3D Vectorization and Generation using Generative Adversarial Networks

Thesis Submitted By:
Aamir Fayyaz
01-243172-001

Supervised By:
Dr. Sumaira Kausar

A dissertation submitted to the Department of Computer Science, Bahria University, Islamabad as a partial fulfillment of the requirements for the award of the degree of Masters in Computer Science

Session (2017-2019)

Bahria University
Discovering Knowledge

MS-13

Thesis Completion Certificate

Scholar's Name: Aamir Fayyaz Registration No. 53728
Programme of Study: MS (CS)
Thesis Title: 3D Vectorization and Generation using Generative Adversarial Networks

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for Evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 13% that is within the permissible limit set by the HEC for the MS/MPhil degree thesis.
I have also found the thesis in a format recognized by the BU for the MS/MPhil thesis.

Principal Supervisor's Signature: ____________

Date: 14th June 19 **Name:** Dr. Sumaira Kausar

Bahria University
Discovering Knowledge

MS-14A

Author's Declaration

I, Aamir Fayyaz hereby state that my MS thesis titled "3D Vectorization and Generation using Generative Adversarial Networks"

is my own work and has not been submitted previously by me for taking any degree from this university

Bahria University Islamabad Campus

or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw/cancel my PhD degree.

Name of scholar: AamirFayyaz
Date: July 2019

Bahria University
Discovering Knowledge

MS-14B

Plagiarism Undertaking

I, solemnly declare that research work presented in the thesis titled "3D Vectorization and Generation using Generative Adversarial Networks" is solely my research work with no significant contribution from any other person. Small contribution / help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Bahria University towards plagiarism. Therefore I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD degree, the university reserves the right to withdraw / revoke my PhD degree and that HEC and the University has the right to publish my name on the HEC / University website on which names of students are placed who submitted plagiarized thesis.

Student / Author's Sign:

Name of the Student:
Aamir Fayyaz

Abstract

Generating and understanding the 3D shape of objects in the world is a crucial step for many areas of robotics. Across object categories, shapes are used for classification. Within each category, fine shape details and textures contribute to successful manipulation. Existing generation methods usually rely on sketches and meshes, new objects generated by obtaining and merging patterns and components from the database. The major drawback of such techniques is that they cannot produce a complete 3D object from a 2D image. Given a 2D image of a chair taken from front view missing back legs, in its corresponding 3D object this information will not be present.The architecture of the proposed model, employees 3D Vectorization and Generation using Generative Adversarial Network, that forms 3D-objects by leveraging the probabilistic space taking advantages from novel developments in volumetric convolutional networks and generative adversarial nets. The main advantages of the proposed method are: It uses an adversarial model that is capable of implicitly getting the object formation and to produce quality 3d objects. A mapping of 3D-object is learned from a low dimensional probabilistic space. The adversarial discriminator gives a compelling 3D shape descriptor.

Acknowledgments

I am very grateful to Almighty Allah, who created this universe. Through his blessings and the knowledge he has given to me, I was able to complete this work. I am obliged to Dr. Sumaria Kausar for her supervision, knowledge, and help through every step of my research work. I would like to say special thanks to my dearest friend, Mr. Munib ul Hassan for his support and motivation in the completion of this work.

I am very thankful to my family for providing me the time, and space needed to accomplish this task. I am thankful for their love and prayers. I pray that may Allah favor them with his most numerous blessings. Ameen

AAMIR FAYYAZ
Bahria University Islamabad, Pakistan

July 2019

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AI	Artificial Intelligence
CNN	Convoluational Neural Network
INN	Interpreter Neural Network
GAN	Generative Adversarial Network
RNN	Recurrent Neural Network
Conv-Net	Convulational Neural Network
CNN	Convulational Neural Network
3D-R2N2	3D Recurrent Reconstruction Neural Network
DCGAN	Deep Convolutional Generative Adversarial Networks
VCN	Volumetric Convolutional Networks

Chapter 1

Introduction

1.1 Introduction

Understanding 3D structure of an object is a challenging task for many areas of computer science like computer vision, artificial intelligence, robotics, and computer graphics. 3D objects are used in many applications of computer science like VR(virtual reality), robotics interaction. In the past, to extract 3D structure from an image different sensors were used(real-scene cameras). Major drawback of these techniques they required a single scene to be captured from different angles to gain depth of the image. Though, It's not feasible to examine each surface of an image before remodeling this guides to inadequate 3D shape with extended holes.

Standard procedures utilize the traditional cost-effective depth-sensing tools like cameras to retrieve 3D object formation from occupied intensity pictures. These methods normally need many depths pictures from various viewing points of an object to appraisal the entire 3D composition [2, 3, 4]. But, in tradition, it's not always possible to examine all surfaces of the subject before regeneration, this drives towards the inadequate 3D objects having occluded areas and big openings. Additionally, obtaining and preparing many depth scenes to demand extra computation power, that is not perfect in several models that demand real-time execution.

Interestingly, people are surprisingly great at answering such doubtfulness by inherently leveraging antecedent experience. For instance, given the picture of the table with two back legs blocked by front legs, people undoubtedly competent to infer the most suitable shape beside apparent pieces. Current progress in deep neural networks and data-based methods give encouraging outcomes.

Figure 1.1: Figure of same table from different view point [1]

Figure 1.1 shows the image of a table from two different preparatives. One on the right side missing the back legs, and one on the left side all four legs are visible. Generating a 3D object for an image on the right side is a challenging task as it contains missing information (back legs are not visible) on the other hand generating a 3D object for an image on the left side is a less challenging task as contains complete information (all four legs are visible). By using the the approaches [2, 3, 4] on these images will produce the different results as they contain different level of information.

We attempt to resolve the dilemma of divining the entire 3D composition of the object. It's a noteworthy job as a partial view of an object (detail of picture from a single perspective) correlate to an endless amount of feasible illustrations. Conventional restoration techniques normally apply interpolation procedures like plane-fitting, Laplacian openings filling [5, 6] to predict the underlying 3D composition. Though, they could only retrieve very inadequate closed or disappeared areas, e.g., tiny openings or cracks due to image compression and decompression artifacts, sensor noise, and inadequate infrastructure details.

While most recent deep learning techniques [9, 10] for 3D-shape restoration from a particular intensity to obtain promising outcomes, are restricted to inadequate intentions, normally at the rate of 32^3 pixel grids. This cultured 3D composition leads to granular and fallacious. To form greater perseverance 3D objects with effective reckoning, Octree description has been lately presented in [11, 12]. Though, raising the frequency of product 3D object also unavoidably impose a big test to acquire the geometric specifications for a high-resolution 3D image, that is still being investigated.

Lately, deep generative models accomplish impressively greater achievement in modeling complex high-dimensional data patterns, with Generative Adversarial Networks (GANs) [13] because it is a robust architecture for generative learning, including an image as well as text creation [14], and also latent space learning [15]. In preceding years, researchers used generative patterns to determine possible space they can describe 3D object forms, in succession doing jobs like novel image creation, object grouping, identification and retrieving shapes.

Forming 3D objects in a generative-adversarial mode allows extra unique benefits. One, it becomes feasible to inspect unique 3D shapes of a probabilistic latent space like Gaussian distribution or the normal distribution. Tow, discriminator of generative-adversarial architecture offers informational characteristics for 3D object identification, From a distinctive viewpoint, alternately learning a singular characteristic illustration for both creating and identifying objects.

The GAN structure gives every object it has undergone to a locality in a latent space. The rest of this space can enable the formation of new lifelike objects, it is only possible when the GAN training has converged fortunately to the data distribution. For uncomplicated data distributions, like objects from a singular group, a simplistic network architecture such as linear interpolation may answer. Complicated data distributions, like those made by the cutting changes among various object classes, guide to difficult training problems.

We tried to illustrate the likely solution for forming objects that are new and vivid by using the general-adversarial setting. Our strategy merges the advantages of both general-adversarial modelings [13] and VCN [16]. Distinctive from common heuristic models, in such a way that generative models present an adversarial discriminator, for the purpose of distinguishing between fake and authentic objects. It can be a helpful architecture for generating 3D shapes. As 3D shapes are well structured, unlike the self-supporting heuristic ones. Generative models have the power to learn and generate 3D shapes.

1.2 Motivation and Problem Description

Generating 3D objects is a challenging task for machine learning. With the emergence of Neural Networks and later Deep Neural Networks it attracted researchers. More recently the Generative Adversarial Networks (GAN's) were given a new direction to the task of image generation and many others like text generation, poetry generation. Take advantage of the generative power of the GAN's we tried to use them for generating 3D objects.

1.3 Research Contribution

The key contributions to this research are as follow:

- An automated system is developed to generate 3D objects.
- New 3D objects can be generated by merging two or more objects. It will lead in creating new designs, that will help the designers to craft new designs.
- A publicly available dataset is used to validate the findings of this research.
- Worked on two objects of the dataset tables and chairs.

1.4 Thesis Organization

Whole work is divided into 5 parts. In chapter 2 review of Literature is presented. The methodology of the proposed solution is discussed in chapter 3. The complete analysis of experiments and results are discussed in chapter 4. Chapter 5 summarises the whole work.

Chapter 2

Literature Review

2.1 Literature Review

Generating and understanding 3D objects is a challenging task in AI, robotics and many other areas of computer vision. In the past researchers used various techniques to generate 3D objects. With the emergence of deep learning, the field attracted more interest of the researchers and a lot of work done by using deep learning. With the emergence of GAN's, it has given totally a new direction to 3D object generation and classification. Following is the overview of the different techniques used by the researchers to generate and classify 3D objects.

2.1.1 Techniques Based on Sketch and Skeleton

Constructing 3D objects remained a vast domain of research within Computer Vision and Artificial Intelligence. Aron et al. used plane fitting that rebuilds the minute unavailable parts [17]. Pauly et al., M.R. Oswald et al. and Sipiran et al. used symmetry of shapes to fill in holes [18, 19, 20]. Such methods expected to disappoint when desiring or closed areas are comparatively large. A related attachment pipeline support database priors. By inputting a partial shape, alike or suitable 3D model is regained and joined by incomplete scan in [21, 22]. Though, these strategies explicitly consider the database comprises same or very related shapes, therefore being incapable of inferring the new objects or classes.

2.1.2 Techniques Based on Deep Learning

H. Su et al. stated a collective embedding space filled with 3D as well as 2D objects, where the distance among the enclosed entries indicates the correlation within the 3D and 2D objects [23]. The embedding space constructed by using a 3D object correlation means.

R Girdhar et al. proposed an architecture known as TL-embedding capable of learning embedding space [24]. It cotains two components, 1 auto-encoder, 2 CNN. The introduced model can also be utilized to divine voxel from 2D shapes.

As point clouds are important data structures. Qi et al. designed a model for classification of objects that uses point clouds [25]. For the object categorization job, the input point cloud is either inspected directly from a shape or pre-segmented from a picture point cloud. It outputs k scores for all k number of groups.

H. Su et al. used 3D models to train CNN for viewpoint estimation [26]. The foremost objective of their study to determine the viewpoint for the 2D input image. The viewpoint appraisal dilemma formalized as organizing camera circumrotation parameters within fine-grained classes. By utilizing fine-grained viewpoint grouping formulation, an evaluation was informational and reliable.

Wu et al. stated a deep convolutional model to describe the geometric 3D appearance being a likelihood distribution regarding twofold variables on the 3D voxel grid [9]. The proposed model capable of collectively understands and restore objects of an indivisible view 2.5D extent map.

Xiang et al. presented Deep Pano an extensive explanation for 3D shape grouping and retrieval [27]. Picturesque scenes are created of 3D objects, and illustrations are acquired. Then a modification of CNN is particularly meant for understanding the deep illustrations straight from these compositions is applied. Distinctive from standard CNN, a max-pooling in a row-wise manner is interpolated among convolution and entirely coupled layers, affecting the extracted descriptions invariant to the circle about a primary axis. The weakness of this model is that it is alike to various past view-based methods, expecting the primary axes of 3D-objects, that can be disappointing in identifying 3D objects by severe deformation.

Choy et al. presented a novel RNN structure they named it 3D-R2N2 [28]. It detects a mapping from pictures of objects so the under consideration 3D molds of a large number of polymerized data. It reads single or multiple pictures of an object from a random view-point and returns restoration regarding object in a state of a 3D possession grid. This model doesn't need either picture explanations or object group tags for training or testing.

Sharma et al. proposed an auto-encoder based model that acquires volumetric illustration from noisy data by calculating voxel possession grids [29]. Given a group of shapes of different objects and their various postures, it detects the image patterns of different groups by prophesying the disappeared voxels of remaining. Next, they used the learned embedding for image identification and incorporation activities. They used the production skills of auto-encoder structure for divining an improved copy of contaminated descriptions.

Rezende et al. introduced a structure that can learn effective extensive generative patterns of 3D compositions, and revive those compositions from 2D pictures through probabilistic assumption [30]. Their model manipulates the generative method, which first forms the 3D illustration and then projects to the region of the examined data.

Yangyan et al. introduced a collective embedding space filled by both 3D objects and 2D shapes of objects, where the gaps among embedded objects display correlation among the objects [31]. The collective embedding expedites similarity among objects of each kind and permits cross-modality extraction. They constructed the embedding location using 3D object correlation ratio since 3D objects remain clear and perfect as compared to representation within images, heading to more vigorous interval metrics. They apply CNN to clarify objects by softening confusing parts. It has the ability to fit the image to a particular location in embedding space, therefore, it comes very closer to the 3D object one represented within the image. This capacity of CNN is achieved by the cooperation with a huge number of training examples containing images manufactured from 3D objects.

2.1.3 Techniques Based on GAN's

I. Goodfellow proposed the structure of GAN [13]. Denton et al. introduced the generative parametric structure that has the ability to generate great quality scenes like natural photographs [32]. Their method utilizes a cascade of CNN inside the Laplacian pyramid ("The Laplacian pyramid is a linear invertible image representation consisting of a set of band-pass images, spaced an octave apart, plus a low-frequency residual [33].") structure to produce pictures in a coarse-to-fine procedure. Using GAN on every step of pyramid a generative Conv-net model was trained.

L. Metz et al. stated a DCGAN("deep convolutional generative adversarial networks"), it has special structural restraints, and express that they remain a powerful competitor for unsupervised learning [34]. Being trained on different image based datasets, they exhibited reliable indication that their deep convolutional adversarial couple determines a hierarchy of descriptions of object elements to views in either of generator and discriminator. Moreover, they used the learned characteristics for different jobs expressing the applicability as comprehensive picture descriptions.

Wang et al. factorize image formation method and introduced S^2-GAN [35]("Style and Structure Generative Adversarial Network"). The model has two parts: Structure-GAN produces a surface normal map, output of first model goes Style-GAN that generates a 2D representation. Aside from an authentic vs. formed loss function, they use loss between estimated surface-normal forms created pictures. In the first step, both networks are trained separately and later joined collectively by collective learning.

Zhu et al. proposed to acquire the original picture manifold right of data applying a generative adversarial neural network [36]. They determined the class of picture modification methods, and restrain their outcome depending on the acquired manifold at all times. The architecture itself can adjust the results retaining each edit as vivid as feasible. All their directions are formulated in the form of restrained optimization and implemented.

Creating lifelike pictures a longstanding purpose regarding machine learning. Im et al. stated a technique to form images using repetitive adversarial network [37]. Their model forms an image in a sequential way. They don't force a coarse to fine (or any other) composition on the production method. Rather they allow the model to learn the best method itself.

Brock et al. presented a variational autoencoder based on voxel and also developed an interface for visualization of the potential scope of 3D generative compositions [38]. They also developed a deep CNN for grouping. They mainly focused on the challenges that are critical to the voxel exhibition and showed their ability to voxel exhibitions within discriminative assignments.

Arsalan et al. used multi-view depth maps for acquiring a generative model and for producing a 3D object from these images they used deterministic rendering procedure [39]. As 2D depth maps contain rich information the generation of 3D objects can be produced much more detail.

Study	Technique	Dataset	Remarks
Denton et al. [32]	Deep Generative Adversarial Network for natural scene generation.	LSUN scene dataset	They used GAN's to generate Natural scene images. The Generated images were real enough that it was difficult for the humans to differentiate.
Radford et al. [34]	DCGAN for unsupervised learning.	Imagenet-1k, Faces dataset.	Deep convolutional adversarial learn a hierarchy of descriptions from image components.
Wang et al. [35]	Style and Structure Adversarial Networks for image generation	NYUv2	They used two different GAN models to generate images by utilizing the concept of image as a group of style and structure.
Zhu et al. [36]	GAN's for Natural Image Manifold.	ImageNet	They used the GAN architecture for editing the image manifold. They also applied color editing and resizing.
Im et al. [37]	Recurrent adversarial networks for image generation.	CIFAR10	They used idea of Recurrent neural networks and combine it with GAN's to generate images.
Arsalan et al. [39]	GAN to model Multi composition depth maps.	ShapeNet Core	

Table 2.1: Illustration of different GAN based techniques

2.2 Summary

Table 2.1 provides a summary of different techniques in which GAN's used. It can clearly be seen that researchers used GAN's for solving different types of problems, and transformed them according to their requirements, and gained promising results. By studying these methods it can be inferred that we can use GAN architecture for generating things like (images, text content, and poetry, etc.). By utilizing the generative power of GAN's we tried to generate 3D objects. They used different datasets to evaluate the performance of their models and complied their results.

Chapter 3

Methodology

3.1 Introduction

This chapter, we discussed the proposed method of generating 3D objects. We try to create 3D objects by using GAN's. Figure 3.1 exhibits a summary of the stated model.

The dataset we use consists of different categories of images of day to day objects. We will use only two categories from it, tables and chairs. The reason for selecting only two categories is to make the generator diverse in generating new 3D objects like these.

3.2 Dataset

To generate a more sensible dataset including authentic 3D objects, we use the ShapeNet table category with 4,704 instances for training and 1,200 for testing. For chair object category with 10,668 instances for training and 1,200 for testing [40]. These 3D CAD models built by human artists to realistically describe original objects. We focus on only these two object categories to explore intra-category variation by getting detailed descriptions for many object instances. We chose the table and chair classes from ShapeNetCore as they have a significant variation in geometry and appearance.

Figure 3.1 shows sample images taken from dataset of chair class. A clear intra class variation can be seen. Chair class contains 10,668 images in training set and 1,200 in test set. These images explains how much different they are from one another. More over each image taken from 12 different preparatives which means that each object has 12 different images taken from 12 different viewpoints.

Figure 3.1: Images from dataset for chair class

Figure 3.2 shows sample images taken from dataset of table class. A clear intra class variation can be seen. Chair class contains 4,704 images training set and 1,200 in test set. Those images explain how much different they are from one another. More over each image taken from 12 different preparatives which means that each object has 12 different images taken from 12 different viewpoints.

Figure 3.2: Images from dataset for table class

3.3 (GAN) Generative Adversarial Net

Ian Goodfellow and his research team introduced the GAN in 2014. Potentially GAN's are very broad, as they can imitate any data configuration. They could be taught to learn and generate the world's any domain. They are like human artists but reboots, and their output is outstanding. A GAN is a category of Machine Learning methods. It has two Neural Networks that compete with one other. The network that generates is called Generator. The network groups the output of Generator is called Discriminator.

3.3.1 The Generator

In GAN settings Generator is a network that forms new objects. Usually, it takes input from a random normal distribution and generates the output.

3.3.2 The Discriminator

In GAN settings Discriminator is a neural network that receives input from the Generator network and the dataset under consideration. It tries to differentiate whether the input is coming from dataset or Generator network.

3.3.3 How it works

The generator creates new data objects. These newly generated objects alongside the objects from the dataset are supplied as input to the Discriminator network that attempts to discriminate among the objects formed by using Generator network and objects from the dataset. The generator creates new objects and expects they will be marked genuine although they not real. The purpose of the generator network is to generate distinct objects in such a way that discriminator cannot catch. On the other hand, the objective of the discriminator network is to identify the objects that are produced by the generator.

- Input to the generator is a random number.
- The generated object alongside the objects in dataset fed to the discriminator.
- The discriminator gets the both real and fake object as input.
- The discriminator returns a number in a range from 0 to 1 where 0 means the input was from the generator a fake object and 1 means input were from dataset under study real.
- The discriminator is in a feedback circle with objects from the dataset

- The generator is in feedback circle with the discriminator so the next time the generator comes with an output that looks more real.

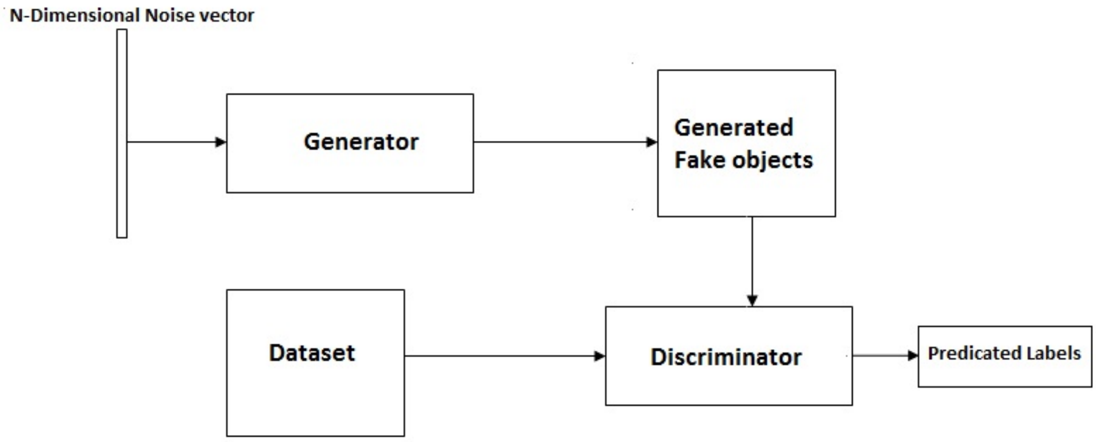


Figure 3.3: Flow layout of GAN

Figure 3.3 shows a flow diagram of the GAN model. The generator is initialized with noise vector taken from a random normal distribution. Gradually it tries to learn the distribution of a dataset. The output of the generator and the samples from the dataset are given as input to discriminator that attempts to differentiate among the examples of the dataset and produced through the generator. With every training, step generator comes up with a sample that looks more likely to samples in the dataset. After every training step, it becomes harder for discriminator network to differentiate among original and bogus objects.

3.3.4 How to Train GAN

Weights of discriminator are updated to maximize the likelihood that any object coming from dataset classified as a real object. While decreasing the likelihood of an object approaching from generator belongs to the dataset. On the other hand weights of the generator are updated to increase the likelihood that any object coming from the generator is classified as it is coming from the dataset.

GANs need more time to train than traditional neural networks. On an individual GPU, this may require hours to train on CPU this may require days to train.

3.3.4.1 Training of Discriminator

Figure 3.4 explains the training procedure of discriminator. The weights of the generator are frozen and the loss of the network computed. Then calculated loss is backpropagated into the network, and weights of the discriminator are updated.

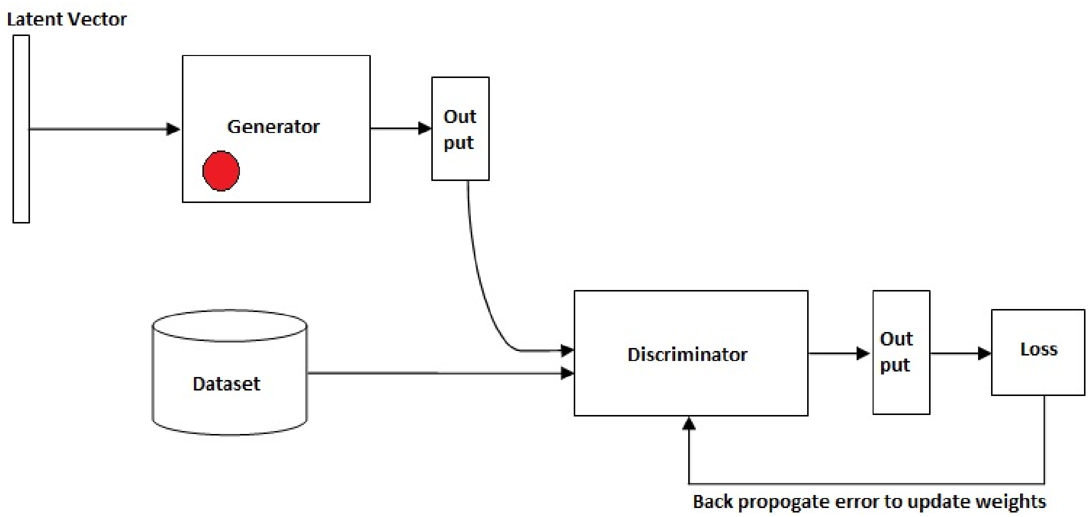


Figure 3.4: Block diagram of training Discriminator

3.3.4.2 Training of Generator

Figure 3.5 explains the training procedure of the generator. Loss of the network is computed by freezing the discriminator's weights. Then calculated loss is backpropagated into the network, and the generator 's weights are updated.

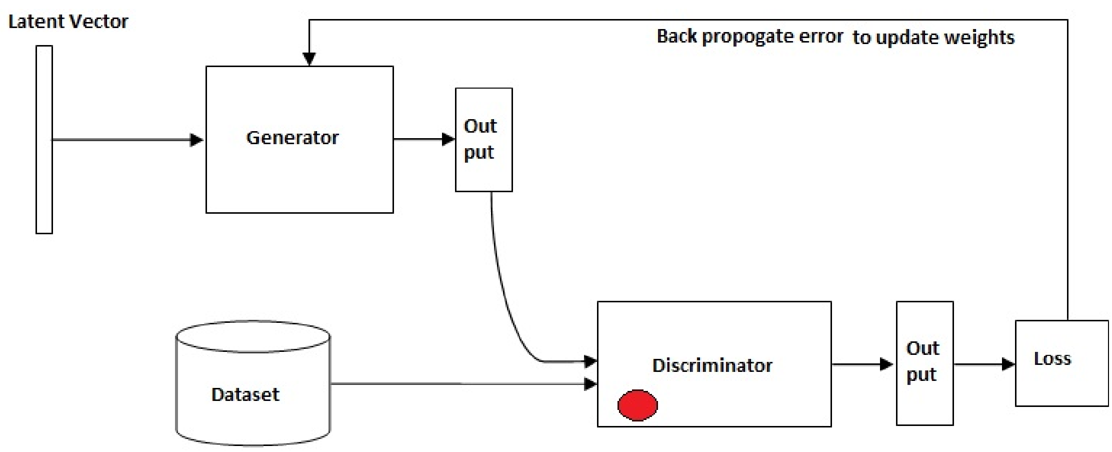


Figure 3.5: Block diagram of training Generator

3.3.4.3 What is Latent Space

Meaning of the word latent is hidden. It is the most commonly used term in machine learning especially in GAN's. In GAN architecture, generative models learn to map points in latent space to produce objects. By itself, latent space has no meaning, slightly the meaning referred to it through GAN's. The association between latent space and GAN quite appealing. While working with images let's say all data fall in some high dimensional hyperplane. It's very challenging to sample from this. Rather we like to draw from some simple distribution and try to map it in the direction of the data distribution using the Generator network. This simple to pick from the population is latent space.

3.4 Proposed Methodology

To do this task, we use GAN formulation it provides better visualization and diversity. We took the advantage at the instance level correspondence between images and 3D shapes. In our first step, we present a way to acquire a collective exhibition directly from pictures and 3D objects followed by image-to-shape formation framework the image is mapped to the latent space and concatenated with noise from the random normal distribution. The resultant vector is pass to the Generator network which tries to generate a plausible shape the formed shape is passed to the Discriminator network finally the loss is computed. Finally, this loss is back-propagated in the system.

First, we will train an auto-encoder which will map 3D image to its corresponding labels. That will map learned features from the auto-encoder to an n-dimensional vector v. Then vector z which will be initialized from random normal distribution concatenated with vecot v and passed as input to the GAN.

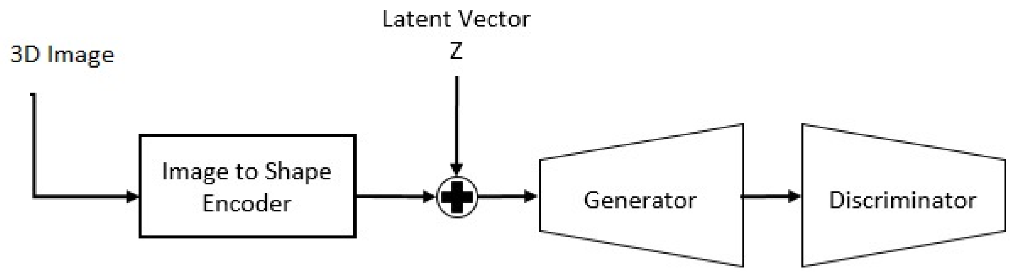


Figure 3.6: Description of the stated method

Figure 3.6 describes of the introduced method. In the first step, a 3D image supplied to the image-to-shape encoder as input. The resulting image mapped to the latent space and concatenated with randomly initialized vector Z which taken from the normal distribution. Then this vector is passed to Generator model. Then the output of the Generator Network becomes the input of the Discriminator Network and overall loss computed.

3.4.1 Model

The proposed model has three parts.

1. **An Encoder E:**
   It consists of 5 convolutional layers having a window dimension of 11, 5, 5, 5, and 8 with a step size of 4, 2, 2, 2, and 1 sequentially. Batch normalization and ReLU applied as an activation map. In the end, a sampler used to sample 100 dimensional vector.

2. **A decoder (the Generator G):**
   It has 5 convolutional layers by a window size of 4x4x4 and steps size of 2. Batch normalization and ReLU applied as an activation map. In the final layer used Sigmoid as an activation function. Figure 3.7 explains the architecture of Generator model.

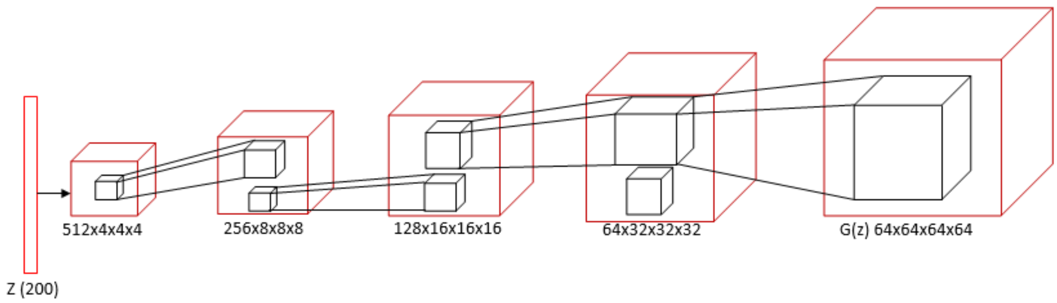


Figure 3.7: Description of the Generator.

3. **A discriminator D:**
   The Discriminator primarily reflects the Generator. The only difference between Generator and Discriminator is the activation function. In Discriminator, we used Leaky ReLU as an activation function.

We didn't use polling or linear layers in our model. Binary-cross-entropy applied as grouping loss and presented overall decline function as:

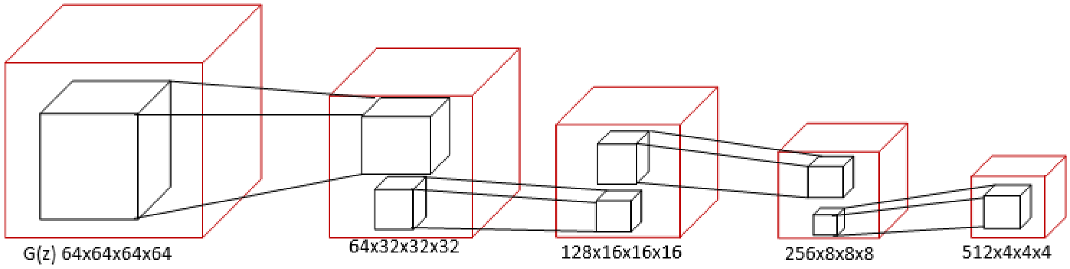


Figure 3.8: Block diagram of the Discriminator

3.4.2 Loss Function

It is the principal function to train a GAN. It gives loss of estimations that are used to compute gradients. Based on loss values the weights are updated. The following equation describes the loss function.

$$L_{3D-GAN} = logD(x) + log(1 - D(G(z)))$$

In which x is an original object in a 64x64x64 space. Z is a noise vector sampled randomly from distribution p(z). Each dimension of z is sampled from a normal distribution across [0;1]. The output from the discriminator is represented with D(x). The output from the generator represented with G(z).

3.4.3 Evaluation Metric

IOU(Intersection over Union) among voxels formed and true shape voxels, indicating the quality of the occupancy of the generated shapes, regardless of color. Models with similar shapes are more likely to have higher IoU than dissimilar models. Mathematically it can be presented as:

$$\text{IoU(A,B)} = \frac{A \bigcap B}{A \bigcup B}$$

It assigns each object a number between 0 and 1. Where 1 means the object exactly match to the object that is in dataset, and 0 means don't match at all.

3.5 Summary

This section gives a comprehensive summary of the proposed model and selected dataset. It also explains how to train a GAN, and the loss function. In the next chapter results and performance analysis of the proposed model is discussed.

Chapter 4

Experiments and Results

4.1 Introduction

This section explains experiments conducted on the proposed model. Discussed how the model was trained and results generated on Shape Net dataset.

4.2 Training

In the first step, we trained a deep convolutional autoencoder for classification. After successfully training, we extracted the learned features from the encoder part of this network. Then we trained our model for two categories from the dataset, which are table and chair. One model trained for each class separately. For production, we sampled a 200 dimensional vector from a random normal distribution and concatenated it with the learned features by the encoder and then passed this vector to the generator.

Figure 4.1 explains the structure of the discriminator network. It has three columns in the first column name of the layer is written, in second column output shape is shown and in third column number of parameters learned by each layer are written. Last three rows explain the total number of parameters, trainable parameters, and non-trainable parameters.

Figure 4.2 explains the structure of the generator network. It has three columns in the first column name of the layer is written, in second column output shape is shown and in third column number of parameters learned by each layer are written. Last three rows explain the total number of parameters, trainable parameters, and non-trainable parameters.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 64, 64, 64, 1)     0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 32, 32, 32, 64)    4160
_________________________________________________________________
batch_normalization_6 (Batch (None, 32, 32, 32, 64)    256
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 32, 32, 32, 64)    0
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 16, 16, 16, 128)   524416
_________________________________________________________________
batch_normalization_7 (Batch (None, 16, 16, 16, 128)   512
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 16, 16, 16, 128)   0
_________________________________________________________________
conv3d_3 (Conv3D)            (None, 8, 8, 8, 256)      2097408
_________________________________________________________________
batch_normalization_8 (Batch (None, 8, 8, 8, 256)      1024
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 8, 8, 8, 256)      0
_________________________________________________________________
conv3d_4 (Conv3D)            (None, 4, 4, 4, 512)      8389120
_________________________________________________________________
batch_normalization_9 (Batch (None, 4, 4, 4, 512)      2048
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 4, 4, 4, 512)      0
_________________________________________________________________
conv3d_5 (Conv3D)            (None, 1, 1, 1, 1)        32769
_________________________________________________________________
batch_normalization_10 (Batc (None, 1, 1, 1, 1)        4
_________________________________________________________________
activation_6 (Activation)    (None, 1, 1, 1, 1)        0
=================================================================
Total params: 11,051,717
Trainable params: 11,049,795
Non-trainable params: 1,922
_________________________________________________________________
```

Figure 4.1: Detail of the Discriminator Network

4.3 Experiments

First, we examined the results of the system to generate objects from the population consisting of tables and chairs. Our method produces quality objects from different viewing points. It was also a remarkably successful effort as the trained model generates diverse and great quality objects. The quality of the produced objects can be using discriminator's

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 1, 1, 1, 200)      0
_________________________________________________________________
conv3d_transpose_1 (Conv3DTr (None, 4, 4, 4, 512)      6554112
_________________________________________________________________
batch_normalization_1 (Batch (None, 4, 4, 4, 512)      2048
_________________________________________________________________
activation_1 (Activation)    (None, 4, 4, 4, 512)      0
_________________________________________________________________
conv3d_transpose_2 (Conv3DTr (None, 8, 8, 8, 256)      8388864
_________________________________________________________________
batch_normalization_2 (Batch (None, 8, 8, 8, 256)      1024
_________________________________________________________________
activation_2 (Activation)    (None, 8, 8, 8, 256)      0
_________________________________________________________________
conv3d_transpose_3 (Conv3DTr (None, 16, 16, 16, 128)   2097280
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 16, 16, 128)   512
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 16, 128)   0
_________________________________________________________________
conv3d_transpose_4 (Conv3DTr (None, 32, 32, 32, 64)    524352
_________________________________________________________________
batch_normalization_4 (Batch (None, 32, 32, 32, 64)    256
_________________________________________________________________
activation_4 (Activation)    (None, 32, 32, 32, 64)    0
_________________________________________________________________
conv3d_transpose_5 (Conv3DTr (None, 64, 64, 64, 1)     4097
_________________________________________________________________
batch_normalization_5 (Batch (None, 64, 64, 64, 1)     4
_________________________________________________________________
activation_5 (Activation)    (None, 64, 64, 64, 1)     0
=================================================================
Total params: 17,572,549
Trainable params: 17,570,627
Non-trainable params: 1,922
_________________________________________________________________
```

Figure 4.2: Detail of the Generator Network

loss. It allowed the model convergence to be traced easily and gives a clear clue when to stop the training.

Figure 4.3 reveals the loss of GAN. It is clear that the generator is trying to maximize its performance to generate objects that look like original ones. On the other hand, Discriminator is trying to minimize the possibility of letting the fake ones go through. This is the way how GAN models work.

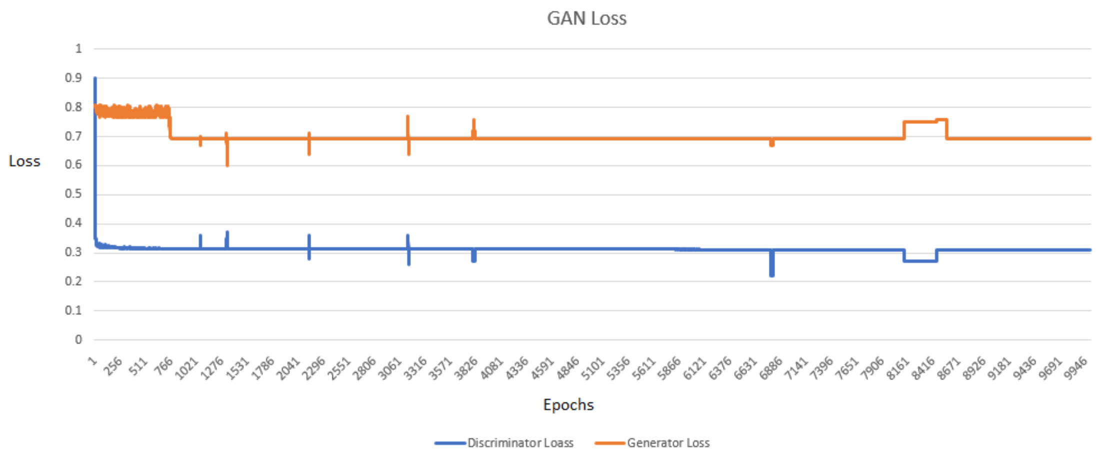


Figure 4.3: Loss of GAN model

4.4 Qualitative Results

Qualitative results of the proposed model are explained. Figure 4.4 and 4.5 show the results of 3D object generation. The generated objects can be envisioned and correlated by the ground-truth values taken from the dataset. The quality of the generated objects can be examined by using IOU evaluation metric.

4.4.1 Generating 3D objects

Results of generated for chair and table class can be visualized in figure 4.4 and figure 4.5 respectively. For the production of new 3D objects, we sampled a 200-dimensional vector from latent space and concatenated with the features learned from 3D Classifier. This given the resulting vector is given as input to the Generator network.

4.4.1.1 Initial Settings

The model was trained for 10000 training steps. For training, it took 10 hours on 1070 Ti GPU system. Initially learning-rate for generator set to 0.008, for discriminator set to 0.000005, and batch-size set to 32. We used only two classes of tables and chairs from the shape net dataset. Figure 4.4 shows generated images for chair class alongside truth images taken from the dataset. Figure 4.5 shows generated images for table class alongside truth images of the table taken from the dataset. Generated objects are of good quality

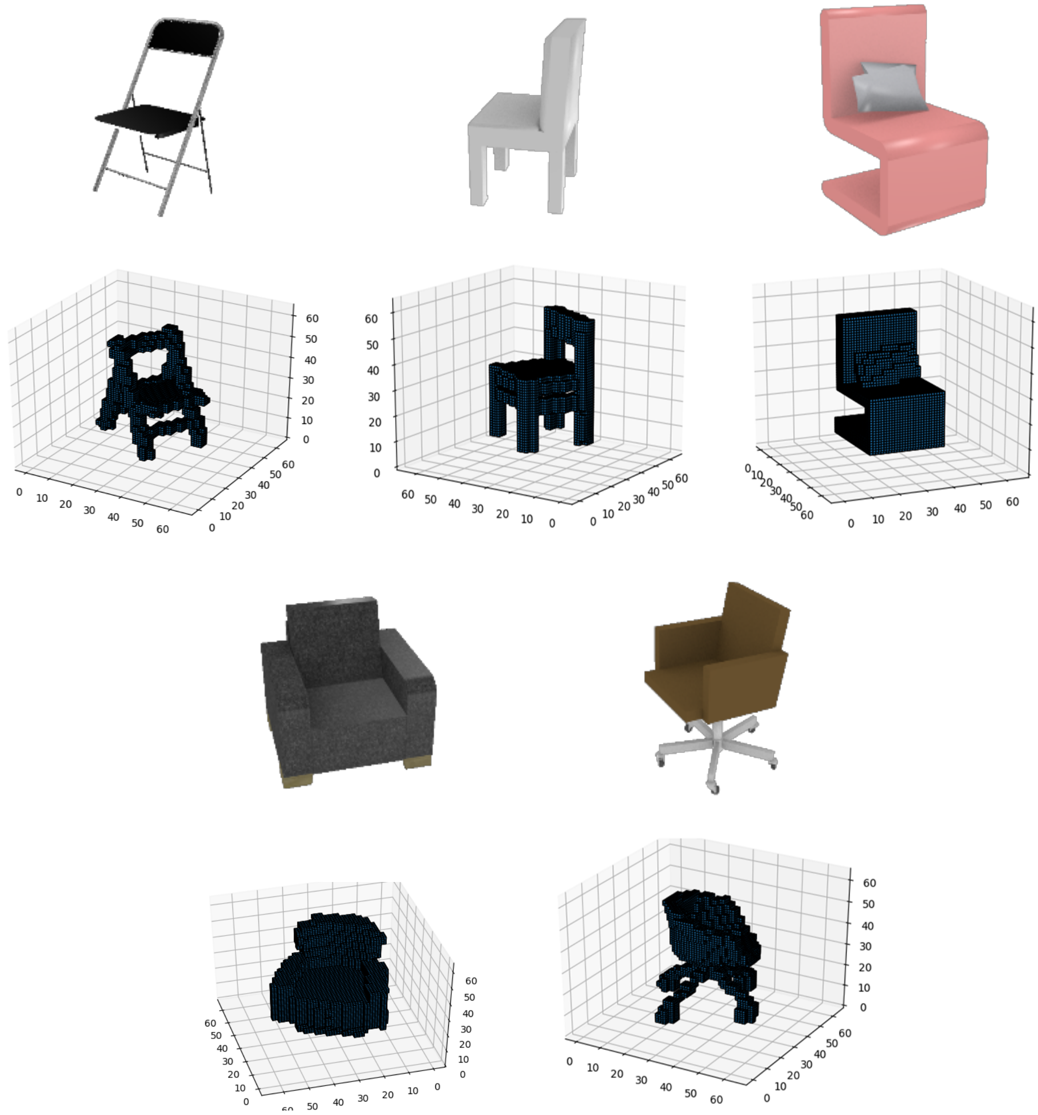


Figure 4.4: Shows the results for chair class. Row one shows images from the dataset. Row two presents images produced with the proposed model. Row three shows images from dataset. Row four shows images generated with the proposed model. The produced images vary in quality. Some of the objects mapped perfectly to 3D object space while others still having some missing information or having some overlapping regions.

this can be verified from figure 4.4 and 4.5. It's also likely to determine the condition of created objects by discriminator's loss.

In comparison with earlier techniques, our model can manufacture quality 3D objects including specified geometries. One thing is noticeable that generating low-quality 3D objects is easy but it's difficult to generate good quality 3D objects because of agile extension in 3D space. Though, object portions are shown in high intention. The main

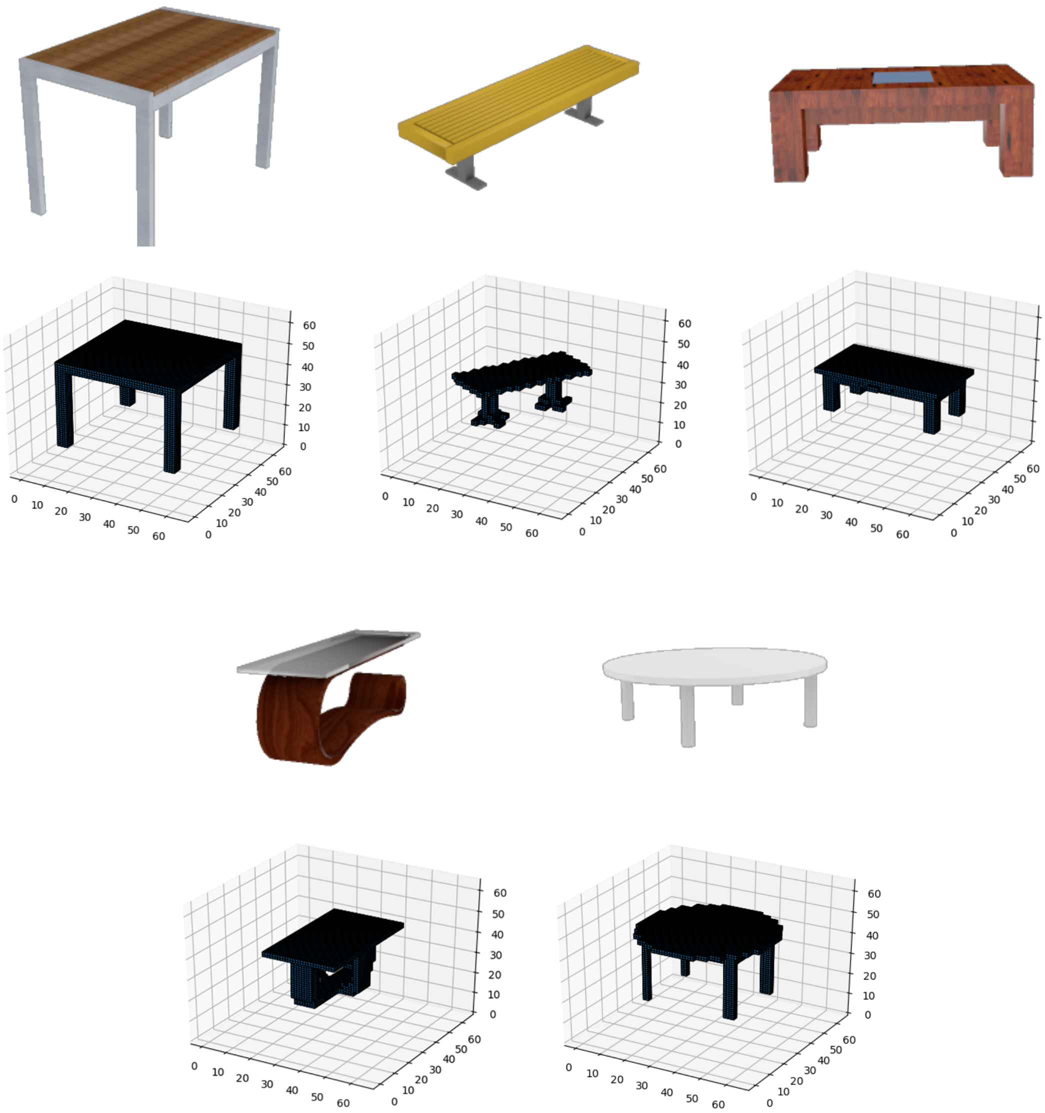


Figure 4.5: Results for table class can be visualized. Row one shows images from the dataset. Row two presents images produced with the proposed model. Row three shows images from dataset. Row four shows images generated with the proposed model. The produced images vary in quality. Some of the objects mapped perfectly to 3D object space while others still having some missing information or having some overlapping regions.

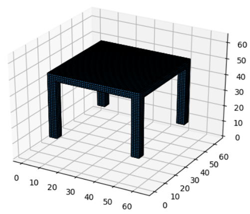
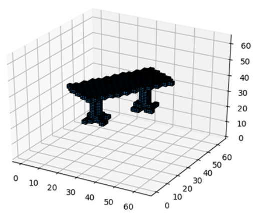
concern of such models whether they are inferring the training data. Such models must be designed and implemented by keeping generalization in mind.

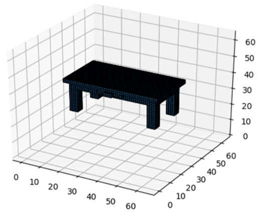
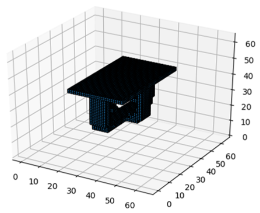
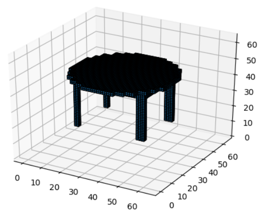
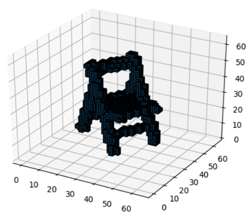
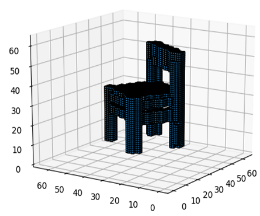
4.5 Quantitative Results

Table 4.1 explains the quantitative results of the purposed model. The main objective here is to verify that the generated objects are worthwhile. Results of IOU represented in table 4.1. It can be noted some of the objects are easier to construct and some are hard. After the optimization to be predictable from object space, it can be seen that it maintains overall regeneration completion.

4.5.1 Results of Intersection Over Union

Table 4.1 shows the results of the IOU. IOU calculated by the ground-truth object taken from the dataset and the generated objects from our system. The table shows how much two objects are similar. IOU is computed in three steps, one intersection of the ground truth and generated objects is computed, two unions of the ground truth and generated object is computed, and the third intersection of two objects is divided by the union of two objects and final results are computed.

Sr. #	**Ground Truth**	**Generated Object**	**IOU Value**
1			0.81
2			0.75

3			0.78
4			0.70
5			0.79
6			0.65
7			0.73

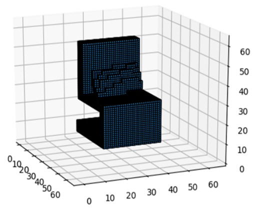
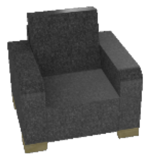
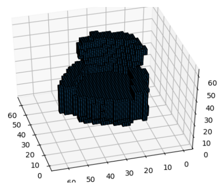
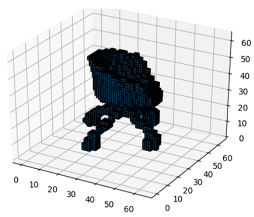
8			0.75
9			0.63
10			0.60

Table 4.1: Results of Intersection Over Union

4.6 Generation of new Objects

Figure 4.6 explains the transition from one table's latent description to another when trained on ShapeNet table class. It explains how new objects are formed that are unique in their representations. Similarly, Figure 4.7 explains the transition from one chair's latent description to another when trained on ShapeNet chair class.

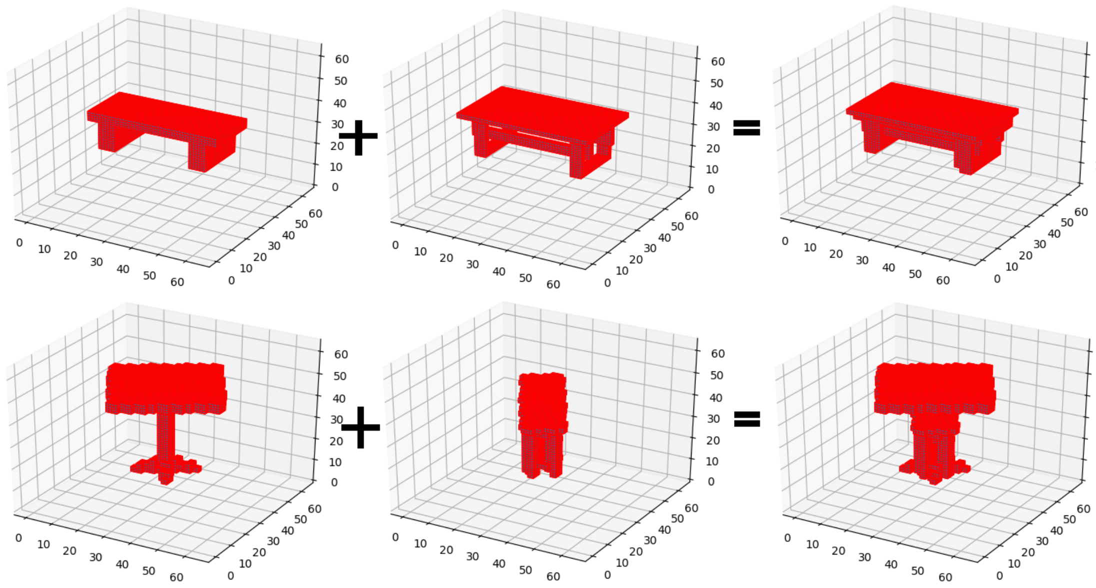


Figure 4.6: New generated objects for table class

It can be seen how it proposed model handles the transition from one unique object class and familiarization over another. The transitions in both of the figures 4.6 and 4.7 are understandable and clear. This is a positive change in primary GAN method, that produces objects with no sense at all. The proposed model's clean and smooth transition from one object to another that it was able to acquire a better arrangement of the observed target location beside the latent scope so it can produce as many as possible unrealistic regions in an object.

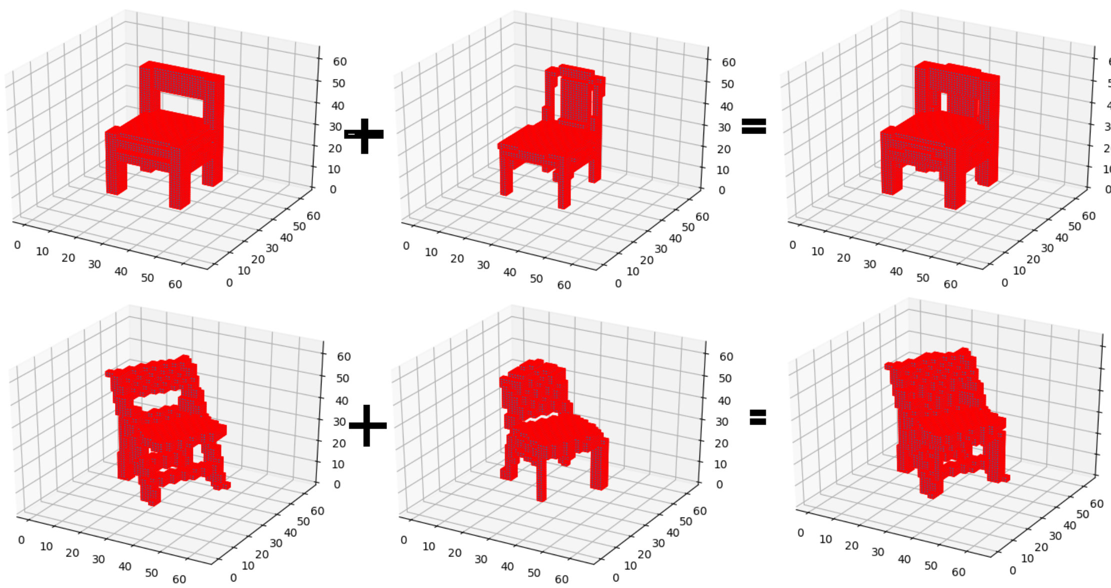


Figure 4.7: New generated objects for chair class

4.7 Classification of 3D objects

Learned representations by the discriminator are evaluated. A typical way of doing so is to use the learned features as classification. To get these features we train the model on to classes table and chair of shape net dataset, in discriminator the output of second, third and fourth Conv layer are concatenated and then apply the max pooling with a step size of 8,4,2. For grouping, we applied linear Support Vector Machine.

Table 4.1 exhibits the classification outcomes on the shape-net dataset. In the proposed model, only table and chair classes of the shape-net dataset are used. We compared our model with most recently proposed methods and our model outperformed these methods. For table class, our model gained the classification accuracy of 74 percent and for chair class 77.79 percent, that is greater than any previously proposed models.

Sr. #	**Method**	**Classification Accuracy**	
		Table	**Chair**
1	Michael et. al [7]	67 percent	66.3 percent
2	Rohit et. al [24]	70 percent	71 percent
3	Abhishek et. al [29]	72 percent	77 percent
4	Ours	74 percent	77.79 percent

Table 4.2: Classification results on shapenet dataset for table and chair class

4.8 Summary

The results still need to be improved quality wise. But we perceive that our method has the capacity to the much of our information is past unexplored. It has the capability to construct a 3D image. If trained properly it can construct any 3D image, so we can say there is no limit on the dataset. Certain baselines, though, are specific answers to one of several jobs we can explain and frequently use extra information. As society begins tackling frequently difficult 3D problems, we consider that our work provides a powerful base to benchmark improvement.

Chapter 5

Conclusions

5.1 Conclusion

The presented scheme provides quality results but still needs a lot of improvements. Once trained, the model can be used for producing new objects. It could be very helpful as if we want to create a new design of table or chair by using this model we can easily achieve this task. If trained properly on any dataset it can generate quality objects accordingly. We hope that our contribution will be a valuable addition to the field.

5.2 Feature Work

As feature work, we want to improve the findings of the proposed model. We also want to use a pre-trained-model for feature extraction use them as input to the generator network.

Bibliography

[1] funkihomes, "funkihomes." Cited on pp. v and 2.

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking.," in *ISMAR*, vol. 11, pp. 127–136, 2011. Cited on pp. 1 and 2.

[3] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, p. 169, 2013. Cited on pp. 1 and 2.

[4] F. Steinbrucker, C. Kerl, and D. Cremers, "Large-scale multi-resolution surface reconstruction from rgb-d sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3264–3271, 2013. Cited on pp. 1 and 2.

[5] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pp. 381–389, ACM, 2006. Cited on p. 2.

[6] W. Zhao, S. Gao, and H. Lin, "A robust hole-filling algorithm for triangular mesh," *The Visual Computer*, vol. 23, no. 12, pp. 987–997, 2007. Cited on p. 2.

[7] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006. Cited on p. 29.

[8] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, p. 29, 2013. No Citations.

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference*

on computer vision and pattern recognition, pp. 1912–1920, 2015. Cited on pp. 2 and 6.

[10] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5868–5877, 2017. Cited on p. 2.

[11] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2088–2096, 2017. Cited on p. 3.

[12] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "Octnetfusion: Learning depth fusion from data," in *2017 International Conference on 3D Vision (3DV)*, pp. 57–66, IEEE, 2017. Cited on p. 3.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Courville, Aand Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014. Cited on pp. 3 and 7.

[14] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017. Cited on p. 3.

[15] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in neural information processing systems*, pp. 2172–2180, 2016. Cited on p. 3.

[16] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, IEEE, 2015. Cited on p. 3.

[17] M. Aron, N. Mellado, G. J. Brostow, and N. J. Mitra, "Rapter: rebuilding man-made scenes with regular arrangements of planes.," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 103–1, 2015. Cited on p. 5.

[18] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3d geometry," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, p. 43, 2008. Cited on p. 5.

[19] P. Speciale, M. R. Oswald, A. Cohen, and M. Pollefeys, "A symmetry prior for convex variational 3d reconstruction," in *European Conference on Computer Vision*, pp. 313–328, Springer, 2016. Cited on p. 5.

[20] I. Sipiran, R. Gregor, and T. Schreck, "Approximate symmetry detection in partial 3d meshes," in *Computer Graphics Forum*, vol. 33, pp. 131–140, Wiley Online Library, 2014. Cited on p. 5.

[21] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas, "Acquiring 3d indoor environments with variability and repetition," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 138, 2012. Cited on p. 5.

[22] Y. Shi, P. Long, K. Xu, H. Huang, and Y. Xiong, "Data-driven contextual modeling for 3d scene understanding," *Computers & Graphics*, vol. 55, pp. 55–67, 2016. Cited on p. 5.

[23] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas, "Joint embeddings of shapes and images via cnn image purification," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, p. 234, 2015. Cited on p. 5.

[24] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *European Conference on Computer Vision*, pp. 484–499, Springer, 2016. Cited on pp. 6 and 29.

[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017. Cited on p. 6.

[26] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2686–2694, 2015. Cited on p. 6.

[27] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015. Cited on p. 6.

[28] 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 *European conference on computer vision*, pp. 628–644, Springer, 2016. Cited on p. 6.

[29] A. Sharma, O. Grau, and M. Fritz, "Vconv-dae: Deep volumetric shape learning without object labels," in *European Conference on Computer Vision*, pp. 236–250, Springer, 2016. Cited on pp. 7 and 29.

[30] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3d structure from images," in *Advances in Neural Information Processing Systems*, pp. 4996–5004, 2016. Cited on p. 7.

[31] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas, "Joint embeddings of shapes and images via cnn image purification," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, p. 234, 2015. Cited on p. 7.

[32] E. L. Denton, S. Chintala, R. Fergus, *et al.*, "Deep generative image models using a lapliacian pyramid of adversarial networks," in *Advances in neural information processing systems*, pp. 1486–1494, 2015. Cited on pp. 7 and 9.

[33] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on communications*, vol. 31, no. 4, pp. 532–540, 1983. Cited on p. 7.

[34] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. Cited on pp. 7 and 9.

[35] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *European Conference on Computer Vision*, pp. 318–335, Springer, 2016. Cited on pp. 8 and 9.

[36] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision*, pp. 597–613, Springer, 2016. Cited on pp. 8 and 9.

[37] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," *arXiv preprint arXiv:1602.05110*, 2016. Cited on pp. 8 and 9.

[38] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," *arXiv preprint arXiv:1608.04236*, 2016. Cited on p. 8.

[39] A. Arsalan Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, "Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1511–1519, 2017. Cited on pp. 8 and 9.

[40] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. Cited on p. 10.

3D Vectorization and Generation using Generative Adversarial Network

ORIGINALITY REPORT

13%	11%	11%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	export.arxiv.org Internet Source	2%
2	arxiv.org Internet Source	2%
3	www.groundai.com Internet Source	1%
4	Mingming Liu, Kexin Zhang, Jie Zhu, Jun Wang, Jie Guo, Yanwen Guo. "Data-driven Indoor Scene Modeling from a Single Color Image with Iterative Object Segmentation and Model Retrieval", IEEE Transactions on Visualization and Computer Graphics, 2019 Publication	1%
5	Po Kong Lai, Weizhe Liang, Robert Laganière. "Additive depth maps, a compact approach for shape completion of single view depth maps", Graphical Models, 2019 Publication	1%
	repository.tudelft.nl	

6	Internet Source	<1%
7	Xinle Li, Wenbo Nie, Wei Zhang, Xiaobo Lin, Yao Yu. "A Shared Model Based Dense Real-Time Semantic SLAM Method Towards Repetitive Scene", 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2019 Publication	<1%
8	Submitted to Higher Education Commission Pakistan Student Paper	<1%
9	Ruben Villegas, Jimei Yang, Duygu Ceylan, Honglak Lee. "Neural Kinematic Networks for Unsupervised Motion Retargetting", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018 Publication	<1%
10	vision.in.tum.de Internet Source	<1%
11	"Computer Vision – ACCV 2018", Springer Science and Business Media LLC, 2019 Publication	<1%
12	eprints.whiterose.ac.uk Internet Source	<1%

oa.upm.es