as object detection and segmentation, using the transfer learning method [8]. These models can serve as feature extraction layers for new deep learning models.

Three-dimensional reconstruction is the process of creating a 3D model of an object or scene using a series of 2D images or measurements. Manually developing 3D computer models is time-consuming. Although novel methods have emerged for generating 3D models from 2D photographs of real-world objects using modern computer methods, producing 3D models using 2D photographs remains a significant challenge. Today, researchers employ both computer vision and deep learning-based approaches [9–13] to generate 3D models from 2D images.

- *The model uses an Encoder-Decoder structure with Inception and ResNet modules.*
- *A Categorical Mean Cross Entropy (CMCE) error function is proposed.*
- *The model's performance is evaluated using the Intersection over Union (IoU) metric.*
- *The model demonstrates improved 3D model generation across several object categories.*

One of the metrics to measure the quality of 3D reconstruction is called "IoU" or "Intersection over Union." The IoU compares the predicted 3D object's shape to the ground truth shape. A higher IoU value (closer to 1) indicates better accuracy, as it shows that the prediction overlaps well with the actual 3D shape. An IoU of 1 means perfect overlap, while an IoU of 0 means no overlap at all. The state of the art has quite a bit of room for improvement, as achieving high IoU values is still a challenge in many 3D reconstruction tasks. Factors such as noise in input data, complex object geometries, partial occlusions, and inaccuracies in the reconstruction algorithm can lead to lower IoU values. In this work, we aimed to improve the IoU by developing a more robust 3D reconstruction algorithm that addresses some of these common challenges.

A significant amount of research has been done to create 3D models from 2D images using deep learning. Some use a single image as input [9, 12, 14–17] while others utilize multiple images as inputs. These studies were typically conducted using categorically supervised methods. Additionally, the McRecon method proposed by Gwak et al. [15] along with the PTN method suggested by Yan et al. [16] successfully generated categorical, supervised voxel-based 3D models using the ShapeNet [18] dataset.

In the studies conducted so far, two primary forms of representation have been favored in deep learning-based approaches to 3D modeling: point cloud representation, as seen in Fig. 1, and voxel representation, as seen in Fig. 2. Additionally, researchers have also employed octree representation, as seen in Fig. 3, which indicates the coordinates of 3D models using octrees. An octree is a tree data structure used to partition a 3D space into smaller regions, often for efficient storage, retrieval, or processing of spatial data. Also, mesh representation, seen in Fig. 4, is used to represent 3D shapes, which creates 3D models using polygons. However, point cloud and voxel representations are generally preferred over mesh and octree representations. Moreover, there is a larger dataset available for point cloud and voxel representations than for mesh and octree representations.

A. *Voxel Representation*

A voxel, short for volumetric pixel or volume element, is a unit of representation in a 3D space, analogous to a 2D pixel in an image. Each voxel represents a value on a regular grid in 3D space and typically corresponds to a small cube or rectangular prism. Voxel-based 3D model representation of a 3D image can be seen in Fig. 2. Voxel-based 3D model representation is

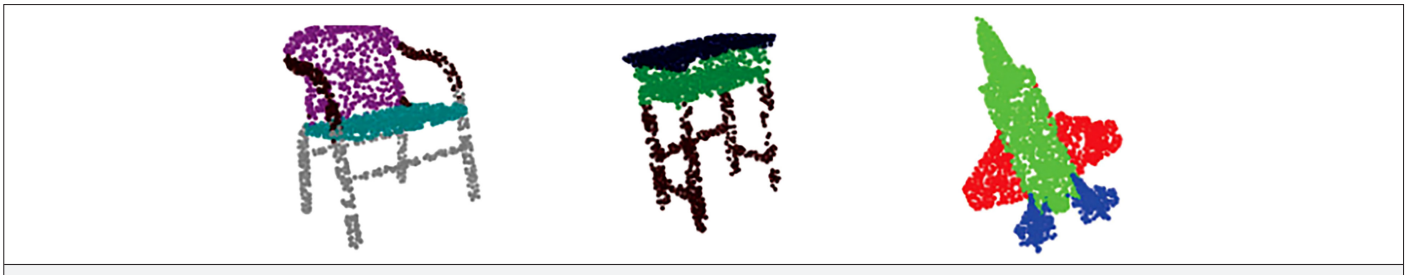


Fig. 1. Point cloud representation. This figure illustrates how 3D models can be represented as a set of points in space.

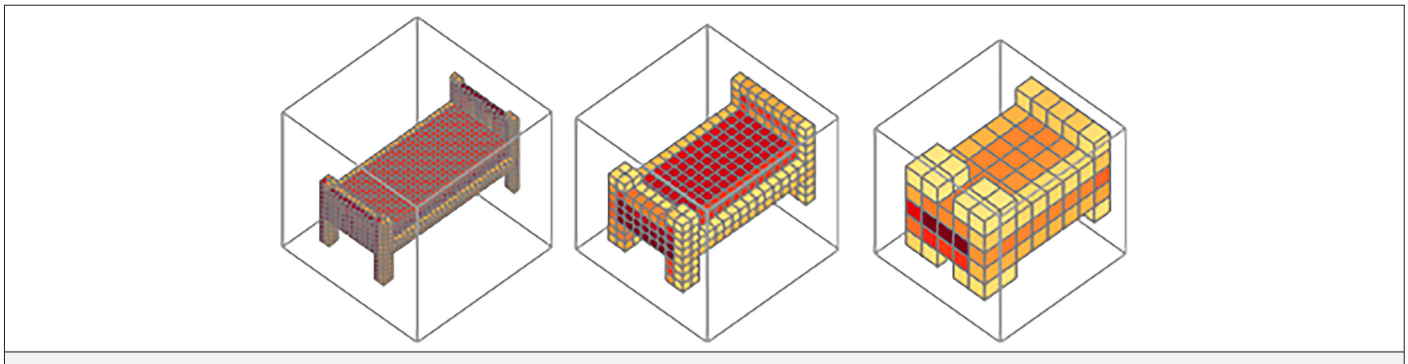


Fig. 2. Mesh representation. This figure shows a 3D model created using polygons.

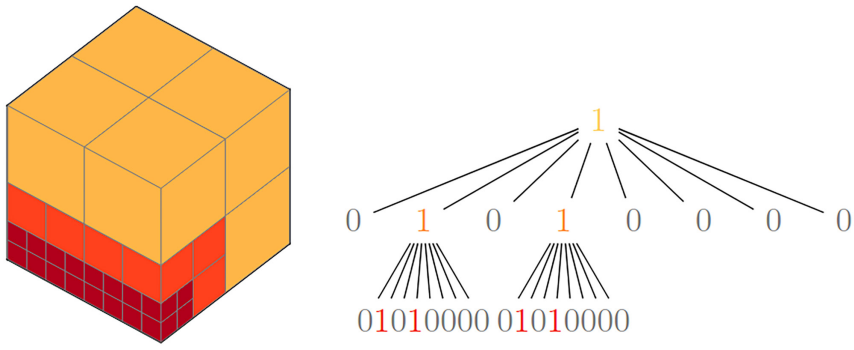


Fig. 3. This figure depicts how octrees can be used to represent the coordinates of three-dimensional models.

characterized by representing models with more data compared to other forms of representation [19]. In addition, voxels represent points in 3D graphics, and each voxel can be independently modified and managed.

For this representation, Chang et al. [18] have created the ShapeNet and ShapeNetCore datasets. The ShapeNet dataset contains a total of 43 784 data points across 14 categories, while the ShapeNetCore dataset, a subset of ShapeNet, includes 51 300 3D models across 55 categories. In the ShapeNet dataset, there are 137 × 137-sized RGB images rendered from 24 different perspectives for each 3D voxel model. For this work, the proposed deep learning model is trained using the ShapeNet dataset.

B. *Residual Network Approach*

In deep learning, the training process encounters a degradation problem when establishing very deep architectures. As the network deepens, it is expected that, if trained with a sufficient amount of data, the performance of a shallower network would be lower than that of a deeper one. However, in reality, due to the degradation problem, a shallower network can exhibit higher performance compared to a deeper one as the network depth increases. To address this issue, He et al. [7] proposed the ResNet approach. In the ResNet approach, a layer establishes a direct connection by skipping intermediate layers (e.g., connecting the fifth layer directly with the third layer) with the layers that precede it. This process prevents the degradation problem and enables deep networks to operate more efficiently.

C. *Network in Network Approach and Inception Module*

In image classification problems, the area covered by pixels belonging to a class in the input data can vary, with some inputs having more coverage than others. Therefore, adjusting the kernel size of filters optimally during the convolution operation is crucial for feature extraction, considering the difference in the area covered by class-related information in the input data. For large features, a larger kernel size should be used, while for smaller features, convolution should be performed with filters having a smaller kernel size. However, filters with large kernel sizes can be computationally expensive. Lin et al. [20] proposed reducing the computational cost of large-sized kernel filters by first applying convolution with $1 \times 1 \times n$ kernel filters, where n is the number of filters, and then using $k \times 1 \times m$ kernel filters. This approach allows the input channels to reach the desired dimensions with lower computational costs through the initial 1×1 kernel convolution operation. On the other hand, Krizhevsky et al. [9] introduced the Inception module, which involves performing convolution with 1×1 kernel filters followed by using filters of different sizes at the same level for feature extraction. This enables feature extraction with multiple kernel sizes at the same layer level and leads to a significant reduction in the number of parameters. The first version of the Inception module is depicted in Fig. 5.

This study describes the creation of 3D models represented by voxels using deep learning, utilizing 2D images of size 128 × 128 captured from four different perspectives. This paper is organized as follows: Section II provides information about the dataset used in the study and describes the deep learning model developed for 3D reconstruction; Section III presents the results of the study; and Section IV concludes the study.

II. MATERIALS AND METHODS

A. *Dataset*

We used the ShapeNet dataset [18] for training the deep learning model developed for 3D reconstruction. The ShapeNet dataset includes 24 RGB images of size 137 × 137 taken from 24 different angles for each data point, along with corresponding voxel-based

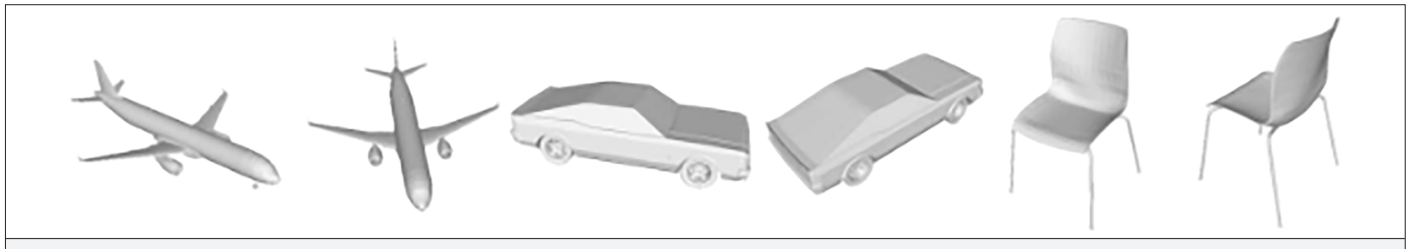


Fig. 4. Three-dimensional model represented using voxels, which are essentially 3D pixels. Voxels are represented as cubes and can be modified individually.

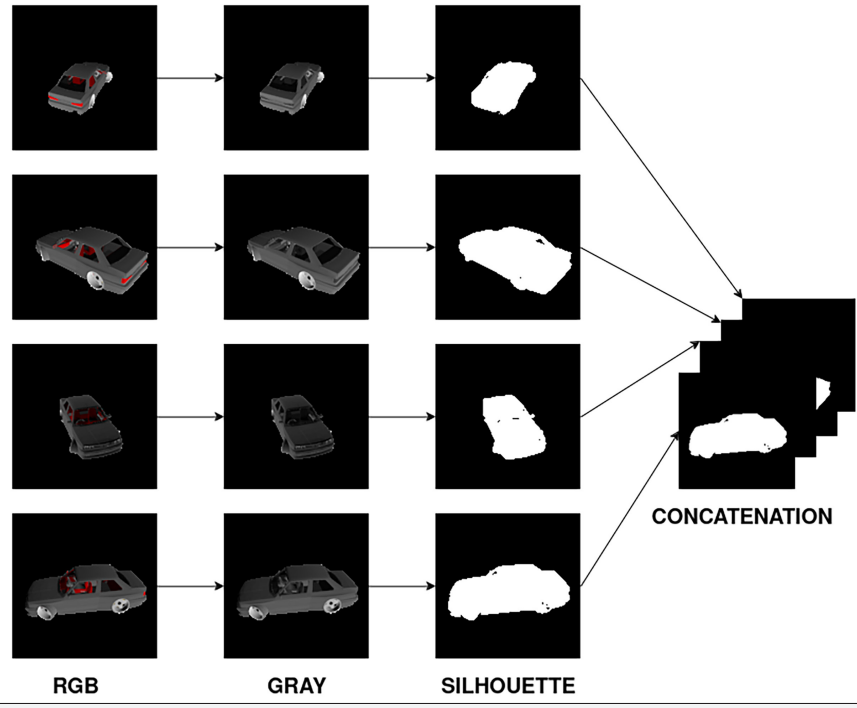


Fig. 5. Preprocessing steps. This figure illustrates the steps used to prepare two-dimensional images for input into the three-dimensionalreconstruction model, including conversion to grayscale, generating silhouette images, resizing, stacking, and normalization.

3D models. A voxel-based 3D model from the ShapeNet dataset is illustrated in Fig. 6, and the set of 24 2D RGB images corresponding to this 3D model is shown in Fig. 7.

During the preparation of the training data, only images captured from four angles out of the 24 angles available in the dataset were used, along with the corresponding voxel-based 3D models. The set of four images used for training data was selected sequentially, starting at a random sample, from the 24 images captured from different angles. The main reasons for selecting four angles out of the 24 angles are, first, our method is independent of the category (plane, car, etc.) and, second, most details on the objects in the ShapeNet dataset can be represented with at least four images.

B. *Preprocessing*

For the selected training dataset, each image captured from four different angles, which is sufficient to represent the objects, was first converted from RGB to grayscale using the International ITU-R BT.601-7 (03/2011) standard given in (1):

$$Grayscale\left(R,G,B\right)=0.299\times R+0.587\times G+0.114\times B \quad (1)$$

In (1), *R*, *G*, and *B* are the pixel values of the Red, Green, and Blue channels, respectively. After this, silhouette images of the pictures were obtained using the thresholding method on the grayscale-converted images. For every image, the threshold value is chosen as the mean intensity value of that image. Subsequently, the 137 × 137-sized images were reduced to 128 × 128 in dimension. The reason for this is that the pretrained models we used only work with images sized 128 × 128. Finally, the silhouette images of the pictures taken from four different angles were stacked on top of each other as different channels. The resulting image has a size of 128 × 128 × 4. The data preprocessing steps are illustrated in Fig. 8. The final image was normalized by dividing it by 255, bringing the pixel values of the images from the range of 0–255 to the range of 0–1. After this process, the data preprocessing steps are finished.

C. *3D Reconstruction*

For the deep learning model for 3D reconstruction, three subnetwork modules were developed: Inception DownSampling 2D (Fig. 9), Inception ResNet 2D (Fig. 10), and Inception UpSampling 3D (Fig. 11).

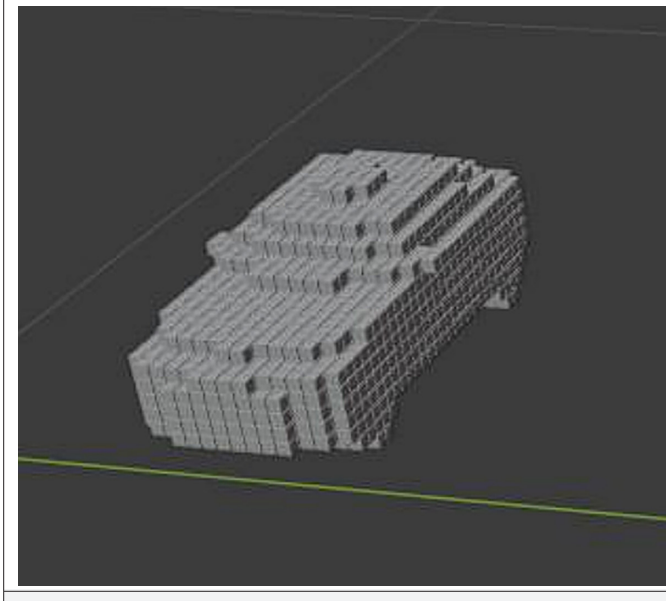


Fig. 6. This figure shows an example of a voxel-based three-dimensional model from the ShapeNet dataset, used for training in this study.

Fig. 7. A set of 24 two-dimensional RGB images corresponding to a three-dimensional model. This figure displays the 24 different two-dimensional images of a three-dimensional model in the ShapeNet dataset, taken from 24 different angles. The study uses four of these images as input for its three-dimensional reconstruction model.

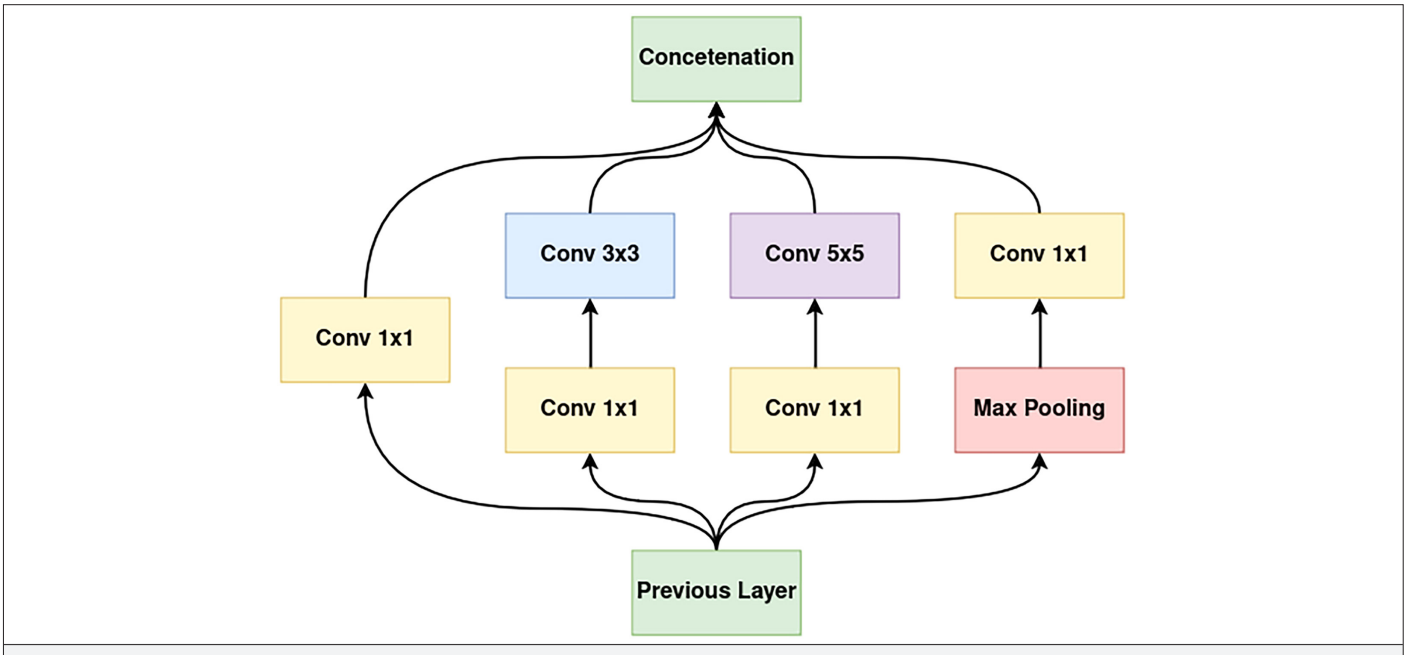


Fig. 8. Inception V1 module, which uses different filter sizes at the same level for feature extraction in a neural network.

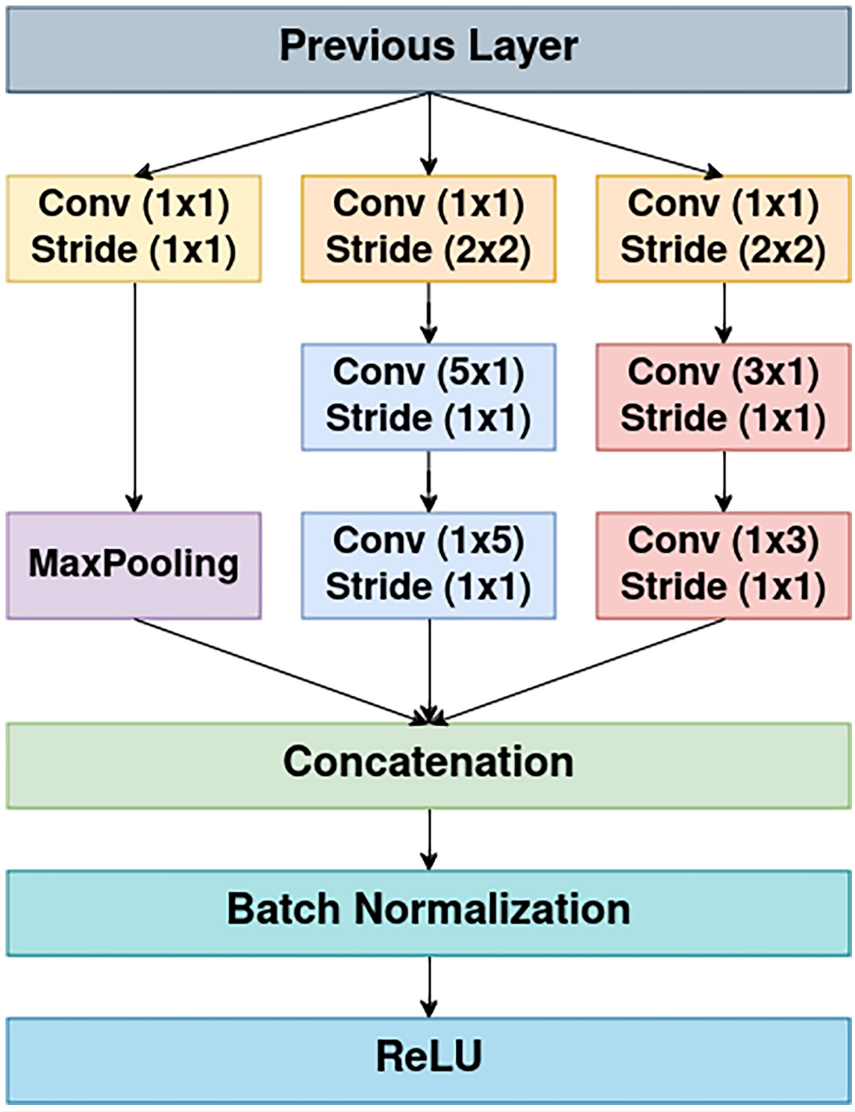


Fig. 9. Proposed model. This figure presents the overall architecture of the deep learning model developed for three-dimensional reconstruction in this study. The model utilizes an Encoder–Decoder approach with Inception and ResNet module.

In the proposed deep neural network model shown in Fig. 12, an Encoder–Decoder approach has been adopted to enhance the prominent features in 2D images and generate coordinate data in the third dimension (3D reconstruction) using the accentuated features. Moreover, within the submodules of the proposed deep learning model, the inception approach has been favored to reduce the number of parameters and capture features of different dimensions. Additionally, to mitigate the issue of gradient vanishing (degradation) in deep networks, the ResNet approach has been used in the "Encoder" part of the network. In the "Decoder" part of the network, the transposed convolution operation has been preferred for the UpSampling process. The subnetwork modules developed for 3D reconstruction in the subject of this study, along with the proposed deep learning model, contain a total of 2 529 282 parameters. The training of the developed deep learning model used categories from the ShapeNet [18] dataset, each containing 2000 or more data. Following this selection process, a total of 34 691 different data were obtained across seven categories. About 60% (20 814) of the collected data were used for training, 20% (6938) for validation, and the remaining 20% (6939) were used as test data. The created dataset was used in a categorically unsupervised manner.

After preparation of the dataset, the proposed deep neural network model was trained with a batch size of 8 for 60 epochs. For training, the RMSprop algorithm with a 10^{-4} learning rate was used for the optimizer.

There are existing studies in the realm of deep learning that address the problem of 3D reconstruction using both single-view images [12, 14–16] and multiple-view images [15, 16]. However, these studies have been developed using categorically supervised methods. Additionally, proposed methods such as McRecon by Gwak et al. [15] and PTN by Yan et al. [16] can generate voxel-based 3D model outputs using the ShapeNet dataset. Both PTN and McRecon methods have versions that use either five images or one image as input. In both PTN and McRecon methods, versions that use five images

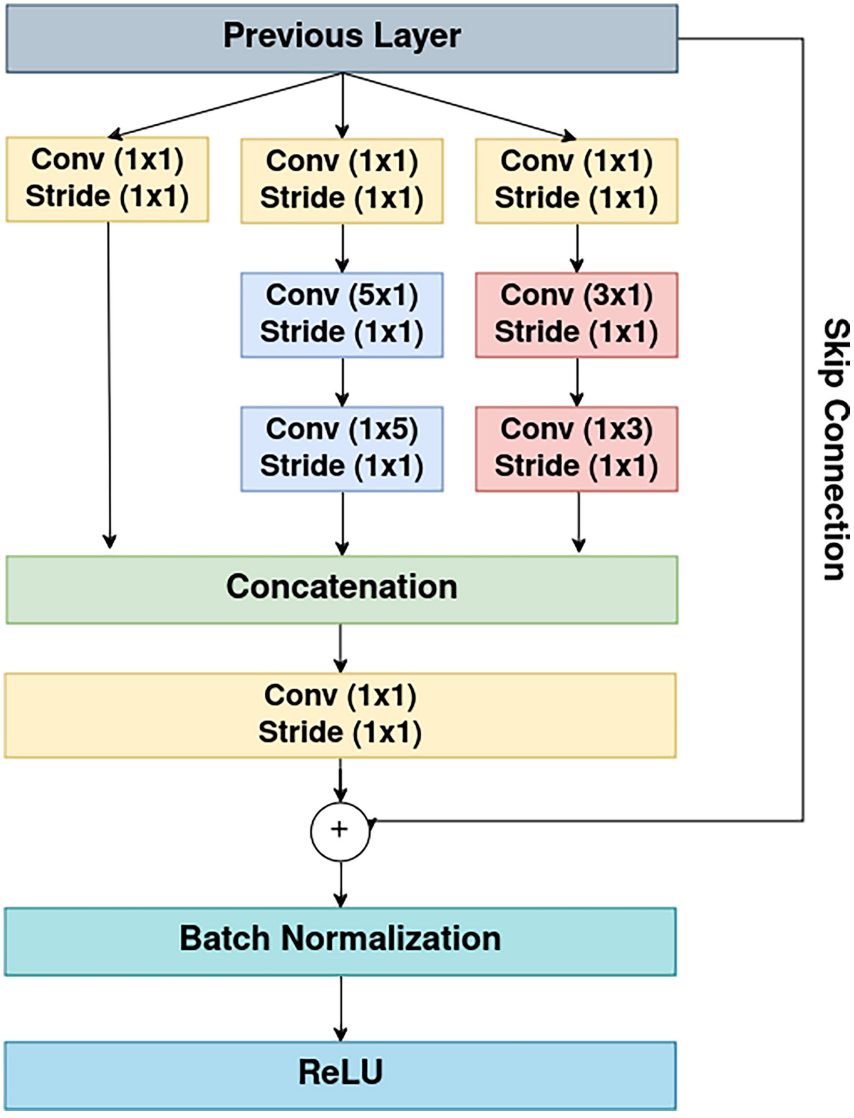


Fig. 10. Inception DownSampling two-dimensional module. This figure details the subnetwork module that extracts features by reducing the dimensions of the two-dimensional inputs using convolution and max pooling.

as input tend to produce better outputs compared to versions that use one image as input. The success of the generated outputs is discussed in detail in Section IV.

D. *Inception DownSampling Two-Dimensional Module*

The Inception DownSampling 2D network module Fig. 9 has been developed to extract features by reducing the dimensions of the height and width axes of 2D inputs and increasing the number of channels. In the Inception DownSampling 2D network module, initially, as proposed in the original Inception method in the study by ref. [21], a 1×1 kernel convolution operation is applied, followed by convolution and Max Pooling layers using 3×1 and 1×3, 5×1, and 1×5 kernels at the same layer levels.

In the Inception DownSampling 2D network module, both Max Pooling layers and 1×1 kernel convolution operations with a 2×2 stride have been employed to achieve DownSampling. Subsequently, the obtained filters are concatenated and sequentially pass through batch normalization and ReLU activation function layers. In each Inception DownSampling 2D network module, the goal is to reduce the dimensions of the input from the previous layer by 2×2, with minimal parameters and the maximum number of filters in various kernel sizes.

In the output of the Inception DownSampling 2D network module, both the dimensions of the input from the previous layer are halved, and the downsampled inputs are intended to extract features of different sizes through convolution operations using kernels of different sizes.

E. *Inception ResNet Two-Dimensional Module*

The Inception ResNet 2D network module Fig. 10 has been developed to extract features using skip connections from 2D inputs. Similar to the Inception DownSampling 2D network module, the Inception ResNet 2D network module uses the Inception approach but does not include a Max Pooling layer. In this module, the input

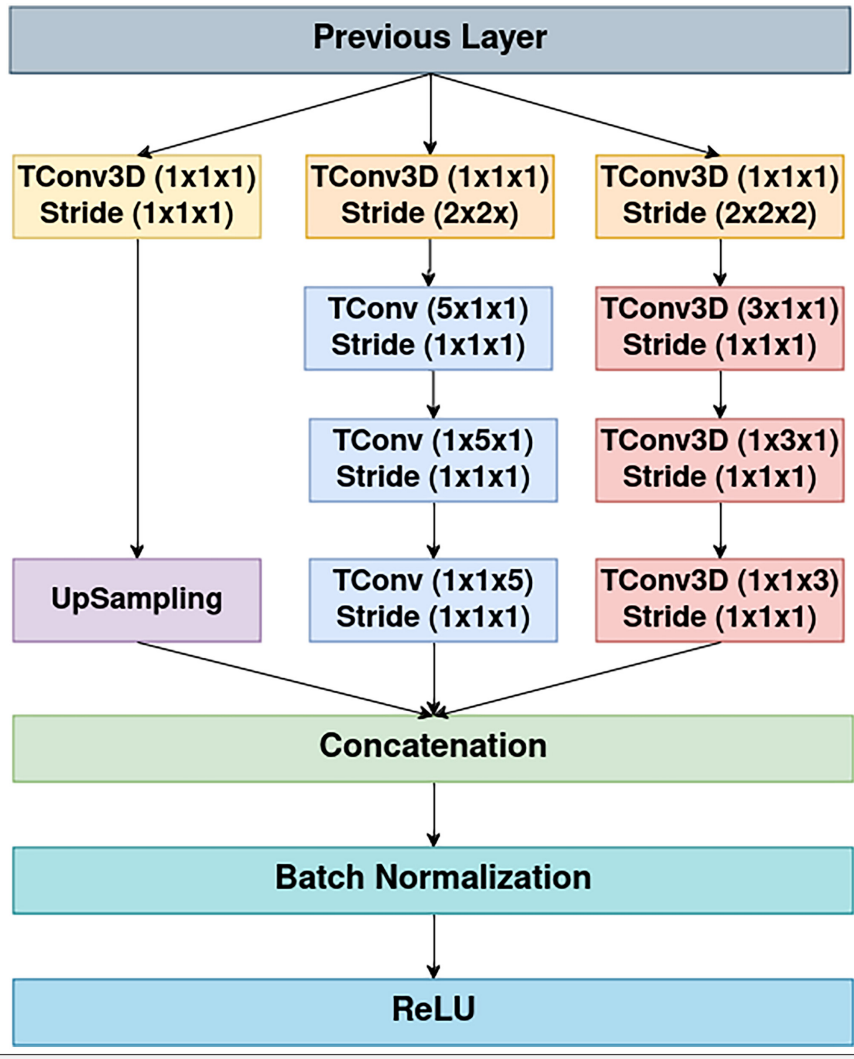


Fig. 11. Inception ResNet two-dimensional module. This figure shows the subnetwork module that uses skip connections to extract features from two-dimensional inputs, helping to minimize the vanishing gradient problem.

data from the previous layer establishes a connection with the output of the module through a skip connection method before being fed into the ReLu activation function [22] to produce the final output. The skip connection aims to minimize the vanishing gradient problem and reinforce the recall of features obtained from filter banks in previous layers, making their impact evident in the final output. No dimension reduction or UpSampling is performed in this network module. The objectives include reducing the vanishing gradient problem and extracting features of different sizes through convolution operations using kernels of different sizes.

F. *Inception UpSampling Three-Dimensional Module*

The Inception UpSampling 3D network module Fig. 11 has been developed to increase the dimensions of the height and width axes of 3D inputs. Similar to the Inception DownSampling 2D network module, dimension manipulation operations are achieved in the Inception UpSampling 3D network module using both strides and an UpSampling layer. For the dimension enlargement process in this module, an UpSampling layer and 3D-transposed convolution operations with a $2 \times 2 \times 2$ stride are employed. Following these operations, 3D-transposed convolution operations are performed at the same layer level using kernel sizes of $3 \times 1 \times 1$, $1 \times 3 \times 1$, $1 \times 1 \times 3$, and $5 \times 1 \times 1$, $1 \times 5 \times 1$, $1 \times 1 \times 5$. Subsequently, similar to the Inception DownSampling 2D module, filter concatenation, batch normalization, and ReLu layers are applied to generate the output. In this network module, the dimensions of the input from the previous layer are doubled. The objective of this module is to both double the dimensions of the input from the previous layer and extract features of different sizes through convolution operations using kernels of various sizes applied to the doubled inputs.

G. *Loss Function*

During the training process, an approach resembling a segmentation problem was initially adopted for the 3D reconstruction problem. In this approach, within the 3D coordinate system $\{x, y, z\}$, coordinates corresponding to the voxels where an object is located were assigned a value of 1, while coordinates corresponding to the background where no object exists were assigned a value of 0.

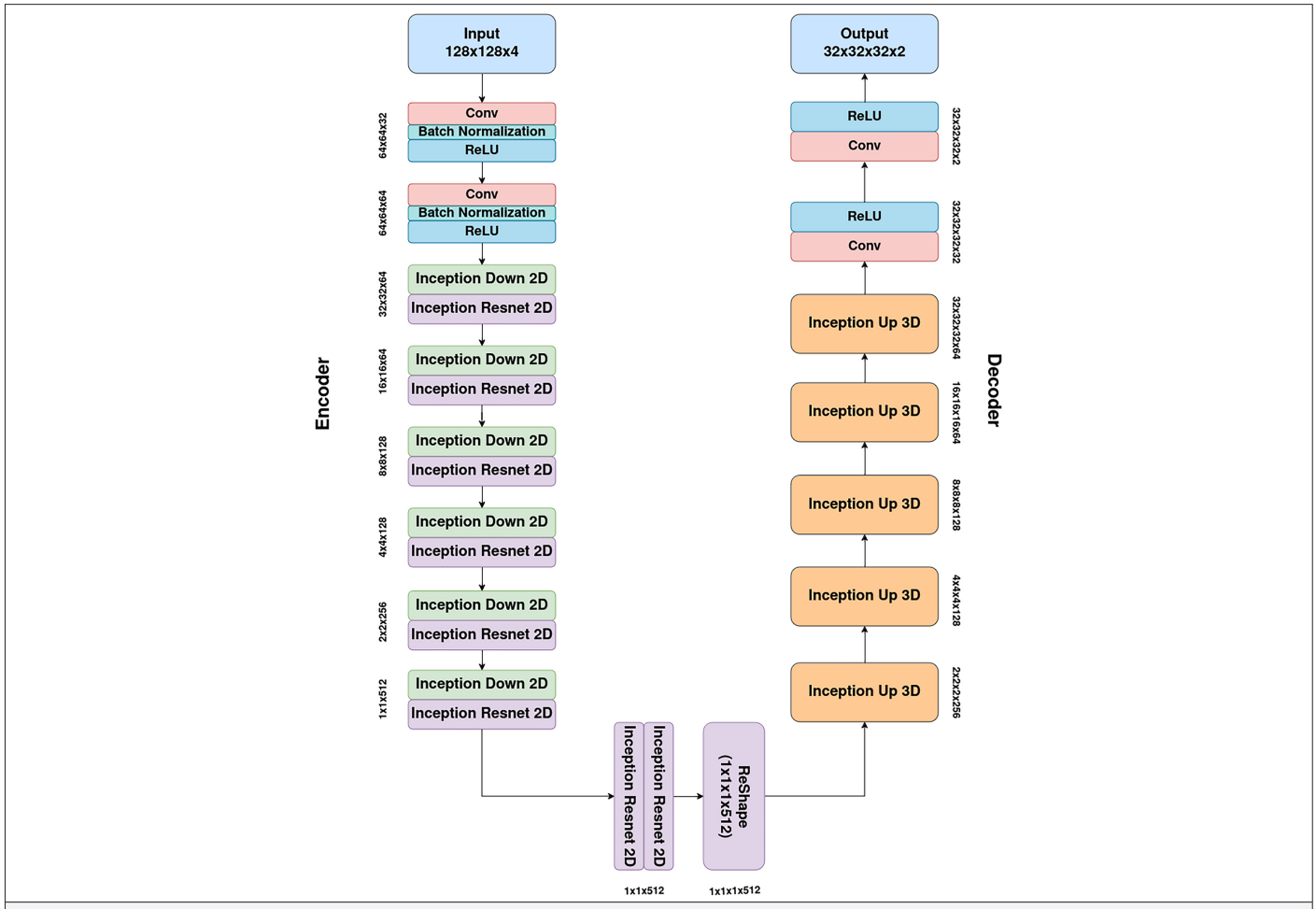


Fig. 12. Inception UpSampling three-dimensional module. This figure shows the subnetwork module that increases the dimensions of three-dimensional inputs using transposed convolution and UpSampling.

At this stage, the data have sparse characteristics, which means that the number of 0 values in the acquired data is significantly higher than the number of 1 values. For example, if a small object is in a dark background, at the end of thresholding, the resulting image will have too many 0 values compared to 1 values. In such a case, classical error functions like mean squared error and CE, when applied to imbalanced label data, result in a greater impact of the dominant label value on the error function. To overcome this issue, we propose the Categorical Mean Cross Entropy (CMCE) error function. The CMCE error function is developed to address the issue of imbalanced distribution between the data in the 0 category and the data in the 1 category within a voxel model. Essentially, the category labels were normalized separately for 0 and 1 values, and the mean cross entropy error was calculated for each category. The final CMCE error function was derived by summing up the mean cross entropy errors obtained for the 0 and 1 values as given in (2). In this equation, n_0 and n_1 are the number of 0 values and 1 values in the image, respectively. y_{i_0} and y_{i_1} are the ground truths for ith pixel among the pixels that have 0 values and ith pixel among the pixels that have 1 values, respectively. y_{i_0} and y_{i_1} are the outputs of the model in a similar manner.

$$CMCE = -\frac{\sum_{i_0=1}^{n_0} y_{i_0} \log\left(\check{y}_{i_0}\right)}{n_0} - \frac{\sum_{i_1=1}^{n_1} y_{i_1} \log\left(\check{y}_{i_1}\right)}{n_1} \quad (2)$$

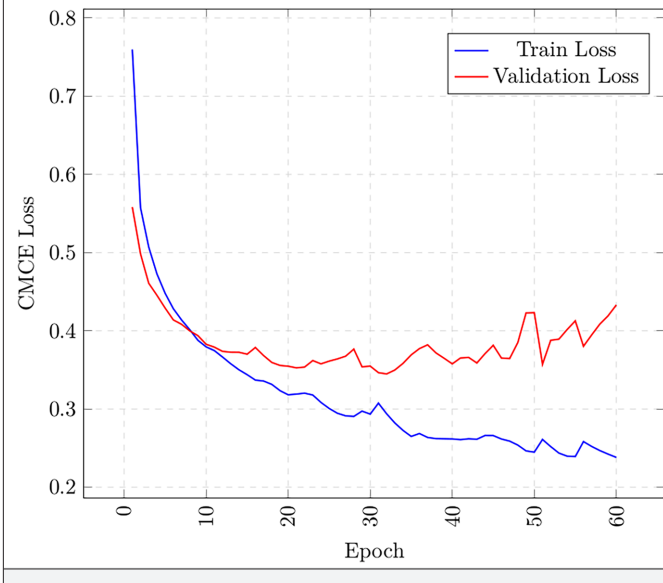


Fig. 13. Categorical mean cross entropy loss during training. This figure shows the training and validation loss curves of the proposed model using the CMCE loss function over 60 epochs. The best results were achieved at epoch 32, with overfitting observed in later epochs.

TABLE I. INTERSECTION OVER UNION PERFORMANCE COMPARISON BY ERROR FUNCTION

	IoU Error	
Category	**CMCE**	**CE**
Car	**0.7597**	0.7124
Plane	**0.4637**	0.4383
Couch	**0.6011**	0.5167
Chair	**0.4291**	0.3953
Table	**0.4791**	0.3979
Lamp	**0.3949**	0.3656
Weapon	0.4440	**0.4598**
Average	**0.5283**	0.4815

CE, cross entropy; CMCE, Categorical Mean Cross Entropy; IoU, intersection over union.
Bold in this table, the best results in each category is shown with bold characters.

III. RESULTS

In this section, the performance metrics of the proposed method have been examined categorically using the IoU metric given in (3). Additionally, the performance of the suggested CMCE loss function has been compared with the commonly used Cross Entropy (CE) loss function based on the IoU metric. The performance metrics of the proposed method have been compared with other studies in the literature using the IoU metric [15, 16], and the unique aspects of the proposed method have been discussed. Finally, the voxel-represented 3D models generated by the proposed model have been compared with reference models using sample input data.

$$IOU = \frac{Volume\,of\,the\,Intersection}{Volume\,of\,the\,Union} \tag{3}$$

As a result of this study, the learning curve obtained from training the proposed deep learning model with the CMCE loss function is shown in Fig. 13. At the end of the training process, CMCE losses of 0.2382 for the training dataset and 0.4333 for the validation dataset were achieved. However, as observed in Fig. 13, the most successful step in training is step 32, and overfitting is observed in subsequent steps. Therefore, the training of the proposed model was terminated at step 60. Additionally, to ensure capturing the step where the validation dataset performs the best during training, a subroutine was developed and integrated into the training procedure. With this subroutine, the step with the lowest CMCE loss in the validation dataset was identified and saved. This enabled recording the most efficient step, and the model file containing the weights obtained at this step was used in subsequent computations. In the model saved at step 32, where the validation dataset performed the best, CMCE losses of 0.2939 for the training dataset and 0.3449 for the validation dataset were obtained.

To evaluate the performance of the proposed deep learning model, the IoU performance metric given in (3) has been used. The IoU performance metric calculates the ratio of the intersection area of the obtained output overlaid with the reference image to the union area of the two sets, resulting in a performance metric that produces a result within the range of 0–1. The category-specific IoU values calculated using the IoU performance metric for the deep learning model obtained in this work are presented in Table I. In this table, the best results in each category is shown with bold characters. According to the results, the best result is achieved with the Car category, reaching an IoU value of 0.7597, while the worst result is observed in the Lamp category with an IoU value of 0.3949. The average IoU value for all seven categories is calculated as 0.5283. Furthermore, in Table II, the performances of the proposed deep learning model trained with the CMCE loss function and the same deep learning model trained with the CE loss function using identical data are compared categorically in terms of the IoU metric. In this table, the best results in each category is shown with bold characters. The comparison reveals that in six out of the seven categories, the model obtained through the training of the proposed deep learning model with the CMCE loss function tends to be more successful in terms of IoU values when compared to the model obtained with the CE loss function. The model trained exclusively with the CE loss function has been determined to outperform the model trained with the CMCE loss function in terms of IoU values only for the "Weapon" category. Furthermore, when considering the average IoU values across all test data, it has been demonstrated that the model trained with the CMCE loss function exhibits a higher average IoU value compared to the model trained with the CE loss function.

TABLE II. COMPARISON OF INTERSECTION OVER UNION VALUES OF METHODS

		Category					
Method	**Number of Images**	**Car**	**Plane**	**Couch**	**Chair**	**Table**	**Average**
PTN [16]	1	0.4437	0.3352	0.3309	0.2241	0.1977	0.2931
McRecon [15]	1	0.5622	0.3727	0.3791	0.3503	0.3532	0.4036
PTN [16]	5	0.6593	0.4422	0.5188	0.3736	0.3356	0.4572
McRecon [15]	5	0.6142	0.4523	0.5458	**0.4365**	0.4204	0.4849
Proposed model + CE	4	0.7124	0.4383	0.5167	0.3953	0.3979	0.4621
Proposed model + CMCE	4	**0.7597**	**0.4637**	**0.6011**	0.4291	**0.4711**	**0.5449**

CE, cross entropy; CMCE, Categorical Mean Cross Entropy.
Bold in this table, the best results in each category is shown with bold characters.

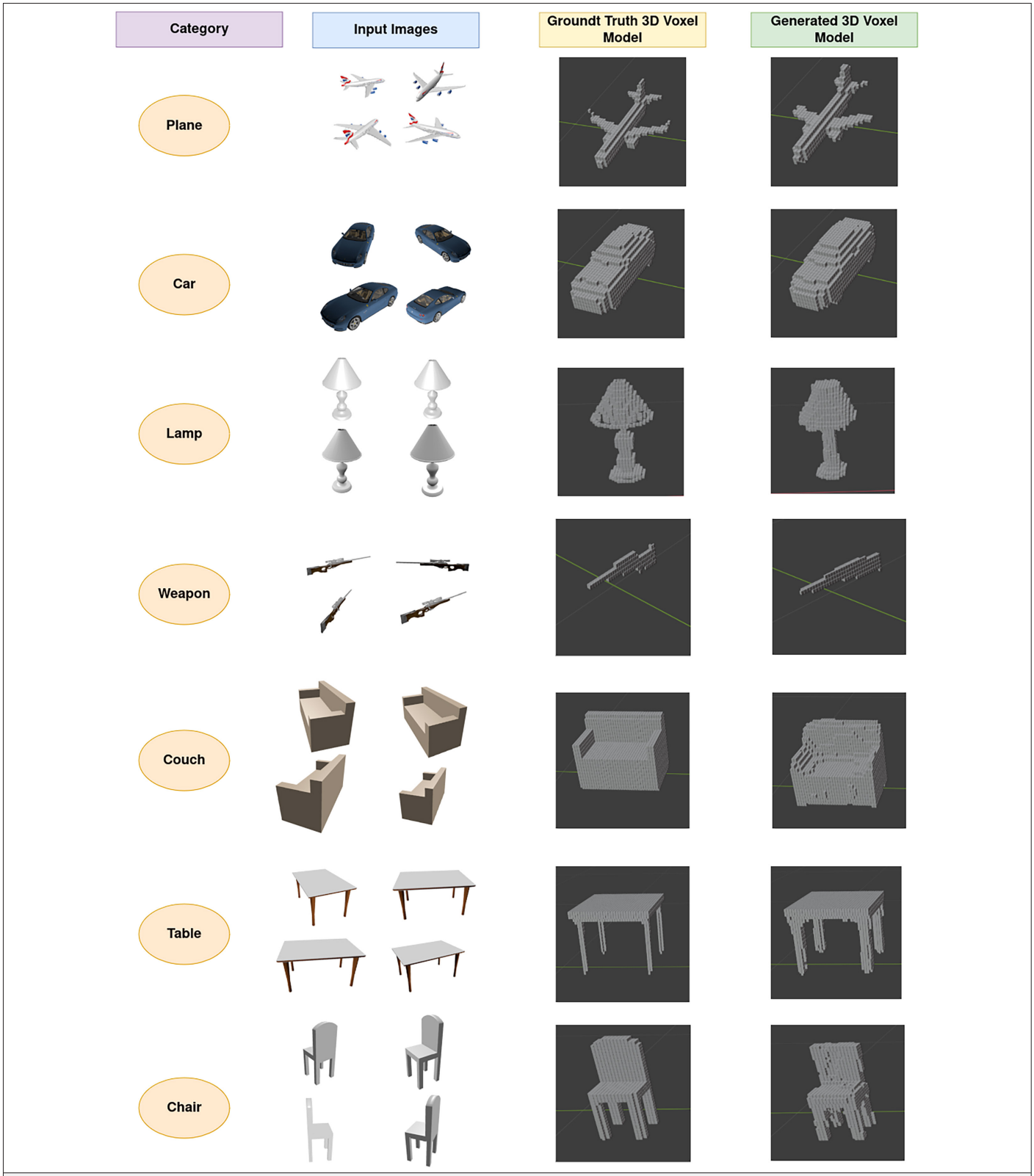


Fig. 14. Sample of generated three-dimensional voxel models. This figure displays sample three-dimensional voxel models generated by the proposed deep learning model compared with reference three-dimensional voxel models.

The comparison of the category-specific IoU values for both the version trained with the CE loss function and the version trained with the CMCE loss function of the proposed method is presented in Table II. The evaluation involves a comparison with the PTN [16] method developed using input of 1 image and 5 images, as well as the McRecon [15] method. In Table II, the IoU performance values reported by Yan et al. [12] for the PTN and McRecon methods were used. The obtained results highlight the outcome of the most successful method in each category in the table. In this comparison, the metric results of the IoU performance are shown, and the set of common categories used in the training of the PTN and McRecon methods on the ShapeNet dataset is specified. In the context of these common category sets, it is observed that the proposed method outperforms both the 1 image and 5 image input versions of both McRecon and PTN methods in categories other than the chair category, achieving higher IoU values.

The proposed method not only attains the highest IoU value solely in the chair category but also demonstrates the second highest IoU value in the chair category when considering the version of the proposed model trained with the CMCE loss function. Furthermore, when focusing on the common category sets and calculating the average IoU values, the version of the proposed deep learning model trained with the CMCE loss function consistently achieves a higher average IoU value compared to other methods, as illustrated in Table II.

The comparison of the 3D voxel models generated in the categories of airplane, car, lamp, weapon, coach, table, and chair using the developed CMCE loss function within the scope of this study, trained with the proposed deep learning model, with the reference 3D Voxel models is illustrated in Fig. 14. The 3D voxel models generated in comparison with the reference 3D voxel models shown in Fig. 14 were opened in a computer environment, and screenshots were taken.

IV. CONCLUSION

We have developed a deep learning model along with a proposed error function to generate voxel-represented 3D models from silhouette images captured from four different perspectives. The average IoU value achieved was 0.5449. After preprocessing the example 2D images, as shown in Fig. 8, and feeding them into the input layer of the trained deep learning model, voxel-represented 3D models were generated, as shown in Fig. 14.

The 3D models were created using images captured from four different perspectives. While some studies in the literature [12, 14–16] have been able to produce 3D models using only a single image, these methods have performed poorly in terms of IoU values. Additionally, they used supervised methods in terms of categorization. Therefore, it is recommended that unsupervised deep learning studies be conducted using three or more images for categorical exploration.

The development of unsupervised deep learning models for categorization is expected to significantly reduce the reliance of these models on predefined categories in the training dataset, such as airplanes and cars. Consequently, these models will be capable of generating 3D models for objects that have never been encountered before, resulting in a more comprehensive and efficient deep-learning model. Therefore, the advancement of unsupervised deep learning models for categorization is essential for future research and industrial applications.

We propose a method that utilizes voxel representation to produce 3D models in this work. However, for industries such as gaming, graphics, and architecture to fully benefit from deep learning-based approaches and to offer more practical solutions, future research should focus on developing 3D models with smoother and more realistic mesh representations. The method proposed in this study does not address these aspects, indicating the need for further refinement in future research.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – M.D., Design – B.A.; Supervision – M.D., Resource – M.D.; Materials - B.A.; Data Collection and/or Processing – M.D., B.A., Ö.C.T.; Analysis and/or Interpretation – M.D., B.A., Ö.C.T.; Literature Search – M.D., B.A., Ö.C.T.; Writing – M.D., B.A., Ö.C.T.; Critical Review – M.D., B.A., Ö.C.T.

Declaration of Interests: The authors have no conflict of interest to declare.

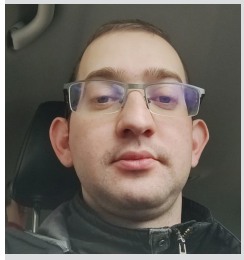
Funding: The authors declare that this study has received no financial support.

REFERENCES

1. W. Lee et al., "3D scan to product design: Methods, techniques, and cases," in Proc. 6th Int. Conf. 3D Body Scanning Technol., Lugano, Switzerland, Oct. 27–28, 2015, pp. 168–174. [CrossRef]
2. T. Bangemann et al., "State of the art in industrial automation," in *Industrial Cloud-Based Cyber-Physical Systems*, A. W. Colombo et al., Ed. Springer Int. Publishing, 2014, pp. 23–47. [CrossRef]
3. M. Bitzidou, D. Chrysostomou, and A. Gasteratos, "Multi-camera 3D object reconstruction for industrial automation," *IFIP Adv. Inf. Commun. Technol.*, Berlin, Heidelberg: Springer , pp. 526–533, 2013. [CrossRef]
4. T. Vernon, and D. Peckham, "The benefits of 3D modelling and animation in medical teaching," *J. Audiov. Media Med.*, vol. 25, no. 4, pp. 142–148, Jan. 2002. (doi: [CrossRef])
5. F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi, "UAV photogrammetry for mapping and 3D modeling – Current status and future perspectives," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XXXVIII–1/C22, pp. 25–31, Sep. 2012. (doi: [CrossRef])
6. K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Representations (ICLR), Sep. 2014.
7. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), IEEE, Jun. 2016, pp. 770–778. [CrossRef]
8. R. Mehrotra, M. A. Ansari, R. Agrawal, and R. S. Anand, "A Transfer Learning approach for AI-based classification of brain tumors," *Mach. Learn. Appl.*, vol. 2, p. 100003, Dec. 2020. [CrossRef]
9. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. (doi: [CrossRef])
10. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), IEEE, Jun. 2016, pp. 779–788. [CrossRef]
11. B. Asiroglu et al., "A deep learning based object detection system for user interface code generation," 2022 Int. Congr. Hum.-Comput. Interact., Optim. Robot. Appl. (HORA), IEEE, Jun. 2022, pp. 1–5. [CrossRef]
12. C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," *Lect. Notes Comput. Sci.*, Springer Publishing, pp. 628–644, 2016. (doi: [CrossRef])
13. H. Gupta, M. T. McCann, L. Donati, and M. Unser, "CryoGAN: A new reconstruction paradigm for single-particle cryo-EM via deep adversarial learning," *IEEE Trans. Comput. Imaging*, vol. 7, pp. 759–774, 2021. [CrossRef]
14. K. Fu, J. Peng, Q. He, and H. Zhang, "Single image 3D object reconstruction based on deep learning: A review," *Multimed. Tools Appl.*, vol. 80, no. 1, pp. 463–498, Jan. 2021. (doi: [CrossRef]).

15. J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese, "Weakly supervised 3D reconstruction with adversarial constraint," Int. Conf. 3D Vision (3DV), IEEE, Oct. 2017, pp. 263–272. [CrossRef]
16. X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: learning single-view 3D object reconstruction without 3D supervision," 30th International Conference on Neural Information Processing Systems, Dec 2016, pp. 1704-1712. (https://dl.acm.org/doi/epdf/10.5555/3157096.3157287).
17. A. Uçar, Y. Demir, and C. Güzeliş, "Object recognition and detection with deep learning for autonomous driving applications," *Simulation*, vol. 93, no. 9, pp. 759–769, Sep. 2017. [CrossRef]
18. A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," Dec. 2015. Available: https://arxiv.org/abs/1512.03012.
19. A. Yuniarti, and N. Suciati, "A review of deep learning techniques for 3D reconstruction of 2D images," 2019 12th Int. Conf. Inf. Commun. Technol. Syst. (ICTS), IEEE, Jul. 2019, pp. 327–331. [CrossRef]
20. M. Lin, Q. Chen, and S. Yan, "Network in network," 2nd Int. Conf. Learn. Representations(ICLR), Dec. 2013. Available: https://arxiv.org/pdf/1312.4400
21. C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), IEEE, Jun. 2015, pp. 1–9. (doi: [CrossRef])
22. K. Fukushima, "Visual feature extraction by a multilayered network of analog threshold elements," *IEEE Trans. Syst. Sci. Cybern.*, vol. 5, no. 4, pp. 322–333, 1969. (doi: [CrossRef])

Mustafa Dağtekin is currently working as an Assistant Professor at İstanbul University-Cerrahpaşa. He has completed his M.Sc. and PhD in 1999 and 2006, respectively, at NC State University, Raleigh, NC, USA.



Batuhan Aşıroğlu is currently a PhD student at İstanbul University-Cerrahpaşa. He has completed his M.Sc. in 2022 at the same institution.

Özgür Can Turna is currently working as an Assistant Professor at İstanbul University-Cerrahpaşa. He has completed his M.Sc. and Ph.D. in 2005 and 2014, respectively, at the same institution.