ABSTRACT

Image-based 3D reconstruction is a very challenging problem in computer vision and deep learning. Since 2015 image-based 3D reconstruction using a convolution neural network has attracted and demonstrated impressive performance. We focus on the work that uses deep-learning techniques to reconstruct the 3D shape of generic objects from single or multiple RGB images. However, unlike 2D images, 3D cannot be represented in its canonical form to make it computationally lean and memory-efficient. This paper proposes Grid/voxel-based 3D object reconstruction from a single 2D image for better accuracy, using the Autoencoders (AE) model. The encoder part of the model is used to learn suitable compressed domain representation from a single 2D image, and a decoder generates a corresponding 3D object. We provide a comprehensive, structured review of the recent advanced 3D objects reconstruction using deep-learning techniques.

Contents □ pages

CHAPTER 1

INTRODUCTION

The goal of image-based 3D reconstruction is to infer the 3D geometry and structure of objects and scenes from one or multiple 2D images. This long-standing ill-posed problem is fundamental to many applications such as robot navigation, object recognition, scene understanding, 3D printing and animation, industrial control, and medical diagnosis. Recovering the lost dimension from just 2D images has been the goal of classic multi-view stereo and shape-from X methods, which have been extensively investigated for many decades. The first generation of methods approached the problem from the geometric perspective; they focused on mathematically understanding and formalizing the 3D to 2D projection process to devise mathematical or algorithmic solutions to the ill-posed inverse problem. Effective solutions typically require multiple images captured using accurately calibrated cameras. For example, stereo-based techniques [1] require matching features across images captured from slightly different viewing angles and then using the triangulation principle to recover the 3D coordinates of the image pixels. Shape-from-silhouette, or shape by-space-carving, methods [2] require accurately segmented 2D silhouettes. These methods, which have led to reasonable quality 3D reconstructions, require multiple images of the same object captured by well-calibrated cameras. This, however, may not be practical or feasible in many situations. In recent years, imaging devices such as cameras have become common, and people have easy access to these devices; however, most of these devices can only capture the scene in 2D format. Initially, the real-world scenes exist in a 3D format, but the third dimension gets lost during image acquisition. The recovery of the lost dimension is essential for many applications such as robotic vision, medical imaging, 3D printing, and the TV industry. In a 2D image, an essential element is known as a pixel having coordinates X and Y.

In contrast, in a 3D model, the basic element is a voxel consisting of three coordinates X, Y, and Z. Interpreting 3D shapes is a primary function of the human visual system. Hence, we can easily infer the object’s 3D shape by viewing it from one or more viewpoints. However, it is quite a trivial task for machines to infer the lost third dimension due to the absence of crucial geometrical information in the 2D format. Literature confirms that different approaches have been employed for 3D reconstruction over the last few decades, such as generating a 3D model from point cloud data and generating a 3D model directly from 2D images. The point-cloud-based approach employs skeletons, meshes, and Voronoi diagrams for 3D reconstruction [3]. The point cloud data are the 3D unstructured data gathered using a 3D laser scanner and 3D cameras [4]. Constructing a 3D model from point cloud data is highly mathematical because complex geometrical information is required. However, some data-driven approaches use machine learning techniques for 3D reconstruction from point cloud data [4]. In the second approach, initially, researchers proposed several methods for 3D reconstruction using an extensive collection of images of the same object. For this purpose, the geometrical properties were extracted from images using direct minimization of projection errors or dense matching. In addition, implicit volumetric reconstruction or explicit mesh-based techniques were also used for 3D reconstruction. However, both cases require much input data and mathematical knowledge to estimate sufficient geometrical properties [5]. Recently, after the availability of large 3D data sets such as ShapeNet [6] and advancements in machine learning techniques, several successful attempts have been made for 3D reconstruction directly from 2D images using learning-based methods. These techniques include multiple-view-based methods [7], panoramic-view-based methods [8], and single-view-based methods [9]. In multiple-view-based methods, particular image capturing devices, such as 3D cameras, are required to capture the multi-view images of an object or scene used for 3D model reconstruction.

In contrast, the panoramic image of a scene or object estimates the geometry and reconstructs the layout in a 3D model.

Both approaches are pretty tedious because extensive mathematical information is required for 3D reconstruction using these methods. Constructing a 3D model from a single view 2D image is more promising because of the easy availability of single-view image capturing devices. Several methods have been proposed for 3D reconstruction from a single 2D image; however, there is still a need to address many issues such as low resolution, inefficiency, and low accuracy of existing methods. This work presents simple-autoencoder (AE)- and variational-autoencoder (VAE)-based methods for 3D reconstruction from a single 2D image. Our contribution is twofold.

First, to the best of the authors' knowledge, it is the first time that VAE has been employed for the 3D reconstruction problem. Second, the model is designed in such a way that it could extract a discriminative set of features for improving reconstruction results. The proposed method is evaluated on the ShapeNet benchmark data set, and the results confirm that it outperforms state-of-the-art methods for 3D reconstruction. The rest of the paper is organized as follows. In Section 2, related work for 3D model reconstruction is presented. In Section 3, we elaborate on the proposed methodology. Section 4 presented our proposed methodology. Experimentation results and discussion are presented in Section 5. Finally, the paper is concluded in Section 6.

1.1 Feature extraction: Features are parts or patterns of an object in an image that help to identify it. For example — a square has 4 corners and 4 edges, they can be called features of the square, and they help us humans identify it’s a square. Features include properties like corners, edges, regions of interest points, ridges, etc.

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is that many variables require many computing resources to process. Feature extraction is the name for methods that select and/or combine variables into features, effectively reducing the amount of data that must be processed while wholly and accurately describing the original data set.

The feature extraction process is useful when you reduce the resources needed for processing without losing essential or relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine’s efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the Machine Learning process**.**

1.2 Practical use of feature extraction :

- **Autoencoder:** The purpose of Autoencoder is Unsupervised Learning of efficient data coding. Feature extraction is used here to identify key features in the data for coding by learning from the coding of the original data set to derive new ones.

- **Bag of Words:** A technique for Natural Language Processing that extracts the words (features) used in a sentence, document, website, etc., and classifies them by frequency of use. This technique can also be applied to image processing**.**

- **Image Processing:** Algorithms are used to detect features such as shaped, edges, or motion in a digital image or video.

1.3 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning Algorithms use historical data as input to predict new output values.

Machine learning is important because it gives enterprises a view of trends in customer behaviour and operational business patterns and supports the development of new products. Many of today's leading companies, such as Facebook, Google, and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches supervised learning, Unsupervised learning, semi-supervised learning, and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

- **Supervised Learning :**

  This machine learning algorithm supplies algorithms with labeled training data and defines the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm are specified.

- **Unsupervised Learning:** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets, looking for any meaningful connection. The data that algorithms train on and the predictions or recommendations they output are predetermined**.**

- **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled Training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

- **Reinforcement Learning:** Machine Learning typically uses Reinforcement Learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

1.4 Deep Learning

Deep learning is a type of Machine Learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an essential element of data science, which includes statistics and Predictive Modelling. It is extremely beneficial to data scientists tasked with collecting, analysing, and interpreting large amounts of data; deep learning makes this process faster and easier.

At its simplest, deep learning can be thought of as a way to automate Predictive Analysis. While traditional machine learning Algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction.

To understand deep learning, imagine a toddler whose first words are *cow, cat, and dog*. The toddler learns what a cow, cat, or dog are -- and are not -- by pointing to objects and saying the words *cow, cat, and dog*. The parent says, "Yes, that is a dog," or, "No, that is not a dog." As the toddler points to objects, he becomes more aware of all of the features of cows, cats, and dogs. What the toddler does, without knowing it, is clarify a complex abstraction -- the concept of a cow -- by building a hierarchy in which each level of abstraction is created with the knowledge gained from the preceding layer of the hierarchy.

- **1.5 Applications:** 3D reconstruction system applications in various fields: medicine, film industry, Robotics, City planning, gaming, virtual environment, earth observation, archaeology, augmented reality, reverse engineering, animation, human-computer interaction, etc., some of the applications. Various comparisons of different methods have been made to improve the quality of the 3D image. The researchers for future studies have suggested many efficient methods for their issues. There are many open issues for 3D reconstruction like face deformations, city planning without sensors, shape, texture, etc., which are challenging for future reviewers in the field of computer vision and computer graphics.

- **1.6 Motivation behind image-based 3D reconstruction:**

This thesis aims to reconstruct and model the 2D images to obtain the 3D form of the object, which will be developed using deep learning algorithms. A 2D image gives the user the freedom to view the object at a 180-degree angle, but a 3D object does a 360-degree view. These 3D images are very much essential in Robotic Vision, Augmented reality, and Virtual reality. in the Early, 2D images which could not provide the actual 3D shape, dimension, and volume of the object, so it is not effective for Robotic vision but in 3D images tells the actual shape, dimensions and volume of the object which gives a better understanding of unknown object; as a result Robot performance gets higher. So, this can be the one objective of reconstructing 3D objects. I chose deep learning because I learned machine learning and had a project in my B. Tech study .so thought to explore deep learning projects as we know that deep learning has high demand in the market and the cutting-edge technology.

Apart from this, 3D reconstruction and modeling are used in many fields, including medicine, film industry,3D printing, City planning, gaming, virtual environment, earth observation, archaeology augmented reality, reverse engineering, animation, human-computer interaction, etc. Nowadays, various algorithms are available to reconstruct image-based 3D objects.

CHAPTER 2

LITERATURE REVIEW

In this section, we provide background information and discuss state-of-the-art methods used for 3D model reconstruction. These methods can be divided into geometry-based reconstruction and learning-based reconstruction methods.

Recovering the lost dimension during image acquisition from any normal camera has been a hot research area in computer vision for over a decade. The literature review shows that the research methodology has changed from time to time. More precisely, we can divide the conversion of 2D images to 3D model reconstruction into three generations. The first generation learns the 3D to 2D image projection process by utilizing the mathematical and geometrical information using some mathematical or algorithmic solution. These types of solutions usually require multiple images that are captured using specially calibrated cameras. For example, using some multi-view of an object with constant angle changing that can cover all the 360 degrees of an object, we can compute the geometrical points of the object [10]. We can join these points using some triangularization techniques to make a 3D model [3]. The second generation of 2D to 3D model conversion utilizes the accurately segmented 2D silhouettes. This generation leads to a reasonable 3D model generation, but it requires specially designed calibrated cameras to capture the image of the same object from every different angle. This technique is not feasible or practical because of the complex image-capturing techniques [10,11]. Humans can assume the object's shape using prior knowledge about some objects and predict what an object will look like from another unseen viewpoint. The computer-vision-based techniques are inspired by human vision to convert 2D images to 3D models. With the availability of large-scale data sets, deep learning research has evolved into 3D reconstruction from a single 2D image. A deep-belief-network-based 3D model was proposed [12] to learn the 3D model from a single 2D image. It is considered one of the earlier neural-network-based data-driven models to reproduce the 3D model. Although the results were not promising, it was considered a good start in 3D reconstruction using the computer-vision-based method.

After this success, another research study based on a recurrent neural network became popular for 3D reconstruction [13]. This method employed encoder–decoder-based architecture while considering single or multiple images as input. The latent vector was produced using input, and then this latent layer vector was given to the decoder module with a residual network to reproduce the 3D model. This also became an achievement in computer vision, but the quality of results depended on the number of images given as input. In this chapter, from sections 2.1.2 to 2.2.3, we discuss single-view 3D reconstruction and multi-view 3D reconstruction representation.

➢ 2.1 Single-view 3D Reconstruction:

Generating 3D reconstruction from just a single image is challenging because the single-view 3D reconstruction problem is ill-posed and ambiguous since the partially predicted points can be associated with an infinite number of 3D models, as mentioned.

➢ 2.1.1 Shape Reconstruction:

Methods have been introduced recently with new data representations for the task of 3D shape reconstruction. These data representations include point clouds [16], meshes [17], and signed distance fields [15]. The PSG method [16] recovers a point cloud from a single RGB image. The method of Pixel2Mesh [17] is the first method in literature for generating a triangular mesh from a single RGB image. The approach of DeepSDF [18] provides the SDF representation of a set of points provided as an input. However, this approach will not work for reconstruction from just an RGB image. The proposed method in this thesis also generates encoded TSDF (Truncated Signed Distance Function) volume in the end. However, the method is not a generative model, unlike DeepSDF, with no probabilistic interpretation. The OGN [19] method uses an octree for handling the memory constraints of large 3D resolutions. Matryoshka

Networks [20] decompose the 3D shape into nested shape layers. The method can outperform octree-based reconstruction methods, and it can generate output resolution as high as 2563

➢ 2.1.2 Scene Reconstruction:

The work of Xie et al. [24] proposes an efficient method for generating TSDF (Truncated Signed Distance Function) volumes and a tree net architecture that solves the scene reconstruction task by splitting channel-wise. This method uses an Autoencoder for efficiently compressing TSDFs of a resolution of 2 x 64. The decoder part of the Autoencoder is used to return to the original resolution, which means it is one of the only methods out there that can generate this high of a resolution. Furthermore, the method also proposes a custom loss shaping function, which penalizes the loss around the surface of an object and the free space before an object. This thesis uses not only the autoencoder for compression and decompression but also a modified version of the TSDF generation pipeline as proposed in Xie et al. [24].

➢ 2.2. Multiview 3D Reconstruction:

Traditional dense 3D reconstruction methods, for example, SfM(Structure from Motion) and vSLAM(Visual Simultaneous Localization and Mapping), require a dense number of RGB images with certain assumptions. These traditional methods involve feature extraction and matching [22] or minimizing reprojection errors [3, 18]. Firstly, the feature matching process can be slow, especially if, for example, SIFT features are calculated, and secondly, the extracted features should cover the whole surface of the 3D object. Otherwise, there may be occlusions or holes in the final 3D reconstruction.

One of the literature's first multi-view deep learning-based methods is the MVCNN [21] network. In MVCNN, 3D geometry is rendered into 2D, after which the 2D features are calculated, followed by max pooling. This approach works suitably well for the classification task but is unsuitable for other upstream 3D tasks, like reconstruction.

➢ 2.2.1 Recurrent Neural Network (RNN) based methods:

The 3D-R2N2 [25] method proposed an RNN for multi-view 3D shape reconstruction where the authors, for each multi-view image, use an RNN module. However, this approach suffers from several issues. Firstly, the approach is order variant, meaning that the generated results depend on the order in which the images of the different viewpoints are given to the network. Secondly, the approach suffers from long-term memory-related issues common in RNNs, which means that the features learned from the initial images might be forgotten. Finally, the approach is not parallelizable and time-consuming since the images are processed sequentially.

The LSM [23] method also uses an RNN for fusing 3D features from different views. However, it addresses the RNN-related problems identified in the approach of 3D-R2N2. The LSM approach also uses feature projection and un-projection along the viewing rays, which needs the camera's intrinsic and extrinsic parameters. As Xie et al. [24] reported, LSM performs better with more than one view than other methods. They argue that for more than one view, the camera's intrinsic and extrinsic help to align the 2D features of multi-view images better. Our proposed approach is not dependent on the view order since the images are processed spatially instead of temporally. Additionally, the proposed approach uses the intrinsic and extrinsic camera, similar to LSM, which are generated, along with the 2D renderings, using BlenderProc [27].

2.2.2 Encoder-Decoder-based methods:

The Pix2Vox [28] method uses an encoder-decoder-based architecture alongside a context-aware module for fusion and a refiner module for correcting wrongly recovered reconstructions. Even though the network produces impressive results, the training process is not end-to-end, where the modules are tried separately. The authors tried to improve their work in a follow-up method named Pix2Vox++ [24] that generates better reconstructions due to improved architectural choices. They also propose a large-scale multi-view 3D shape reconstruction dataset named Things3D, based upon the SUNCG [29] dataset, which unfortunately is no longer available. Spezialetti et al. [30] proposed multi-view 3D shape reconstruction with the added task of estimating the relative pose image pairs used for reconstruction. Unlike the encoder-decoder-based approaches, our approach uses a 2D network that calculates 2D features directly associated with the 3D reconstruction using camera intrinsic and extrinsic parameters. Furthermore, a new dataset is proposed with the output target having a TSDF representation with the same categories and data splits as in 3D-R2N2 [25].

2.2.3 Attention-based methods:

The work of Yang et al. [31] proposed an attention aggregation module named AttSets and a training algorithm named FASet. The work claims to have an aggregation approach comparable to pooling-based approaches, such as average and max pooling. Most recently, Transformer Networks have been used for multi-view 3D shape reconstruction in the work of Yagubbayli et al. [32] and Wang et al. [33]. The Transformer Networks again have the advantage of using attention for view aggregation. Howeve1 our approach uses 30 voxel-based max pooling for view aggregation to avoid the dependency on the number of views.

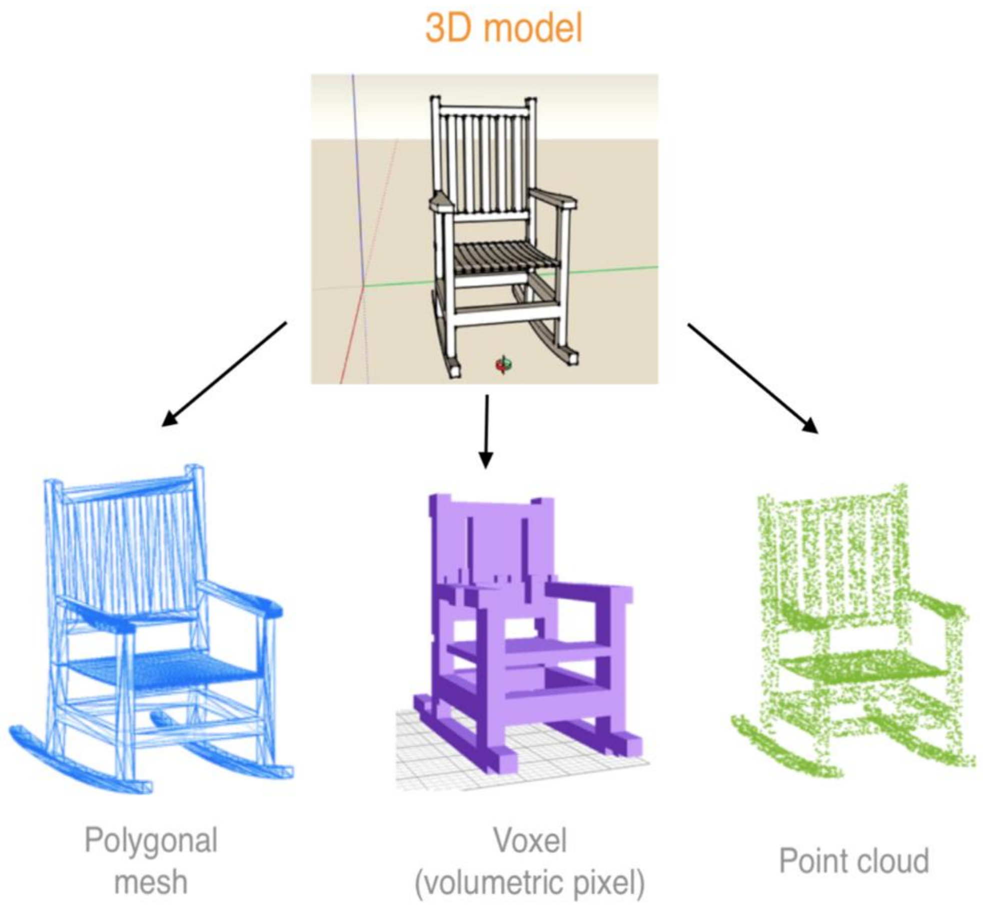
All these approaches use occupancy grid representation with a resolution of, except for the work of Xie et al. [24], which also presented some results for an output resolution of. With this small resolution, objects with fine details cannot be represented. Additionally, the surfaces are not smooth in occupancy grid representation, as shown in Figure 2.1. Instead, the proposed approach uses a TSDF (Truncated Signed Distance Function) based representation, which is a more dense and smooth representation than an occupancy grid representation.

CHAPTER 3

3D Reconstruction from still images

□ 3.1 3D Data Representations:

There exist several ways to represent 3D data. The underlying methods change depending on the type of 3D data representation used. The most common 3D data representations below these are shown in Figure 3.8.

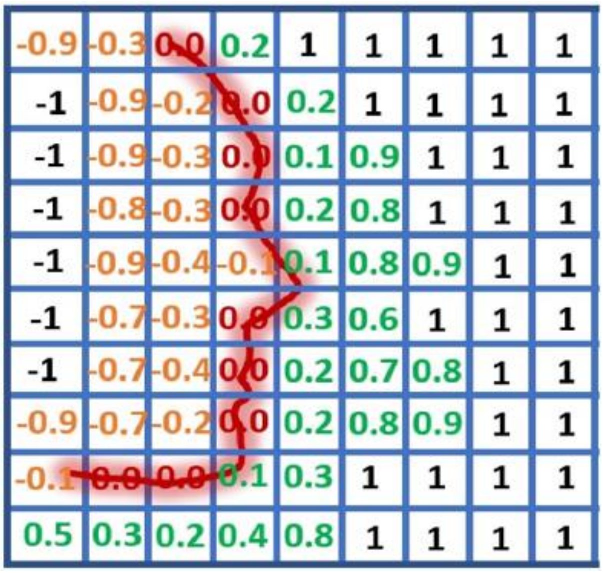


https://medium.com/vitalify-asia/create-3d-model-from-a-single-2d-image-in-pytorch

fig 3.1 Various Representation of 3D data.

➢ 3.2 Binary Occupancy Grid/Voxel Grid:

A binary occupancy grid or voxel grid is a 3D representation that encodes the geometry as a 3D grid. Each cell in the 3D grid encodes whether it is occupied or empty. A value of zero is used for empty cells, and a value of one is used for filled cells. The cells maintain spatial information, which means that traditional deep learning architectures can also be applied to this data type. A binary occupancy grid does not capture fine geometric details for smaller resolutions. As the resolution size increases, the representation suffers from computation and memory-related issues because of the encoding of the occupied and the free space.



-0.9	-0.3	0.0	0.2	1	1	1	1	1
-1	-0.9	-0.2	0.0	0.2	1	1	1	1
-1	-0.9	-0.3	0.0	0.1	0.9	1	1	1
-1	-0.8	-0.3	0.0	0.2	0.8	1	1	1
-1	-0.9	-0.4	-0.1	0.1	0.8	0.9	1	1
-1	-0.7	-0.3	0.0	0.3	0.6	1	1	1
-1	-0.7	-0.4	0.0	0.2	0.7	0.8	1	1
-0.9	-0.7	-0.2	0.0	0.2	0.8	0.9	1	1
-0.1	0.0	0.0	0.1	0.3	1	1	1	1
0.5	0.3	0.2	0.4	0.8	1	1	1	1

Fig3.2. 2D TSDF with a red curve depicting the surface with all the values inside the surface being negative, on the surface being zero, and outside the surface being positive. As can be seen in the figure, a truncation between -1 and 1 is applied. The same concept can also be extended to 3D. **The image is taken from the Arm Community website** [44].

3.2.1 Truncated Signed Distance Function (TSDF):

This 3D data representation encodes geometry as a gradient field in terms of the signed distance to the closest surface where all the values inside the surface are negative, the values on the surface are zero, and the values outside the surface are positive. Since these signed distance values can be unbounded, the values are usually truncated in a range, for example, -1 to 1, as shown in Figure 3.6; hence, the name Truncated Signed Distance Function (TSDF). TSDF is usually a dense representation; thus, it provides a better gradient flow through Deep Neural Network than a binary occupancy grid representation.

Like a binary occupancy grid, traditional deep learning operations like convolution can be applied directly to this 3D data representation. Similarly, it also suffers from the same computation and memory-related issues. Thus, it is desirable to have a binary occupancy grid and a TSDF grid of higher resolution for capturing more finer details. Figure 3.7 shows the memory comparison between binary occupancy /voxel grid and TSDF volume for different resolutions. TSDF volumes take up more memory for higher resolutions than binary occupancy grids. To obtain a mesh from a TSDF volume, a mystification algorithm, like Marching Cubes [42], can be used.

3.2.2 Advantage and Disadvantage of Voxel/Grid :

- Voxel/Grid are more accurate 3D building blocks than any other modeling type, as they mimic particles.
- Voxel/Grid Unlocks new simulation techniques that would be impossible with other modeling methods.
- Voxels are the quickest way to quickly model and visualize volumetric data (especially naturally or organically).
- Without using prohibitively, Expensive training like 3D scanning, it is much harder to build
- Complex objects using voxels
- Voxel modeling lacks the mathematical precision of BRep Modeling.

- Current computer hardware is optimized for rendering polygons, and we don't have specialized to
  Efficiently render high-resolution voxels.

3.3 Mesh

A mesh representation explicitly encodes the surface representation using polygons. It is chal-lenging to design deep learning networks that handle mesh data directly because applying convolutional, and pooling operations is not directly possible, and most previous works generated mesh from an intermediate TSDF representation. However, there are already works in literature, like MeshNet [36] and MeshCNN [37], that take mesh data directly as an input for a Deep Neural Network. There are also methods like Deep Marching Cubes [38] that provide meshes directly as output from well-sampled point clouds.

➢ 3.4 Point Cloud:

A point cloud is an unstructured and memory-efficient representation that expresses the shape geometry as 3D straight points. However, point clouds do not have the concept of free space and do not capture geometry well due to the spacing of the points. However, point clouds have received much attention recently, and there are many methods in the literature, like PointNet [34] and PointNet++ [35], which work directly on point clouds as an input**.**

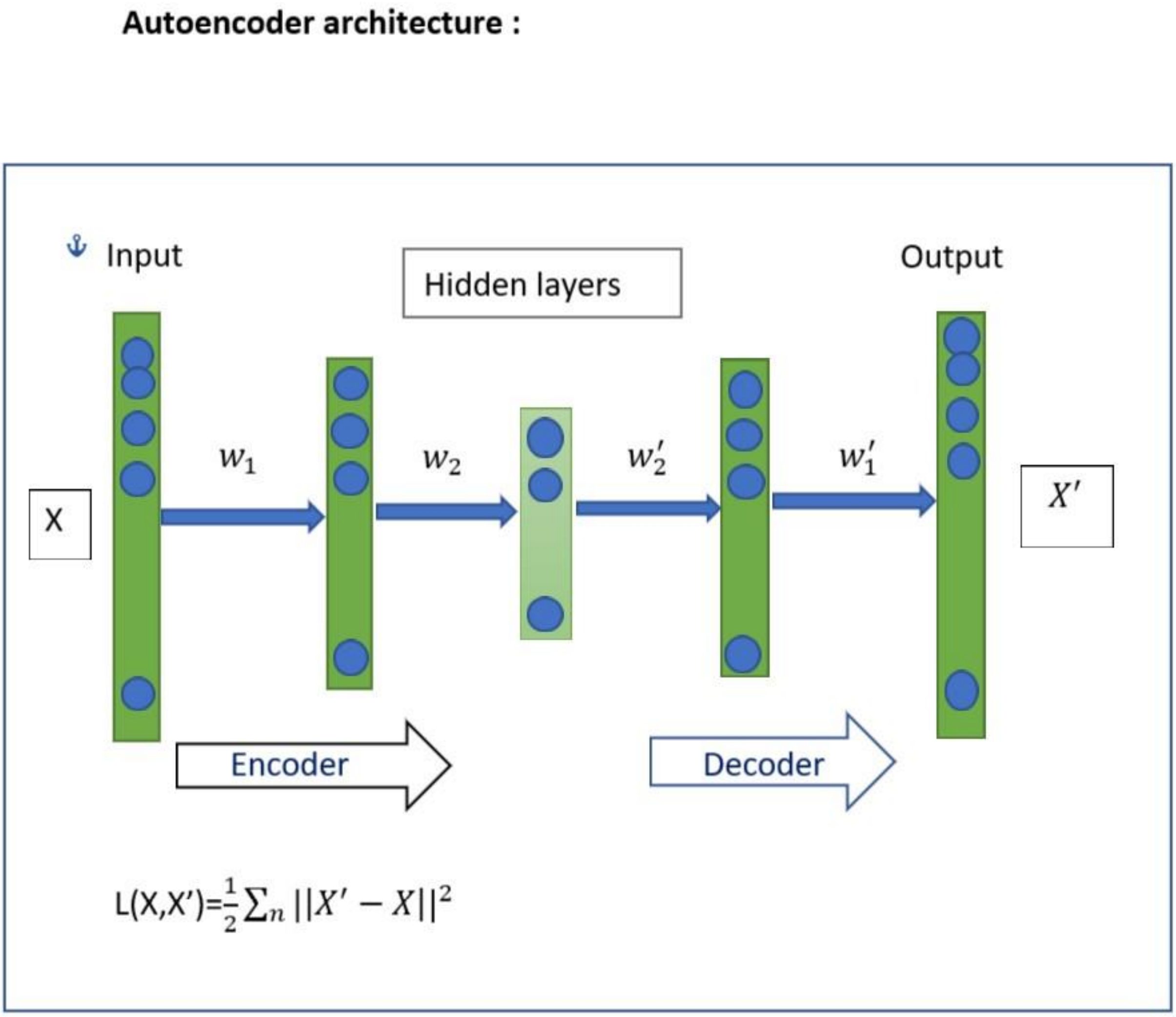
➢ 3.4 Some Deep Learning Reconstruction Methods :

Deep learning is a class of machine learning algorithms that use artificial neural networks to learn data representations. Neural networks saw their resurgence thanks to the availability of large datasets, better algorithms, and powerful computational resources. Since then, deep learning has succeeded in problems like machine translation, visual object recognition, and 3D reconstruction. For a more detailed discussion about deep learning, the reader is requested to refer to **the Deep Learning book by Goodfellow et al.** [36].

➢ 3.4.1 Autoencoder:

Autoencoders, first introduced in Rumelhart et al. [39], are a type of unsupervised neural networks that are used to encode an input into a compressed and useful representation, also called a latent representation, then decode **it** back, such that the reconstructed input is as similar as possible to the original input. The encoding part of the autoencoder is known as an encoder, while the decoding part of the autoencoder is known as a decoder, as shown in Figure 3.3. The other components in an autoencoder architecture are an input layer, a hidden layer, and an output layer. A similarity loss function is applied to the output of the autoencoder and the original input during the training process.

For image data, it is more common to use convolutional layers as the data is not only processed spatially but also reduces parameters in both the encoder and decoder parts of the autoencoder. There are different types of autoencoders like Sparse Autoencoder (SAE), Denoising Autoencoder (DAE), and Variational Autoencoder (VAE). The VAE is an autoencoding approach where the latent representation learns probability distribution parameters for modeling the input data, as proposed by Kingma et al. [40] in 2014. Some prominent application areas in which autoencoders have been used extensively include anomaly detection, dimensionality reduction, image processing, and machine translation.



Machine Translation

Fig 3.3 Autoencoder Architecture

- ### 3.4.2 3D GAN:

The adversarial architecture was first proposed by Goodfellow *et al*. [44], and its main idea is to simultaneously train two models, the generator and the discriminator, and make them both stronger in adversarial learning.

GANs are a powerful recent innovation in the domain of deep learning, and they are used for unsupervised learning. GANs consist of two models, namely, the generative model and the discriminator model. The generative model is responsible for creating fake data instances that resemble your training data. On the other hand, the discriminator model behaves as a classifier that distinguishes between real data instances from the generator's output. The generator attempts to deceive the discriminator by generating real images as far as possible, and the discriminator tries to keep from being deceived. The discriminator penalizes the generator for producing an absurd output. At the initial stages of the training process, the generator generates fake data, and the discriminator quickly learns to tell that it's fake. But as the training progresses, the generator moves closer to producing an output that can fool the discriminator. Finally, if generator training goes well, the discriminator performance worsens because it can't quickly tell the difference between real and fake. It starts to classify the fake data as real, and its accuracy decreases.

3D-GAN [48] applied GAN in learning latent 3D space, and it can generate 3D voxel models from the latent space by extending 2D convolution into 3D convolution. 3D-GAN with WGAN-GP and 3D-IWGAN [49] can generate high-quality 3D models with a more stable training process. Wang *et al*. [50] utilized an encoder-decoder as a generator of the adversarial network to address 3D shape inpainting. Then a long-term recurrent convolutional network (LRCN) was employed to refine the generated results to obtain more complete 3D models in higher resolution. Chen *et al*. [51] proposed a text2shape system that combined 3D generation with natural language processing. The network encoded the text, regarded the results as a condition, and utilized WGAN to decode it into a 3D model related to the input text.

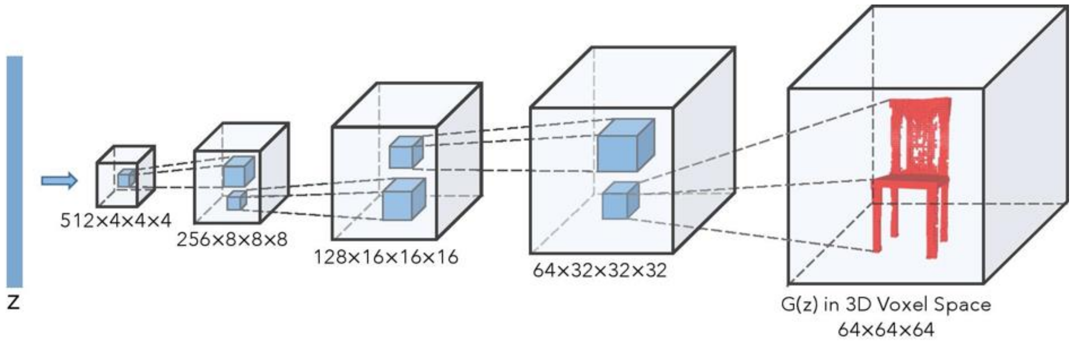


Fig3.4: the structure of the 3D Generative Adversarial Network(3d-GAN)

“3dgan.csail.mit.edu”

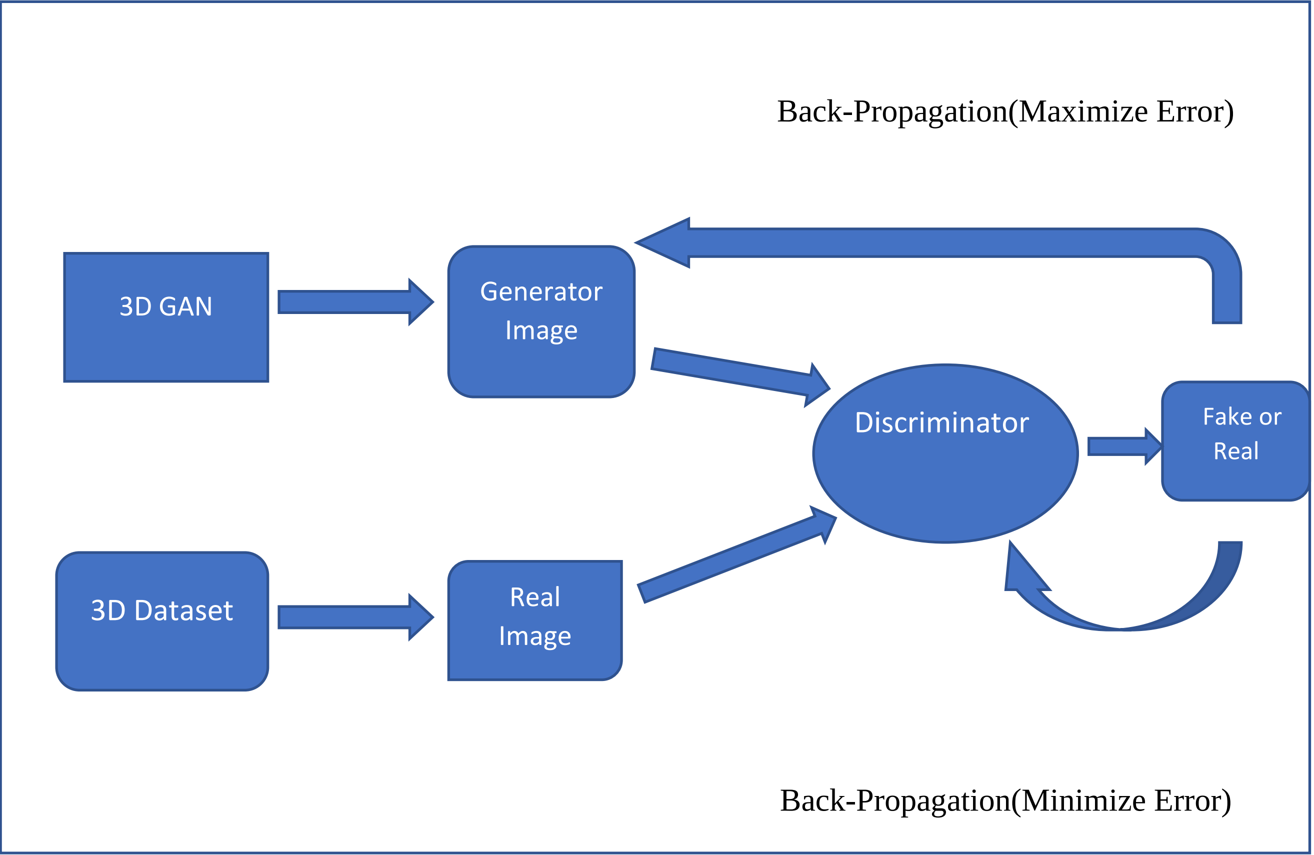


Fig3.5: Shows the basic Generator and Discriminator model of 3D GAN

➢ 3.4.3 3D Recurrent Reconstruction Neural Network:

We introduce a novel architecture named the 3D Recurrent Reconstruction Network (3D-R2N2), which builds upon the standard LSTM and GRU. The goal of the network is to perform both single- and multi-view 3D reconstructions. The main idea is to leverage the power of LSTM to retain previous observations and incrementally refine the output reconstruction as more observations become available

The network is made up of three components: a 2D Convolutional Neural Network (2D-CNN), a novel architecture named 3D Convolutional LSTM (3DLSTM), and a 3D Deconvolutional Neural Network (3D-DCNN) (see Fig. 2). Given one or more images of an object from arbitrary viewpoints, the 2D-CNN first encodes each input image x into low dimensional features T (x). Then, given the encoded input, a set of newly proposed 3D Convolutional LSTM (3D-LSTM) units either selectively update their cell states or retain them by closing the input gate. Finally, the 3D-DCNN decodes the hidden states of the LSTM units and generates a 3D probabilistic voxel reconstruction.

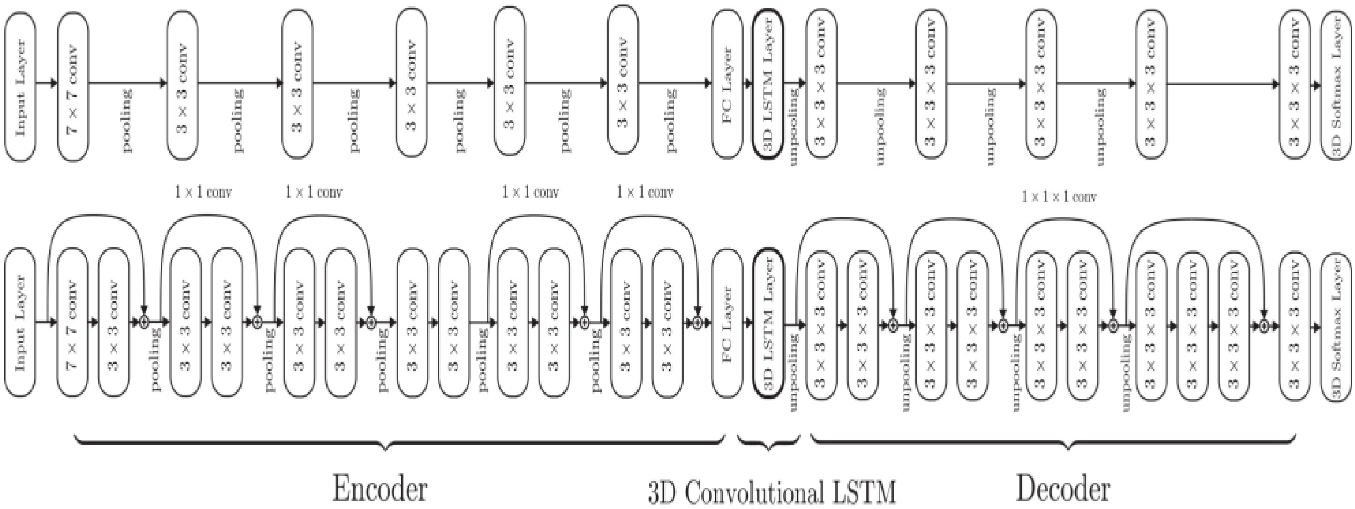


Fig 3.6:shows that a 3D R2N2 consists of an Encoder, a Recurrent Unit (LSTM), and a Decoder. This is the output result of the paper by Kar et al. [30]; here, we have shown the 8 different inputs and their corresponding GT, 3D R2N2.

Inputs Ground Truth 3D R2N2 Kar et al[30] inputs ground truth 3D R2N2 Kar et al[30]

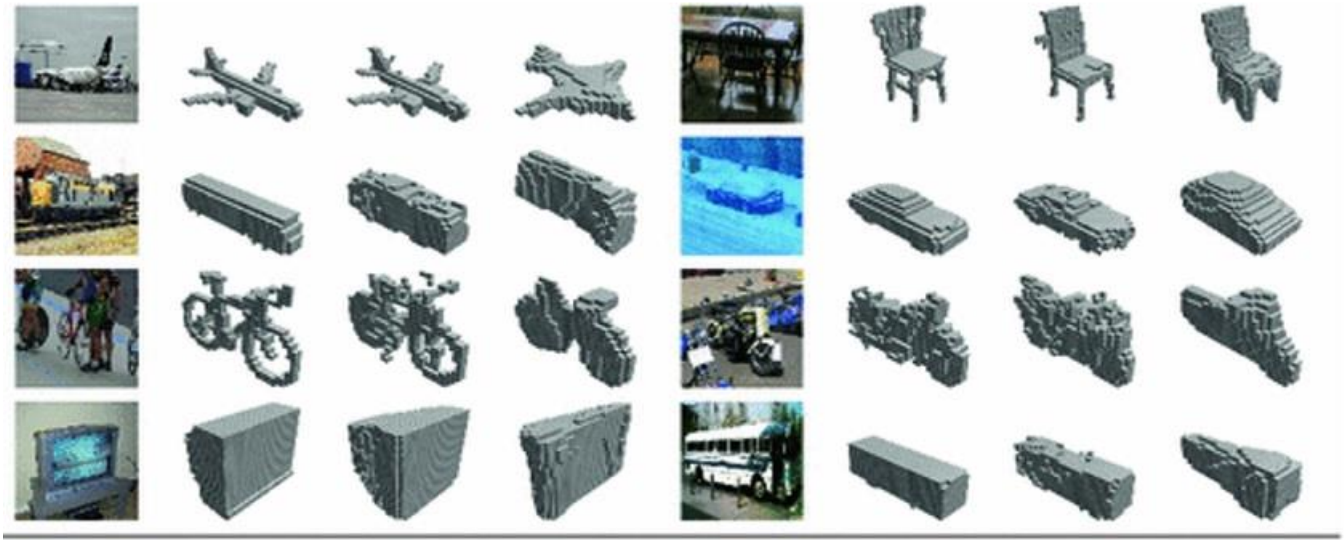


Fig 3.7: shows the occupancy grid output of resolution 32^3 of the different multi-view 3D shape reconstruction methods on the dataset introduced in 3D-R2N2 from kar et al. [30]. The image is taken from "**https://link.springer.com/chapter/10.1007/978-3-319-46484-8_38**".

CHAPTER 4

3D reconstruction from still images using Deep Learning

In this suggested paper, I use Autoencoder as a deep–learning model to reconstruct 3D objects from multiple 2D images. Autoencoder unsupervised learning where neural networks are subject to the task of representation learning. Autoencoder has two blocks: encoder block and decoder block and in between two blocks, we have one imposed bottleneck layer. Encoder block encodes sequentially in each layer and makes a bottleneck layer, a compressed domain representation using feature extraction of the input. The bottleneck layer forces a compressed knowledge representation of the input. Then this bottleneck layer is forwarded to the decoder block and decoder block using deconvolution and up- sampling technique to make the output as same as the input. In the early stage, Autoencoder was used for applications like dimension reduction, image -compression, and image -denoising, and now it is used for image Based 3D reconstruction from single or multiple 2D images.

➢ 4.1 Types of Autoencoder :

a. Sparse Autoencoder

b. Denoising Autoencoder

c. Convolutional Autoencoder.

d.Contracted Autoencoder

4.1.1 Sparse Autoencoder :

- An interesting feature can be learned even when several nodes in the hidden layer are large.

- Introduced sparsity constraint on the hidden layer nodes that penalize activations function
  In a layer
- Network learn encoding and decoding that relies on activating a small number of neuron

4.1.2 Denoising Autoencoder :

- The Autoencoder learns a generalizable encoding–decoding stage.
- An approach – using corrupt data as input but output as uncorrupted original data while training.
- The model can not memorize the training data as input, and the target output differs.
- The model learns a vector field to map the input data towards a low-dimensional manifold.

4.1.3 Convolutional Autoencoder :

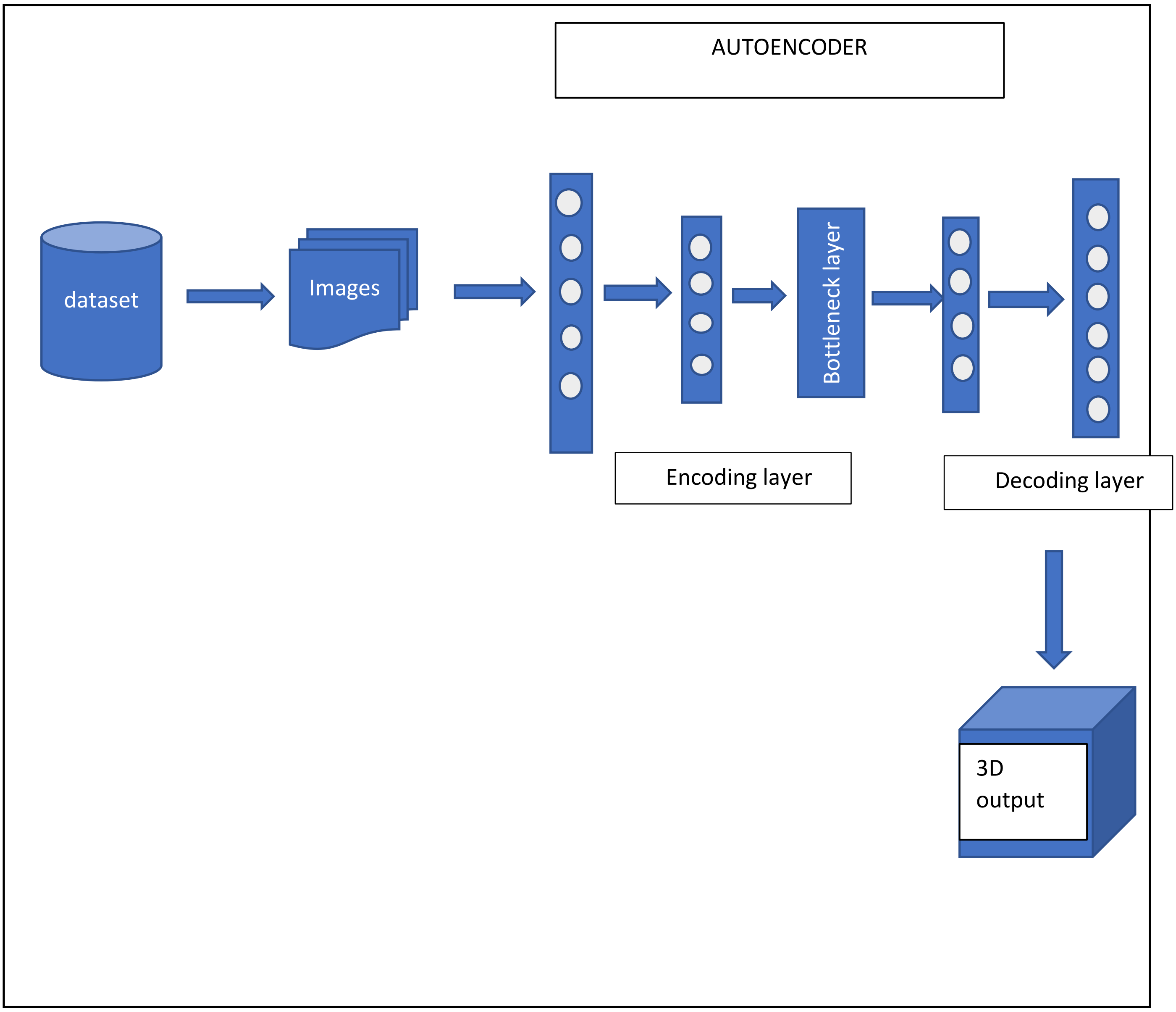
Convolutional Autoencoder learns to encode the input in a set of simple signals and reconstruct the input from then. In addition, we can modify the geometry or generate the reflectance of the image by using a Convolutional autoencoder. In This type of Autoencoder, the Encoder layer is known as the convolution layer, and the decoder layer is also called the deconvolution layer. The deconvolution layer is also known as upsampling or transpose layer.

4.1.4 Contracted Autoencoder :

- For similar inputs -learned encoding (compressed domain representation should also be very similar)
- Hidden layer activation variation with input should also be small

Effectively the models learn to contract a neighborhood of inputs to a small neighborhood of output.

4.2 Planning of the implementation:



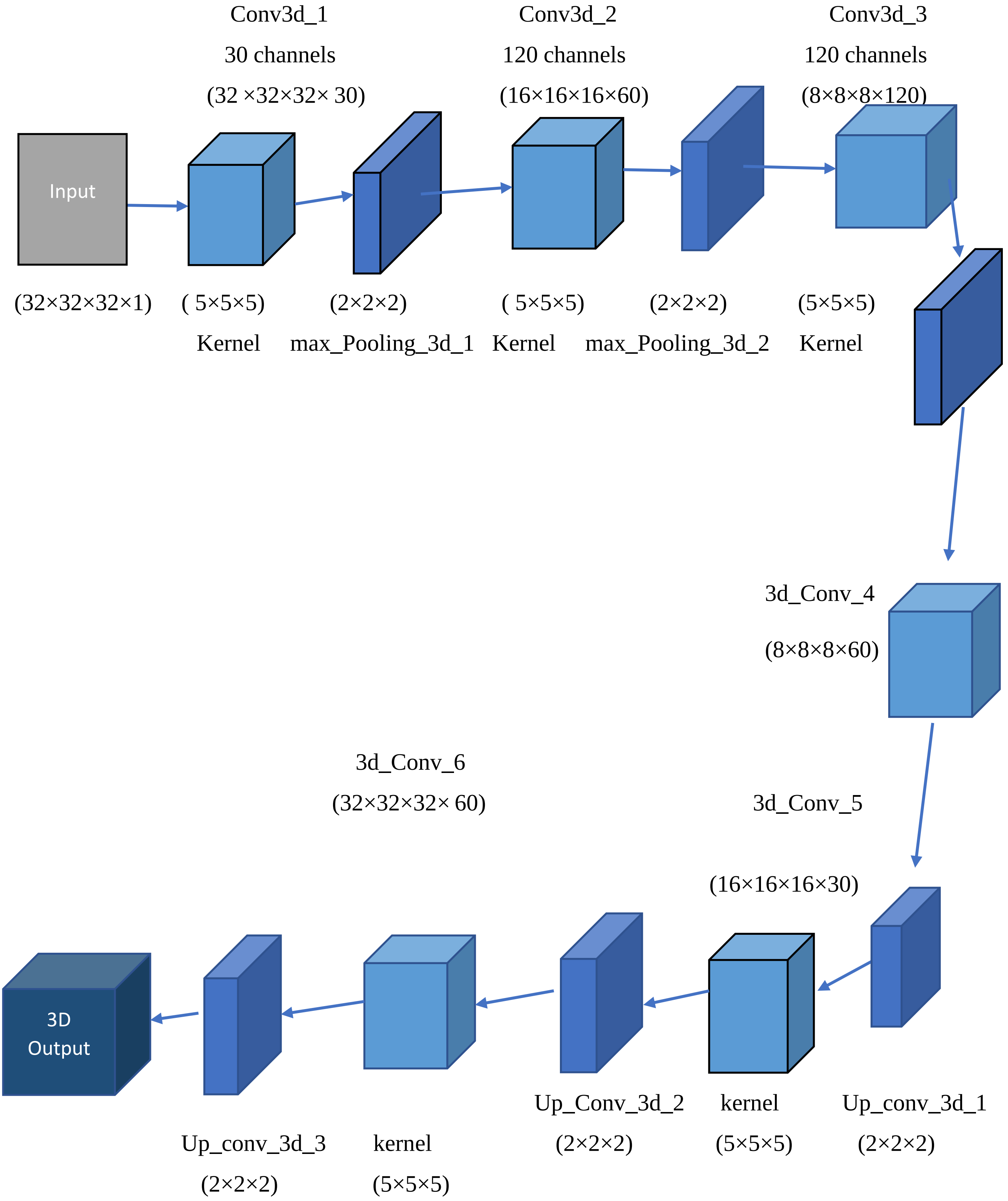


Fig 4.2: the layer-wise architecture of the model.

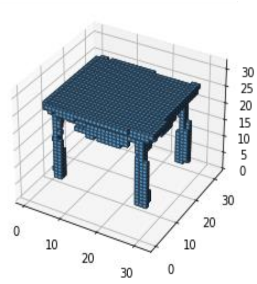
4.3 Steps of the Process:

- **Dataset:** Dataset modelNet10 is collected from the “https://modelnet.cs.princeton.edu” website. The dataset contains the input of 10 categorical objects. These objects are chair, table, desk, bed, sofa dresser, monitor, toilet, nightstand, and bathtub. These objects are in object file format, and the file contains geometry values and surface coordinates.

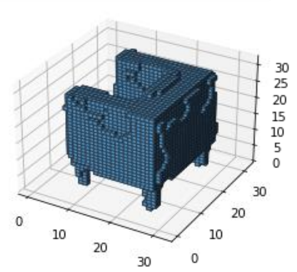
- **Encoder:** This dataset feed into the Autoencoder model. The first block of the model is Encoder, and the Encoder consists of convolutional layers, which convolute layer by layer using kernel and make an activation map of input feature .this process goes on until the model gets the best prominent feature that is called bottleneck layer.

- **Decoder:** After that, this bottleneck layer passes through the decoder layer, which does the opposite of the encoded input feature to reconstruct the 3D object.

- **Output:** some output I am showing here



Table



Chair

CHAPTER 5

EXPERIMENTAL RESULT

5.1 Dataset: The dataset I used is ModelNet10, which has images of 10 objects these are chair, table, desk, sofa, bed, dresser, monitor, nightstand, bathtub, and toilet. The dataset contains 4931 sample images having 176 for testing and 4975 for training, and 176 for validation. So 78% for training and 22% for testing. Details structure are given below.

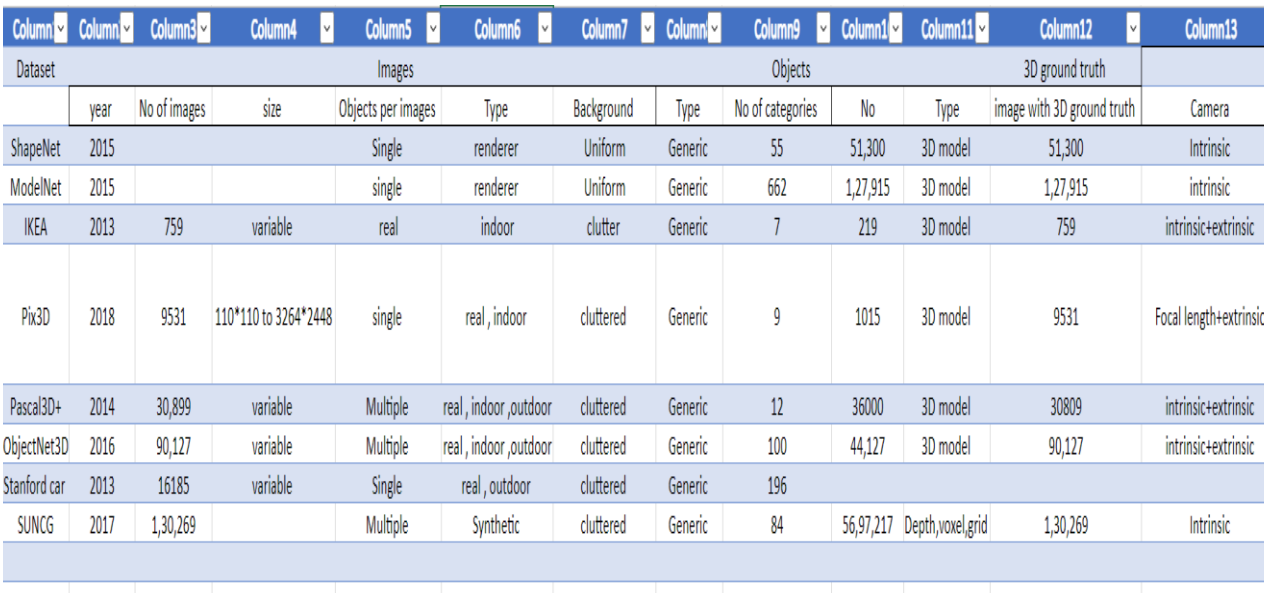
Object name	Train samples	Testing samples
Chair	498	78
table	499	30
desk	499	29
sofa	500	30
bed	524	33
dresser	498	29
monitor	499	29
nightstand	499	29
bathtub	416	36
toilet	499	29
Total	4755	176

The success of deep learning-based 30 reconstruction algorithms depends on the availability of large training datasets.

Supervised techniques require images of their corresponding 3D annotations in the form of 3D models represented as volumetric grids, triangular meshes, point clouds, or depth maps, which can be dense or sparse. On the other hand, weakly supervised and unsupervised techniques rely on additional supervisory signals such as the extrinsic and intrinsic camera parameters and segmentation masks.

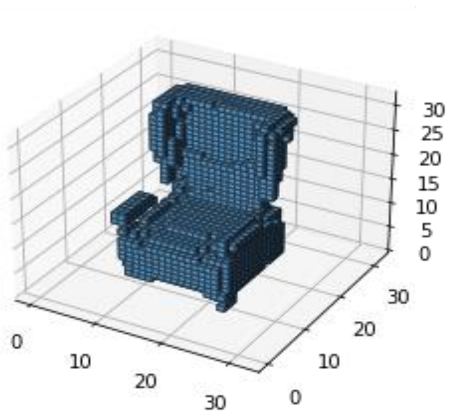
5.2 more datasets and their models : Some of the dataset ,for example ShapeNet , ModelNet10, IEKA, Pix3D , Pascal3D , ObjectNet3D ,

Stanford Car, SUNCG details are given. These details are like their publishing year, no. of images per dataset, size, objects per image, type, background, number of categories, **3D GT(ground truth) number**, 3D GT type, images with 3D ground Truth and camera provided, these datasets can be used for image-based 3D reconstruction.

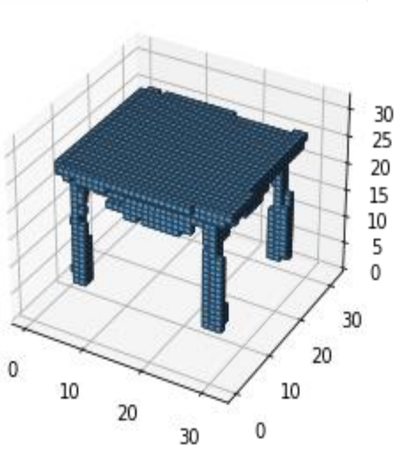


Column	Column	Column3	Column4	Column5	Column6	Column7	Column	Column9	Column1	Column11	Column12	Column13
Dataset				Images				Objects			3D ground truth	
	year	No of images	size	Objects per images	Type	Background	Type	No of categories	No	Type	image with 3D ground truth	Camera
ShapeNet	2015			Single	renderer	Uniform	Generic	55	51,300	3D model	51,300	Intrinsic
ModelNet	2015			single	renderer	Uniform	Generic	662	1,27,915	3D model	1,27,915	intrinsic
IKEA	2013	759	variable	real	indoor	clutter	Generic	7	219	3D model	759	intrinsic+extrinsic
Pix3D	2018	9531	110*110 to 3264*2448	single	real , indoor	cluttered	Generic	9	1015	3D model	9531	Focal length+extrinsic
Pascal3D+	2014	30,899	variable	Multiple	real , indoor ,outdoor	cluttered	Generic	12	36000	3D model	30809	intrinsic+extrinsic
ObjectNet3D	2016	90,127	variable	Multiple	real , indoor ,outdoor	cluttered	Generic	100	44,127	3D model	90,127	intrinsic+extrinsic
Stanford car	2013	16185	variable	Single	real , outdoor	cluttered	Generic	196				
SUNCG	2017	1,30,269		Multiple	Synthetic	cluttered	Generic	84	56,97,217	Depth,voxel,grid	1,30,269	Intrinsic

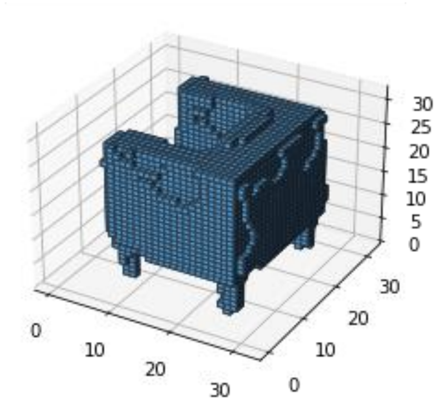
5.3 Outputs: output is in a 3D grid/voxel format. Some output figures are given below:



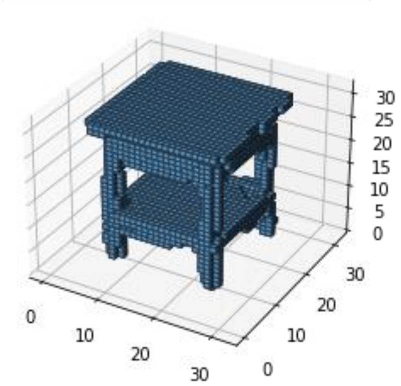
Chair



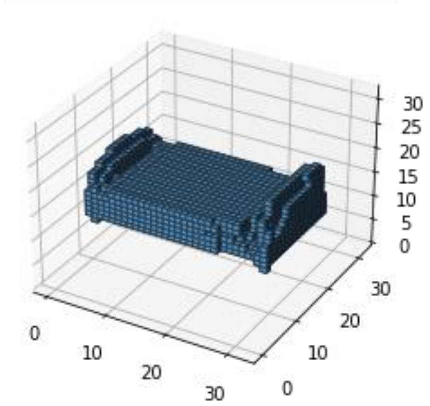
Table



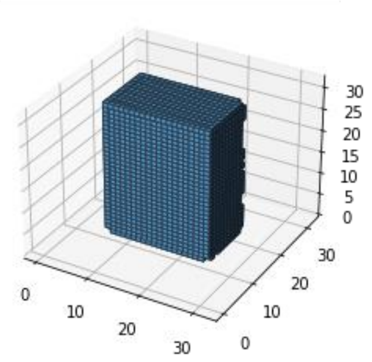
Sofa



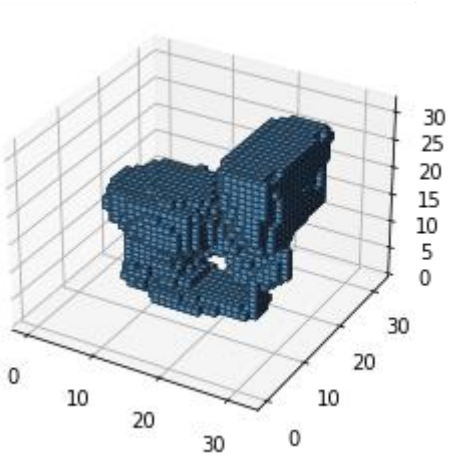
Table



Bed



Book Dresser



Toilet

5.4 Confusion matrix: A confusion matrix sometimes represents classifier performance based on the four values **true positive, true negative, false positive, and false negative. These are plotted in a table as.**

		Actual Value	
		Negative	Positive
Predicted value	Positive	FP	TP
	Negative	TN	FN

5.5 Accuracy: The formula is:

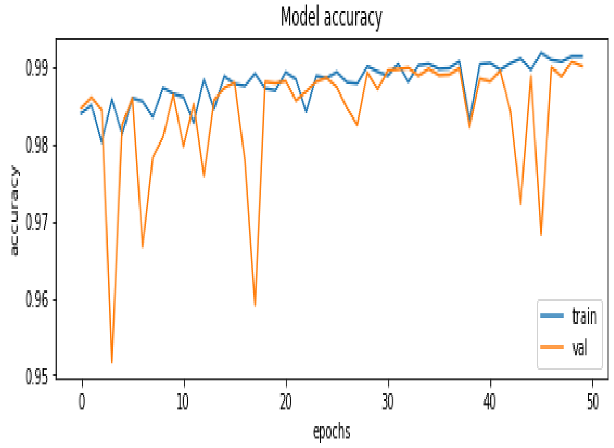
$$= \frac{Total\ number\ of\ correct\ prediction}{total\ number\ of\ prediction}$$

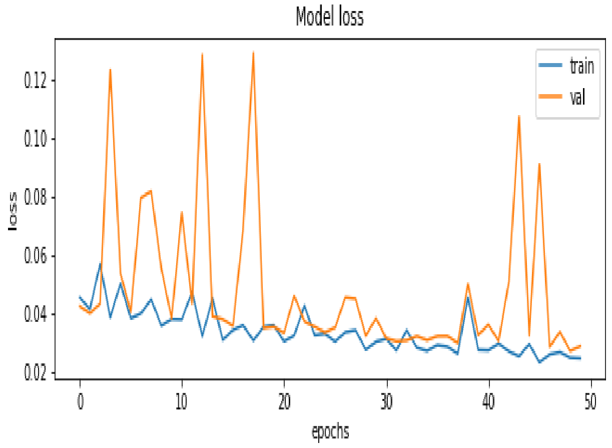
This can be written as from the above table

$$= \frac{TP+TN}{TP+TN+FN+FP},$$

Where TP=True positive, TN=True Negative, FN=False Negative, FP=False Positive.

Validations accuracy is almost 98%, and validation loss is 0.04.





5.6 Precision: Precision is how many positive predictions are made correctly. This can be written as

$$= \frac{True\, positive}{true\, positive + false\, positive}$$

5.7 Recall: recall is how many positive cases the classifier correctly predicted over all the positive cases in the data. The formula is:

$$= \frac{True\, positive}{true\, positive + false\, negative}$$

CHAPTER 6.

FUTURE SCOPE & CONCLUTION

6.1 Future research direction: In the light of extensive research undertaken in the past 5 years, image-based 3D reconstruction using deep learning techniques has achieved promising results. The topic is still in its infancy, and further development is yet to be expected. We present some of the current issues and highlight directions for future work.

6.2 Training data issue: the success of deep learning techniques depends heavily on the availability of training data. Unfortunately, the publicly available datasets that include images and their 3D annotations are small compared to the training datasets used in tasks such as clas-sification and recognition. 2D supervision techniques have addressed the lack of 3D training data. Many of them rely on silhouette-based supervision, and thus they can only reconstruct the visual hull. As such, we expect to see in the future more papers proposing new large- scale datasets, new weakly-supervised and unsupervised methods that leverage various visual cues, and new domain adaptation techniques where networks trained with data from a certain domain, *e.g.,* synthetically rendered images, are adapted to a new domain, *e.g.,* in-the-wild images, with minimum retraining and supervision. Research on realistic rendering techniques that can close the gap between real and synthetically rendered images can poten-tially contribute to addressing the training data issue.

6.3 ***Generalization to unseen objects:*** Most state-of-the- art papers split a dataset into three subsets for training, validation, and testing, *e.g.,* ShapeNet or Pix3D, then report the performance on the test subsets. However, it is not clear how these methods would perform on a completely unseen object/image category. The ultimate goal of the 3D reconstruction method is to be able to reconstruct any arbitrary 3D shape from arbitrary images. Learning-based 3D reconstruction performs well only on images and objects spanned by the training set. Some recent papers started to address this issue. However, an interesting direction for future research would be to combine traditional and learning-based techniques to improve the generalization of the latter methods.

6.4 ***Fine-scale*** **3D** ***reconstruction*****:** Current state-of-the-art tech-niques can recover the coarse 3D structure of shapes. Although recent works have significantly improved the res-olution of the reconstruction by using refinement modules, they still fail to recover thin and small parts such as plants, hair, and fur.

6.5 ***Specialized instance reconstruction:*** We expect in the fu-ture to see more synergy between class-specific knowledge modeling and deep learning-based 3D reconstruction to leverage domain-specific knowledge. There is an increasing interest in reconstruction methods specialized in specific classes of objects such as human bodies and body parts (which we need to cover in this survey briefly), vehicles, animals [52], trees, and build-ings. Specialized methods exploit prior and domain-specific knowledge to optimize the network architecture and training process. As such, they usually perform better than the general framework. However, similar to deep learning- based 3D reconstruction, modeling prior knowledge, *e.g.,* by using advanced statistical shape models [53],[54] requires 3D annotations, which are not easy to obtain for many classes of shapes, *e.g.,* animals in the wild.

6.6 Conclusion: This paper comprehensively surveys the past eight years' developments in image-based 3D object reconstruction using deep learning techniques. We classified the state-of-the-art into volumetric, surface-based, and point-based techniques. We then discussed methods in each category based on their input, the network archi-tectures, and the training mechanisms they use. We have also discussed and compared the performance of some key methods. This survey focused on methods that define 3D recon-struction as the problem of recovering the 3D geometry of objects from one or multiple RGB images. There are, however, many other related problems that share similar solutions. To improve the network, we need to open up the model and check the parameters, hyperameter, and weights to maintain the back propagation algorithm and reduce the error. We can use multiple models to improve the network. The closest topics include depth reconstruction from RGB images, which deep learning has recently addressed.

REFERENCES

[1] image-based 3D reconstruction: state of art and Trends in the deep learning era. Xian-Feng han*, hamid laga*, Mohammed Bennamoun senior member IEEE.

[2] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning . MIT Press, 2016.

[3] A. Laurentini, “The visual hull concept for silhouette-based image understanding,” IEEE TPAMI, vol. 16, no. 2, pp. 150–162, 1994.

[4] Berger, M.; Tagliasacchi, A.; Seversky, L.; Alliez, P.; Guennebaud, G.; Levine, J.; Sharf, A.; Silva, C. A survey of surface reconstruction from point clouds. Comput. Graph. Forum **2017**, 36, 301–329.

[5] Goel, S.; Bansal, R. Surface Reconstruction Using Scattered Cloud Points. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2013**, 3, 242–245.

[6] Lee, P.; Huang, J.; Lin, H. 3D model reconstruction based on multiple view image capture. In Proceedings of the 2012 International Symposium on Intelligent Signal Processing and Communications Systems, Tamsui, Taiwan, 4–7 November 2012; pp. 58–63.

[7] Chang, A.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. arXiv **2015**, arXiv:1512.03012

[8] Liu, J.; Yu, F.; Funkhouser, T. Interactive 3D modeling with a generative adversarial network. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 126–134.

[9] Zou, C.; Colburn, A.; Shan, Q.; Hoiem, D. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2051–2059.

[10] Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.

[11] Liu, S.; Acosta-Gamboa, L.; Huang, X.; Lorence, A. Novel low cost 3D surface model reconstruction system for plant phenotyping. J. Imaging **2019**, **3**, 39

[12] Gwak, J.; Choy, C.; Chandraker, M.; Garg, A.; Savarese, S.Weakly supervised 3d reconstruction with adversarial constraint. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 263–272.

[13] Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920

[14] Choy, C.; Xu, D.; Gwak, J.; Chen, K.; Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In Proceedings of the European Conference On Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 628–644.

[15] M. Denninger and R. Triebel. "3D Scene Reconstruction from a Single Viewport." In: *Proceed- ings of the European Conference on Computer Vision (ECCV).* 2020.

[16] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann. "DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction." In: *Advances in Neural Information Processing Systems* 32. 2019.

[17] **H.** Fan, H. Su, and L. Guibas. "A Point Set Generation Network for 3D Object Reconstruction from a Single Image." In: *Computer Vision and Pattern Recognition.* 2017.

[18] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images." In: *ECCV.* 2018.

[19] **J.** J. Park, P. Florence, **J.** Straub, R. Newcombe, and S. Lovegrove. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation." h1: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2019.

[20] M. Tatarchenko, A. Dosovitskiy, and T. Brox. "Octree Generating Networks: Efficient Convo- lutional Architectures for High-resolution 3D Outputs." In: *IEEE International Conference on Computer Vision (ICCV).* 2017.

[21] S. R. Richter and S. Roth. "Matryoshka Networks: Predicting 3D Geometry via Nested Shape Layers." In: *CVPR.* 2018.

[22] H. Su, S. Maji, E. Kalogerakis, and E.G. Learned-Miller. "Multi-view convolutional neural networks for 3d shape recognition." In: *Proc. ICCV.* 2015.

[23] 0. Ozye il, V. Voroninski, R. Basri, and A. Singer. "A survey of structure from motion." In: *Acta Numerica* (2017).

[24] A. Kar, C. Hane, and J. Malik. "Learning a Multi-View Stereo Machine." In: *NIPS.* 2017.

[25] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun. "Pix2Vox++: Multi-scale Context-aware 3D Object Reconstruction from Single and Multiple Images." In: *International Journal of Computer Vision (IJCV)* (2020).

[26] 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." In: *Proceedings of the European Conference on Computer Vision (ECCV).* 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In: *"CVPR".* 2015.

[28] M. Denninger, M. Sundermeyer, D. Winkelbauer, D. Olefir, T. Hodan, Y. Zidan, M. Elbadrawy, M. Knauer, H. Katam, and A. Lodhi. "BlenderProc: Reducing the Reality Gap with Photorealistic Rendering." In: *Robotics: Science and Systems (RSS).* 2020.

[29] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang. "Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images." In: *ICCV.* 2019.

[30] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. "Semantic Scene Completion from a Single Depth Image." In: *CVPR* (2017).

[31] R. Spezialetti, D. J. Tan, A. Tonioni, K. Tateno, and F. Tombari. "A Divide et lmpera Approach for 3D Shape Reconstruction from Multiple Views." In: 2020 *International Conference on 3D Vision (3DV).* 2020.

[32] B. Yang, S. Wang, A. Markham, and N. Trigoni. "Robust Attentional Aggregation of Deep Feature Sets for Multi-view 3D Reconstruction." In: *IJCV.* 2019.

[33] F. Yagubbayli, A. Tonioni, and F. Tombari. *LegoFormer: Transformers for Block-by-Block Multi- view* 3D *Reconstruction* .

[34] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images." In: ECCV. 2018.

[35] C. R. Qi, H. Su, K. Mo, and L. **J.** Guibas. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." In: CVPR (2016).

[36] C.R. Qi, L. Yi, H. Su, and L. J. Guibas. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space." In: (2017).

[37] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge university press, 2003.

[38] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. "MeshCNN: A Network with an Edge." In: ACM Transactions on Graphics (TOG) (2019).

[39] Y. Liao, S. Donne, and A. Geiger. "Deep Marching Cubes: Learning Explicit Surface Repre- sentations." In: Conference on Computer Vision and Pattern Recognition (CVPR). 2018.

[40] D. E. Rumelhart and J. L. McClelland. "Learning Internal Representations by Error Propagation." In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations.* 1987

[41] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. "Visualizing the loss landscape of neural nets." In: *Advances in Neural Information Processing Systems.* 2018

[42] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes." In: *2nd International Con- ference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.* 2014.

[43] W. E. Lorensen and H. E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm." In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques.* 1987.

[44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, in Advances in Neural Information Processing Systems (NIPS) (Montreal, Canada), 2014, pp. 2672-2680.

[45] A. Radford, L. Metz, and S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. arXiv preprint arXiv: 1511.06434

[46] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein gan, 2017. arXiv preprint arXiv: 1701.07875

[47] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A.C. Courville, Improved training of wasserstein gans, in Advances in Neural Information Processing Systems (NIPS) (Long Beach, California), 2017, pp. 5767-5777. acmid: 3295327

[48] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, in Advances in Neural Information Processing Systems (NIPS) (Barcelona, Spain), 2016, pp. 82-90. acmid: 3157106

[49] E. Smith and D. Meger, Improved adversarial systems for 3d object generation and reconstruction, in Conference on Robot Learning (CoRL) (Mountain View, California), 2017, pp. 87-96.

[50] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, Shape inpainting using 3d generative adversarial network and recurrent convolutional networks, in Proceedings of the International Conference on Computer Vision (ICCV) (Venice), 2017, pp. 2317-2325.

[51] K. Chen, C.B. Choy, M. Savva, A.X. Chang, T. Funkhouser, and S. Savarese, Text2shape: generating shapes from natural language by learning joint embeddings, in Proceedings of Asian Conference on Computer Vision (ACCV) (Cham), 2018, pp. 100-116.

[52] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning Category-Specific Mesh Reconstruction from Image Collections," *ECCV,* 2018.

[53] G. Wang, H. Laga, J. Jia, N. Xie, and H. Tabia, "Statistical modeling of the 3d geometry and topology of botanical trees," *CGF,* vol. 37, no. 5, pp. 185-198, 2018.