DEEP AUTOENCODERS FOR CROSS-MODAL RETRIEVAL

By

AYESHA SIDDIQUA

Bachelor of Science in Computer Science and Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
2007

Master of Science in Electrical and Computer Engineering
Oklahoma State University
Stillwater, Oklahoma
2011

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 2019

DEEP AUTOENCODERS FOR CROSS-MODAL RETRIEVAL

Dissertation Approved:

Dr. Guoliang Fan

Dissertation Advisor

Dr. Martin Hagan

Dr. Weihua Sheng

Dr. Christopher John Crick

ACKNOWLEDGMENTS

I would first like to express my sincere gratitude to my advisor Dr. Guoliang Fan for his continuous support during my graduate studies through his knowledge, motivation, and patience. I am grateful for his insightful suggestions and encouragement to overcome many difficulties during this research endeavor. His enthusiasm and dedication to this research work was a constant motivation for me to push boundaries to successfully complete this research project.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Dr. Martin Hagan, Prof. Dr. Weihua Sheng, and Prof. Dr. Christopher Crick for their prompt response to any of my e-mails and committing their valuable time for this project. I have learned a lot from their insightful questions which encouraged me to delve deeper into my research from different perspective to improve my understanding of the project.

I am in debt to Dr. Martin Hagan for my fundamental understanding of Neural Network and Deep learning through the classes he has offered. These classes gave me a foundation to build upon. I would also like to thank my committee members Dr. Christopher John Crick for his dedication and commitment throughout the review of this dissertation. I greatly appreciate their extremely helpful guidance during the past few years. I am also grateful to Dr. Weihua Sheng for being kind enough to agree to be one of my committee members in such a short notice.

I owe a great deal to Nate Hannan for his time and effort on reviewing and proof

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

reading my thesis. I would like to thank Guo Lin for his insightful comments on my thesis and Mahdi Yazdanpour for his help and support. I would also like to take this opportunity to thank Dr. Vijay Venkataraman, Dr. Yi Ding, Dr. Xin Zhang, Dr. Liangjiang Yu for their kind support and invaluable suggestions during my early graduate life. Thank you to all VCIPL lab members including Dr. Wei Liu, Yong Li, Le Zhou, Ryan Swann, Andersen Lin.

Last but not the least, I appreciate all my family members for their continuous and unconditional support. I would like to thank my parents, specially my father, without his inspiration I would not have started the degree. This acknowledgement would be incomplete if I don't mention two of the most important people in my life - my husband, Arif, without his support I could not have finished this work and my precious little girl Ayaat who has been a great source of joy for me.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: Ayesha Siddiqua

Date of Degree: May, 2019

Title of Study: DEEP AUTOENCODERS FOR CROSS-MODAL RETRIEVAL

Major Field: ELECTRICAL AND COMPUTER ENGINEERING

Abstract: Increased accuracy and affordability of depth sensors such as Kinect has created a great depth-data source for 3D processing. Specifically, 3D model retrieval is attracting attention in the field of computer vision and pattern recognition due to its numerous applications. A cross-domain retrieval approach such as depth image based 3D model retrieval has the challenges of occlusion, noise, and view variability present in both query and training data. In this research, we propose a new supervised deep autoencoder approach followed by semantic modeling to retrieve 3D shapes based on depth images. The key novelty is the two-fold feature abstraction to cope with the incompleteness and ambiguity present in the depth images. First, we develop a supervised autoencoder to extract robust features from both real depth images and synthetic ones rendered from 3D models, which are intended to balance reconstruction and classification capabilities of mix-domain data. We investigate the relation between encoder and decoder layers in a deep autonecoder and claim that an asymmetric structure of a supervised deep autoencoder is more capable of extracting robust features than that of a symmetric one. The asymmetric deep autoencoder features are less invariant to small sample changes in mixed domain data. In addition, semantic modeling of the supervised autoencoder features offers the next level of abstraction to the incompleteness and ambiguity of the depth data. It is interesting that, unlike any other pairwise model structures, the cross-domain retrieval is still possible using only one single deep network trained on real and synthetic data. The experimental results on the NYUD2 and ModelNet10 datasets demonstrate that the proposed supervised method outperforms the recent approaches for cross modal 3D model retrieval.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

The emergence of inexpensive depth sensors in recent years has made depth image based 3D model retrieval a competitive approach for cross-modal retrieval. It is a more affordable approach than sketch-based 3D model retrieval and RGB-D image based 3D model retrieval. Scanning and rebuilding 3D scenes using 3D CAD model database has many applications in many areas such as robotics, surveillance, object detection, interior design and providing user interaction interface facility. Successful depth image based 3D model retrieval has many powerful applications and can also help in the fields of computer vision, product design, architecture, information retrieval, augmented reality and many others. Existing 3D model retrieval processes require digitization to eventually produce a draft to retrieve CAD models. This process is time consuming and requires manual work. Since RBG-D cameras are becoming handy and available at a low cost, depth image based 3D model retrieval is becoming more practical and desirable by returning a set of relevant existing 3D models quickly and easily. The retrieved CAD models can be directly modified for effective product design process.

In this work, for the first time, we study the application of a supervised deep autoencoder for a cross-modal retrieval (3D model retrieval based on real depth images) followed by semantic modeling. Supervised and semi supervised autoencoders have been used in face recognition, speech recognition, sentiment analysis, text classification, phone classification, multi-task learning and many other areas [1–8]. The deep autoencoder is a class of deep learning. An autoencoder tries to learn an approximation to the identity function in an unsupervised way which means only the data is available without a label. By constraining the hidden layer with different number of neurons, sparsity, weight decay or corrupted input, we can reveal strong feature structure present in the data. A supervised deep autoencoder can restructure features by providing label information when the data is incomplete and occluded. The deep supervised structure of an autoencoder can effectively learn complicated structures from raw depth images and requires less domain knowledge. To compare the effectiveness of a supervised autoencoder over an unsupervised autoencoder, we propose one unsupervised deep autoencoder approach and two supervised deep autoencoder approaches for retrieval. Supervising an autoencoder with label information has turned out to be a useful method for retrieving 3D models based on incomplete depth images.

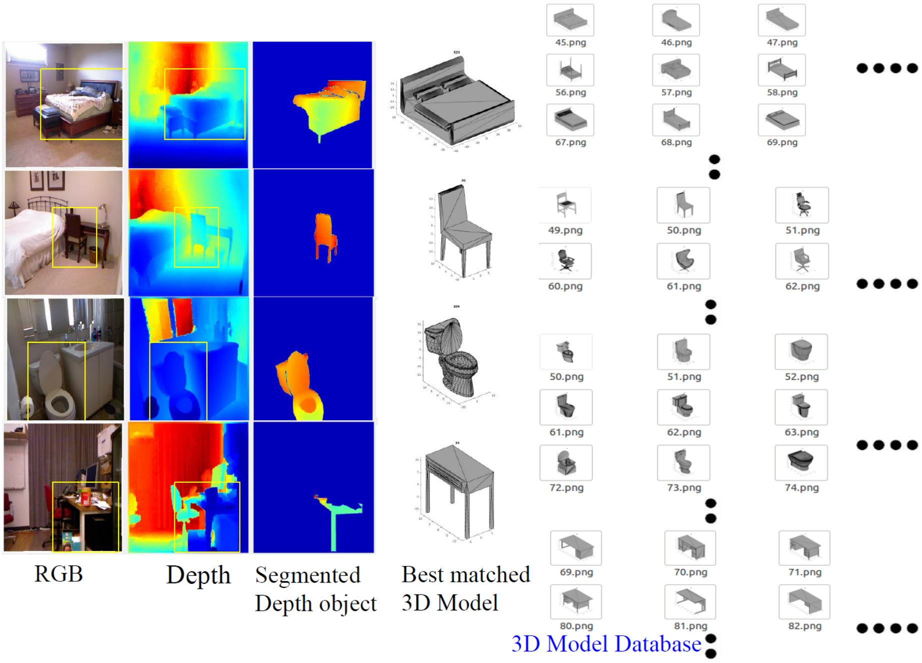


Figure 1.1: Research problem: retrieve the best matched 3D-models from the 3D model database for the given depth image object.

Usually cross-modal retrieval methods involve training two different domains with a parallel or paired network structure [7, 9–11]. The cross-modal retrieval is based on the assumption of training two different supervised networks for two different domains which facilitates transfer learning. We show that training one single deep autoencoder with rendered and real depth images together is more efficient to bring cross domains in single feature space. One other positive side of our model is that there is no restriction of having equal number of depth images and 3D models. On the contrary, most of the cross-modal retrieval approaches such as correlation modeling (CM) [12] and semantic correlation modeling (SCM) [12] algorithms are restricted to have the same number of text and images due to the dimension requirement of canonical correlation analysis (CCA). In our proposed approaches, we take depth images as the direct input rather than handcrafted features.

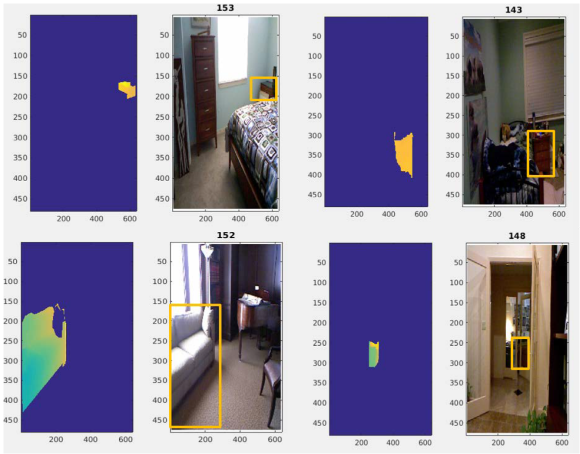


Figure 1.2: Challenges of incompleteness, noise, occlusion and partial view present in the real depth images.

Though supervised autoencoder features can deal with the incompleteness present in the data, those features do not have perfect image interpretation due to ambiguities present among the depth image objects. Semantic modeling over supervised autoencoder features provides higher level abstraction of the ambiguities present in the depth images which is helpful for cross modal retrieval. 3D models projected on to multiple 2D views are reasonable to match with 2D depth image objects. We have rendered ninety five different views which have improved 3D model retrieval for incomplete and partial depth images. Since retrieval would be done using two different domains, we have used two different databases. The first domain is depth image database and the second domain is 3D model database. For depth images we have used NYD Depth Dataset V2(NYD2) [13]. NYD2 is a database of RGBD images from a variety of indoor scenes. The images have been collected using a Microsoft kinect. NYUD2 has 1449 densely labeled pairs of aligned RGB and depth images. Only depth images have been used. The depth images containing any of the ten indoor objects (bathtub, bed, chair, desk, dresser, nightstand, table, toilet, monitor and sofa) is considered as candidate inputs. The input depth image has one object per image. We have selected 3D models from ModelNet10 [14] benchmark database belonging to the ten indoor objects. In [9], they have rendered line drawings from 3D model view. In [7], they have projected 3D shape into many 2D depth images for 3D shape retrieval.

1.1 Research Goal

We address the challenging cross-modal retrieval problem of depth image based 3D model retrieval with significant occlusion. With the advent of the Kinect, comes a great depth-data source for 3D model retrieval due to it's increased accuracy at a lower cost. 3D model retrieval is attracting the attention of various fields due to its numerous applications in computer vision and artificial intelligence. A cross-domain retrieval approach like depth image based 3D model retrieval has the challenges of occlusion, noise and view variability present in the real depth images. In this research, we aim to articulate a new supervised deep autoencoder approach followed by semantic modeling to retrieve 3D shapes based on depth images. Retrieving 3D models based on depth images can be considered as a transfer learning or cross-domain approach since real depth and rendered depth have different depth quality. Real depth images have noise and incomplete object information. We aim to bring two variant domains (depth image and 3D model) into one single feature space by training one single supervised deep autoencoder with real depth images and rendered depth images from 3D models to bring those cross-modal data in the same feature space. It is interesting that, unlike any other pairwise model structures, cross-domain retrieval is still possible using only one single deep network in our model. Rendering different views for each 3D model matching with the real depth image scenario is an effective way for 3D model retrieval. And again training rendered and real depth images together is promising way to bridge the gap between 3D models and depth data.

The goal of our research is to achieve a two-fold abstraction of incompleteness and ambiguity present in the real depth images by the supervised autoencoder and semantic matching of the supervised autoencoder features respectively. Previous research has shown that supervised and unsupervised approaches using deep autoencoders are able to capture details present in the data. But those research works were performed for same-domain retrieval. We aim to show that supervision on the autoencoder can improve the cross-domain retrieval performance by restructuring the features by providing label information. Though supervised deep autoenocder deals with incompleteness and occlusion to a certain extent this is mainly data driven. Semantic modeling on the supervised features is capable to offer the next level of abstraction of ambiguity by forcing the features to cluster to a certain category. In a nutshell, two-level abstraction is expected to improve 3D model retrieval accuracy by minimizing incompleteness and view variability present in the real depth images. We plan to evaluate the effectiveness of our model on two different data domains: NYUD2 real depth image dataset and ModelNet10 models for ten indoor object categories. The proposed supervised method outperforms the recent approaches for cross modal 3D model retrieval based on depth images.

1.2 Our Contributions

We can summarize our contributions from different points of view stated below:

Objective function: We propose a novel supervised model using a single supervised deep autoencoder to address the challenging 3D model retrieval problem based on depth images with significant occlusion. The proposed supervised single deep autoencoder objective function has an exclusive combination of sigmoid cross entropy loss and softmax loss.

Only one network for both the domains: We demonstrate the idea that instead of using parallel networks to get features from different domains, transfer learning is still possible in the case of cross-domain retrieval using only one single deep network. Also we show the strength of a supervised autoencoder for cross-modal 3D-model retrieval. To bridge the gap between two different domains and bring those into a single feature space, a single supervised deep autoencoder can effectively learn a single feature space from raw depth images and rendered 3D models. Bringing 3D models and 2D depth images in the same feature space helps to reduce their differences. Many researchers have brought 3D and 2D representation in the same space for successful cross-modal 3D model retrieval. However, in our research, the assumption is that the hidden layer feature of a deep autoenoder would bring two different domains in same feature space. In the encoded feature space, we can compare the similarity of 3D model and depth image in a meaningful way.

Asymmetric structure: We exhibit that an asymmetric deep structure of an autoencoder leads to a more robust latent space less invariant to small changes in samples than symmetric ones for cross modal retrieval.

Dealing with ambiguity: We have dealt with the incompleteness and ambiguity present in the depth images by adding higher level abstraction in two stages. Supervising the autoencoder with label information deals with incompleteness and occlusion present in the data and can be considered as the first stage. In the second stage, to downsize ambiguities we derive semantic concepts of the supervised autoencoder features by learning a subspace S. We assume that real depth images and rendered depth images are in the same space facilitating 3D model retrieval based on real depth images. One other strong advantage of our model is low dimensional features. Compared to 1000D dimensional features used in [10], 15D or 30D features work meaningfully in our model. We use the hidden encode layer features of a deep autoencoder to compare and retrieve 3D models. Low dimensional features are inexpensive with respect to time complexity to compare and retrieve 3D models.

Flexibility: Our model is not restricted to an equal number of depth images and 3D models. For retrieval, we can train our model with any number of depth image and models present in the database. In [12], CM and SCM algorithms are restricted to have same number of model and depth present in two different domains since they use canonical correlation analysis(CCA) [12]. Direct usage of images rather than image features, is another mighty advantage of this work. Compared to handcrafted features and shallow learning frameworks, the deep structure of an autoencoder can effectively learn complicated structures from raw depth images, which requires less domain knowledge. The deep network of our model can handle a huge number of training depth images. In our network, we have used around fifty nine thousand images for training compared to other cross domain models [10, 12, 15].

We address the challenging 3D model retrieval problem based on depth images with significant occlusion. Our model takes depth images as the direct input rather than handcrafted features and it is not restricted to an equal number of depth-model pairs for two different domains. Unlike other cross-domain retrieval approaches, our model can handle a large number of 3D models present in the database. Compared to handcrafted features and shallow learning frameworks, the deep structure of an autoencoder can effectively learn complicated structures from raw depth images. We outperformed all the state of the art methods using parallel network for cross-modal retrieval.

1.3 Proposed Retrieval Algorithms

Single deep unsupervised autoencoder approach (SDA): SDA is an unsupervised approach which is a purely data-driven, the real depth images and rendered depth images from 3D models make separate clusters because they belong to different data domains. The loss function for SDA's unsupervised autoenocder only depends on reconstruction loss sigmoid cross entropy instead of euclidean loss.

Single supervised deep autoencoder approach (SSDA-1): The proposed SSDA-1 is a cross-modal retrieval approach for retrieving 3D models given a real depth image query. SSDA-1 has a classification network to supervise the autoencoder with label information. Since the supervised approach guides the autoencoder with label information, real and rendered depths cluster in similar regions as desired for retrieval. Training the supervised deep autoencoder with two different domains gives the opportunity to determine a stopping criteria where the real depths are being classified with a good accuracy. The objective function we propose for single supervised deep autoencoder model SSDA-1 is a unique combination of sigmoid cross entropy loss and softmaxwithloss for reconstruction and classification of a supervised autoencoder respectively.

Single supervised deep autoencoder approach (SSDA-2): The proposed supervised autoencoder in our SSDA-2 model is not symmetric since it has three encoder layers and four decoder layers. In SSDA-2, instead of adding a classification network with the encoder we have added a classification network after the decoder to supervise the autoencoder with label information. This supervision indicates how well the reconstruction is being classified into different categories. To portray the basic difference between SSDA-1 and SSDA-2 in detail we can say that the classification network is tightly tied with the reconstruction in SSDA-2. In other words, the relationship between reconstruction and classification is straightforward. And in the case of SSDA-1 we classify the features instead of the reconstruction of the features, which makes the relationship between classification and reconstruction tricky or indirect.

Single supervised deep asymmetric autoencoder approach (SSDAA): The proposed asymmetric supervised deep autoencoder (SSDAA) for 3D model retrieval is an asymmetric autoencoder having less decoder layers than encoder layers. We have analyzed the relation between symmetric and asymmetric autoencoder structures and have shown that asymmetric deep autoencoders lead to more robust embedding and less invariant to small variation of samples. We have compared asymmetric autoencoder retrieval results with symmetric autoencoder and found that asymmetric deep autoencoder overcoming the symmetric deep autoencder for improving retrieval accuracy.

Semantics-enhanced single deep supervised autoencoder (S3DA and S3DAA): Semantic modeling can help a supervised autoencoder handle the ambiguity by providing the next-level abstraction. Previous research has shown that increased level of abstraction leads to better image or text retrieval . Semantic modeling applied over supervised-autoencoder features offers two stages of high level abstractions. The first abstraction comes from supervised autoencoder. The next abstraction comes from the semantic modeling of supervised autoencoder features. Supervised autoencoders restructure the autoencoder features with label information. When we apply semantic modeling on asymmetric structure SSDAA, we get our new model S3DAA where AA stands for asymmetric autoencoder. All the proposed approaches presented in this section are summarized in Table 1.1.

Proposed models	**Autoencoder structure**	**Classification layer**	**Semantic modeling**
SDA	Unsupervised	No classification layer	Not applied
SSDA-1	Supervised	Added to encoder	Not applied
SSDA-2	Supervised and asymmetric (less encoders than decoders)	Added to decoder	Not applied
SSDAA	Supervised and asymmetric (more encoders than decoders)	Added to decoder	Not applied
S3DA	SSDA-2	Added to decoder	Applied
S3DAA	SSDAA	Added to encoder	Applied

Table 1.1: Proposed models.

1.4 Applications

Due to the emergence of low cost depth sensors in recent years, depth image based 3D model retrieval is not an expensive approach for cross-modal retrieval. Rather it is a more affordable approach than sketch-based 3D model retrieval and RGB-D image based 3D model retrieval. To scan and rebuild 3D scenes using a 3D CAD model database has many applications in many areas such as robotics, surveillance, object detection, interior design and providing user interaction interface facility.

Augmented reality: Augmented reality (AR) is a mixture of reality with virtuality where people are assumed to interact with virtual objects along with real objects in the real world. Accurately finding the correct position of the virtual object in real world scene is the key to an effective AR application [16]. Depth image based 3D model retrieval can help find an object relative to other virtual and physical objects in a scene.

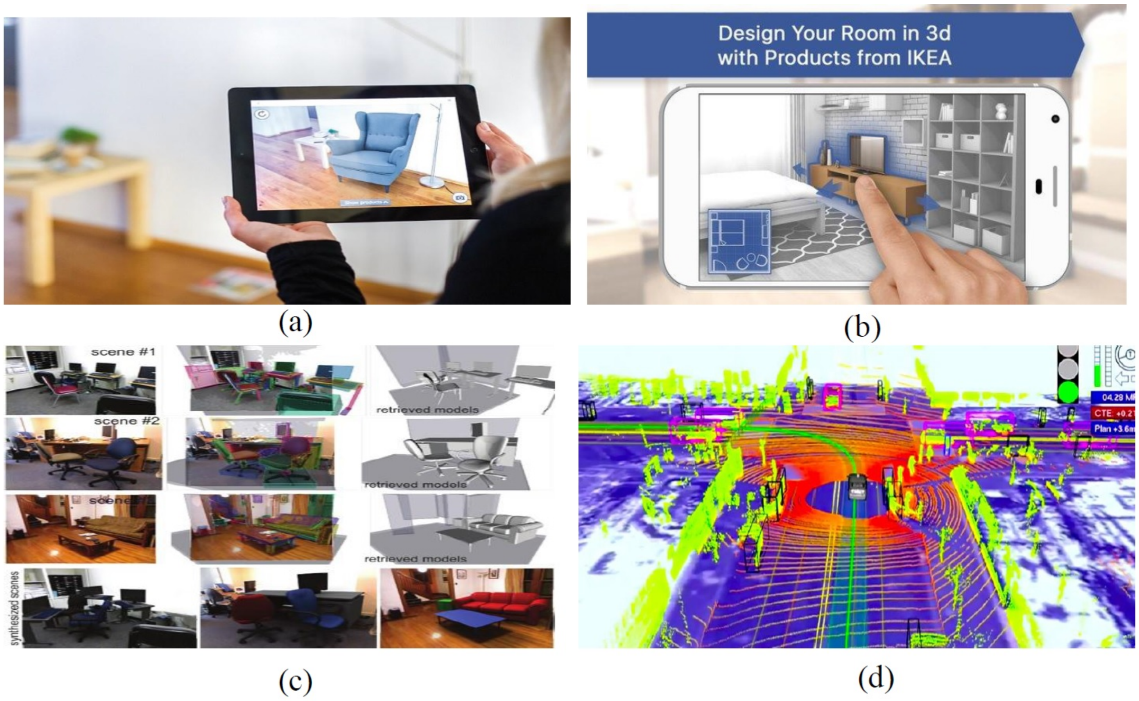


Figure 1.3: (a)Augmented reality [17], (b)Product design [18], (c)3D model retrieval and scene rearrangement [19], (d)Google's self-driving car uses lidar to create 3D image of its surroundings [20].

For example, a user can search for a sofa model that is similar to his sofa at home, and then virtually move the sofa around through augmented reality scene and see where he can place it. With the rapid growth of mobile devices with 3D sensors, 3D model retrieval and its derived applications can be as accessible and enjoyable as taking a picture.

Scene modeling: By retrieving the corresponding 3D models for the depth image objects present in the scene, scene modeling or 3D scene rebuilding can be done. Success in scene reconstruction and scene rearrangement depend on 3D model retrieval performance since 3D models offer high-level structure, complex and finer material features and critical details of entities. Depth image based 3D model retrieval can lower the cost of building the 3D model database search engine [21] instead of depending on sketch-based search engine.

Self driving car: Depth image based 3D model retrieval can help to calculate the distance between a self-driving car and the obstacle on its path. To understand the depth and size of the obstacle, retrieved 3D models can be employed as samples to train any neural network. In creating the surroundings as shown in Figure 1.3(d), 3D model retrieval based on depth image can help.

Product design industry and architecture: Depth image based 3D model retrieval can help product design industry by efficient and effective generation of new models based on the existing ones. Thus, it is a major aspect of new product development. Retrieving full 3D models for furniture can help home decoration and housekeeping where the user can edit the scene by rearranging furniture. IKEA has developed a smart phone app to help customers make their own furnishing choice. It is also possible to render a scene by adding retrieved synthetic objects into existing scenes which can help architectural designs.

3D model editing, printing and extrapolation: The growing popularity of depth image based 3D model retrieval explores the opportunity to help 3D content editing tasks. To handle the challenges of occlusion, object segmentation and producing correct perspective for 3D model editing and printing, 3D models retrieved for coarse depth object can be exploited.

CHAPTER II

RELATED WORK

The literature of this section discusses broadly different neural network and deep network autoenocder approaches and retrieval algorithms using deep networks. We present the related concepts from four aspects: unsupervised-autoencoder approaches, supervised-autoencoder approaches, pairwise-models for retrieval and semantic model approaches for retrieval purpose.

2.1 Unsupervised deep-autoencoder approaches

Stacked autoencoders (SAE) are a typical class of deep learning algorithms [22, 23]. Recent 3D shape feature retrieval approaches use unsupervised deep autoencoders to get 3D shape features [7, 8] for 3D model retrieval.

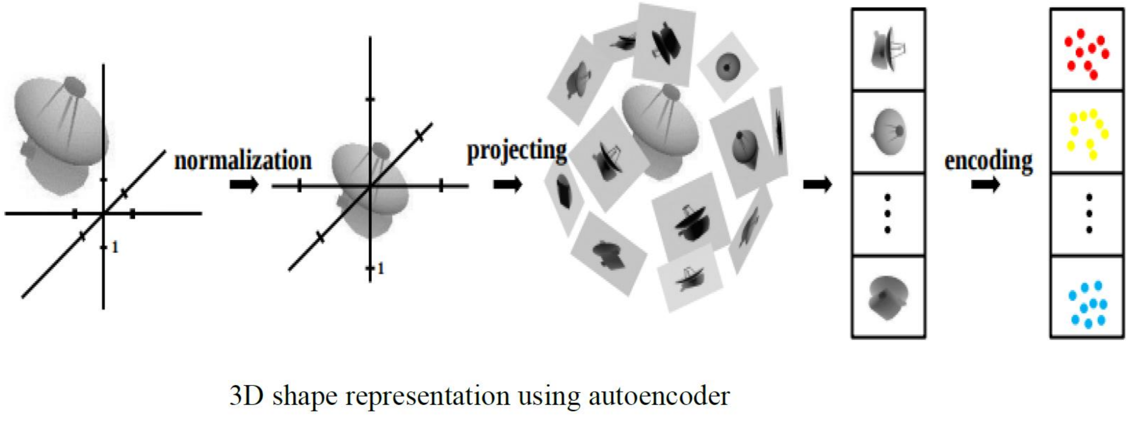
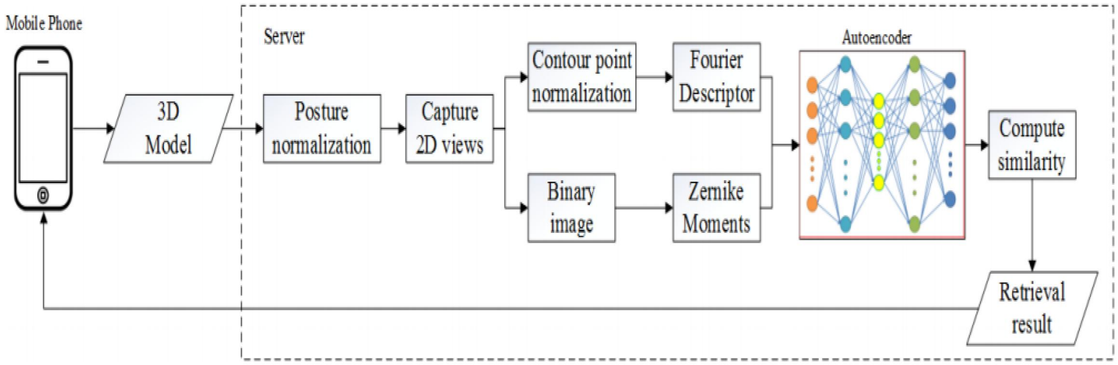


Figure 2.1: Deep learning representation using autoencoder for 3D shape retrieval [7].

In [7], an unsupervised autoencoder has been used for 3D shape retrieval. The

3D model is normalized, scaled and projected on to many 2D views and uses autoencoder for getting features for every view/2D image. The autoencoder is initialized by rbm (restricted boltzman machine) as the best performance was observed by this initialization. Figure 2.1 shows the flow chart of 3D shape representation using an autoencoder. First, pose normalization is done to remedy the differences in translation and scale present in the 3D models. Next, each 3D shape is represented by a set of depth-buffer images. Finally, all the projections are used to train the autoencoder to acquire a low-dimensional representation of the depth images and conduct 3D shape retrieval. The similarity we have with this this work is that they also have projected 3D shapes into 2D space and used an autoencoder for learning features from 2D images. Our work differs in that we do not normalize the 3D models and our input images are not complete. We have used noisy, partial and occluded images.



By sending a query 3D model to the server the retrieval result is obtained immediately.

Figure 2.2: 3D model retrieval based on deep autoencoder neural networks [8].

In [8], many views of the same 3D model have been rendered and normalized. In Figure 2.2 we see that different features of the generated views have been extracted before passing the views to the deep unsupervised-autoencoder for getting features for 3D model retrieval. In our work, we have not normalized or used different features of 2D views. Our approach uses direct image as input to the autoencoder. The similarity they have with our work is using many rendered views of same 3D model, and the dissimilarity is they have normalized the images and calculated different kind of features before passing to the autoencoder.

2.2 Supervised deep-autoencoder approaches

We have discussed supervised autoencoder approaches from the aspects of network architecture, number of hyper parameters and how the autoencoder is being supervised. In Figure 2.3, we have presented some examples of supervised autoencoder approaches where we have showed the objective function information and how our objective is different.

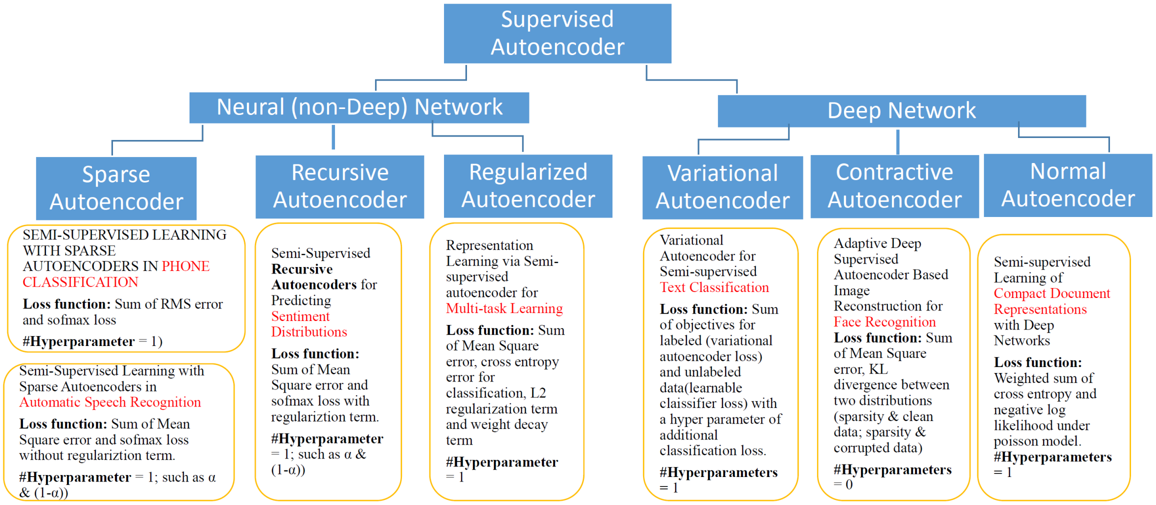

Supervised Autoencoder
Neural (non-Deep) Network
Deep Network
Sparse Autoencoder
Recursive Autoencoder
Regularized Autoencoder
Variational Autoencoder
Contractive Autoencoder
Normal Autoencoder
SEMI-SUPERVISED LEARNING WITH SPARSE AUTOENCODERS IN PHONE CLASSIFICATION
Loss function: Sum of RMS error and sofmax loss
#Hyperparameter = 1)
Semi-Supervised Learning with Sparse Autoencoders in Automatic Speech Recognition
Loss function: Sum of Mean Square error and sofmax loss without regulariztion term.
#Hyperparameter = 1; such as α & (1-α))
Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions
Loss function: Sum of Mean Square error and sofmax loss with regulariztion term.
#Hyperparameter = 1; such as α & (1-α))
Representation Learning via Semi-supervised autoencoder for Multi-task Learning
Loss function: Sum of Mean Square error, cross entropy error for classification, L2 regularization term and weight decay term
#Hyperparameter = 1
Variational Autoencoder for Semi-supervised Text Classification
Loss function: Sum of objectives for labeled (variational autoencoder loss) and unlabeled data(learnable claissifier loss) with a hyper parameter of additional classification loss.
#Hyperparameters = 1
Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition
Loss function: Sum of Mean Square error, KL divergence between two distributions (sparsity & clean data; sparsity & corrupted data)
#Hyperparameters = 0
Semi-supervised Learning of Compact Document Representations with Deep Networks
Loss function: Weighted sum of cross entropy and negative log likelihood under poisson model.
#Hyperparameters = 1

Figure 2.3: Example of supervised autoencoder approaches.

A semi-supervised single-layer sparse-autoencoder has been used in [3] for improving the performance of modeling speech recognition. Figure 2.4 shows the flow chart to calculate cost. The loss function is the weighted sum of sum-square loss (reconstruction error) and softmax-cross entropy loss (classification error) with one hyperparameter. In [5], a semi-supervised recursive-autoencoder has been used for sentence level prediction of sentiment label distributions. The cost function in this

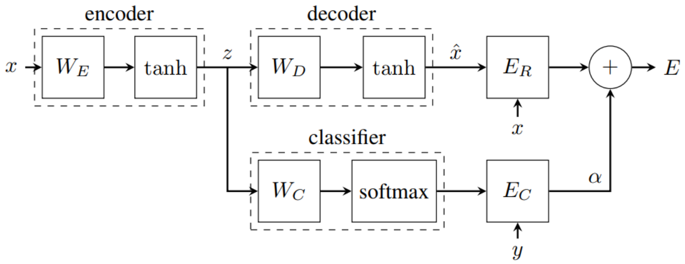


Figure 2.4: Semi supervised learning with sparse autoencoders in phone classification [3].

work is the weighted sum of reconstruction error (mean square error), softmax cross entropy error and a regularization term. The reconstruction error and cross-entropy error are weighted by one hyperparameter. In [1], the semi-supervised autoencoder has been used for multi-task learning. From Figure 2.5, we can see the framework of the model where encoding weights are shared by all the tasks which in turn share common representation. The loss function of this model is the summation of the sum square error term for reconstruction, the softmax regression error term weighted by one hyperparameter, the L2 norm regularization error term and a decay term.

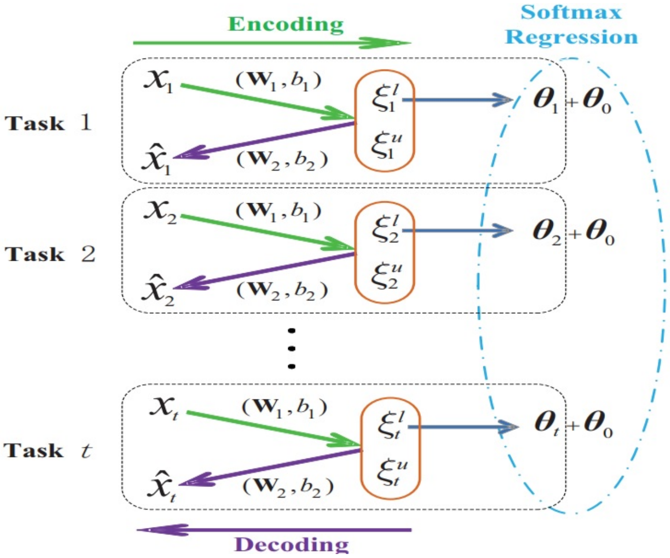


Figure 2.5: The framework of representation learning via semi-supervised autoencoder for multi-task learning [1].

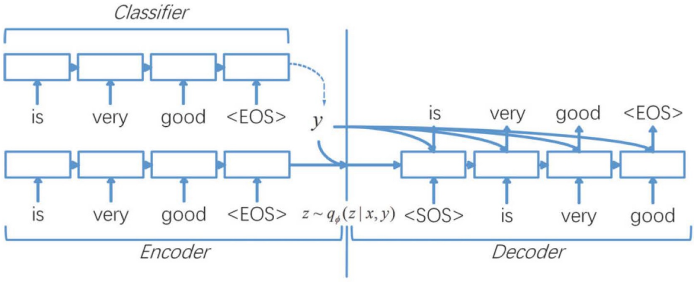
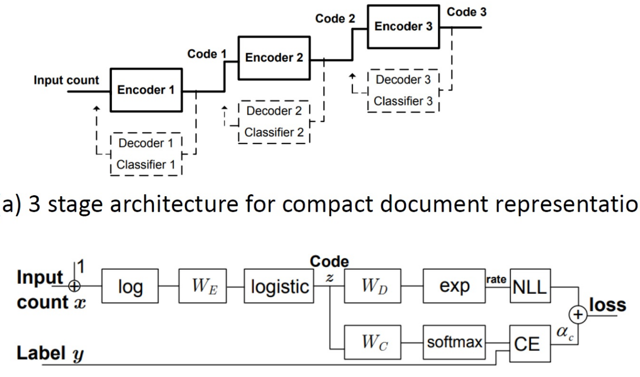


Figure 2.6: The framework of semi-supervised variational autoencoder for text classification [2].

In [2], the loss function is the summation of objectives for labeled data (variational autoencoder loss) and unlabeled data(learnable claissifier loss) with an additional classification loss weighted by one hyper parameter as shown in Figure 2.6. In [24], a supervised deep autoencoder has been used in facial recognition and the model is supervised by reconstructing the clean input data from a corrupted version of it. The objective function is the sum of mean squared error and Kullback-Leibler divergence between two distributions (sparsity and clean data; sparsity and corrupted data) with no hyperparameter. The sparsity mentioned here is usually a constant.



(a) 3 stage architecture for compact document representation

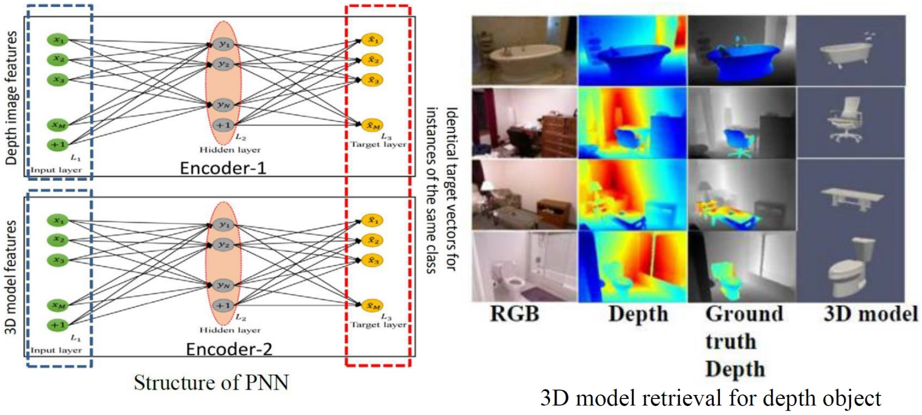
(b) Architecture of first stage with an encoder, a decoder and a classifier

Figure 2.7: The framework for semi-supervised learning of compact document representation with deep networks [25].

In [25], text document representation has been learned by a supervised deep autoencoder. The autoencoder is supervised by label information. From Figure 2.7, we see that the model has three stages where the n-th layer provides codes to train the layer above. The first stage loss function is the weighted sum of softmax cross-entropy (classification) and negative log-likelihood (Poisson regressor for reconstruction) under the Poisson model whereas the other two loss functions are the summation of softmax cross entropy and a Gaussian regressor. In our model, the loss function is the weighted sum of sigmoid cross-entropy and softmax. We have used a supervised deep autoencoder for the first time for 3D model retrieval and our approach differs from all the approaches stated in above in terms of objective function, number of hyper parameters and network structure.

2.3 Pairwise models for cross-domain retrieval

Recently deep learning has achieved great success in object recognition in computer vision. In our research, we have used a deep autoencoder which is a typical class of deep learning algorithms. Normally, cross modal approaches use a pairwise network model for retrieval but ours is a single model approach where we train both real depth and rendered depth images using one single network.



Structure of PNN

3D model retrieval for depth object

Figure 2.8: Pairwise neural network for depth image-based 3D model retrieval [10].

Our research problem is closely related with the pairwise neural network (PNN) approach [10]. The algorithm is a transfer learning based retrieval approach. In Figure 2.8 of PNN, a pairwise neural network encoder model has been used for 3D model retrieval. PNN has used LD-sift [26] for 3D model features and ScSPM for depth image features [27]. In the PNN approach, model features and depth features are classified to the same output vector if their category is same. Unlike PNN, we have employed one single deep autoencoder to train real and rendered depth images together. Unlike the PNN regression model, our single autoencoder works as a pure autoenocoder, it reconstructs the inputs and uses the hidden layer features for retrieval.

In sketch based 3D model retrieval [9], they have used a pairwise Siamese network

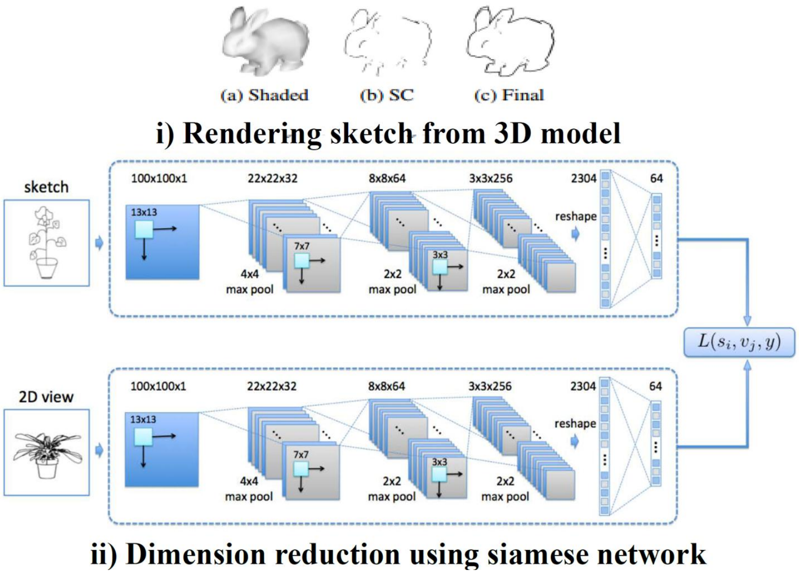


Figure 2.9: Pairwise siamese network model for sketch-based 3D model retrieval [9].

with a specific loss function. Figure 2.9 shows two identical deep networks known as siamese network have been used for sketch and rendered views. Unlike our model, they have used just two views to retrieve 3D models based on sktech images, whereas we have rendered ninety five views for one 3D model. The similarity that we have with this research is that, 3D models and sketch images are used in the same domain for cross modal retrieval. They have rendered sketch images from 3D models to retrieve 3D models. In pairwise relation learning approach [11], parallel autoencoders have modeled relationship for handwritten digit subtraction. First, using different autoencoders, features have been extracted. Those features are fed into another deep network to learn the relational model. Unlike this work, we have not used a two step process to get features. Ours is a single autoencoder which is used to get features for both the domains.

In domain invariant feature learning for depth image-based 3D model retrieval [28], again a pair-wise neural network model has been proposed. From Figure 2.10 we see that two networks are dedicated for depth image (ScSPM) and 3D model (LD Sift)

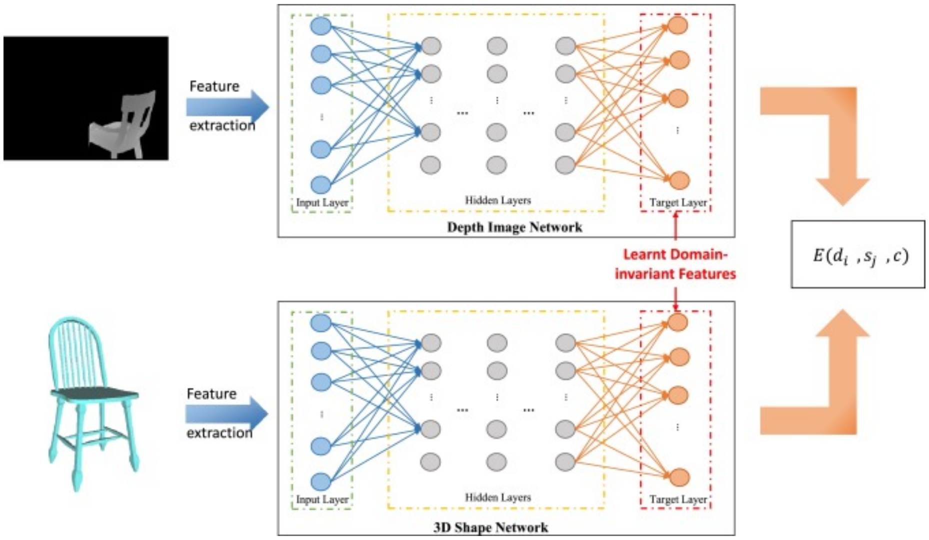
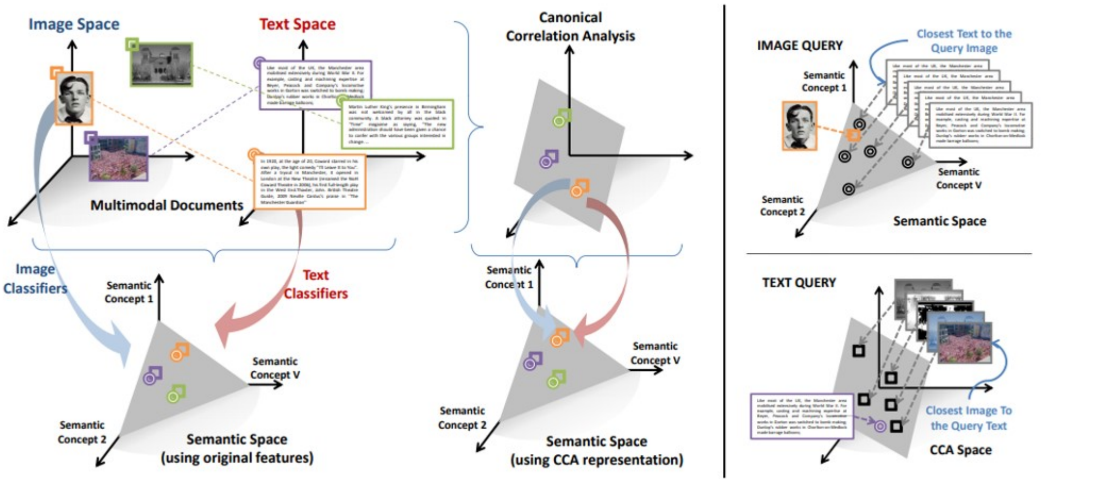


Figure 2.10: Learning domain invariant features using pairwise neural network model for depth image-based 3D model retrieval [28].

features respectively. The loss function in this pairwise model tries to minimize intra-class distance while maximizing inter-class variance to generate hand-crafted features in the hidden layer of each network.

2.4 Semantic models for cross-domain retrieval

Semantic matching is a new approach for cross-modal retrieval. In [12], image based text or text based image retrieval has been proposed using canonical correlation analysis (CCA), semantic matching (SM) and semantic correlation matching (SCM). We have two similarities with this research. First one is the application of semantic matching on features and second one is the dealing with cross-modal retrieval problem. However, there are several dissimilarities such as semantic matching is being applied on two different features (image features and text features) to bring those in a single feature space. Whereas in our work, we apply semantic matching on the same domain supervised features of real and rendered depth images.



Mapping text and image to CCA space and semantic space and a semantic space learned using CCA

Top: Retrieve text given image query

Down: Retrieve image given a text query

Figure 2.11: A new approach to Cross-Modal Multimedia Retrieval [12].

Figure 2.11 describes the semantic matching approach along with CCA. In [12], we have one other dissimilarity with the retrieval method which is the application of CCA on semantic matching. CCA restricts the SCM model to an equal number of training cross domain input pairs.

2.5 Asymmetric and symmetric autoencoder approaches

Autoencoders could be symmetric or asymmetric in its network structure. In the previous research using autoencoder, symmetry used to be enforced. Perhaps the reason behind this strict symmetry is for using tied weights between the encoder and decoder [29]. The unsupervised autoencoders used in 3D shape retrieval approaches [7,8] are all symmetric in nature. The semi-supervised and supervised autoencoders used in speech recognition, text representation, sentiment analysis, sentence level prediction, multitask learning and face recognition [1,3,5,6,24,25] basically have symmetric network other than the associated supervision network. Convolutional autoencoers have convolutional layers in the encoder, deconvolutional layers in the decoder and pooling-unpooling layers in the deep autoencoder network structure [30]. Recently researchers have been using asymmetric deep autoencoders [31] and upsampling layers instead of the decoder layers [32] to get rid of artifacts problems seen in convolutional deep autoencoders. Using asymmetry in deep autoencoder structure is a very current approach where researchers claim that symmetry is a forced restriction and not a requirement [33–36]. In [33] classification accuracy has been improved by using asymmetric deep autoencoder. In [34] the encoder is composed of several LSTM layers while the decoder is a few layers of fully-connected neurons. The particular structure of deep autoencoder is capable of recognizing detailed patterns to extract features from educational data. In [35] more robust and effective features have been learned by the application of asymmetric deep autoencoder.

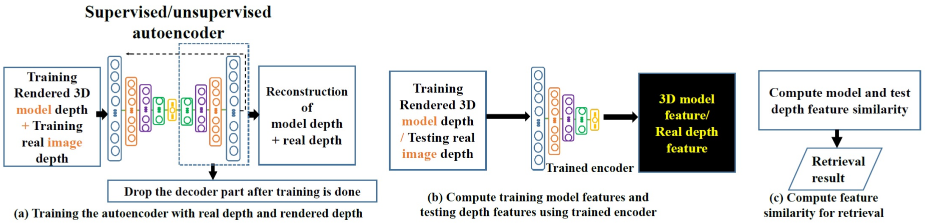
CHAPTER III

DEEP AUTOENCODERS FOR CROSS MODAL RETRIEVAL

3.1 Background

An unsupervised autoencoder is a purely data driven approach for learning features and clustering the data into different groups. But when the data is incomplete relying entirely on the data leads to incorrect grouping or extracting improper features. Supervision on an autoencoder with label information helps us in restructuring the features into correct groups. To reveal the usefulness of supervised learning over unsupervised for retrieval, we propose one unsupervised single deep-autoencoder (SDA) model and two supervised single deep-autoencoder (SSDA-1 and SSDA-2) models. A cross-modal retrieval approach using a **single** supervised autoencoder trained by two different domains(real and rendered depths) can be advantageous in two ways. First, it provides insight whether two different domains with same category are being grouped in the same group or not. Second, it gives the opportunity to visualize the training process for determining a stopping criteria where the real depths are being classified with a good accuracy. We consider the problem of 3D model retrieval from a database $M = M_1, M_2, ..., M_n$ of 3D models for a given query depth image.

The two advantages of the supervised approach have been demonstrated with examples and experiments in the following subsections of this chapter. The objective function we propose for single suprevised deep autoencoder models SSDA-1 and SSDA-2 is a unique combination of sigmoid cross entropy loss and softmaxwithloss for reconstruction and classification of a supervised autoencoder respectively. Our supervised models train rendered depth images and real depth images together. Fig-


Supervised/unsupervised autoencoder
Training Rendered 3D model depth + Training real image depth
Reconstruction of model depth + real depth
Drop the decoder part after training is done
Training Rendered 3D model depth / Testing real image depth
Trained encoder
3D model feature/ Real depth feature
Compute model and test depth feature similarity
Retrieval result

(a) Training the autoencoder with real depth and rendered depth

(b) Compute training model features and testing depth features using trained encoder

(c) Compute feature similarity for retrieval

Figure 3.1: General (supervised/unsupervised) single autoencoder approaches for 3D model retrieval.

ure 3.1 shows our general framework for unsupervised and supervised autoencoder approaches for 3D model retrieval.

Unsupervised feature learning is relatively new and very exciting area in Machine learning. An autoencoder tries to learn an approximation to the identity function in an unsupervised way. By unsupervised way we mean only the data is available without any training labels. An autoencoder reconstructs it's input. The learning algorithm of an autoencoder applies backpropagation where the target output is equal to the input. The encoder part of autoencoder maps the input to a hidden representation: *E= f(Wx+b)*, where f is a non-linear function. The decoder part tries to map the hidden representation back to the original input: *D=f(W'x+b')*. The parameters of the network are optimized in a way that the reconstruction error is minimized.

In the hidden layers of the autoencoder data is compressed in a way that reconstruction is possible from the compression. An autoencoder tries to cluster similar data in same region. It is part of a pre-training regime which helps to learn the weights and biases rather than randomly initializing them. The feature learning by an autoencoder can be done in a comprehensive way by altering the hidden layer with different number of neurons, sparsity, weight decay or corrupted input.

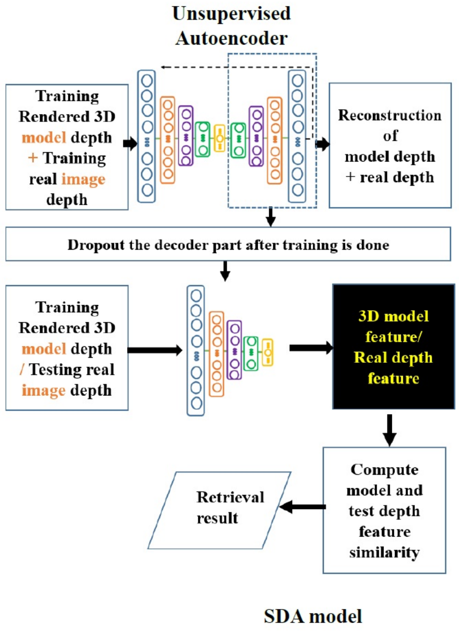

Unsupervised Autoencoder
Training Rendered 3D model depth + Training real image depth
Reconstruction of model depth + real depth
Dropout the decoder part after training is done
Training Rendered 3D model depth / Testing real image depth
3D model feature/ Real depth feature
Compute model and test depth feature similarity
Retrieval result
SDA model

Figure 3.2: Single Deep Autoencoder (SDA) approach for 3D model retrieval.

3.2 Single deep unsupervised autoencoder model (SDA)

Our single deep unsupervised autoencoder (SDA) model utilizes an unsupervised autoencoder to extract the depth image features for retrieval. The unsupervised autoencoder is entirely data driven and tries to predict the label for the data depending on it's data structure even though the data is occluded and incomplete. With an unsupervised autoencoder the real depths and rendered depth make separate clusters because those belong to different data domains. Since the supervised approach guides the autoencoder with label information, real and rendered depths cluster in similar regions as desired for retrieval. The loss function for SDA's unsupervised autoencoder only depends on reconstruction loss sigmoid cross entropy instead of euclidean loss. SDA model presented in Figure 3.2 describes the research problem.

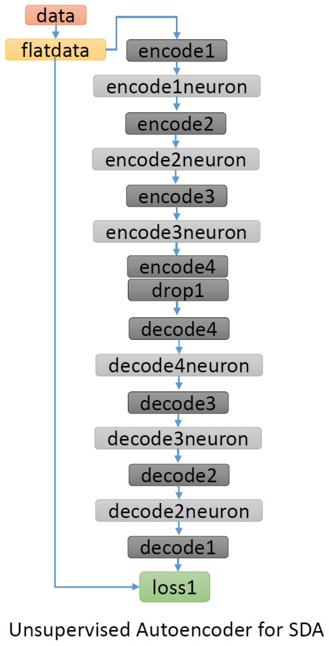


Figure 3.3: Deep network architecture for unsupervised autoencoder.

3.2.1 Network architecture for SDA

The deep unsupervised autoencoder in SDA has four encoder layers and four decoder layers. Therefore, the deep network has a total of eight inner product layers. To prevent overfitting, a dropout layer was used under the fourth encoder layer [37]. Figure 3.3 shows the architecture of our unsupervised network. SDA network autoencoder consists of encode1(2000), encode2(1000), encode3(500), encode4(30), decode4(500), decode3(1000), decode2(2000) and decode1(19200) layers. The kernel filter is gaussian type for all the layers. We input rendered 3D model depths and real depths together to train the autoencoder.

3.2.2 Loss function for SDA

The loss function E_u for the unsupervised autoenocder network is sigmoid cross entropy function and defined as below:

$$E_u = -\frac{1}{N}\sum_{n=1}^{N}[t_n \log \hat{p}_n + (1-t_n)\log(1-\hat{p}_n)], \tag{3.1}$$

where the prediction $\hat{p}_n$ is the sigmoid function applied on the last decoder's inner-product layer having the same dimension as input, t_n is the original input, and N is the number of training samples.

3.2.3 Optimization and retrieval using SDA

The unsupervised autoencoder model is trained optimizing the reconstruction error term E_u. E_u is the sigmoid cross entropy loss function and corresponds to loss1 in Figure 3.3. We have used AdaDelta [38] as an optimization function since it gave the lowest reconstruction error. After training the network, we dropped the decoder portion. Using the encoder part of the autoencoder as a deploy network, we get 30D rendered 3D model features $\mathbf{M}_u$ and 30D real test depth image features $\mathbf{D}_u$ from the fourth encoder layer. The distance between a model M and a query depth image D is the distance between the autoencoder feature M_u for the model and the autoencoder feature D_{qu} for the depth and defined as below:

$$Dist_{sda}(M, D) = d(M_u, D_{qu}), \tag{3.2}$$

where $d(.)$ is the distance between M_u and D_{qu} . We refer to this type of retrieval as retrieval by SDA.

3.3 Single supervised deep autoencoder model (SSDA-1)

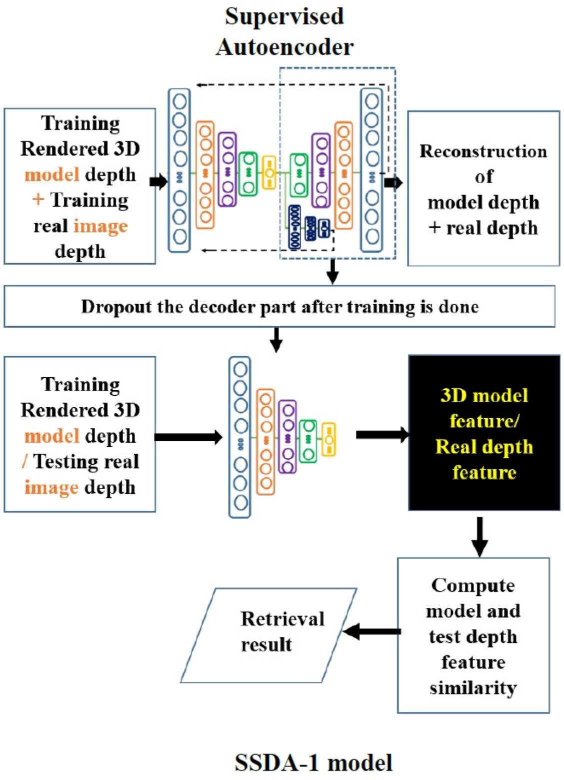

Supervised Autoencoder
Training Rendered 3D model depth + Training real image depth
Reconstruction of model depth + real depth
Dropout the decoder part after training is done
Training Rendered 3D model depth / Testing real image depth
3D model feature/ Real depth feature
Compute model and test depth feature similarity
Retrieval result
SSDA-1 model

Figure 3.4: Supevised autoencoder network with dropout for SSDA-1.

The proposed SSDA-1 is a cross-modal retrieval approach for retrieving 3D models given a real depth image query. SSDA-1 consists of a **single** supervised deep autoencoder trained by two different domains(real and rendered depths). As a supervised model during the training of SSDA-1 we can determine a stopping criteria or a suitable trained network with which the real depths are being classified with a good accuracy. The autoencoder in the SSDA-1 model is supervised by a classification network. This network supervises the autoencoder with label information. Since our real depth objects are incomplete and partial, classification information helps to restructure the autoencoder features to facilitate the retrieval performance. The objective function we propose for single suprevised deep autoencoder model SSDA-1 is a unique combination of sigmoid cross entropy loss and softmaxwithloss for reconstruction and classification in a supervised autoencoder respectively. SSDA-1 model presented in Figure 3.4 describes the research problem.

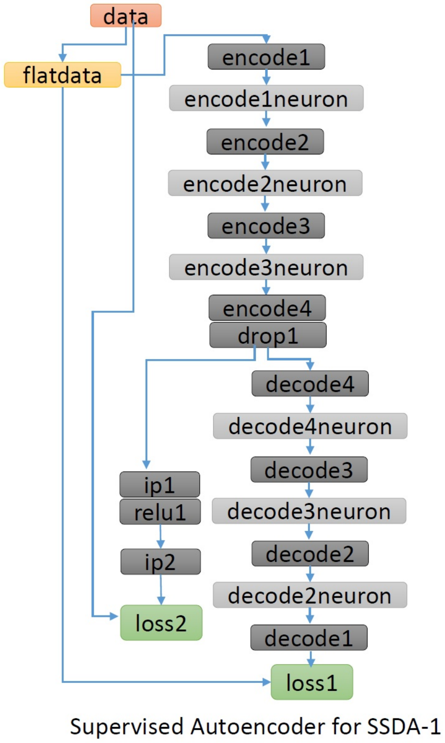


Figure 3.5: Deep network architecture for supervised autoencoder of SSDA-1.

3.3.1 Network architecture for SSDA-1

The supervised autoencoder has four encoder layers and four decoder layers. In this network, after the fourth encoder layer we have added two innerproduct layers to backpropagate the classfication error. Therefore, the network has a total of ten inner product layers. To prevent overfitting we have added one dropout layer after fourth encoder layer with fifty percent dropout ratio. Figure 3.5 shows the architecture of our SSDA-1 network. SSDA-1 network autoencoder consists of enocode1(1000), encode2(500), encode3(250), encode4(30), decode4(250), decode3(500), decode2(1000), decode1(19200), ip1(500) and ip2(5/7/10) layers. We have defined an objective function for the supervised autoencoder for adding classification information to the autoencoder network.

3.3.2 Loss function for SSDA-1

The cost function E_s for the supervised autoenocoder network is an weighted sum of sigmoid cross entropy loss and softmax regression loss and defined as below:

$$E_s = \alpha E_1 + \beta E_2, \tag{3.3}$$

where the reconstruction error term E_1 is the sigmoid cross entropy loss function from the autoencoder and corresponds to loss1 in Figure 3.5 and defined below:

$$E_1 = E_u = -\frac{1}{N}\sum_{n=1}^{N}[t_n \log \hat{p}_n + (1 - t_n)\log(1 - \hat{p}_n)], \tag{3.4}$$

where the prediction $\hat{p}_n$ is the output from the last decoder layer having the same dimension as the input, t_n is the input and N is the number of training samples.

$$\hat{p}_n = g(\mathbf{w}.\mathbf{x}_n) = \frac{1}{(1 + e^{-w.x_n}),} \tag{3.5}$$

where $g(z)$ is the sigmoid function applied on the hidden layer's (decoder) output, $\mathbf{w}$ are the weights. And the classification loss term E_2 is the softmax loss function from classifier and corresponds to loss2 in Figure 3.5 and defined as below:

$$E_2 = -\frac{1}{N}\sum_{n=1}^{N}\log(\hat{p}_{n,l_n}), \tag{3.6}$$

where $\hat{p}_{n,l_n}$ is the softmax output of the classifier defined in equation (3.7) below. In $\hat{p}_{n,l_n}$ the subscripts n, l_n indicate the nth training sample and the category label respectively. N is the number of training samples. The hyper-parameters in (3.3), α and β, control the trade-off between the two costs. To compute the softmax loss, the softmax is performed first on the hidden units of the layer named ip2 as below:

$$\hat{p}_{n,l_n} = \frac{\exp(x_{n,l_n})}{\sum_{k'=1}^{K}\exp(x_{n,k'})}, l_n \in [0, 1, 2, ..., K - 1], \tag{3.7}$$

where x_{n,l_n} is the score of each neuron computed for nth sample. In our case, we have ten categories, so in our case $K = 10$. Then, the multinomial logistic loss (-log likelihood) is computed on the softmax output as below:

$$\therefore \qquad E_2 = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{\exp(x_{n,l_n})}{\sum_{k'=1}^{K} \exp(x_{n,k'})} = -\frac{1}{N}\sum_{n=1}^{N}(x_{n,l_n} - \log \sum_{k'=1}^{K} \exp(x_{n,k'})), \tag{3.8}$$

where SoftmaxWithLoss function E_2 is the mean over all training examples and is basically multinomial logistic regression used for predicting a single class of K mutually exclusive classes.

3.3.3 Optimization and retrieval using SSDA-1

The supervised-autoencoder model is trained optimizing the cost function in equation 3.3. For the 5 and 7-category experiments we use $\alpha = 0.8$ and $\beta = 0.2$. For the 10-category experiment we set $\alpha = 1$ and $\beta = 1$. The two hyper-parameters α and β are optimized empirically from the training data. In this supervised case AdaDelta [38] which is a gradient based optimization method optimizes the cost function. After training the network, we dropped the decoder portion. In SSDA-1 we define $\mathbf{M}_s$ as 3D model features (rendered depth image features) and $\mathbf{D}_s$ as testing real depth image features. Using the encoder part of the autoencoder as a deploy network, we get 30D rendered 3D model features $\mathbf{M}_s$ and 30D real test depth image features $\mathbf{D}_s$ from the fourth encoder layer. The distance between a model M and a query depth image D is the distance between the autoencoder feature M_s for the model and the autoencoder feature D_{qs} for the depth and defined as below:

$$Dist_{ssda1}(M, D) = d(M_s, D_{qs}), \tag{3.9}$$

where $d(.)$ is the distance between M_s and D_{qs} . We refer to this type of retrieval as retrieval by SSDA-1.

3.4 Single supervised deep autoencoder model (SSDA-2)

We have proposed two single supervised deep autoencoder models SSDA-1 and SSDA-2. SSDA-2 has some differences with SSDA-1. Unlike SSDA-1 the supervised autoencoder in our SSDA-2 model is not symmetric since it has three encoder layers and four decoder layers. In SSDA-2, instead of adding a classification network with the encoder we have added a classification network after the decoder to supervise the autoencoder with label information. This supervision indicates how well the reconstruction is being classified into different categories correctly. To portray the basic difference between SSDA-1 and SSDA-2 in details we can say that the classification network is tightly tied with the reconstruction in SSDA-2. In other words, the relationship between reconstruction and classification is straightforward. And in the case of SSDA-1 we classify the features instead of the reconstruction of the features, which makes the relationship between classification and reconstruction tricky or indirect. SSDA-2 has better retrieval accuracy than SSDA-1. The training time is slightly less in SSDA-2 is because of lower number layer or neuron number than SSDA-1.

3.4.1 Network architecture for SSDA-2

The supervised autoencoder in our SSDA-2 model has three encoder layers and four decoder layers, one encoder layer less than the previous supervised autoencoder of SSDA-1. After the fourth decoder layer we have added two innerproduct layers to backpropagate the classfication error. Therefore, the network has a total of ten inner product layers and a dropout layer after third encoder layer to prevent overfitting. SSDA-2 network autoencoder consists of encode1(1000), encode2(500), encode4(30), decode4(250), decode3(500), decode2(1000), decode1(19200), ip1(500) and ip2(5/7/10) layers. Figure 3.7 shows the architecture of our SSDA-2 autoencoder network. We input 3D model depths and real depths together to train the autoencoder.

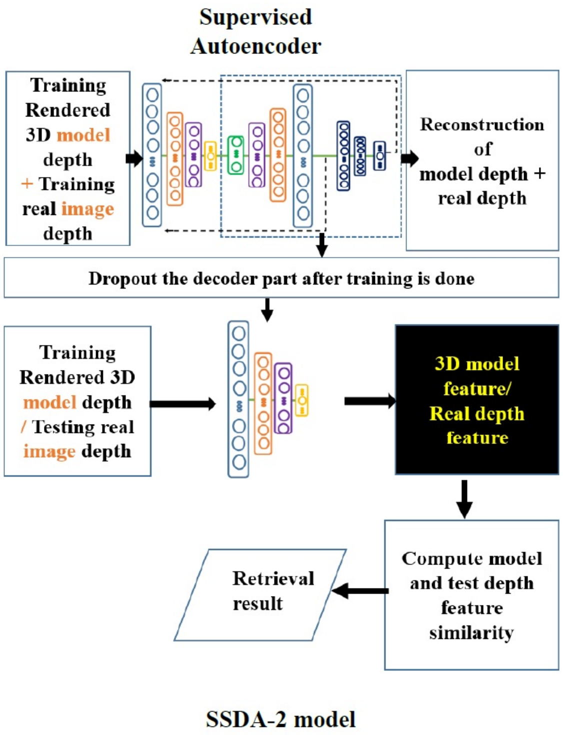

Supervised Autoencoder
Training Rendered 3D model depth + Training real image depth
Reconstruction of model depth + real depth
Dropout the decoder part after training is done
Training Rendered 3D model depth / Testing real image depth
3D model feature/ Real depth feature
Compute model and test depth feature similarity
Retrieval result
SSDA-2 model

Figure 3.6: Supevised autoencoder network with dropout for SSDA-2.

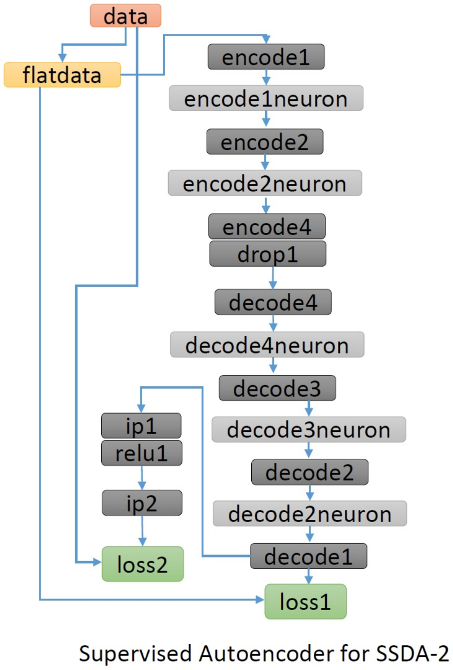


Figure 3.7: Deep network architecture for the supervised autoencoder of SSDA-2.

3.4.2 Loss function for SSDA-2

The cost function for the supervised autoencoder in the SSDA-2 model is defined as below:

$$E_{s2} = E_1 + E_2, \tag{3.10}$$

where the reconstruction error term E_1 is the sigmoid cross entropy loss function from the autoencoder and corresponds to loss1 in Figure 3.7 and defined in equation 3.4. The classification loss term E_2 is the softmax loss function from the classifier and corresponds to loss2 in Figure 3.7 and is defined in equation 3.6. Unlike the SSDA-1 model, there is no hyper-parameter in the supervised autoencoder of SSDA-2 model.

3.4.3 Optimization and retrieval using SSDA-2

The cost function in equation 3.10 is optimized using AdaDelta [38]. After training the network, we dropped the decoder portion. In SSDA-2 we define $\mathbf{M}_{s2}$ as 3D model features (rendered depth image features) and $\mathbf{D}_{s2}$ as testing real depth image features. Using the encoder part of the autoencoder as a deploy network, we get 30D rendered 3D model features $\mathbf{M}_{s2}$ and 30D real test depth image features $\mathbf{D}_{s2}$ from the fourth encoder layer. The distance between a model M and a query depth image D is the distance between the autoencoder feature M_{s2} for the model and the autoencoder feature D_{qs2} for the depth and defined as below:

$$Dist_{ssda2}(M, D) = d(M_{s2}, D_{qs2}), \tag{3.11}$$

where $d(.)$ is the distance between M_{s2} and D_{qs2} . We refer to this type of retrieval as retrieval by SSDA-2.

3.5 Experiments

3.5.1 Experimental setup

We performed 5, 7, and 10 category experiments to show the scalability of the proposed methods. We have chosen python and caffe deep learning framework to train and test the deep networks. The experiments performed are: experiments on different learning rates and experiments on adding a dropout layer to prevent overfitting. We have also shown how to choose the number of training iteration as a stopping criteria. We have used minibatches of size 80 and 100. The learning rate, which gives a reconstruction error neither too high nor too low, has been chosen. To prevent overfitting and extrapolation we have used a dropout layer with 50% dropout ratio. We have listed our experimental setup in Table 3.1.

Chosen Framework	Caffe, python
Training Networks	1. Unsupervised single deep autoencoder(SDA), encode1(1000), encode2(500), encode3(250), encode4(30), decode4(250), decode3(500), decode2(1000) and decode1(19200) 2. Supervised single deep Autoencoder(SSDA-1), encode1(1000), encode2(500), encode3(250), encode4(30), decode4(250), decode3(500), decode2(1000), decode1(19200), ip1(500) and ip2(5/7/10) 3. Supervised single deep Autoencoder(SSDA-2). encode1(1000), encode2(500), encode4(30), decode4(250), decode3(500), decode2(1000), decode1(19200), ip1(500) and ip2(5/7/10)
Validation data	Real part of the training data (real depth)
Performance of the Network	Retrieval performance measured by 5 well recognized metrics: NN, FT, ST, DCG and mAP
Experiment type	5, 7 and 10-category experiments
Training experiments	Experimented on: different learning rates and dropout layer
Batch size	SDA:100, SSDA-1:100, SSDA-2:80
Stopping criteria	SDA: 7000, 10000, 12000 iterations for 5, 7 and 10 category experiment respectively SDA-1 and SDA-2: Depends on real training depth
Prevention of over-fitting	Dropout layer

Table 3.1: Experimental setup.

Datasets

Our first data domain is real depth image database. We have used NYUD2 data set [13] which is comprised of RGBD images from a variety of indoor scenes recorded by Microsoft Kinect. It has 1449 densely labeled pairs of aligned RGB and depth images. We have only worked with depth images. We have used ground truth segmentation of NYUD2 depth images to verify our results. The depth images containing any of the ten indoor objects (bathtub, bed, chair, desk, dresser, nightstand, table, toilet, monitor and sofa) are considered candidate inputs. The input depth image has one object per image. For 5 categories we have 1260 training depth and 927 testing depth images and for 10 categories we have 1907 training depth and 1440 testing depth images. In Figures 3.8 and 3.9, we have given examples of some candidate depth images from the depth channel of the given given RGB-D image.

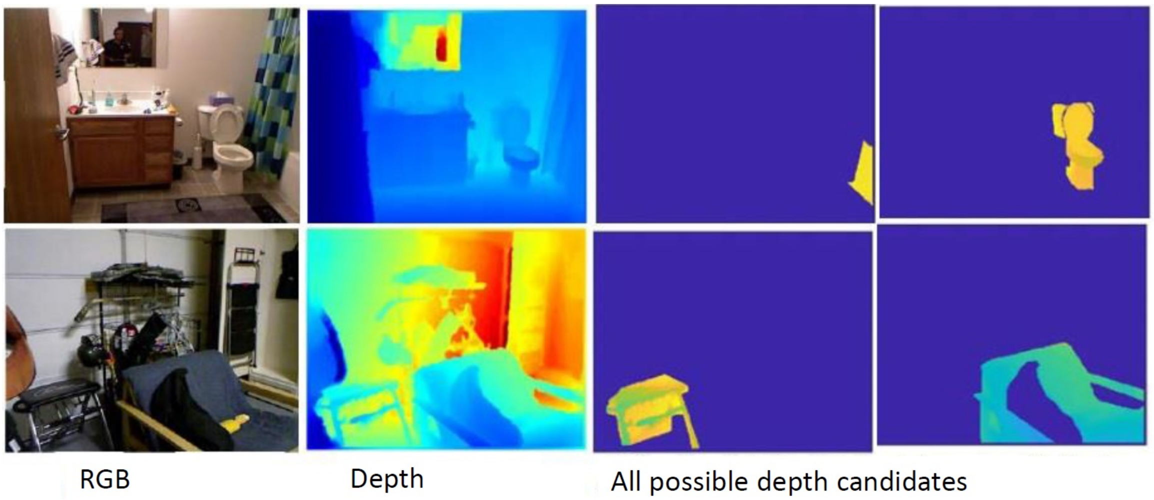


Figure 3.8: Example of segmented objects from 3 NYUD2 depth images. A candidate input depth image contains one of the ten objects (bathtub, bed, chair, desk, dresser, monitor, nightstand, sofa, table and toilet). The depth images only have two candidate depth images each.

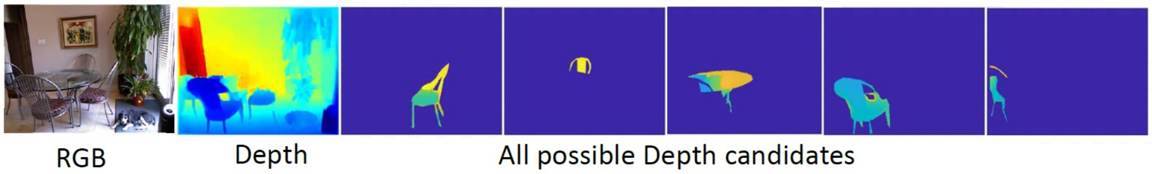


Figure 3.9: Example of segmented objects from three NYUD2 depth images. A candidate input depth image contains one of the ten objects (bathttub, bed, chair, desk, dresser, monitor, nightstand, sofa, table and toilet). The depth image has five possible candidate depth images.

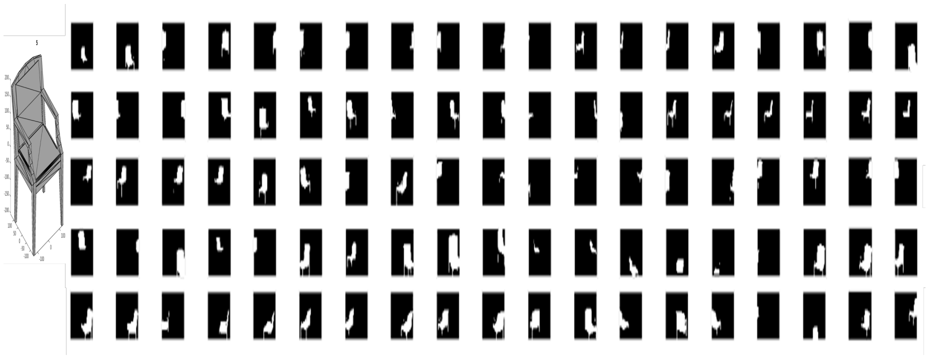


Figure 3.10: The rendered depth images under 95 views of a 3D chair model from ModelNet10.

Our second database domain is 3D model for which we have used ModelNet10 [14]. In ModelNet10 benchmark, 3D models are organized by common indoor scene objects with categorization labels. We downloaded 605 3D models belonging to the ten categories. From one model we have rendered ninety five depth images varying tilt, scale, orientation and 3D location. We have chosen those parameters based on the object statistics and observation of NYUD2 [13] training images. An example of rendered views has been given in Figure 3.10. We have followed the assumption that most objects are aligned in the direction of gravity as suggested in sliding shapes [39]. Therefore, we have 28940 rendered depth images for training the 5 category experiment and 59382 rendered depth images for training the 10 category experiment. In Table 3.2, we have listed our training and testing depth image list.

Categories	**Number of depth images** (# of training/# of testing)	**Number of 3D models**
1. bathtub	65(35/30)	26
2. bed	321(161/160)	37
3. chair	1355(826/529)	113
4. desk	341(176/165)	51
5. dresser	105(62/43)	57
6. monitor	177(78/99)	51
7. nightstand	153(86/67)	41
8. sofa	256(146/110)	90
9. table	502(300/202)	105
10. toilet	72 (37/35)	34
Total 10 categories	Total Training depth: 1907	Total Training model:
	Total Testing depth: 1440	605*95 =57,475

Table 3.2: NYUD2 and ModelNet10 dataset.

Evaluation criteria

The evaluation metrics used to measure the retrieval accuracy of 3D models in this research include five standard quantitative statistics which are Nearest Neighbor, First Tier, Second Tier, Discounted Cumulative Gain and Average Precision. We have also shown the retrieval performance using 11 point Precision Recall (PR) curves. These metrics are well recognized metrics in information retrieval field for 3D shape retrieval performance evaluation [40]. The 3D shape retrieval evaluation procedure is straightforward. In response to a given set of queries by a user, an algorithm searches the database of 3D model features and returns an ordered list of 3D models. Since this list is ordered we call it a ranked list.

For 3D model retrieval, the idea of a ranked list is very important. The greater the ranked position of a relevant object the less valuable it is for the user. Since it is less likely that the user will put time and effort into examining the objects ranked on the later section of a list. An example of a relevance list is given below:

Rel=<1, 1, 0, 0, 0, 1, 1, 0, 1, 0 . . . . >

where the binary value 1 denotes that the retrieved model is relevant and and 0 denotes irrelevancy. The cumulative gain CG at position n of this list Rel is the summation of relevance list from position 1 to n.

Nearest Neighbor (NN): NN is the average percentage of the closest K matches that belong to the same category as the depth image query. Greater percentage of this measure means better retrieval result where the highest limit of this metric is 100%. In our work, the first position of the cumulative gain vector is taken as NN measurement which means K=1.

First Tier (FT): FT is the percentage of the closest K matches that belong to the same category as the depth image query where K depends on how many models are present in the database which match the query's class. Greater percentage of this measure means better retrieval result where the highest limit of this metric is 100%. Usually for first tier, $K = |C| - 1$ where C is the number of class members for a specific class.

Second Tier (ST): ST is the percentage of the closest K matches of 3D models that belong to the same category as the depth image query where K depends on how many models are present in the database which match the query's class. Greater percentage of this measure means better retrieval result where the highest limit of this metric is 100%. Usually for second tier, $K = 2 * |C| - 1$ where C is the number of class members for a specific class. Second tier is little less stringent than first tier since K is twice as bigger as second tier.

Discounted Cumulative Gain (DCG): DCG is a metric which progressively reduces the importance of the object retrieved as the rank goes higher but this reduction is not too sharp or steep. The algorithm of DCG includes the ranked list G. DCG is defined as follows [41]:

$$DCG_i = \begin{cases} G_1, & \text{if } i = 1, \\ DCG_{i-1} + \frac{G_i}{\log_2(i)}, & \text{otherwise.} \end{cases} \tag{3.12}$$

The final DCG is divided by the maximum possible DCG defined as below:

$$DCG = \frac{DCG_k}{1 + \sum_{j=2}^{|C|} \frac{1}{\log_2(i)}} \tag{3.13}$$

where k is the number of 3D models present in the database. The usual convention is to look at the normalized DCG in which DCG value is scaled down by the average over all algorithms.

Mean Average Precision (mAP): Average precision (AP) is the average of precision scores at the rank of a relevant object retrieved. This measure uses a ranked list and changes its value at a rank where the object is relevant. In a ranked list where relevant objects are retrieved at the rank of 1, 2, 4, 7, 10; the AP at each rank of the ranked list would be 1, 1, 0.75, 0.57, 0.50. The average precision (AP) function is a monotonous decreasing function. Mean average precision is the mean of AP over all the queries.

Precision-Recall (PR) curves: Precision is the ratio of retrieved-relevant objects to all the retrieved objects. And Recall is the ratio of retrieved-relevant objects to all the relevant objects. If A is the set of all the relevant objects and B is the set of all the retrieved objects then $Precision = (A \cap B)/B$ and $Recall = (A \cap B)/A$. We have visualized our retrieval performance using 11-point precision-recall curve. Precision-recall curves are the relation between precision and recall for all depth image queries indicating retrieval performance. The interpolated average of precision at different recall levels (0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0) are calculated for an 11-point PR curve. For 11-point PR curve interpolated average of precision at different recall levels of 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0 are calculated. Recall is plotted on the x-axis and precision is plotted on the y-axis.

Training supervised and unsupervised deep autoencoders

In this section we discuss training the unsupervised autoencoder in SDA, the supervised autoencoder in SSDA-1, and the supervised autoencoder in SSDA-2 for five, seven, and ten categories. We also show the experiments to choose the appropriate learning rate. The learning rate we have chosen in unsupervised section has also been used in supervised section. This is because, in supervised autoencoder model, the autoencoder part plays the major role. We do not interrupt the autoencoder reconstruction and have chosen the same learning rate for unsupervised and supervised model for 3D model retrieval. Additionally we discuss the with-dropout and without-dropout comparisons for the autoencoder networks.

Training experiments with different learning rates

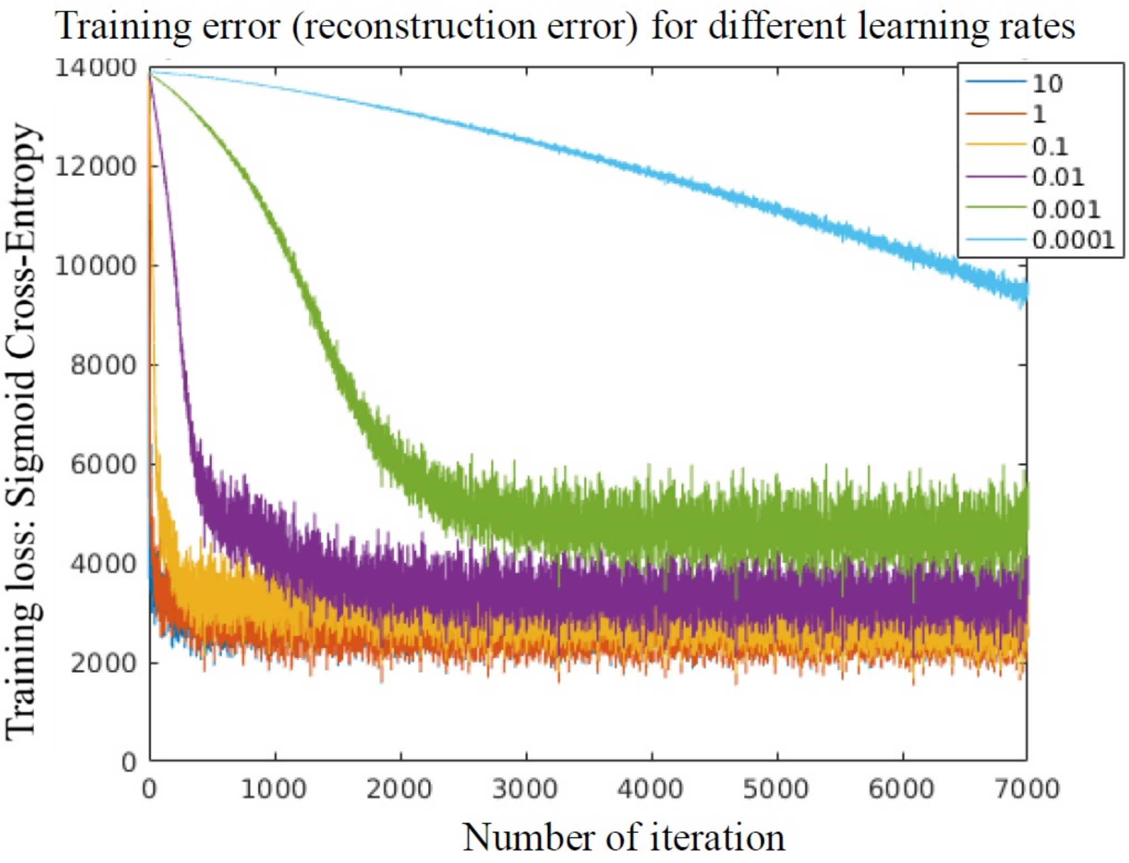


Figure 3.11: Learning rate experiment for unsupervised autoencoder for 5 categories.

In Figure 3.11, the experiment of choosing the learning rate for unsupervised deep autoencoder for five categories has been shown. We have chosen a learning rate of 0.01 because in this case, the learning is not too slow or too fast. Also, training error is not too low which is good for preventing overfitting.

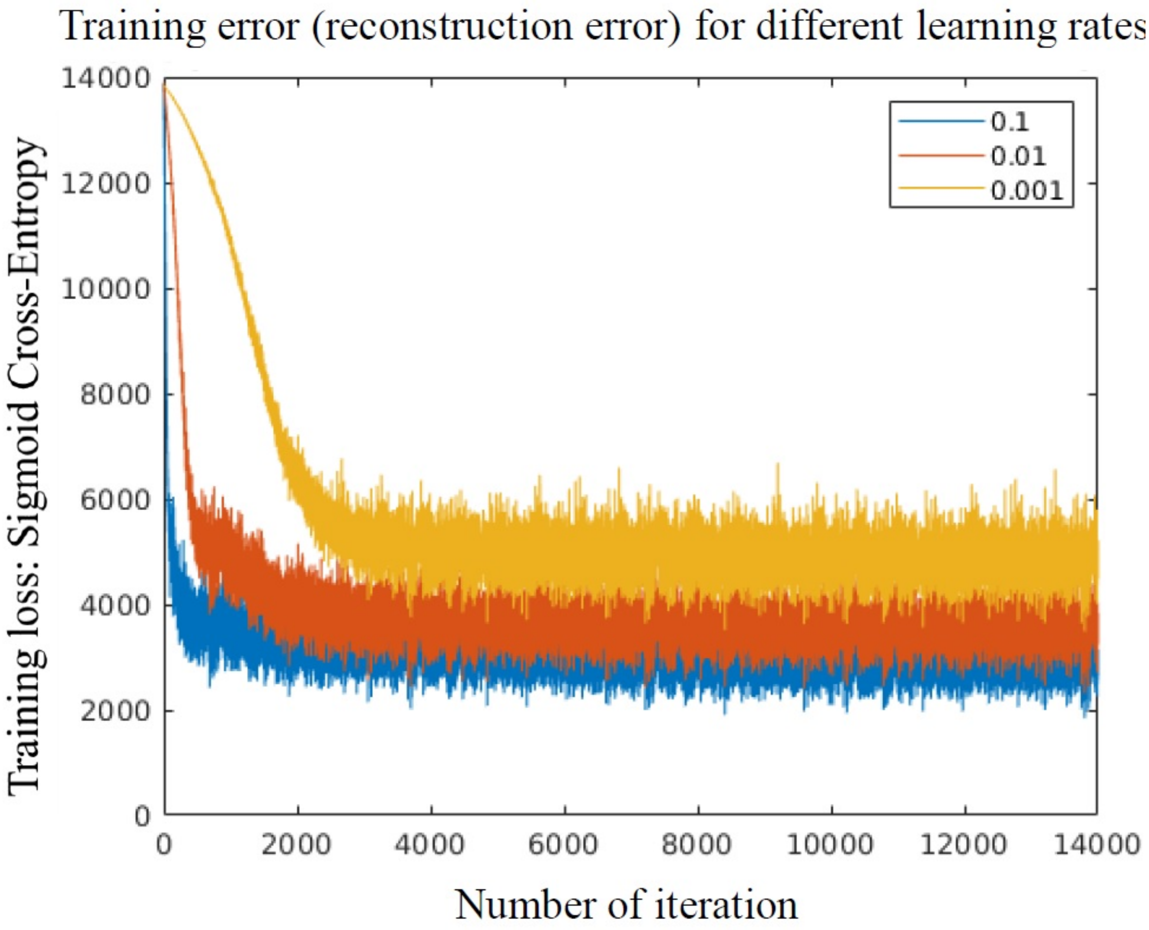


Figure 3.12: Learning rate experiment for unsupervised autoencoder for 10 categories.

In Figure 3.12, the experiment of choosing the learning rate for unsupervised autoencoder for ten categories has been shown. Like the five category experiment, we have chosen a learning rate of 0.01 because in this case, the learning rate is not too slow or too fast. Also, the training error is not too low which is good for preventing overfitting. This learning rate was used for the seven category experiment as well.

Training experiments with dropout layer

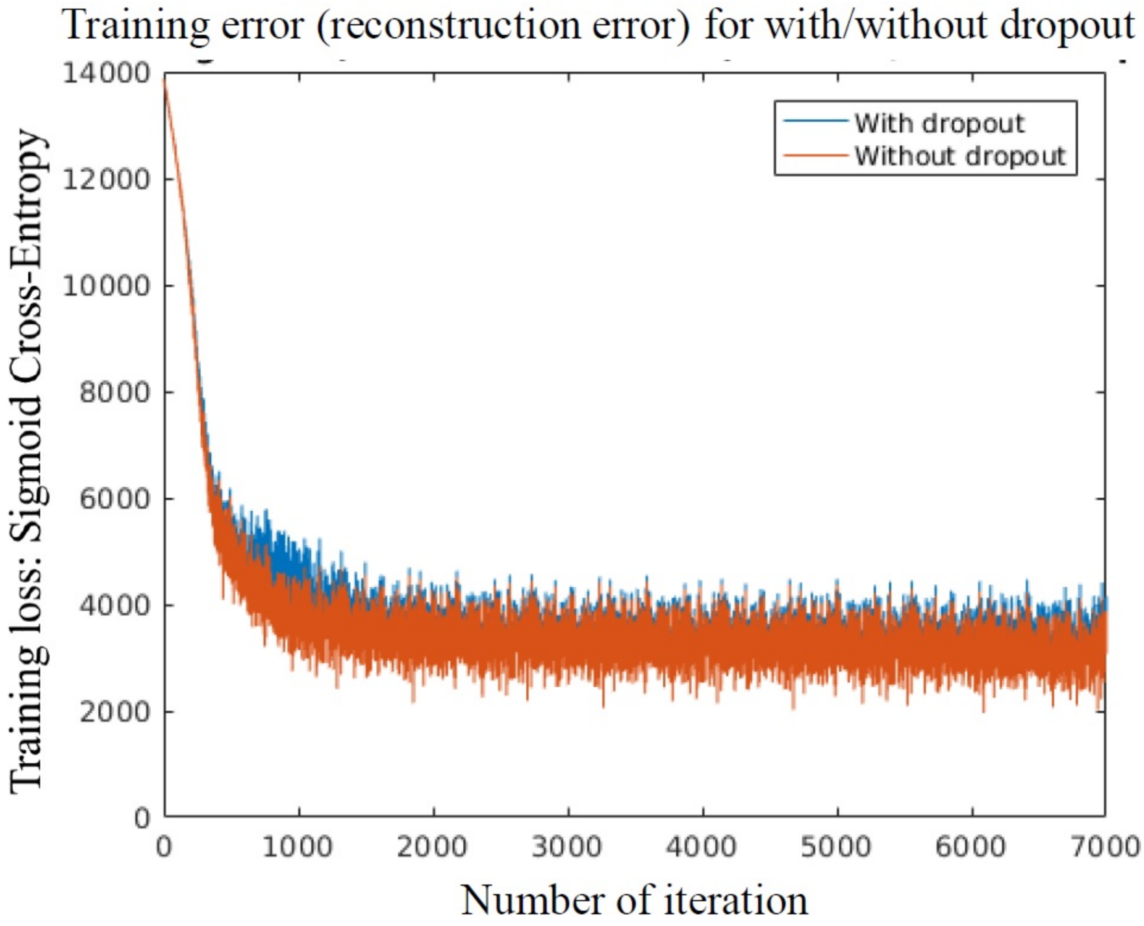


Figure 3.13: Dropout layer experiment for unsupervised autoencoder for 5 categories.

In Figure 3.13, we have shown the effect of adding a dropout layer. The experiment shows adding a dropout layer increases the error. We see that loss rate is higher in a with-dropout deep autoencoder network. We have added one dropout layer with 50% dropout ratio. Adding more than one dropout layer drops important information for retrieval. For SSDA-1 and SSDA-2 we choose the learning rate and dropout layer to be the same as SDA.

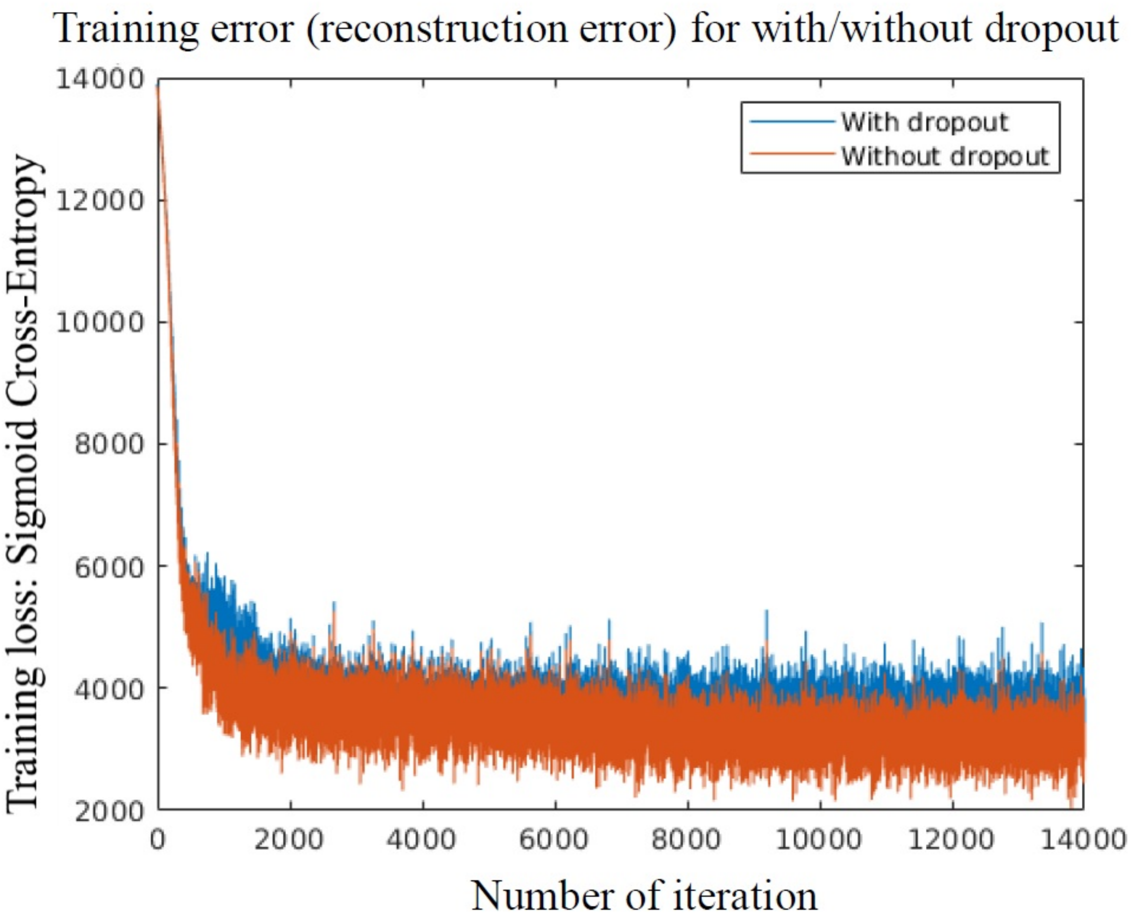


Figure 3.14: Dropout layer experiment for unsupervised autoencoder for 10 categories.

In Figure 3.14, we have shown the effect of adding a dropout layer. We see that loss rate is higher in a with-dropout deep autoencoder network. We have added one dropout layer with 50% dropout ratio. Adding more than one dropout layer drops important information for retrieval. For SSDA-1 and SSDA-2, we choose the learning rate and dropout layer to be the same as SDA.

Training unsupervised autoencoder

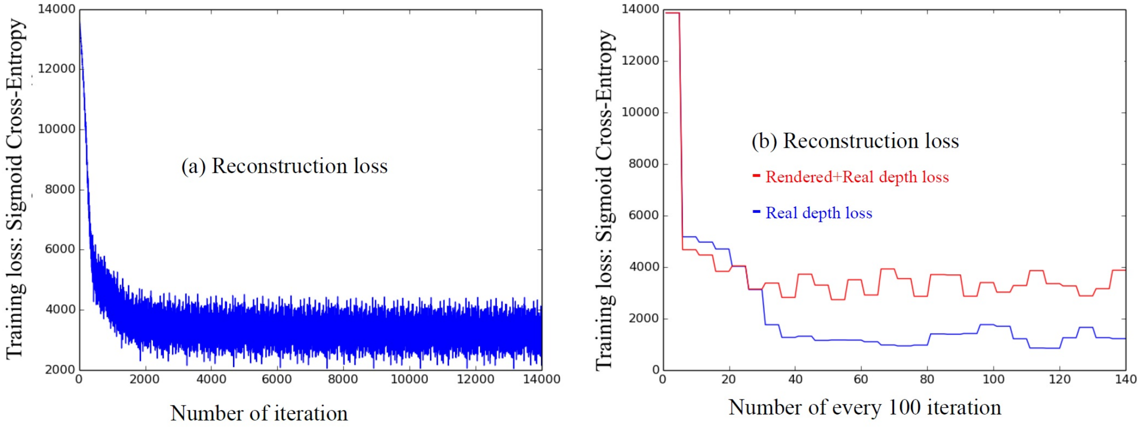


Figure 3.15: Training unsupervised autoencoder for 5 categories.

In Figure 3.15(a), we have shown the pattern of sigmoid cross entropy loss for 14000 iterations for a 5-category experiment. We see the error is decreasing which is an indication that loss function has been able to decrease error over iterations and becomes stable. In Figure 3.15(b), the red line shows the training loss for both real and rendered depth images and the blue line shows the training loss for only real depth images.

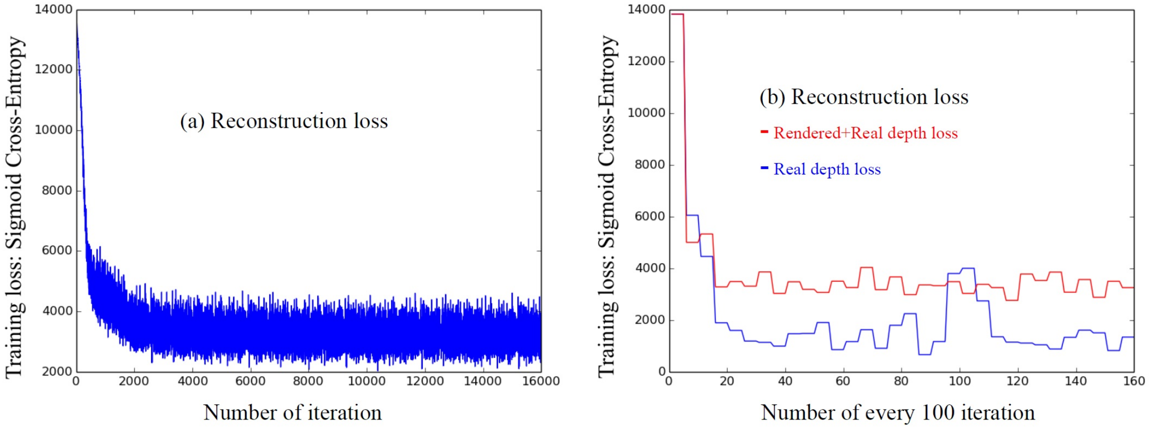


Figure 3.16: Training unsupervised autoencoder for 7 categories.

In Figure 3.16(a), we have shown the plot of sigmoid cross entropy loss for 16000 iterations for a 7-category experiment. We see the error is decreasing which is an indication that loss function has been able to decrease error over iterations and becomes stable and not decrease further. In Figure 3.16(b), the red line shows the training loss for both real and rendered depth images and the blue line shows only the training loss for real depth images.

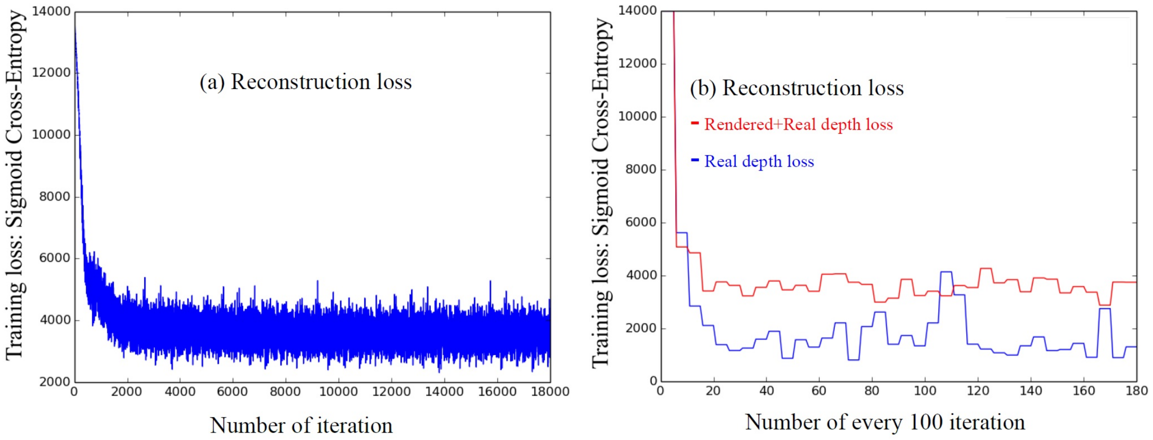


Figure 3.17: Training unsupervised autoencoder for 10 categories.

In Figure 3.17(a), we have shown the plot of sigmoid cross entropy loss for 18000 iterations for a 10-category experiment. We see the error is decreasing which is an indication that loss function has been able to decrease error over iterations and becomes stable and not decrease further. In Figure 3.17(b), the red line shows the training loss for both real and rendered depth images and the blue line only shows the training loss for real depth images.

Training supervised (SSDA-1) autoencoder

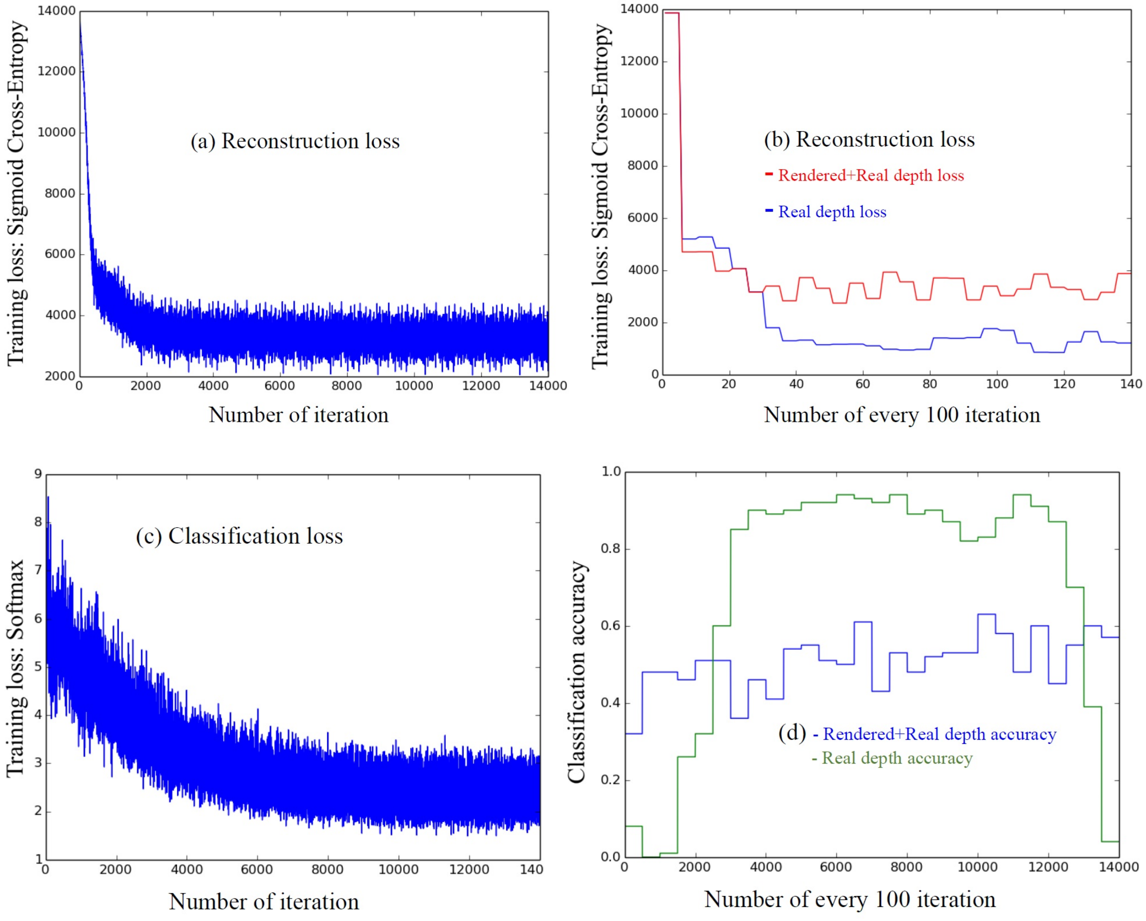


Figure 3.18: Training supervised autoencoder (SSDA-1) for 5 categories.

In Figure 3.18, we have shown the training of the supervised deep autoencoder in SSDA-1 for 5 categories. The supervised autoencoder has two loss terms, reconstruction loss and classification loss shown in Figure 3.18(a) and (c) respectively. In the training experiment we have also included classification accuracy presented in Figure 3.18(d) to see how well the features are being classified in a category. The classification accuracy is an indicator of how well the features being clustered into correct regions in order to retrieve a model from that region. In Figure 3.18(d), we observe the accuracy of real depth images separately so that we can infer on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither too high nor too low. Avoiding an iteration with overly high accuracy can avoid overfitting and avoiding an iteration with extremely low accuracy can avoid underfitting.

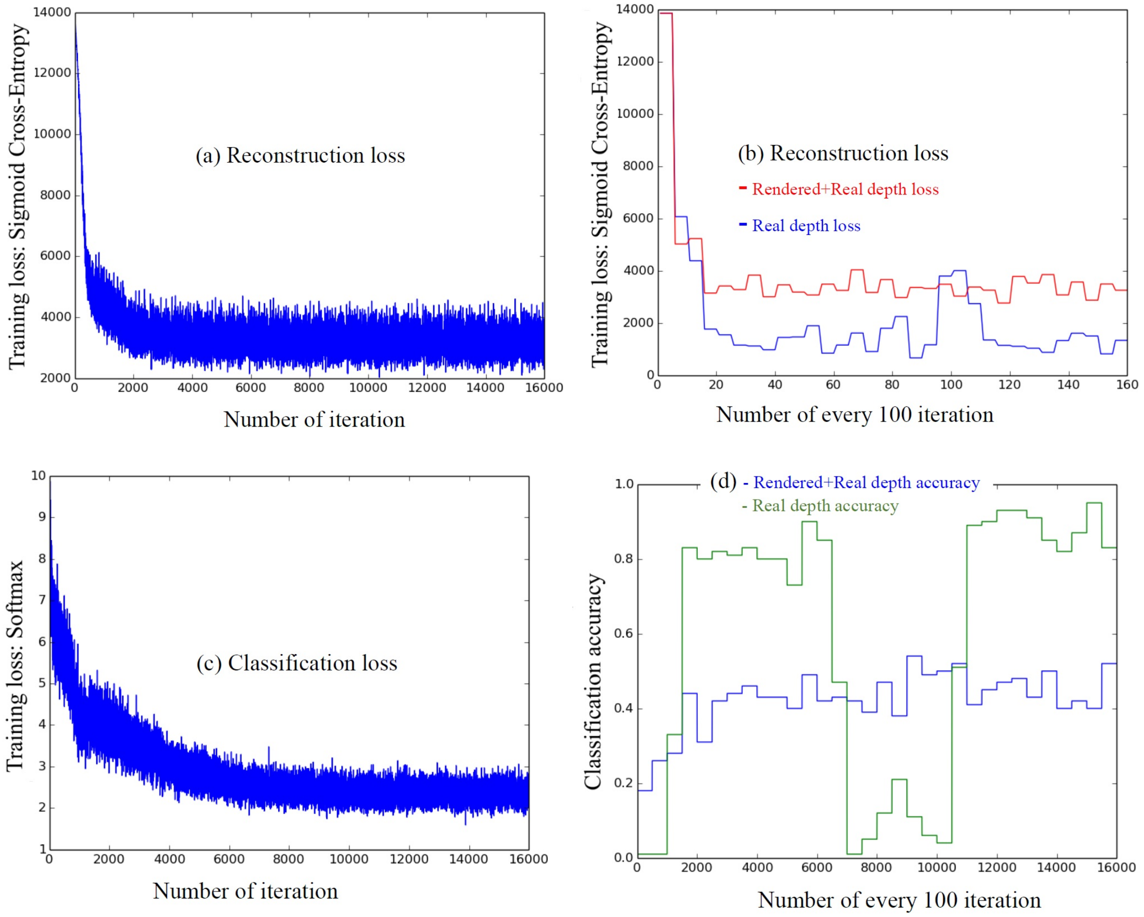


Figure 3.19: Training supervised autoencoder (SSDA-1) for 7 categories.

In Figure 3.19, we have shown the training of a supervised autoencoder for 7 categories. The supervised autoencoder has two loss terms, reconstruction loss and classification loss shown in Figure 3.19(a) and (c) respectively. In the training experiment we have also included classification accuracy presented in Figure 3.19(d) to see well the features are being classified in a category. The classification accuracy is an indicator of how well the features being clustered into correct regions in order to retrieve a model from that region. In Figure 3.19(d), we observe the accuracy of real depth image separately so that we can infer on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither to high nor too low. Avoiding an iteration with overly high accuracy can avoid overfitting and avoiding an iteration with extremely low accuracy can avoid underfitting.

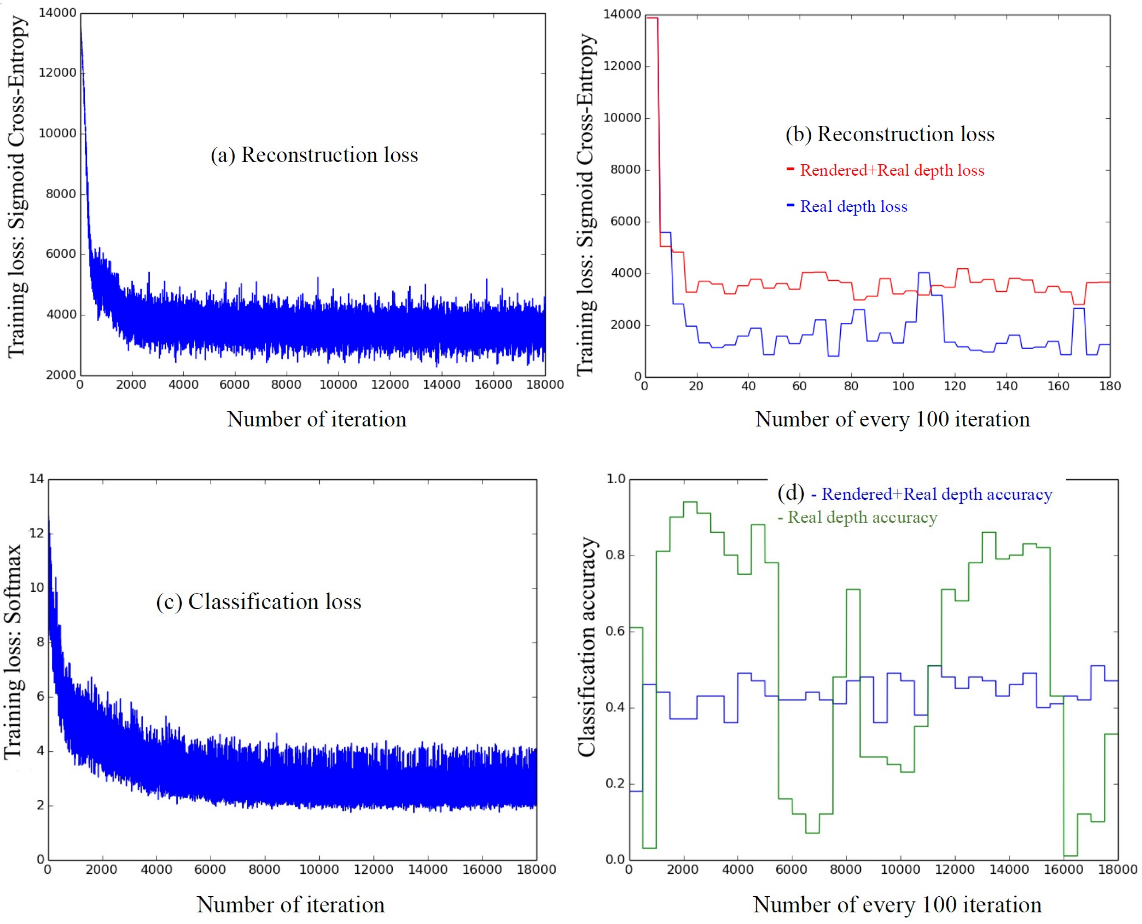


Figure 3.20: Training supervised autoencoder (SSDA-1) for 10 categories.

In Figure 3.20, we have shown the training of a supervised autoencoder for 10 categories. The supervised autoencoder has two loss terms, reconstruction loss and classification loss shown in Figure 3.20(a) and (c) respectively. In the training experiment we have also included classification accuracy presented in Figure 3.20(d) to see how well the features are being classified in a specific category. The classification accuracy is an indicator of how well the features being clustered into different correct regions in order to retrieve a model from that region. In Figure 3.20(d), we observe the accuracy of real depth image separately so that we can infer on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither to high nor too low. Avoiding an iteration with too high accuracy can avoid overfitting and avoiding an iteration with too low accuracy can avoid underfitting.

Training supervised (SSDA-2) autoencoder

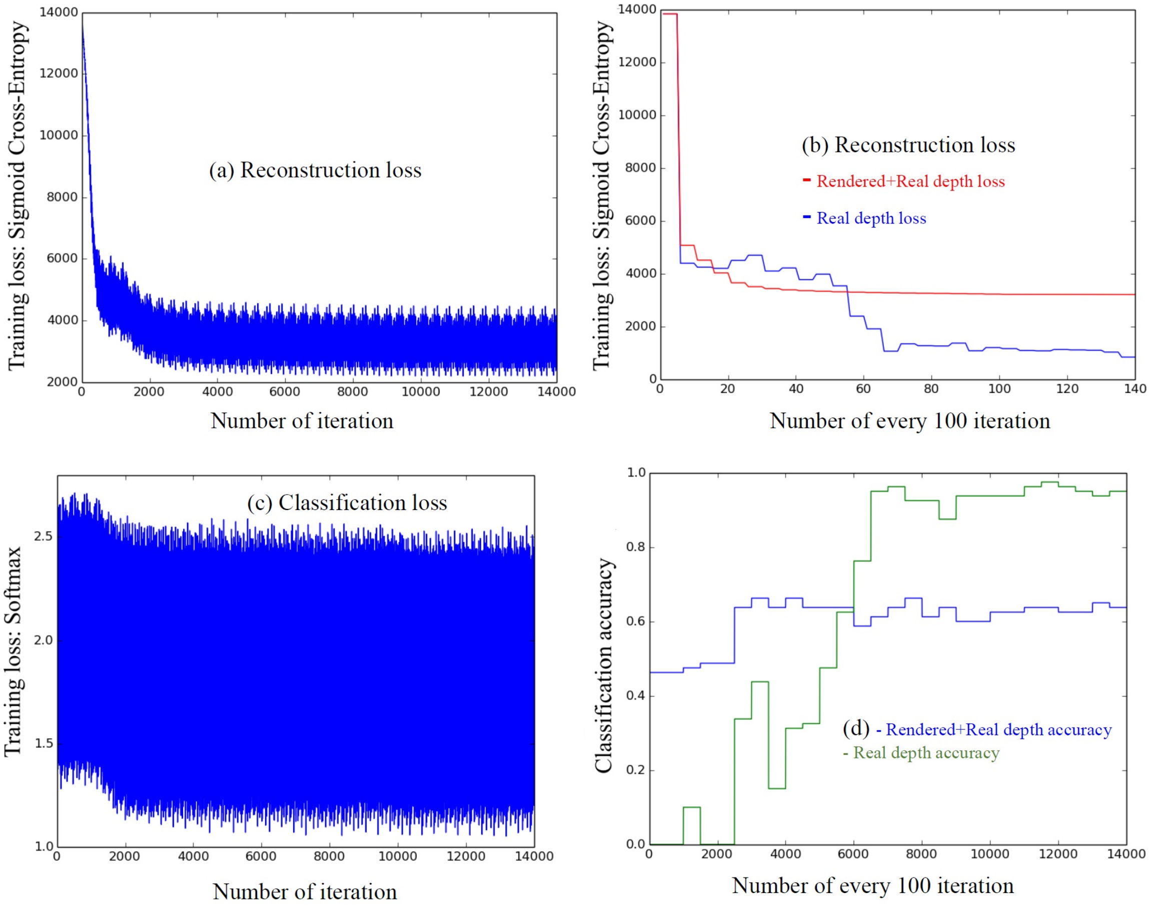


Figure 3.21: Training supervised autoencoder (SSDA-2) for 5 categories.

In Figure 3.21, we have shown the training of a supervised autoencoder in SSDA-2 for 5 categories. The supervised autoencoder has two loss terms, reconstruction loss and classification loss shown in Figure 3.21(a) and (c) respectively. In the training experiment we have also included classification accuracy presented in Figure 3.21(d) to see how well the features are being classified in a category. The classification accuracy is an indicator of how well the features being clustered into different correct regions in order to retrieve a model from that region. In Figure 3.21(d), we observe the accuracy of real depth image separately so that we can infer on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither to high nor too low. Avoiding an iteration with overly high accuracy can avoid overfitting and avoiding an iteration with extremely low accuracy can avoid underfitting.

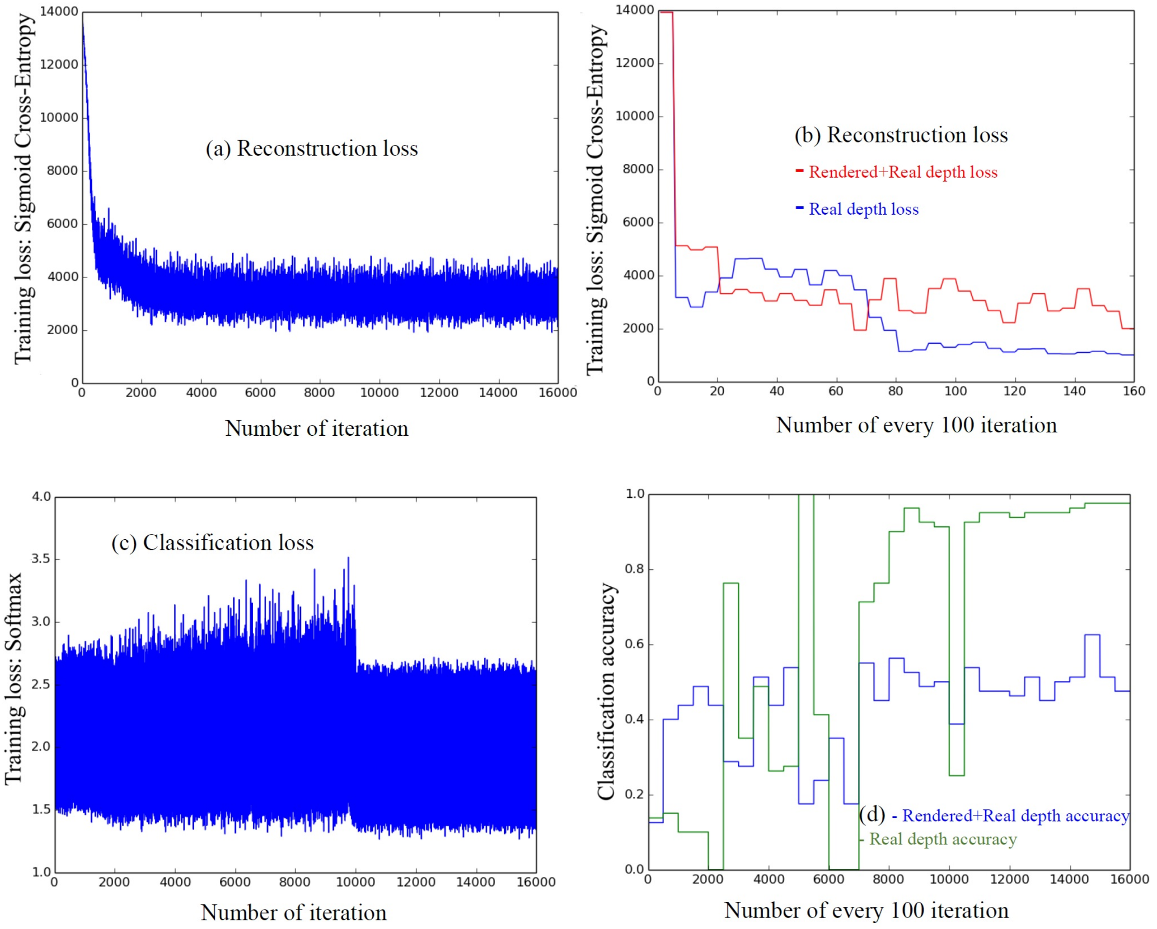


Figure 3.22: Training supervised autoencoder (SSDA-2) for 7 categories.

In Figure 3.22, we have shown the training of a supervised autoencoder for 7 categories. The supervised autoencoder has two loss terms, reconstruction loss and classification loss shown in Figure 3.22(a) and (c) respectively. In the training experiment we have also included classification accuracy presented in Figure 3.22(d) to see how the features are doing in being classified into a specific category. The classification accuracy is an indicator of how well the features being clustered into different correct regions in order to retrieve a model from that region. In Figure 3.22(d), we observe the accuracy of real depth image separately so that we can see on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither to high nor too low. Avoiding an iteration with too high accuracy can avoid overfitting and avoiding an iteration with too low accuracy can avoid underfitting.

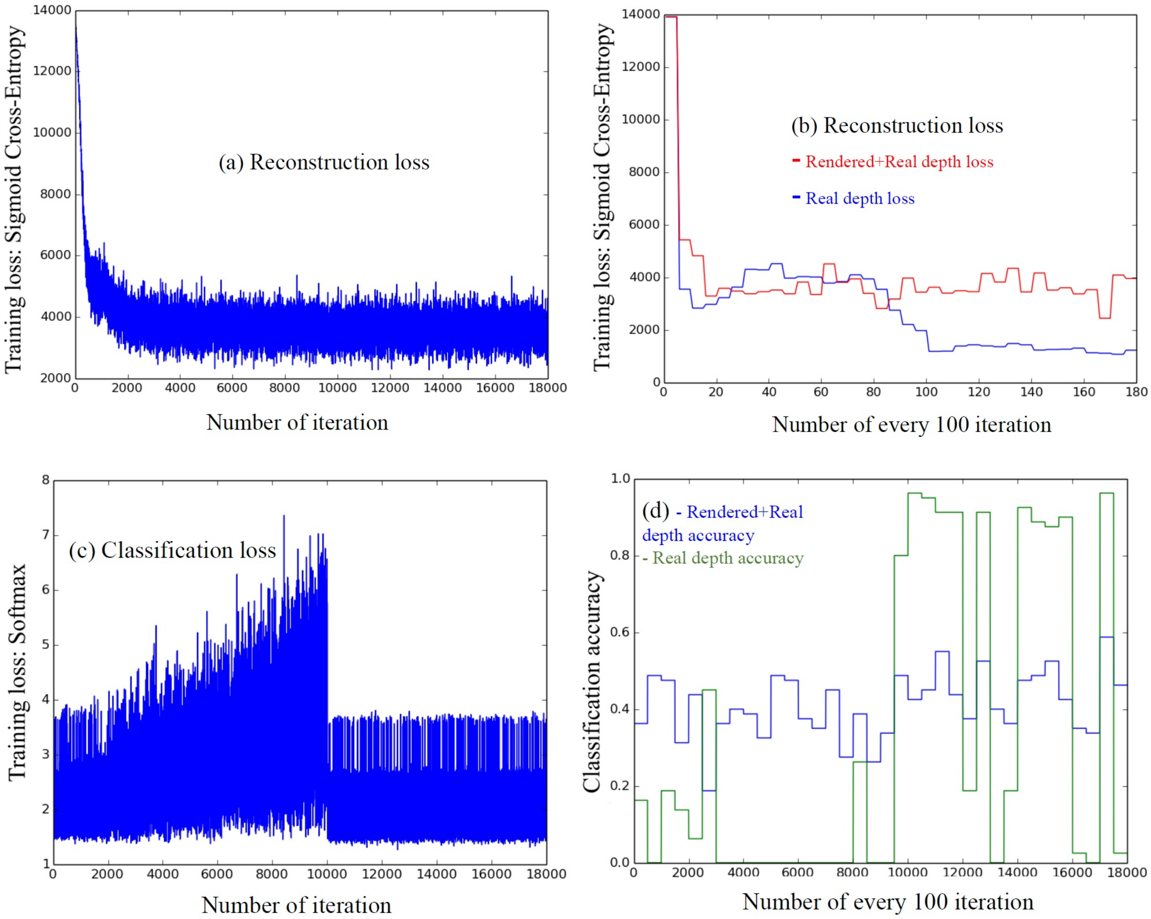


Figure 3.23: Training supervised autoencoder (SSDA-2) for 10 categories.

In Figure 3.23, we have shown the training of a supervised autoencoder for 10 categories. The supervised autoencoder has two loss terms, reconstruction loss and classification loss shown in Figure 3.23(a) and (c) respectively. In the training experiment we have also included classification accuracy presented in Figure 3.23(d) to see how well the features are being classified in a category. The classification accuracy is an indicator of how well the features being clustered into different correct regions in order to retrieve a model from that region. In Figure 3.23(d), we observe the accuracy of real depth image separately so that we can infer on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither to high nor too low. Avoiding an iteration with overly high accuracy can avoid overfitting and avoiding an iteration with extremely low accuracy can avoid underfitting.

3.5.2 Performance comparison

	NN	FT	ST	DCG	AP
		5 categories			
PNN [10]	0.1974	0.3533	0.5529	0.6754	0.3798
SDA [42]	0.5631	0.4877	**0.6963**	0.8927	0.5351
SSDA-1 [42]	0.6084	0.5250	0.6769	0.8937	0.5811
SSDA-2	**0.6160**	**0.5274**	0.6719	**0.8972**	**0.5811**
		7 categories			
PNN [10]	0.0659	0.1843	0.3400	0.5947	0.2255
SDA [42]	0.4995	0.3746	0.5586	0.8532	0.4194
SSDA-1 [42]	0.5178	0.3919	0.5683	0.8576	0.4273
SSDA-2	**0.5178**	**0.3936**	**0.5815**	**0.8583**	**0.4287**
		10 categories			
PNN [10]	0.0729	0.1115	0.2326	0.5517	0.1373
SDA [42]	0.3472	0.2429	0.4090	0.8089	0.2606
SSDA-1 [42]	0.3535	0.2557	0.4221	0.8133	0.2734
SDA-2	**0.3986**	**0.2624**	**0.4265**	**0.8172**	**0.2796**

Table 3.3: Performance metrics comparison of depth-image based 3D model retrieval on the NYU Depth V2 dataset and the ModelNet10 benchmark.

We have compared our results with the PNN approach [10] and the supervised autoencoder approach [42]. These approaches represent the most recent approaches for depth image based 3D model retrieval. The experimental results are reported in Table 3.3. The table suggests that supervised models perform better than the unsupervised model. But if the number of categories are increased, supervised models perform similar to the unsupervised ones due to ambiguity and uncertainty coming from increased number of model categories.

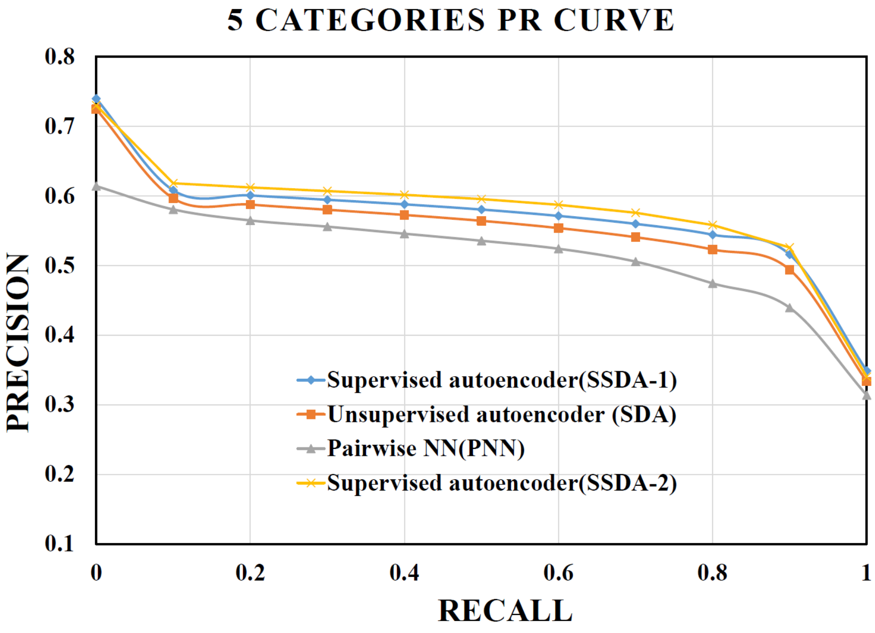


Figure 3.24: PR curve for 5 category experiment.

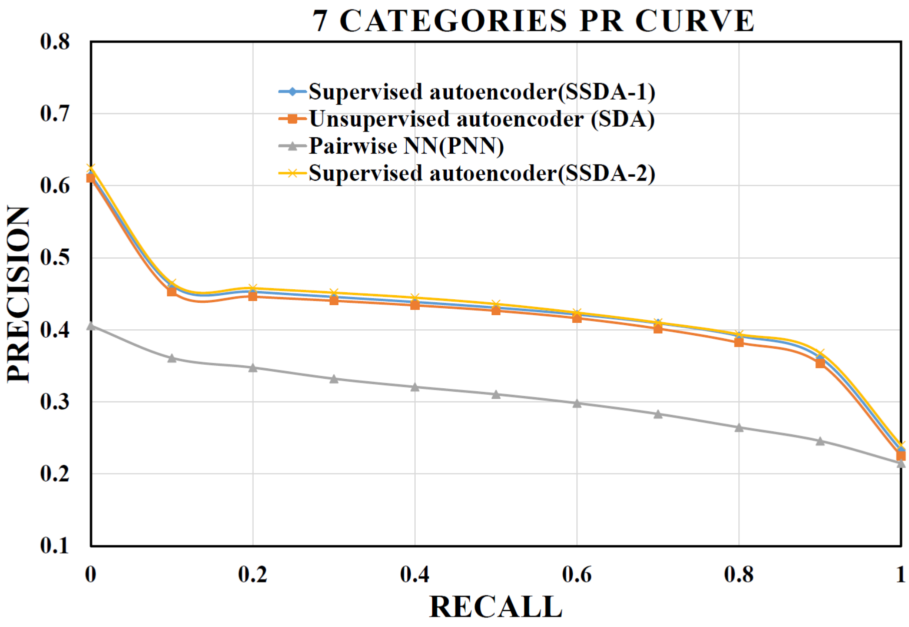


Figure 3.25: PR curve for 7 category experiment.

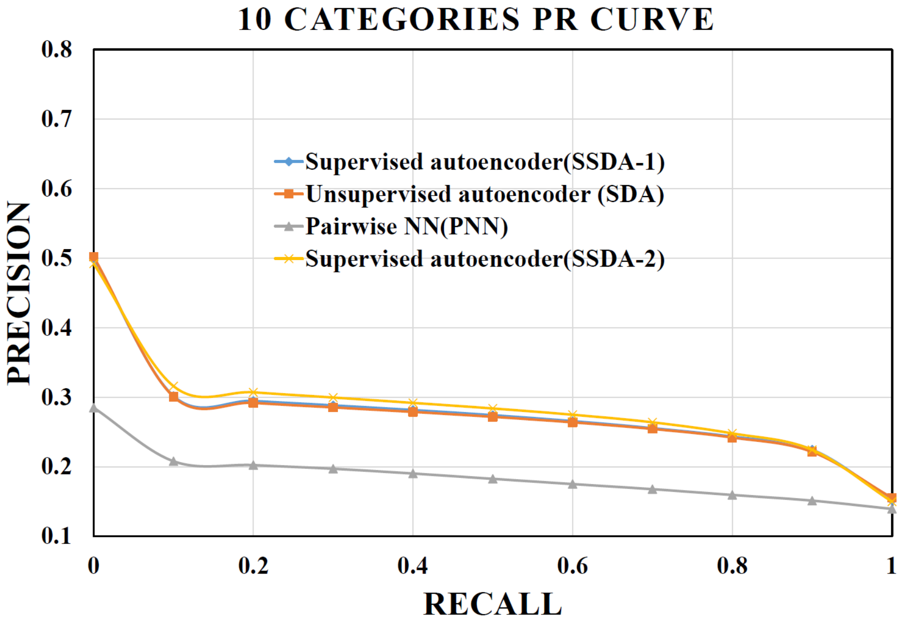


Figure 3.26: PR Curve for 10 category experiment.

PR curves in Figures 3.24 and 3.26 show that the single autoencoder PR curves are higher than PNN for the 5, 7 and 10-categories experiments. Figure 3.25 and 3.26 show that SDA and SSDA-1 performances are similar since the supervision is not effective for training almost 60,000 depth images. The supervised approach SSDA-2 performs better than the other supervised approach SSDA-1. In other words, the supervision in SSDA-2 is more effective than SSDA-1 when more ambiguity is added to the data. The training time is less in SSDA-2 is because of the lower number parameters. Also SSDA-2 retrieval accuracy is slightly better than SSDA-1 using the nearest neigbor (NN) metric. Our experimental results suggest that supervised approaches are better than the unsupervised approach and the unsupervised approach performs better than PNN. We have trained with 59382 depth images for 10 categories, 37144 depth images for 7 categories and 28240 depth images for 5 categories which is around 95 times greater than PNN.

3.5.3 Retrieval results

In this section some retrieval examples of the supervised approach(SSDA-1) and the unsupervised approach (SDA) have been shown. We have shown the first ten retrieved 3D models in the Figures 3.27 to 3.32. In those figures, the second and the third rows show the depth images retrieved by unsupervised model and the corresponding 3D models respectively; the fourth and the fifth rows show the depth images retrieved by supervised model and the corresponding 3D models respectively.

In Figure 3.27, we see that SDA has given better performance than SSDA-1 by retrieving relevant 3D models. SDA has retrieved four bathtub models among the first ten retrieved models. On the other hand, SSDA-1 has retrieved only three bathtub models. SDA has retrieved bathtub models on the third, fifth, ninth and tenth ranked positions. And SSDA-1 has retrieved bathtub models on the first, second and eighth ranked positions. One other thing, we observe from this retrieval is that the bed models retrieved by SSDA-1 have more shape similarity to the segmented bathtub

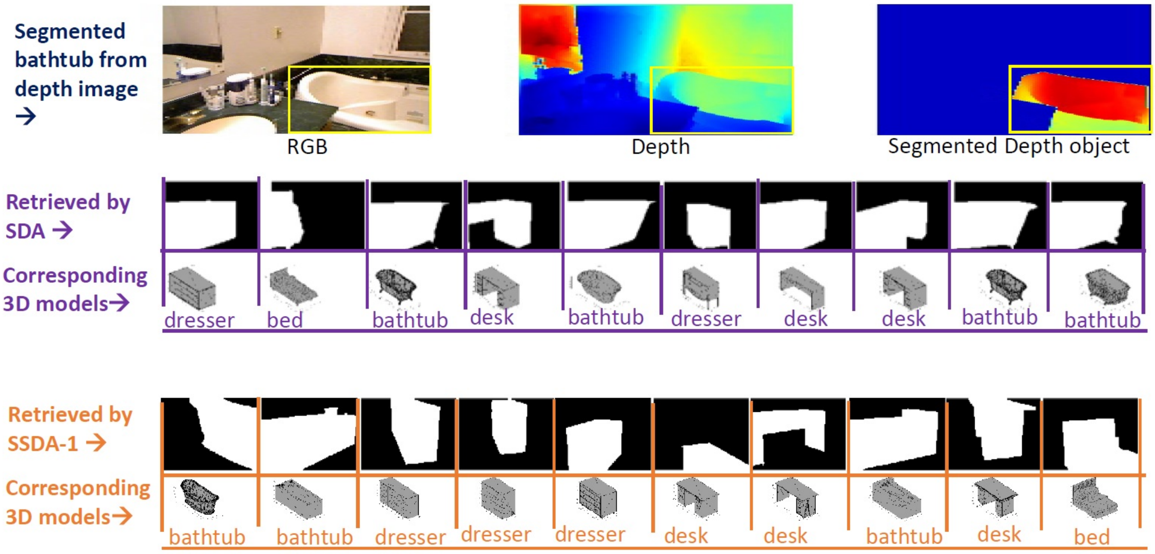


Figure 3.27: Retrieval example: bathtub.

from the depth image than SDA. We can also observe that the rank positions are higher in SSDA-1 since bathtubs are retrieved in the first two positions by SSDA-1.

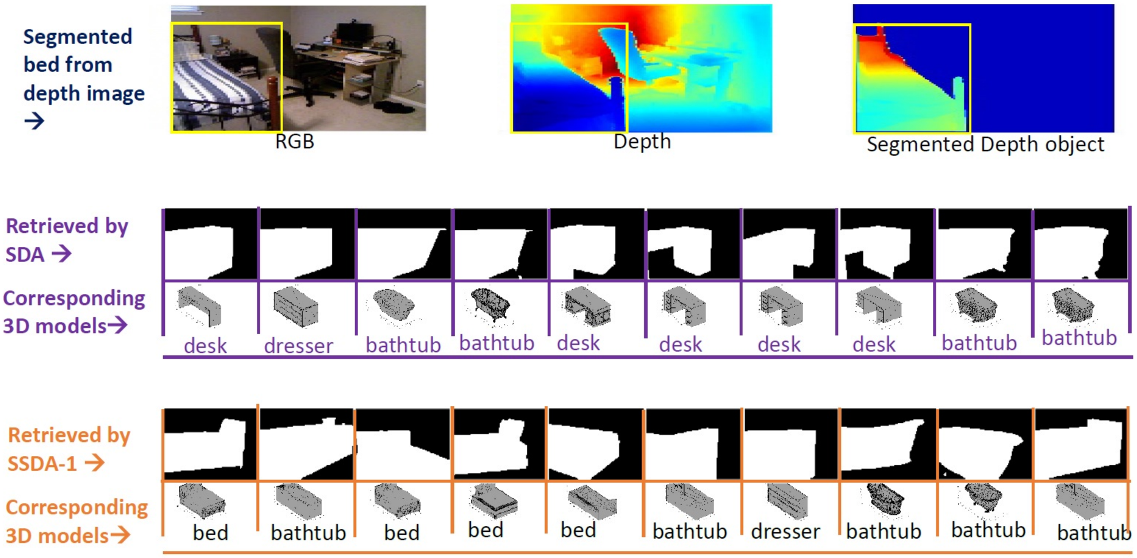


Figure 3.28: Retrieval example: bed.

In Figure 3.28, we see that SSDA-1 has given better performance than SDA by retrieving relevant 3D models. SSDA-1 has retrieved four bed models among the first ten retrieved models. On the other hand, SDA has retrieved no bed models at all among the first ten retrieved models. SSDA-1 has retrieved bed models on the first, third, fourth and fifth ranked positions. One other thing, we observe from this retrieval is that the bed models retrieved by SSDA-1 have more shape similarity to the segmented bed from the depth image than SDA.

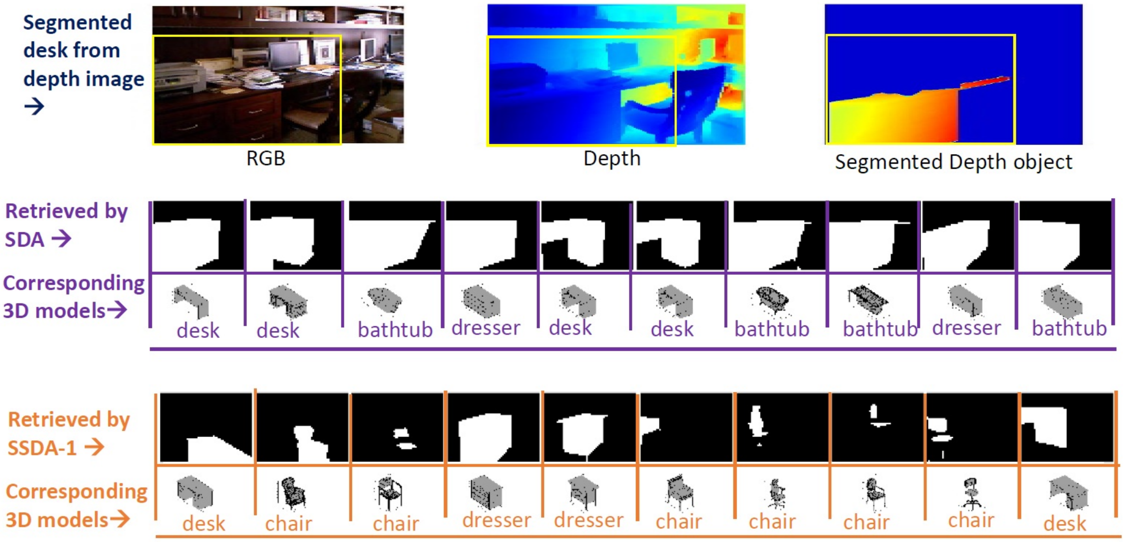


Figure 3.29: Retrieval example: desk.

In Figure 3.29, we see that SDA has given better performance than SSDA-1 by retrieving relevant 3D models. SDA has retrieved four desk models among the first ten retrieved models. On the other hand, SSDA-1 has retrieved only two desk models. SDA has retrieved desk models on the first, second, sixth and seventh ranked positions. And SSDA-1 has retrieved desk models on the first and tenth ranked positions. We observe from this retrieval is that the desk models retrieved by SDA have more shape similarity to the segmented desk from the depth image than SSDA-1.

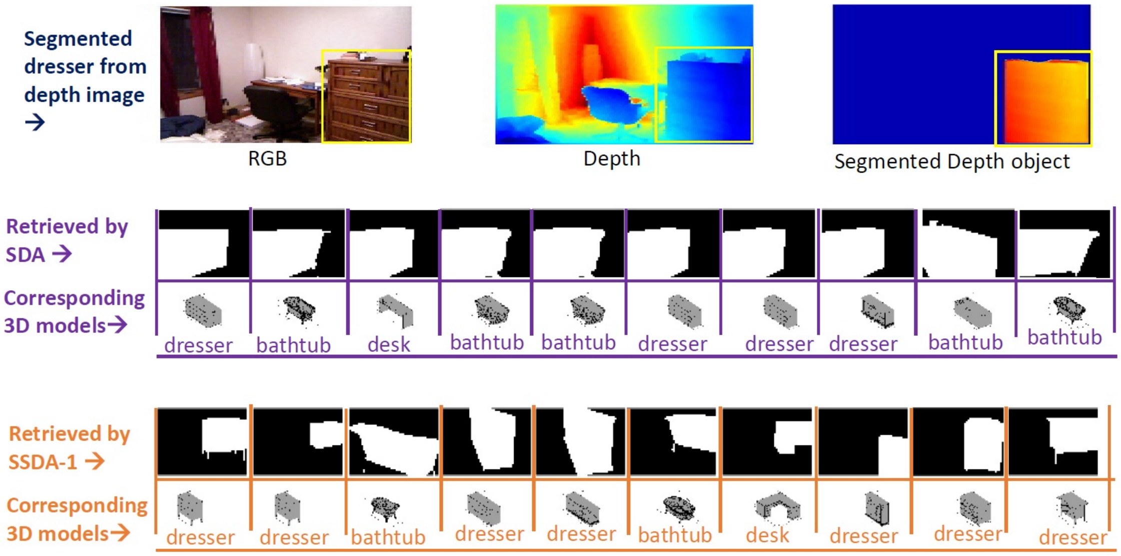


Figure 3.30: Retrieval example: dresser.

In Figure 3.30, we see that SSDA-1 has given better performance than SDA retrieving relevant 3D models. SSDA-1 has retrieved seven dresser models among the first ten retrieved models. On the other hand, SDA has retrieved only four dresser models. SSDA-1 has retrieved dresser models on the first, second, fourth, fifth, eighth, ninth and tenth ranked positions. And SDA has retrieved dresser models on the first, sixth, seventh and eighth ranked positions. One other thing, we observe from this retrieval is that the dresser models retrieved by SSDA-1 have more shape similarity to the segmented dresser from the depth image than SDA. We can also observe that the rank positions are higher in SSDA-1 since dressers are retrieved on the first two positions by SSDA-1.

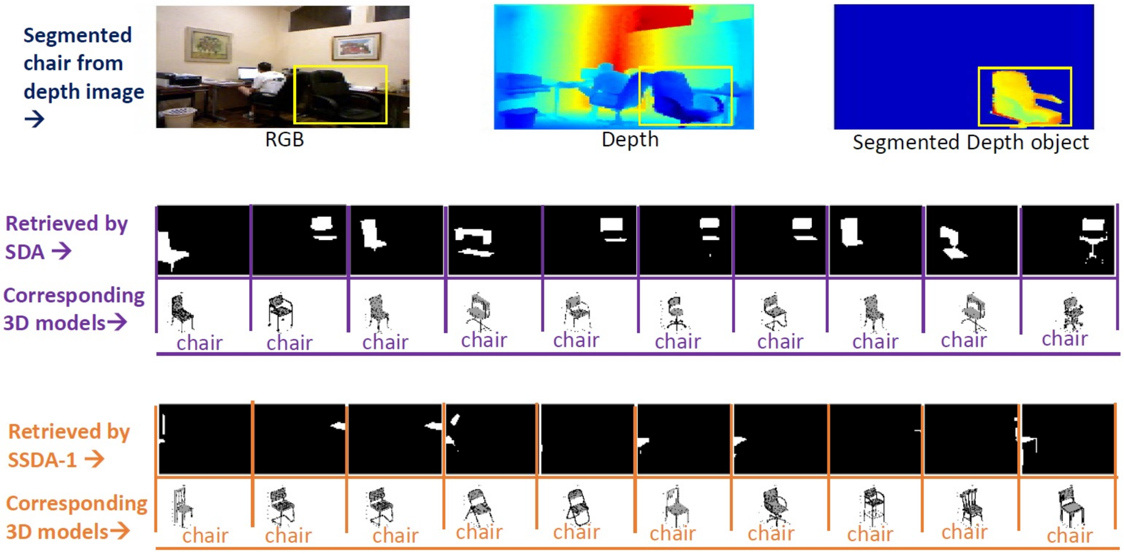


Figure 3.31: Retrieval example: chair.

In Figure 3.31, we see that both SDA and SSDA-1 has given better performance by retrieving relevant 3D models. SDA and SSDA-1 has retrieved ten chair models among the first ten retrieved models. We observe from this retrieval is that the chair models retrieved by SDA has more shape similarity to the segmented chair from the depth image than SSDA-1.

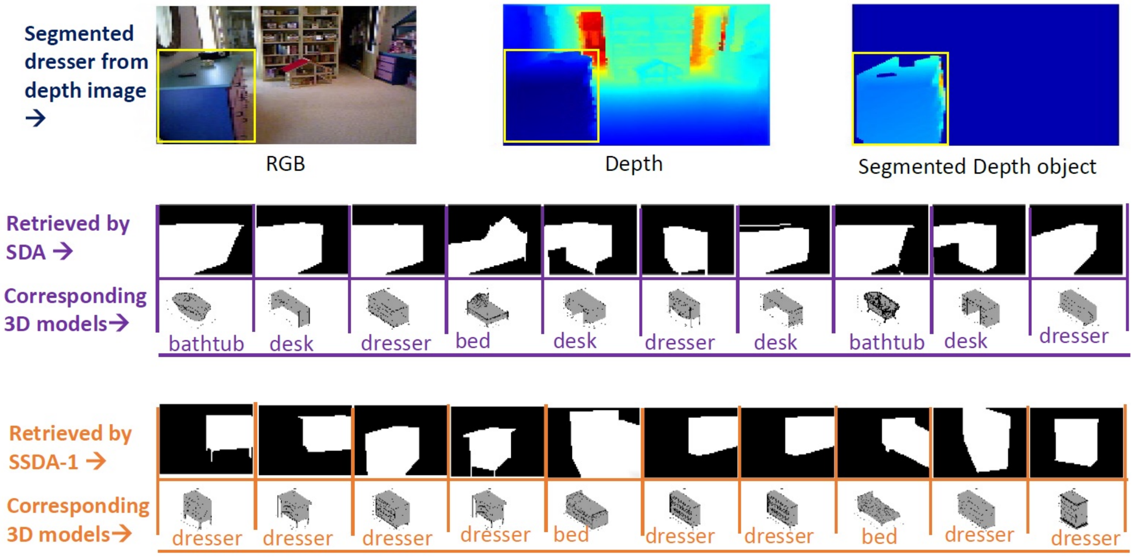


Figure 3.32: Retrieval example: dresser.

In Figure 3.32, we see that SSDA-1 has given better performance than SDA retrieving relevant 3D models. SSDA-1 has retrieved eight dresser models among the first ten retrieved models. On the other hand, SDA has retrieved only three dresser models. SSDA-1 has retrieved dresser models on the first, second, third, fourth, sixth, seventh, ninth and tenth ranked positions. And SDA has retrieved dresser models on the third, sixth and tenth ranked positions. One other thing, we observe from this retrieval is that the dresser models retrieved by SSDA-1 have more shape similarity to the segmented dresser from the depth image than SDA. We can also observe that the rank positions are higher in SSDA-1 since dressers are retrieved on the first four positions by SSDA-1.

3.5.4 Analysis and discussion

1. Computational cost

The training time is measured on a PC with 1.2GHz CPU and GTX 750 GPU. We do not preprocess features from image. The training time is proportional to the number of training iterations. We train with both NYUD2 depth images and rendered 3D model depth images in the same network. Training the supervised autoencoder in SSDA-1 takes approximately 16 minutes for 7000 iterations for 28940 depth images (5 categories) and 18.2 minutes for 7000 iterations for 59382 depth images (10 categories). Training the supervised autoencoder in SSDA-2 takes approximately 15 minutes for 7000 iterations for 28940 depth images (5 categories) and 18.1 minutes for 7000 iterations for 59382 depth images (10 categories).

2. Feature space comparison

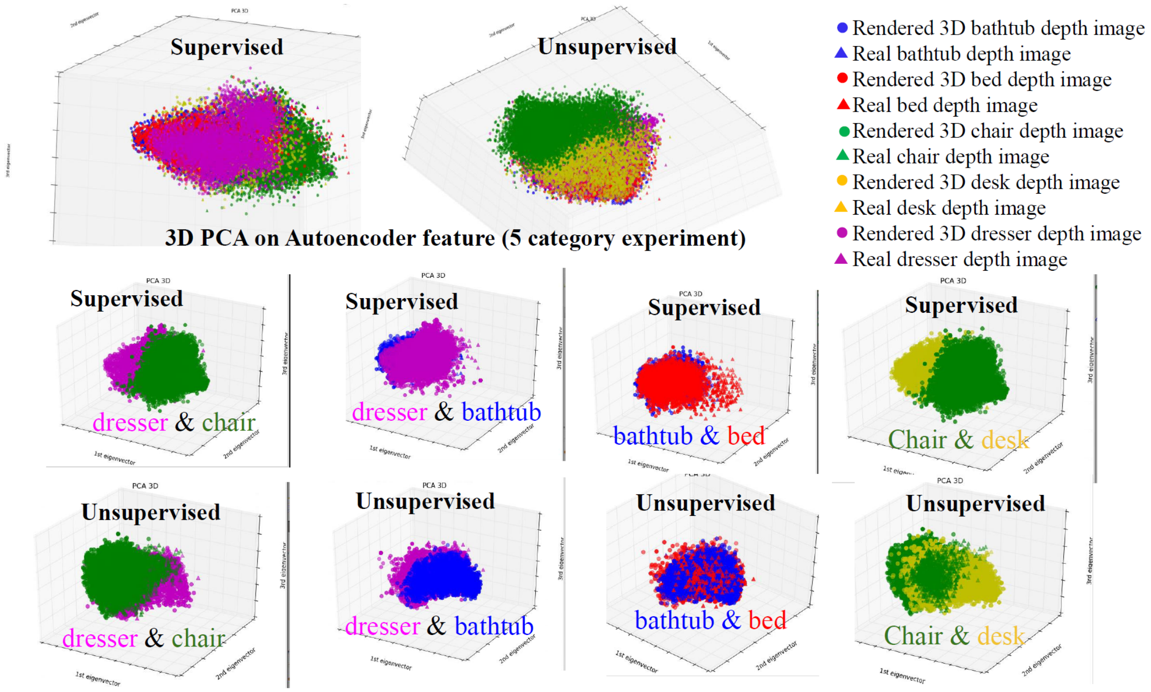


Figure 3.33: Feature space visualization: 3D PCA on 30D SDA (unsupervised) feature and 30D SSDA (supervised) feature.

We have applied 3D PCA on 30D autoencoder features for visualizing autoencoder features in a low dimensional space. In Figure 3.33, the first row shows 5 clusters for 5 objects. We see that the clusters are not well separated, the clusters are overlapped because of object similarity and incompleteness. For visualization clarity we have displayed 2 clusters at a time. In Figure 3.33, we see that dresser and chair make two different clusters with a little overlap. Bathtub and bed clusters overlap badly but SSDA-1 (supervised model) shows a little better performance than SDA. Again in Figure 3.33, we see that dresser and bathtub clusters overlap more than dresser and chair clusters. Chair and desk clusters have overlapped more in SDA than SSDA-1.

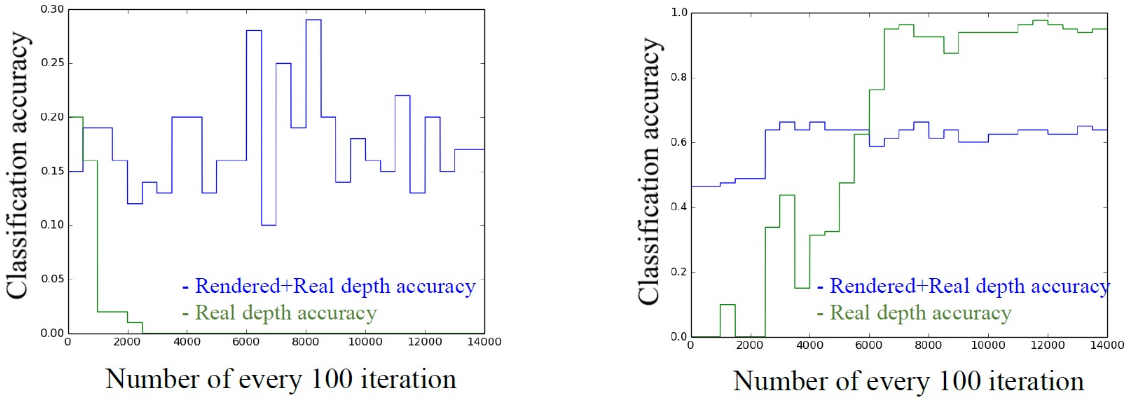
3. Advantage of supervised autoencoder over unsupervised

An autoencoder is a type of neural network that works on the data in an unsupervised manner. It can cluster the data into groups based on the data coding specially when the data is available without any labels. In our research, we deal with real depth images from indoor scenes which suffer noise, incompleteness and occlusion. Relying entirely on incomplete and occluded data to extract feature by an unsupervised autoencoder leads to incorrect grouping or improper features. Moreover we train our autoencoder with mix-domain data. Which means we train our autoencoder with real and rendered depth images together. An autoencoder working on mixed data has a tendency to group real and rendered depth images with the same label in different clusters which is undesirable for effective retrieval. In our proposed models we supervise an autoencoder with label information. For a cross-modal retrieval approach using a single supervised autoencoder trained by mixed domains(real and rendered depths) can favor the retrieval in two ways. First, it provides insight whether two different domains with same category are being clustered in the same group or not. And second, it yields the opportunity to visualize the training process for determining a stopping criteria. Determining a stopping criteria means deciding an iteration having a good real depth image accuracy. These two advantages have been demonstrated with examples and experiments in the following two subsections.

In a nutshell, our proposed supervised autoencoder depends mostly on the data where we rely on reconstruction and reconstruction aware clasification to deal with ambiguity and complexity present in real and rendered depth images..

4. Determining stopping criteria from supervised autoencoder

Training a supervised autoencoder gives us the facility to see how real training depth features or reconstructions are being classified into appropriate categories. Since we can see the real depth accuracy, we can determine a stopping criteria. We can determine a point from the training where real depth images of the training data are being classified with a proper accuracy. This is not possible for unsupervised approach since we do not classify the features so we cannot determine a good stopping criteria with good accuracy of real depth. This concept is further demonstrated in Figure 3.34.



(a) Feature classification accuracy for deep unsupervised autoencoder (SDA)

(b) Feature classification accuracy for deep supervised autoencoder (SSDA-2)

Figure 3.34: Effect of supervision in cross modal retrieval: determining stopping criteria for autoencoder training based on an adequate accuracy. From supervised training presented in (b), we can choose trained network parameters at a certain iteration point (such as 3000 or 6000) based on real training depth accuracy.

In the training experiment we have included classification accuracy presented in Figure 3.34 to see how the features are being classified into a specific category. The classification accuracy is an indicator of how well the features are being clustered into different regions in order to retrieve a model from that region.

In Figure 3.34(b), we observe the accuracy of real depth image separately so that we can see on which iteration the real training depth images are being classified with an adequate accuracy. This is an advantage because we can choose an appropriate iteration where the accuracy is neither too high nor too low. Avoiding an iteration with overly high accuracy can avoid overfitting and avoiding an iteration with extremely low accuracy can avoid underfitting. In Figure 3.34(a) we observe that the unsupervised deep autoencoder is not suitable to pick a good training point. In Figure 3.34(b), we show that we can choose an iteration of 3000 or 6000 as a stopping criteria since real training depth images are being classified with an adequate accuracy and in this way we can also avoid overfitting to the training data.

5. SSDA-2 vs SSDA-1

We have proposed one unsupervised deep autoencoder model SDA and two single supervised deep autoencoder models SSDA-1 and SSDA-2. Unlike SSDA-1, the supervised autoencoder in our SSDA-2 model is not symmetric since it has three encoder layers and four decoder layers. The differences are listed in the Table 3.4. In addition SSDA-2, instead of adding a classification network with the encoder we have added a classification network after the decoder to supervise the autoencoder with label information. This supervision indicates how well the reconstruction is being classified into different categories. After the fourth decoder layer we have added two innerproduct layers to backpropagate the classfication error. Therefore, the network has a total of ten inner product layers and a dropout layer after third encoder layer to prevent overfitting.

Categories	**SSDA-1**	**SSDA-2**
1. Symmetry	Symmetric	Non-symmetric
2. Classification layer	Classifies encoder features	Classifies decoder reconstruction
3. Network Architecture	1000-500-250-30- 250-500-1000- 19200-1000-500-10	1000-500-30- 250-500-1000- 19200-1000-500-10
4. Parameters	24240	23990
5. Training time for 7000 iteration (28940 depth images for 5-category experiments)	16 minutes	15 minutes
6. Training time for 7000 iterations (59382 depth images for 10-category experiments)	18.2 minutes	18.1 minutes
7. Total layers without dropout layer	10	9
8. 5-category retrieval result	NN-0.6084	NN-0.6160
10. 10-category retrieval result	NN-0.3535	NN-0.3986

Table 3.4: SSDA-1 vs SSDA-2.

To portray the basic difference between SSDA-1 and SSDA-2 in detail we can say that the classification network is tightly tied with reconstruction in SSDA-2. In other words, the relationship between reconstruction and classification is straightforward. In case of SSDA-1, we classify the features instead of the reconstruction of the features, which makes the relationship between classification and reconstruction tricky or indirect.

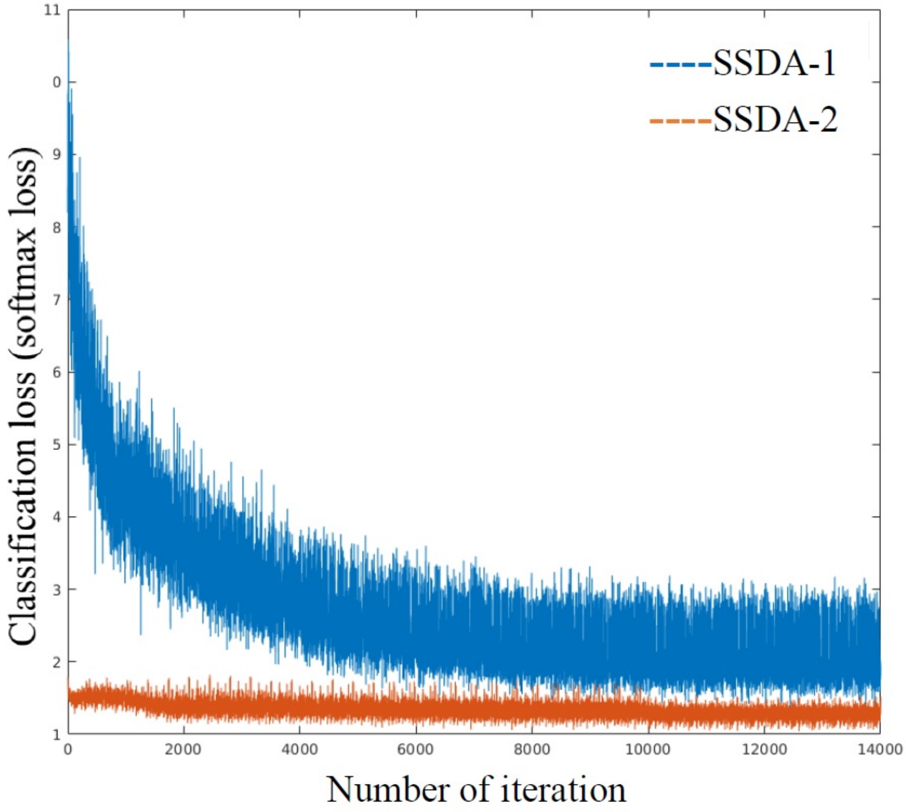


Figure 3.35: Classification loss (Softmax loss), SSDA-1 vs SSDA-2.

In case of SSDA-1, classification is directly based on the features without involving the reconstruction results, while SSDA-2 is proposed to allow the classification network to be more closely associated with the reconstruction. In other words, classification in SSDA-2 is based on the reconstruction results of autoencoder features. Compared with SSDA-1, the relationship between reconstruction and classification is more balanced and integrated in SSDA-2 for joint optimization. SSDA-2 has two advantages over SSDA-1: one is less training time and the other is improved retrieval accuracy. The training time is less in SSDA-2 because of less parameters. Also the classification loss is less in SSDA-2 than SSDA-1 shown in Figure 3.35. The cost function of SSDA-2 is the same as that in SSDA-1 defined in equation 3.3 and optimized using the same optimizing function.

6. Pairwise vs non-pairwise models

Usually cross-modal retrieval methods involve training two different domains with a parallel or paired network structure [7, 9–11]. The assumption of training two different supervised networks facilitating the transfer matching between two different domains leads to a conjecture. We address the pairwise models for cross-domain retrieval as a conjecture because this is an assumption or speculation without enough experimental proof or validation. In this research we show that training one single deep autoencoder with rendered and real depth images together is more efficient to bring cross domains in single feature space. We also show that a supervised deep autoencoder is a better approach for cross modal retrieval than an unsupervised autoencoder. Our model is not restricted to an equal number of depth images and 3D models. On the contrary, in [12], correlation modeling (CM) and semantic correlation modeling (SCM) algorithms are restricted to have the same number of 3D models and depth images due to the dimension requirement of canonical correlation analysis (CCA). Our model takes depth images as the direct input rather than handcrafted features. The deep structure of an autoencoder can effectively learn complicated structures from raw depth images and requires less domain knowledge.

5-category experiment	Testing data retrieval accuracy (NN)
Training: real and rendered depth (SSDA-1) [42]	0.6084
Training: real and rendered depth (SSDA-2)	0.6160
Training: real and rendered depth(#630) (SSDA-2)	0.6030
Training: rendered depth only (SSDA-2)	0.5976
7-category experiment	NN
Training: real and rendered depth (SSDA-1) [42]	0.5178
Training: real and rendered depth (SSDA-2)	0.5178
Training: real and rendered depth(#712) (SSDA-2)	0.4785
Training: rendered depth only (SSDA-2)	0.4822
10-category experiment	NN
Training: real and rendered depth (SSDA-1) [42]	0.3535
Training: real and rendered depth (SSDA-2)	0.3986
Training: real and rendered depth(#954) (SSDA-2)	0.3132
Training: rendered depth only (SSDA-2)	0.1611

Table 3.5: Performance comparison between training with and without real depth images. Nearest neighbor metric comparison of depth-image based 3D model retrieval on the NYU Depth V2 dataset and the ModelNet10 becnhmark.

The experimental results in Table 3.5 supports our assumption of single network retrieval strategy. Introducing real depth images in the training data is an efficient way to have the same features for same classes for two different domains such as real depth image domain and rendered depth image domain. Figure 4.1 will help to get the idea of this assumption. Table 3.5 indicates that training rendered and real depth images together shapes the feature in a way that bridges the gap between the real data and rendered data. The retrieval result using the nearest neighbor (NN) metric with real depth data shows better performance than without real depth data. This better result is also an additional advantage besides the advantage of less training time. Training a single network is obviously less expensive than dual network. The metric nearest neighbor is an indicator of whether the model is able to pick a model with correct label in the first ranked position.

In this work, we have proposed an unsupervised and a novel supervised model for cross-modal 3D model retrieval using a deep autoencoder from depth images. We have shown the strength of supervision on autoencoder by doing classification aware reconstruction. We have studied the retrieval of ten indoor objects (bathtub, bed, chair, desk, dresser, night stand, table, toilet, monitor and sofa). The synthetic and real depth images are used together to train our models whereas all the existing 3D model retrieval approaches use a pairwise network or a separate network to train each domain. We demonstrate that training together increases the possibility to bring two different domains in a single feature space. We have shown the effectiveness of our models on NYUD2 depth image dataset and ModelNet10.

CHAPTER IV

ASYMMETRIC SUPERVISED DEEP AUTOENCODERS

4.1 Background

In a deep autoencoder the features of data such as edges, contour and patterns are learned layer by layer . The deeper layers learn more complex and high-order features. Also by altering the hidden layers of a deep autoencoder with different number of neurons, sparsity, weight decay or corrupted input might reveal interesting structures present in the data. Again a supervised deep autoencoder can restructure features by providing label information when the data are incomplete or occluded [42]. Supervised and semi-supervised autoencoders have been used in face recognition, speech recognition, sentiment analysis, text classification, phone classification, multi-task learning and many other areas [1–8].

Having an equal number of encoders and decoders before and after the central embedding layer in a deep autoencoder is not necessarily required, rather it relates more to greedy training of stacked autoencoders since the encoders and decoders used to be built by nesting one into the other making the symmetric structure inevitable. Researchers have shown that an asymmetric structure leads to more robust latent space representation and invariant to small changes in samples [33–36].We have proposed a unique supervised deep asymmetric autoencoder for 3D model retrieval based on depth images for the first time. We demonstrate the fact that it might be prudent to develop a less complex decoder than the encoder to bridge the domain gap in a single embedding. Our work also shows that asymmetric deep autoencoder for the 3D model retrieval problem outperforms symmetric deep autoencoder by overcoming negative

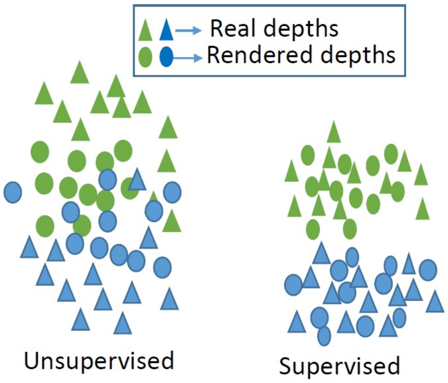


Figure 4.1: Effect of supervision: supervised autoencoder groups real and rendered depths of same category in the same region.

impacts of symmetrical architecture which are not useful for transfer learning.

4.2 Single supervised deep asymmetric autoencoder(SSDAA)

4.2.1 Supervised structure

Autoencoder is a widely used unsupervised data driven approach to find a latent lower dimensional space of data. But relying entirely on occluded and incomplete data leads to incorrect grouping or improper features. A supervised autoencoder with label information tends to correctly restructure the features into correct groups. In a cross-modal retrieval approach using an asymmetric *single* supervised autoencoder trained by mixed datasets (real and rendered depths) can be advantageous in two ways. First it provides insight as to whether two different domains with the same category are being flocked in the same group or not. Second it allows us to determine a stopping criteria where the real depth images are being classified with good accuracy. To reveal the usefulness of supervised learning over unsupervised for retrieval, we show the example of Figure 4.1. The effect of supervision in cross modal retrieval could be stated as: real and rendered depths tend to cluster separately in the unsupervised approach whereas supervised autoencoder groups real and rendered depths of same category in the same region.

We have proposed a supervised single deep asymmetric autoencoder approach for retrieval SSDAA. In the case of SSDA-1 [42], classification is directly based on the features without involving the reconstruction results, while SSDAA is proposed to allow the classification network to be more closely associated with the reconstruction. In SSDAA, instead of adding a classification network with the encoder, we have added a classification network after the decoder to supervise the autoencoder with label information. The proposed SSDAA is not symmetric since it has less decoder layers than encoder layers. SSDAA has two advantages over SSDA-1 [42], less training time and improved retrieval accuracy.

4.2.2 Asymmetric structure

This chapter introduces asymmetric supervised deep autoencoder for 3D model retrieval purpose based on depth images. Resesarches have shown that asymmetric deep autoencoders lead to more robust embedding and invariant to small variation of samples [34, 35]. All the previous research in the area of 3D model retrieval employed symmetric deep autoencoders that had an equal number of encoders and decoders. We have compared asymmetric autoencoder retrieval results with symmetric autoencoder and found that asymmetric deep autoencoder overcoming the symmetric deep autoencder for improving retrieval accuracy depicted in the result section. Our supervised single deep asymmetric autoencoder has three encoders and two decoders. Reducing the number of decoder layers reduces the number of parameters (network weights) to learn [33] which eventually leads to reduced over-fitting. Improving generalizability and classification accuracy are also the outcome of reduced number of parameters. One more advantage of the asymmetry is that each decoder layer is subjected to add reconstruction loss to the network. So reducing the number of decoder layers helps to minimize the loss which can be easily verified by comparing reconstruction error between symmetric and asymmetric autoencoder [33].

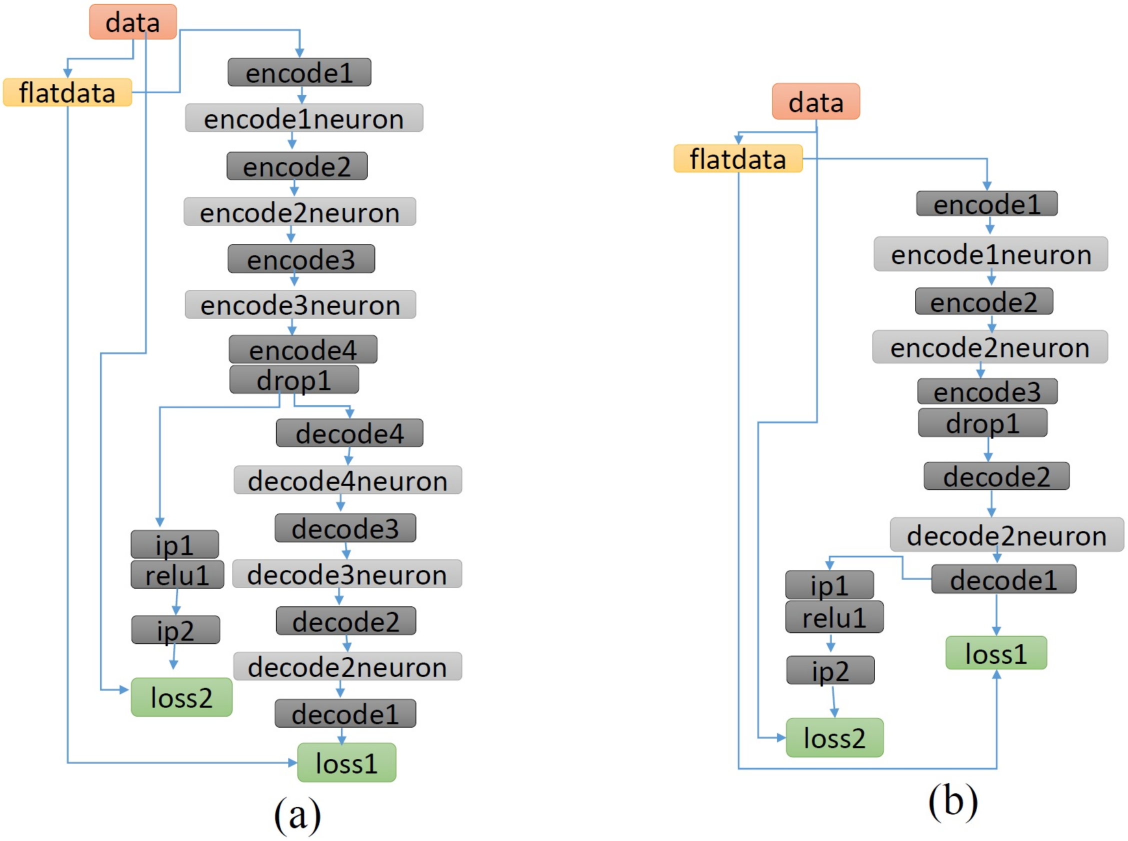


Figure 4.2: Deep network architecture, (a) Supervised symmetric autoencoder SSDA-1 [42], (b)Supervised asymmetric autoencoder SSDAA network: enocode1(1000), encode2(500), encode3(30), decode2(500), decode1(19200), ip1(500) and ip2(5/7/10).

4.2.3 Network structure and loss function

The proposed SSDAA has a classification network to supervise the autoencoder with label information. The network architecture of these supervised autoencoders has been given in Figure 4.2. The first part of the loss function coming from the unsupervised branch of the network is given below:

$$E_{uasym} = -\frac{1}{N}\sum_{n=1}^{N}[t_n \log \hat{p}_n + (1-t_n)\log(1-\hat{p}_n)], \tag{4.1}$$

where the prediction $\hat{p}_n$ is the sigmoid function applied at the last decoder inner-product layer that has the same dimension as the input, t_n is the original input, and N is the number of training samples. The reconstruction error E_{uasym} is the sigmoid cross entropy loss function which corresponds to loss1 in Figure 4.2(b).

The total objective function we propose for single suprevised deep asymmetric autoencoder model SSDAA is a unique combination of sigmoid cross entropy loss and softmaxwithloss for the reconstruction and the classification respectively and is defined as below:

$$E_{Sasym} = \alpha E_{uasym} + \beta E_{casym}, \tag{4.2}$$

where the reconstruction error term E_{uasym} is the sigmoid cross entropy loss function from the autoencoder and corresponds to loss1 in Figure 4.2(b) and is defined in (4.1). The hyper-parameters in (4.2), α and β, control the trade-off between the two costs. The classification loss term E_{casym} is the softmax loss function from classifier and corresponds to loss2 in Figure 4.2(b) and is defined as below:

$$E_{casym} = -\frac{1}{N}\sum_{n=1}^{N} \log(\hat{p}_{n,l_n}), \tag{4.3}$$

where $\hat{p}_{n,l_n}$ is the softmax output of the classifier defined in equation (3.7). In $\hat{p}_{n,l_n}$ the subscripts n, l_n indicate the nth training sample and the category label respectively. N is the number of training samples. E_{casym} is the mean over all training examples and is basically multinomial logistic regression used for predicting a single class of K mutually exclusive classes. After training the network using AdaDelta [38], we got 30D features (encoder output) $\mathbf{M}_{as}$ for rendered 3D model depth images and 30D $\mathbf{D}_{as}$ for test depth images. The distance between a model M and a query depth image D is the distance between the autoencoder feature M_{as} for the model and the autoencoder feature D_{qas} for the depth and defined as below:

$$Dist_{ssdaa}(M, D) = d(M_{as}, D_{qas}), \tag{4.4}$$

where $d(.)$ is the distance between M_{as} and D_{qas} where 'as' means asymmetric supervised autoencoder. We refer to this type of retrieval as retrieval by SSDAA.

4.3 Experiments

4.3.1 Experimental setup

Datasets

We have used ground truth segmentation to create multiple depth images each of which has one a specific object from the NYUD2 data set [13]. Totally, we created 3347 depth images that are split into 1907 for training and 1440 for testing. The depth images containing any one of the ten indoor objects (bathtub, bed, chair, desk, dresser, nightstand, table, toilet, monitor and sofa) are considered as our candidate image. From the ModelNet10 benchmark [14], we obtained 605 3D models belonging to the ten categories, and generated around 57475 rendered depth images as suggested in sliding shapes [39]. An example of rendered synthetic images under different views is given in Figure 3.10. We have chosen those views based on the object statistics and observation of training images in NYUD2 [13].

Evaluation criteria

Our evaluation of retrieval performance includes 5 well recognized retrieval metrics (NN, FT, ST, DCG and mAP) having a maximum of 100% and a 11-point precision-recall curve. Nearest Neighbor (NN) is the average percentage of the closest K matches that belong to the same category as the depth image query. First tier (FT) and Second tier (ST) are the percentages of the closest K matches that belong to the category of the depth image query. K depends on how many models (M) are present in the database for a particular category. In our experiments, $K = 1$ for NN, $K = M - 1$ for FT and $K = 2 * (M - 1)$ for ST. Discounted Cumulative Gain (DCG) is a metric which progressively reduces the importance of the object retrieved as the rank goes higher. Mean Average Precision (mAP) uses ranked list and changes its value at a rank where the object is relevant [40].

4.3.2 Performance comparison

	NN	FT	ST	DCG	AP
		5 categories			
PNN [10]	0.1974	0.3533	0.5529	0.6754	0.3798
SDA [42]	0.5631	0.4877	0.6963	0.8927	0.5351
SSDA-1 [42]	0.6084	0.5250	0.6769	0.8937	0.5811
SSDAA	0.**6343**	**0.5387**	**0.6798**	**0.8967**	**0.5960**
		7 categories			
PNN [10]	0.0659	0.1843	0.3400	0.5947	0.2255
SDA [42]	0.4995	0.3746	0.5586	0.8532	0.4194
SSDA-1 [42]	0.5178	0.3919	0.5683	0.8576	0.4273
SSDAA	**0.5453**	**0.4035**	**0.5729**	**0.8601**	**0.4411**
		10 categories			
PNN [10]	0.0729	0.1115	0.2326	0.5517	0.1373
SDA [42]	0.3472	0.2429	0.4090	0.8089	0.2606
SSDA-1 [42]	0.3535	0.2557	0.4221	0.8133	0.2734
SSDAA	**0.3618**	**0.2759**	**0.4526**	**0.8192**	**0.2998**

Table 4.1: Performance metrics comparison of depth-image based 3D model retrieval on the NYU Depth V2 dataset and the ModelNet10 benchmark.

The retrieval results using asymmetric autoencoder approach SSDAA is presented in Table 4.1 using five well known metrics. We have compared our results with the PNN approach [10] and the unsupervised and supervised autoencoder approaches [42] referred to as SDA and SSDA-1 respectively.

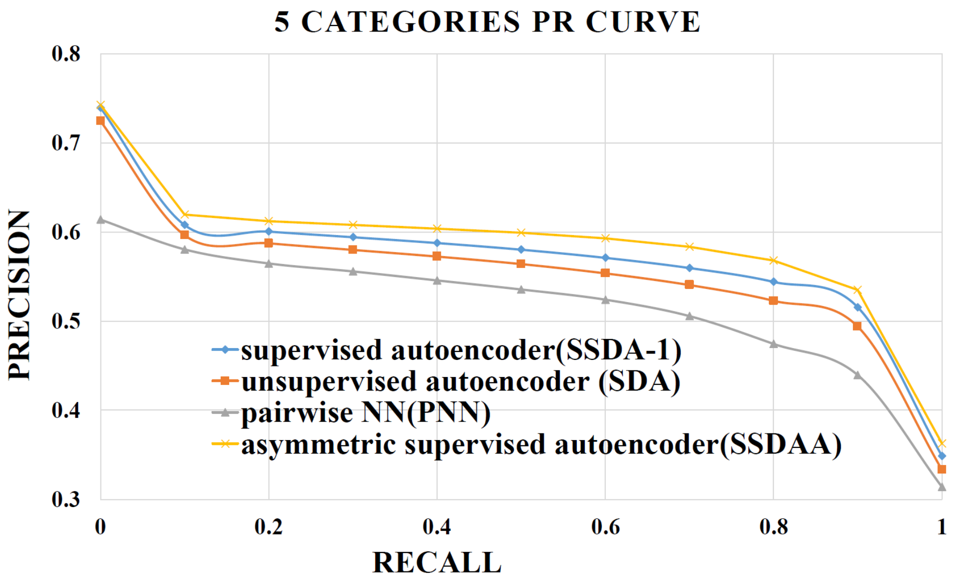


Figure 4.3: PR curve for 5-category experiment.

Precision-recall (PR) curve performance comparison for 3D model retrieval

We have reported the comparison using PR-curves in Figures 4.3, 4.4 and 4.5 which show the results of 5, 7 and 10-category experiments respectively. The 10-category PR-curve shown in Figure 4.5 suggests that SDA and SSDA-1 are similar in performance, SSDAA performs the best, which means the supervision in SSDAA is more effective than that in SSDA to deal with more categories. This is mainly owing to the reconstruction-aware classification involved in training. Moreover, the training time is less in SSDAA because of less parameters. We have trained 59382 training depth images for 10 categories, 37144 depth images for 7 categories and 28240 depth images for 5 categories which are around 95 times greater than the depth images used in PNN that also involves LD-sift features extracted from 3D models for training.

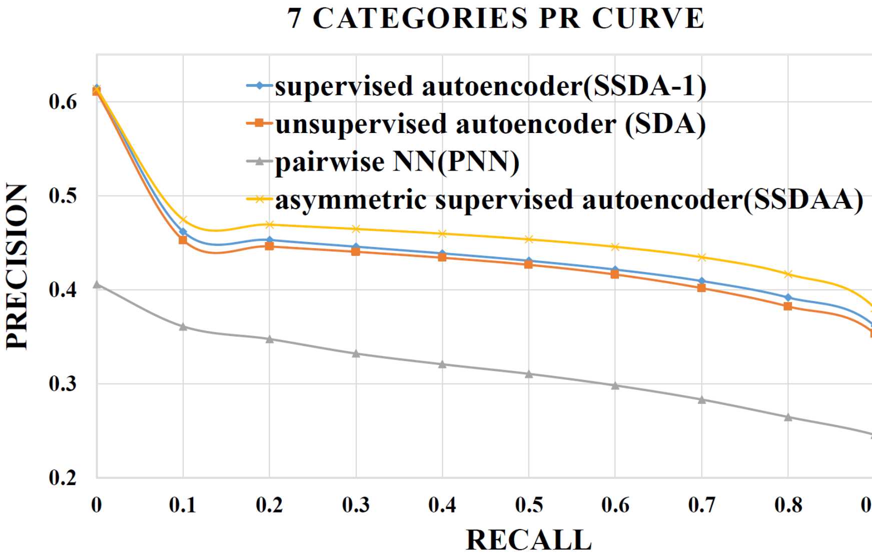


Figure 4.4: PR curve for 7-category experiment.

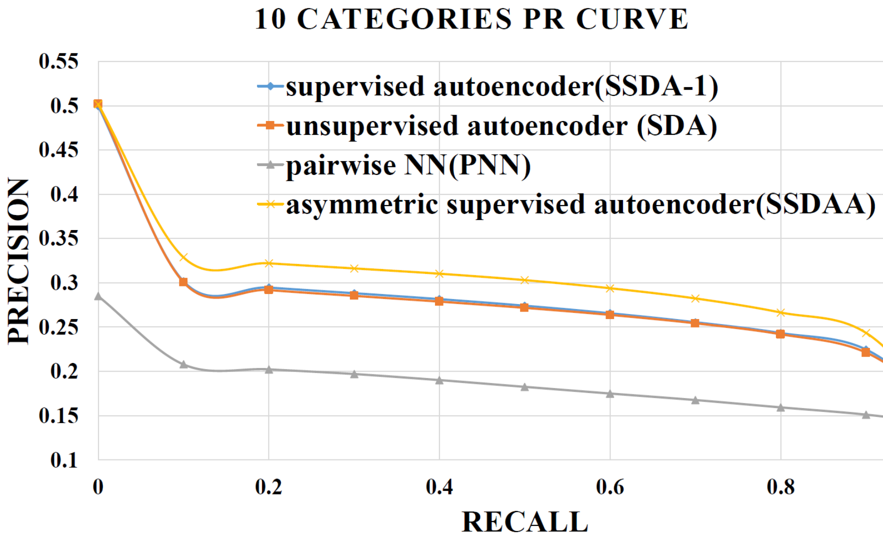


Figure 4.5: PR curve for 10-category experiment.

4.3.3 Retrieval results

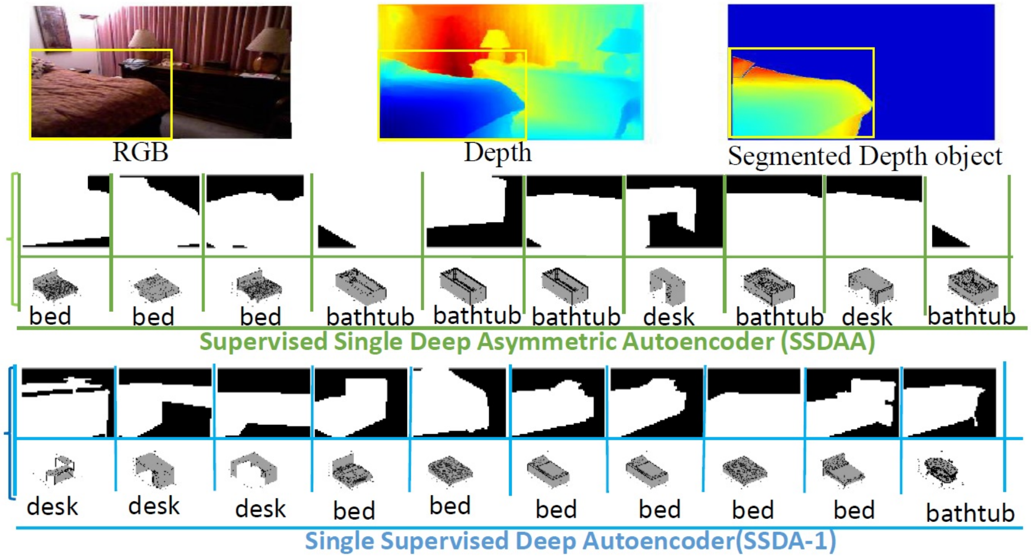


Figure 4.6: Retrieval example: bed.

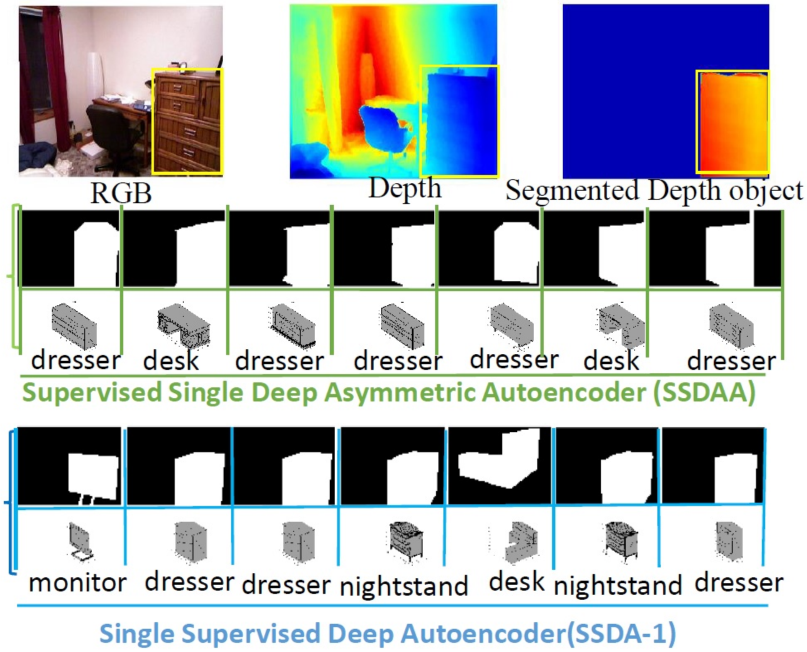


Figure 4.7: Retrieval example: dresser.

In Figure 4.6 shows that asymmetric approach SSDAA has retrieved three bed models in the first, second and third positions. Though symmetric approach SSDA-1 has retrieved six bed models in the first ten retrieved models yet asymmetric approach SSDAA has retrieved the beds models in higher ranks than SSDA-1. Again in Figure 4.7 asymmetric approach SSDAA has retrieved five dresser models whereas symmetric approach has retrieved only three dresser models in the first seven retrieved models.

4.3.4 Analysis and discussion

1. Computational cost

The training time is measured on a PC with 1.2GHz CPU and GTX 750 GPU. Training SSDAA takes approximately 15 minutes for 7000 iterations for 28940 depth images (5 categories) and 18.1 minutes for 7000 iterations on 59382 depth images (10 categories). We added one dropout layer with 50% dropout ratio to reduce overfitting. For the 5, 7, and 10-category experiment we use $\alpha = 1$ and $\beta = 1$.

2. Relation between encoder-decoder layers

In this section we have analyzed relation between the number of encoder-decoder layers. We have compared different number of encoder and decoder combinations for 3D model retrieval. The symmetric-asymmetric autoencoder comparisons for 5, 7, and 10 category experiments have been discussed in this section.

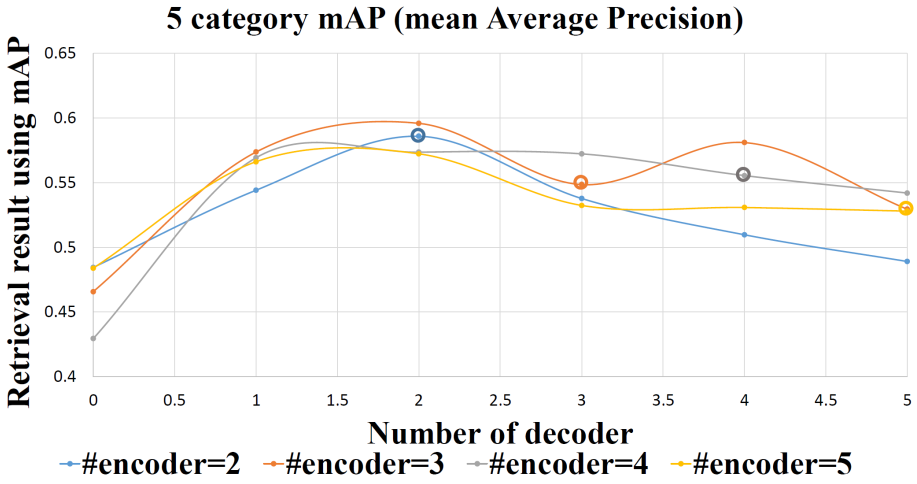


Figure 4.8: Retrieval results in mAP for different number of encoder-decoder autoencoder structure.

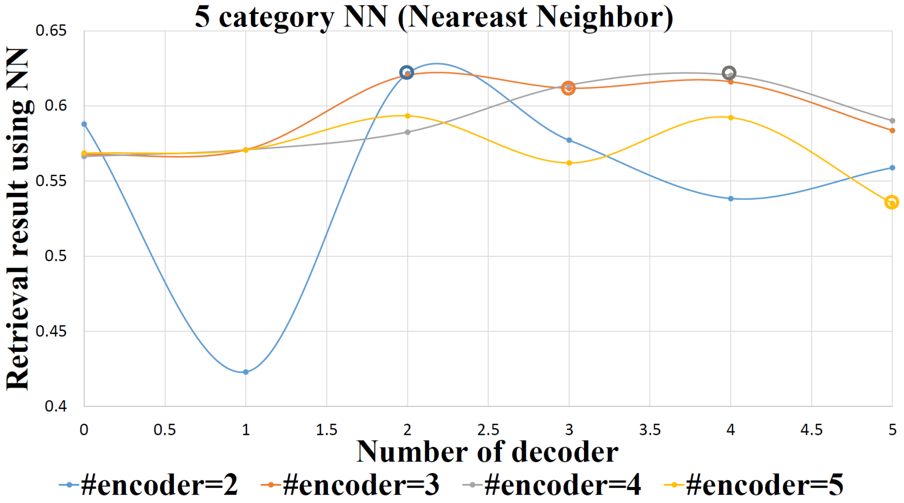


Figure 4.9: Retrieval results in NN for different number of encoder-decoder autoencoder structure.

In Figures 4.8 and 4.9, if we observe the symmetric encoder-decoder combinations such as 2-2(blue point), 3-3 (orange point), 4-4 (ash point), and 5-5 (yellow point), none of those is the highest point. In other words, none of the symmetric autoencoder structures indicates the best retrieval result using the metrics mAP and NN. Rather having two decoders and three encoders gives the best retrieval results using mAP and NN. This scenario indicates that having an asymmetric autoencoder with less decoder layers than encoder layers is a good option for cross modal retrieval.

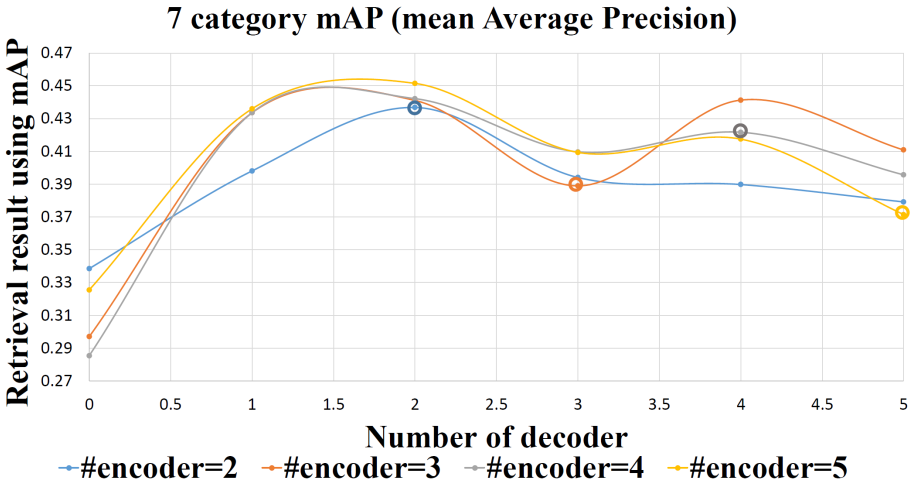


Figure 4.10: Retrieval results in mAP for different number of encoder-decoder autoencoder structure.

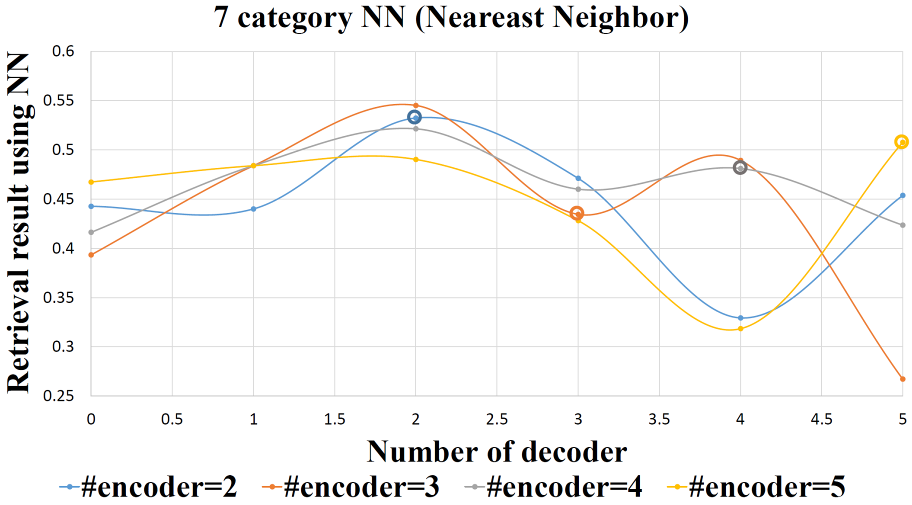


Figure 4.11: Retrieval results in NN for different number of encoder-decoder autoencoder structure.

For the 7-category experiments shown in Figures 4.10 and 4.11, we again see that none of the symmetric encoder-decoder combinations such as 2-2(blue point), 3-3 (orange point), 4-4 (ash point), and 5-5 (yellow point) is the highest point. The points mentioned are symmetric structures and do not have the best retrieval using the metrics mAP and NN. A supervised autoencoder structure having two decoders and three encoders gives the best retrieval results using mAP and NN. This scenario also is an indicator to asymmetric structure being a good option for cross modal retrieval.

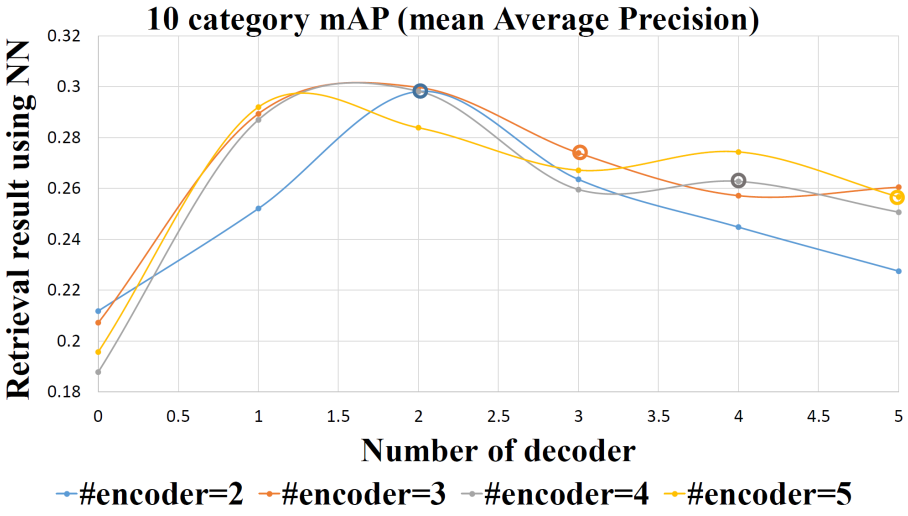


Figure 4.12: Retrieval results in mAP for different number of encoder-decoder autoencoder structure.

We see the same scenario for 10-category experiments shown in Figures 4.12 and 4.13 as we saw for 5 and 7-category experiments. None of the symmetric autoencoder structure has the highest retrieval result. An autoencoder having two decoders and three encoders gives the best retrieval result using mAP. In the case of NN having four encoders and two decoders gives the best result which indicates that having less decoder layers than encoder is the best asymmetric structure for our cross modal retrieval. One other observation is very clear from all the 5,7 and 10 category experiments that having an autoencoder with no decoder (encoder as classifier only) layer is not capable of better cross modal retrieval.

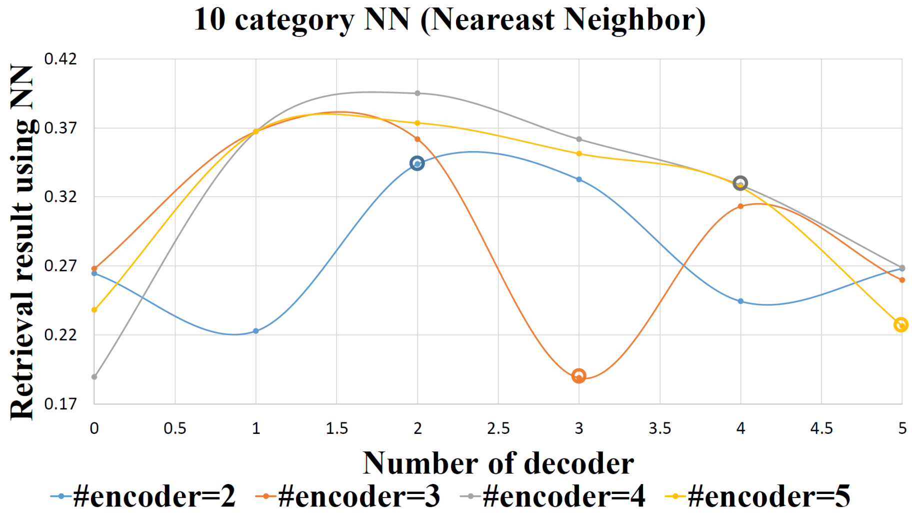


Figure 4.13: Retrieval results in NN for different number of encoder-decoder autoencoder structure.

In this chapter, we have proposed a new asymmetric supervised deep autoencoder by investigating the relation between the number of encoders and decoders. We claim that asymmetric structure of a supervised deep autoencoder learns more robust and effective unified embedding to bridge the gap between cross-domains if those different domains are trained together. We have compared supervised asymmetric and supervised asymmetric structure of a deep autoencoder for depth image based 3D model retrieval and shown that asymmetric structure leads to stabler features to improve 3D model retrieval accuracy. Our proposed supervised structure combines both the reconstruction loss and the reconstruction-aware classification loss in a unique way for the abstraction of incompleteness, ambiguity, occlusion and noise present in the mixed domain data. We performed 5, 7 and 10-category experiments to show the scalability of our algorithm. We present empirical results of the effectiveness of our models on NYUD2 depth image dataset and ModelNet10 that outperforms recent 3D model retrieval methods.

CHAPTER V

SEMANTICS-ENHANCED AUTOENCODERS

5.1 Background

In this chapter, we propose a new supervised deep autoencoder approach followed by semantic modeling to retrieve 3D shapes based on depth images named Semantics-enhanced single deep supervised autoencoder approach (S3DA). The key novelty of this approach is the two-fold abstraction of incompleteness and ambiguity present in the real depth images by the supervised autoencoder and semantic matching of the supervised autoencoder features respectively. As the first step of this approach we investigate both supervised and unsupervised approaches using an autoencoder to show that supervision on the autoencoder can capture details present in the data and restructure the features to improve the retrieval performance. Semantic modeling on the supervised features offers the next level of abstraction of ambiguity of the depth images. Concisely, by providing abstraction in two steps we can minimize incompleteness and view variability present in the real depth images and improve 3D model retrieval accuracy.

Semantic modeling can help supervised autoencoders to handle ambiguity by providing next-level abstraction. Previous research has shown that increased level of abstraction leads to better image or text retrieval [12, 43]. In our approach, the first abstraction comes from the supervised autoencoder. The next abstraction comes from the semantic modeling of supervised autoencoder features. A supervised autoencoder restructures the autoencoder features with label information. However, if we add more categories of objects then we see that a supervised autoencoder does not have much performance difference with an unsupervised autoencoder because of increased ambiguity added by more object classes. Figure 5.1 narrates the general framework for our S3DA model.

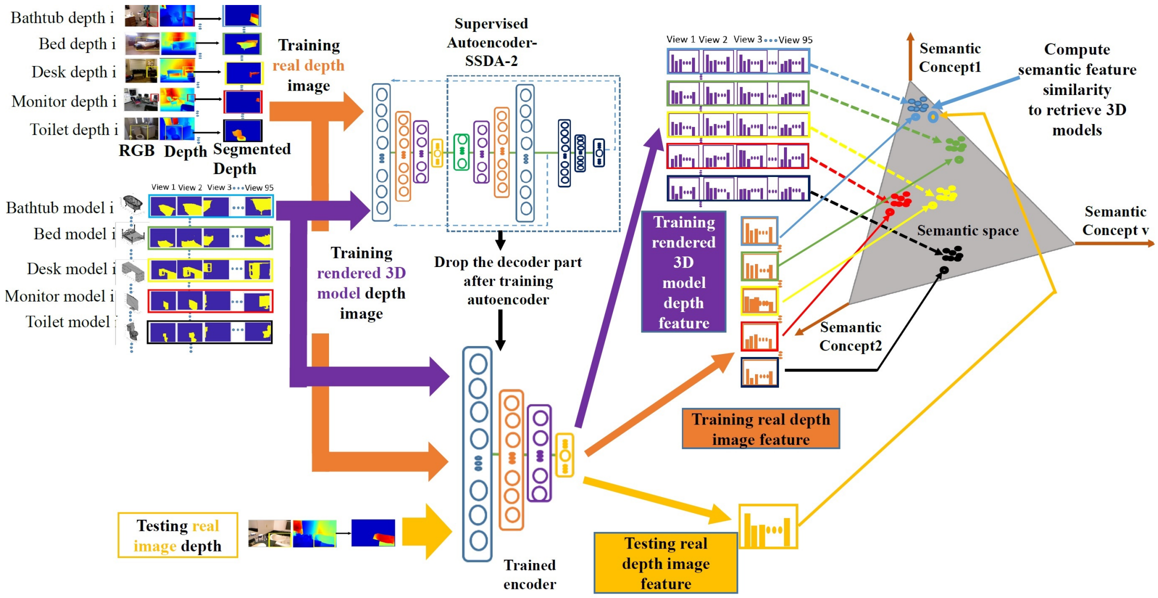

Bathtub depth i
Bed depth i
Desk depth i
Monitor depth i
Toilet depth i
RGB
Depth
Segmented Depth
Training real depth image
Supervised Autoencoder-SSDA-2
View 1 View 2 View 3 ••• View 95
Semantic Concept1
Compute semantic feature similarity to retrieve 3D models
Bathtub model i
Bed model i
Desk model i
Monitor model i
Toilet model
Training rendered 3D model depth image
Drop the decoder part after training autoencoder
Training rendered 3D model depth feature
Semantic space
Semantic Concept v
Semantic Concept2
Training real depth image feature
Testing real image depth
Trained encoder
Testing real depth image feature

Figure 5.1: Semantics-enhanced supervised deep autoencoder model (S3DA) for 3D model retrieval.

Semantic modeling is a fundamental technique used in computer vision used to identify and asociate semantically realted information. For cross-modal retrieval it can bring two different modalities in a subspace where those modalities have natural correspondence. In [12], a subspace of semantic concepts was learned using multi-class logistic regression. In this work, semantic matching was used for image based text retrieval. They have also proposed an effective combination of CCA and semantic matching named semantic correlation matching for cross-modal retrieval. In [43], increased level of abstraction by the use of a semantic space has shown better performance for feature based image retrieval. In our work we have combined a supervised deep autoencoder and semantic modeling by learning a database of semantic concepts based on supervised autoencoder features. The semantic concepts consist of different indoor object classes such as bed, bathtub, chair, dresser.

5.2 Semantics-enhanced single deep supervised autoencoder ($S3DA$)

5.2.1 Building the semantic space

A semantic space consists of semantic concepts. Building a vocabulary $V = \{v1, ..., vk\}$ of semantic concepts for real depth and rendered depth image spaces manages to have a natural correspondence between those domains. The semantic concepts v_i in a database can be grouped in a way that a single depth object is grouped under "Bathtub" or "Bed". To learn the vocabulary V we learn two mappings L_D and L_M. L_D maps a real depth image $D \in \mathbf{D}_{s2}$ into a vector of posterior probabilities $P_{V|D}(v_i|D), i \in \{1, ..., K\}$ in the space S_D with respect to each of the classes in V and L_M maps a rendered depth image $M \in \mathbf{M}_{s2}$ into a vector of posterior probabilities $P_{V|M}(v_i|M), i \in \{1, ..., K\}$ in the space S_M with respect to each of the classes in V. The space S_D consisting of $P_{V|D}$ and the space S_M consisting of $P_{V|D}$ are referred to be the same semantic space S. The mappings L_D and L_M are defined below:

$$L_D : D_{s2} \rightarrow S_D$$

$$L_M : M_{s2} \rightarrow S_M$$

We learn the posterior probabilities $P_{V|M}$ and $P_{V|D}$ through multi-class logistic regression which ends up being a linear classifier with a probabilistic interpretation. The posterior probability for class i is defined as below:

$$P_{V|X}(i|x; w) = \frac{1}{Z(x, W)} \exp(w_i^T x), \tag{5.1}$$

where $Z(x, W) = \sum_{j=1}^{K} \exp(w_j^T x^i)$ is a normalization constant, V is the object category, X is the feature vectors for real depth images or rendered depth images, and $W = \{w_1, ..., w_k\}$, with w_i a vector of parameters for class i.

The following cost function is minimized to find the weights in multinomial logistic regression:

$$J(w) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\{V^{(i)} = k\} log \frac{\exp(w_k^T x^i)}{\sum_{j=1}^{K}\exp(w_j^T x^i)}], \tag{5.2}$$

where m is the number of training samples and $\sum_{j=1}^{K}\exp(w_j^T x^i) = Z(x, w)$ is the normalization constant summed over all the categories. From softmax regression we have that

$$P_{V|X}(i|x; w) = \frac{1}{Z(x, W)}\exp(w_i^T x) = \frac{\exp(w_i^T x)}{\sum_{j=1}^{K}\exp(w_j^T x^i)}, \tag{5.3}$$

where there is no known closed form solution to estimate the parameters that minimize the cost function $J(w)$ analytically and thus we need to use an iterative optimization algorithm such as gradient descent. The iterative algorithm requires us taking the partial derivative of the cost function which is equal to:

$$\nabla_{w^{(k)}} J(w) = -\frac{1}{m}\sum_{i=1}^{m}[x^{(i)}(1\{V^{(i)} = k\} - P_{V|X}(i = k|x^{(i)}; w))]. \tag{5.4}$$

where $\nabla_{w^{(k)}} J(w)$ is itself a vector, so that its j-th element is $\frac{\partial J(w)}{\partial w_{lk}}$ the partial derivative J(w) with respect to the j-th element of $w^{(k)}$.

5.2.2 Retrieval using the semantic space

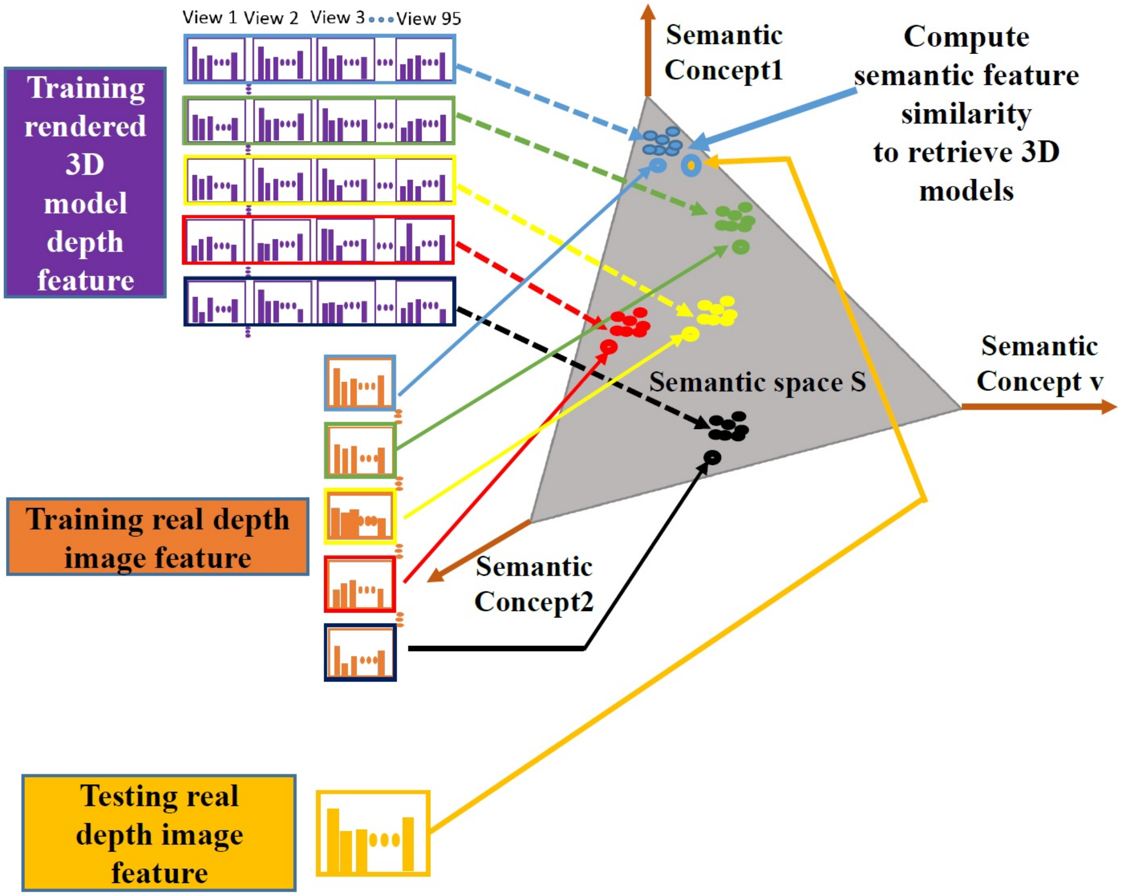

View 1 View 2 View 3 View 95
Training rendered 3D model depth feature
Training real depth image feature
Testing real depth image feature
Semantic Concept1
Compute semantic feature similarity to retrieve 3D models
Semantic space S
Semantic Concept v
Semantic Concept2

Figure 5.2: Semantics space S consisting of S_M and S_D.

The supervised-autoencoder features M_{s2} and D_{s2} though supervised still have no obvious image interpretation due to incompleteness of the image objects. Semantic modeling of supervised autoencoder features has two advantages. The first advantage is that the semantic features in S_M and S_D are semantic concept probabilities (e.g. the probability that the depth image object belongs to the "Bed" or "Bathtub" categories) providing higher level abstraction for ambiguities present in the occluded depth images. The other advantage is that the semantic spaces S_M and S_D are isomorphic. The isomorphic space S_M consists of posterior probability vectors $\Pi_M = P_{V|M}$

and S_D consists of posterior probability vectors $\Pi_D = P_{V|D}$. The spaces S_M and S_D can be assumed as a single seamless semantic space S, i.e. $S_D = S_M$, illustrated in Figure 5.2. Given a query real depth image D represented by a probability vector $\Pi_{Dq} \in S_D$, our retrieval process finds the best matched rendered 3D model depth image M represented by a probability vector $\Pi_M \in S_M$ having the lowest distance d defined below:

$$Dist_{sm}(M, D) = d(\Pi_M, \Pi_{Dq}), \tag{5.5}$$

where $d(.)$ is the distance between Π_M and Π_D . We refer to this type of retrieval as S3DA where S3 means supervised single semantic modeling of features.

5.3 Experiments

5.3.1 Experimental setup

Datasets

We have used the same datasets of the NYUD2 data set [13] and ModelNet10 benchmark [14] as we described in chapter 3 and chapter 4. Totally, we created 3347 depth images that are split into 1907 for training and 1440 for testing of ten indoor objects (bathtub, bed, chair, desk, dresser, nightstand, table, toilet, monitor and sofa) and 605 3D models belonging to the ten categories. From those 605 models we generated around 57475 rendered depth images as suggested in sliding shapes [39].

Evaluation criteria

Our evaluation of retrieval performance includes 5 well recognized retrieval metrics (NN, FT, ST, DCG and mAP) having a maximum of 100% and a 11-point precision-recall curve. The definition and scope of these metrics have been described twice in chapter 3 and chapter 4.

5.3.2 Performance comparison

	NN	FT	ST	DCG	AP
		5 categories			
PNN	0.1974	0.3533	0.5529	0.6754	0.3798
SDA [42]	0.5631	0.4877	0.6963	0.8927	0.5351
SSDA-1 [42]	0.6084	0.5250	0.6769	0.8937	0.5811
SSDA-2	0.6160	0.5274	0.6719	0.8972	0.5811
SSDAA	0.6343	0.5387	0.6798	0.8967	0.5960
S3DAA	0.6375	0.5601	0.6782	**0.9011**	**0.6154**
S3DA	**0.6893**	**0.5632**	**0.6792**	0.9005	0.6138
		7 categories			
PNN	0.0659	0.1843	0.3400	0.5947	0.2255
SDA	0.4995	0.3746	0.5586	0.8532	0.4194
SSDA-1	0.5178	0.3919	0.5683	0.8576	0.4273
SSDA-2	0.5178	0.3936	0.5815	0.8583	0.4287
SSDAA	0.5453	0.4035	0.5729	0.8601	0.4411
S3DAA	0.5727	**0.4357**	**0.5719**	**0.8635**	**0.4758**
S3DA	**0.5746**	0.4259	0.5700	0.8622	0.4669
		10 categories			
PNN	0.0729	0.1115	0.2326	0.5517	0.1373
SDA	0.3472	0.2429	0.4090	0.8089	0.2606
SSDA-1	0.3535	0.2557	0.4221	0.8133	0.2734
SSDA-2	0.3986	0.2624	0.4265	0.8172	0.2796
SSDAA	0.3618	0.2759	0.4526	0.8192	0.2998
S3DAA	0.4188	**0.2976**	**0.4647**	**0.8240**	**0.3230**
S3DA	**0.4326**	0.2925	0.4588	0.8223	0.3151

Table 5.1: Performance metrics comparison of depth-image based 3D model retrieval on the NYU Depth V2 dataset and the ModelNet10 becnhmark.

We have compared our results with the PNN approach [10] and the supervised autoencoder approach [42]. These approaches represent the most recent approaches for depth image based 3D model retrieval. The experimental results are reported in Table 5.1 for 5, 7 and 10 categories. The experimental results show that S3DA approach performs the best among PNN, SDA, SSDA-1 and SSDA-2. The Table 5.1 suggests that supervised models perform better than the unsupervised model. But if the number of categories are increased supervised models perform similar to the unsupervised ones due to ambiguity and uncertainty coming from increased number of model categories. From Table 5.1 we can see that SSDA-2 performs better than SSDA-1. Again we see that the asymmetric supervised deep autoencoder approach performs the best among all other supervised deep autoencoder approaches. S3DA improves the retrieval result of SSDA-2 and S3DAA improves the retrieval result of SSDAA. Overall we can say that the semantic modeling over supervised autoencoder approaches S3DA (applied over SSDA-2) and S3DAA(applied over SSDAA) are the best performing approaches.

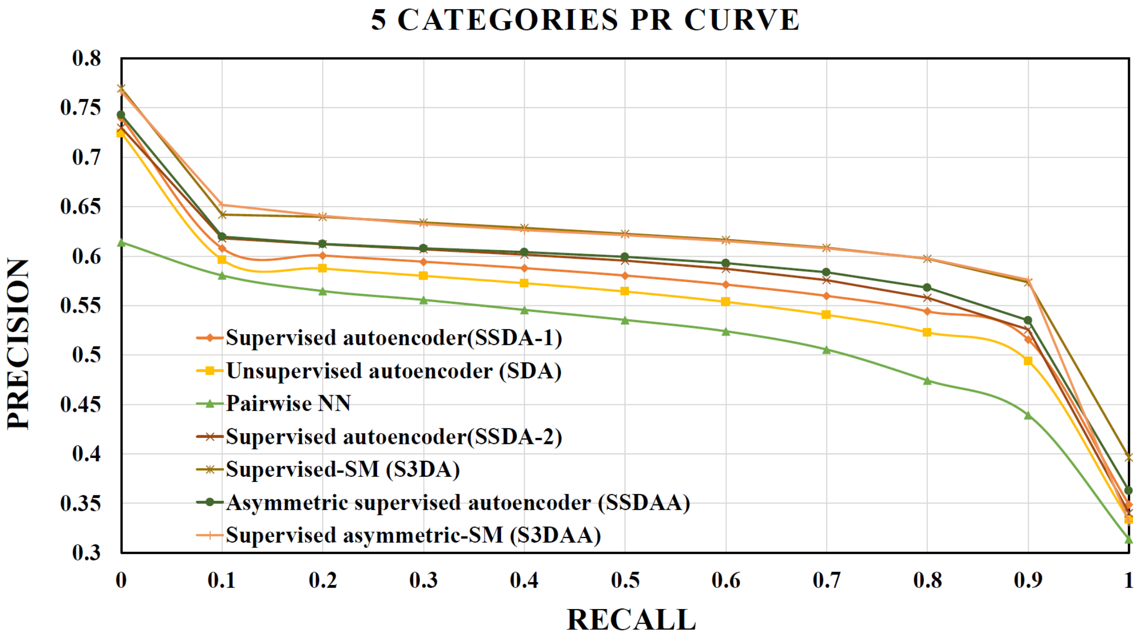


Figure 5.3: PR curve for 5-category experiment.

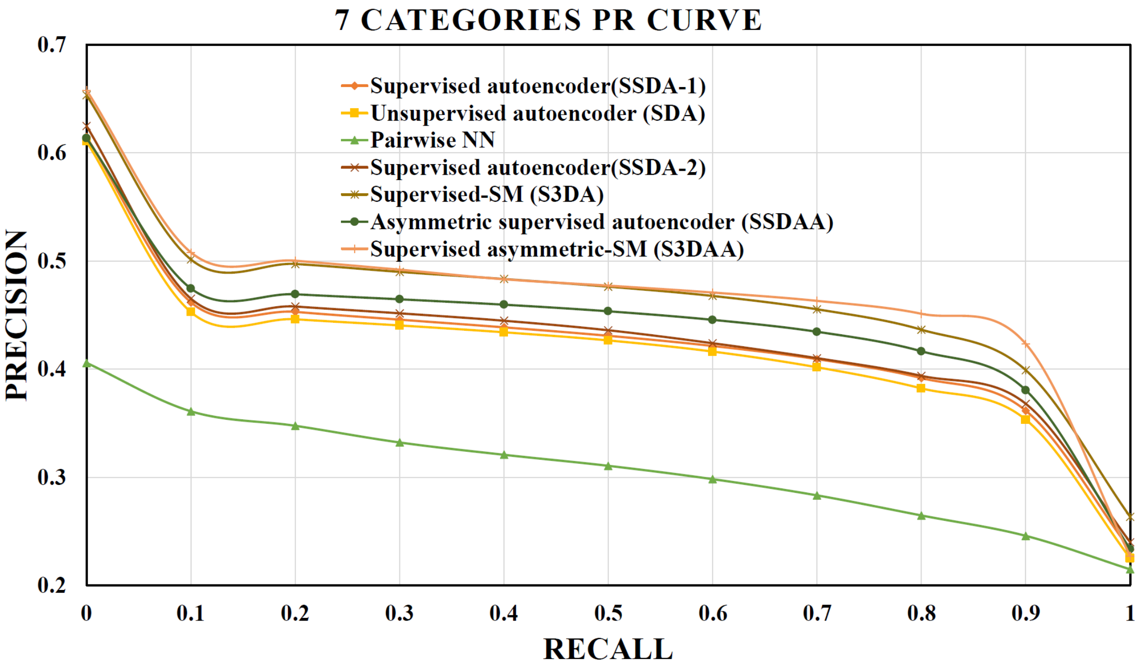


Figure 5.4: PR curve for 7-category experiment.

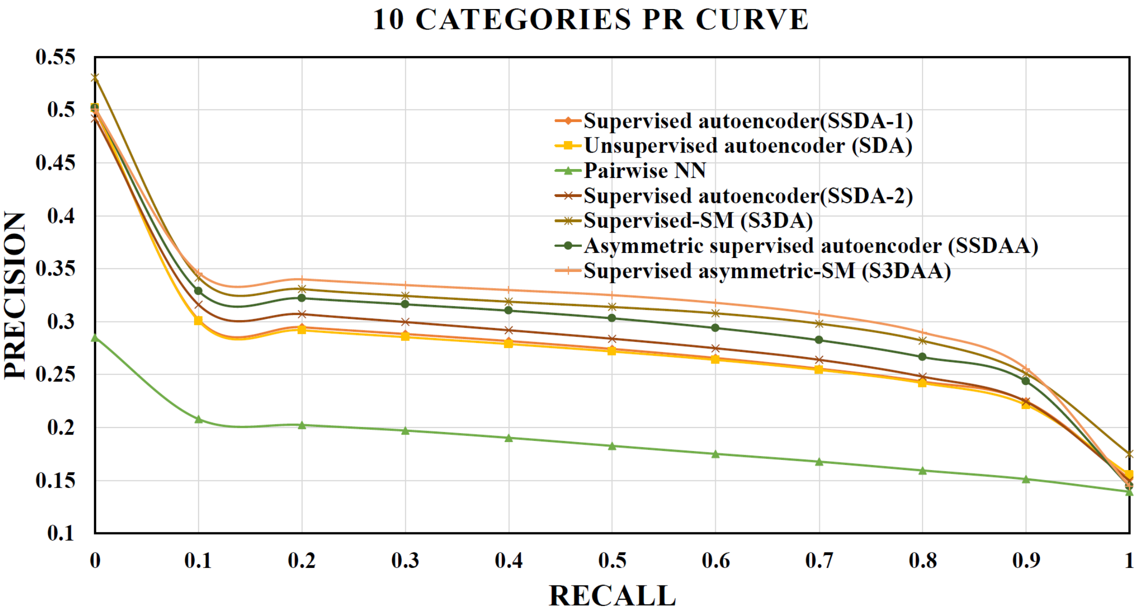


Figure 5.5: PR curve for 10-category experiment.

PR curves in Figures 5.3, 5.4 and 5.5 for 5, 7 and 10 category experiments show that S3DA and S3DAA approaches perform the highest among PNN, SDA, SSDA-1 and SSDA-2. Figure 5.4 and 5.5 show that SDA and SSDA-1 performances are similar since the supervision is not effective for training almost 60,000 depth images. But the supervised approach SSDA-2 performs better than the other supervised approach SSDA-1, which means the supervision in SSDA-2 is more effective than SSDA-1 when more ambiguity is added to the data. Also SSDA-2 retrieval accuracy is slightly better than SSDA-1. Again the asymmetric approach SSDAA performs better than SSDA-2. Our PR curves suggest that S3DA performs better than the supervised approaches, supervised approaches are better than the unsupervised approach and unsupervised approach performs better than PNN. The proposed S3DAA performs the slightly better than S3DA. Overall we can say that the semantic model over supervised deep autoencoder approaches S3DA and S3DAA perform the best among all the proposed methods. We have used 59382 depth images for 10 categories, 37144 depth images for 7 categories and 28240 depth images for 5 categories which is around 95 times greater than PNN.

5.3.3 Retrieval results

In this section, we have shown some examples of the first ten retrieved 3D models by SSDA-2 and S3DA. In those figures, the first row shows the depth object we are interested to retrieve 3D models for. The second and third rows outlined in green belong to S3DA retrieval. The second row displays what the S3DA model has retrieved and the third row displays the corresponding 3D models. The fourth and fifth rows outlined in blue belong to SSDA-2 retrieval. The fourth row displays what the SSDA-2 model has retrieved and the fifth row displays the corresponding 3D models.

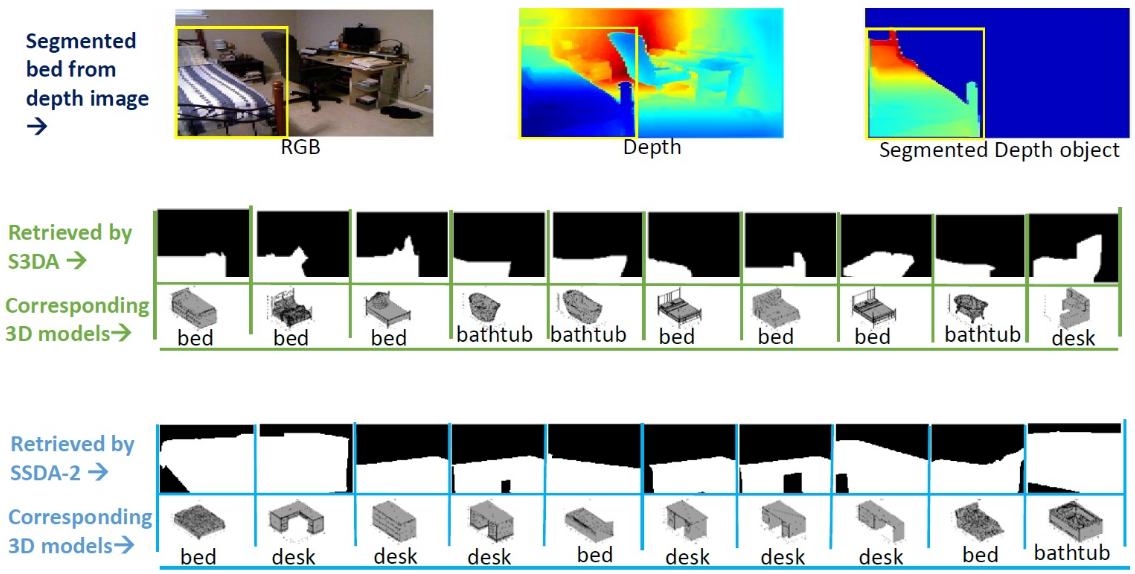


Figure 5.6: Retrieval example: bed (first ten retrieved bed models).

In Figure 5.6, we see that S3DA has given better performance than SSDA-2 by retrieving relevant 3D models. S3DA has retrieved six bed models among the first ten retrieved models. On the other hand, SSDA-2 has retrieved only three bed models. S3DA has retrieved bed models on the first, second, third, sixth and eighth ranked positions. And SSDA-2 has retrieved bed models on the first, fifth and seventh ranked positions. One other thing we observe from this retrieval is that the bed models retrieved by S3DA have more shape similarity to the segmented bed from the depth image than SSDA-2.

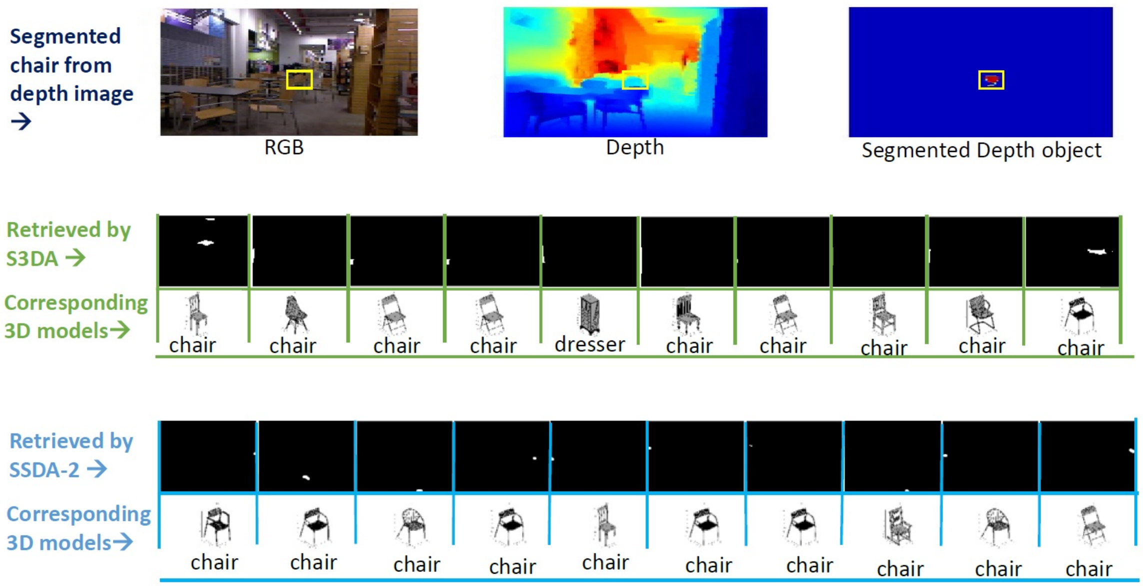


Figure 5.7: Retrieval example: chair (first ten retrieved chair models).

In Figures 5.7 we see the first ten retrieved 3D models for a partial chair. Both S3DA and SSDA-2 have performed well and have retrieved a lot of 3D chair models. Though the information for the chair is partial and very small, we still see that both the supervised models have not confused the partial chair with other categories.

In Figure 5.8, SSDA-2 has confused dresser with bathtub and bed mostly and S3DA has confused dresser with chair. The reason for this bad performance could be the ambiguity present in the dresser due to occlusion. In this case of dresser retrieval, SSDA-2 has given comparably better performance than S3DA by retrieving two dressers in the second and third rank positions. S3DA has retrieved only one dresser and the rank of position of that dresser model is eighth.

In Figure 5.9, we see that S3DA has given better performance than SSDA-2 by retrieving relevant 3D models. S3DA has retrieved five sofa models among the first ten retrieved models. On the other hand, SSDA-2 has retrieved only two sofa models. S3DA has retrieved sofa on the second, fifth, sixth, seventh and tenth ranked positions and SSDA-2 has retrieved sofa on the third and fifth positions. One other thing, we

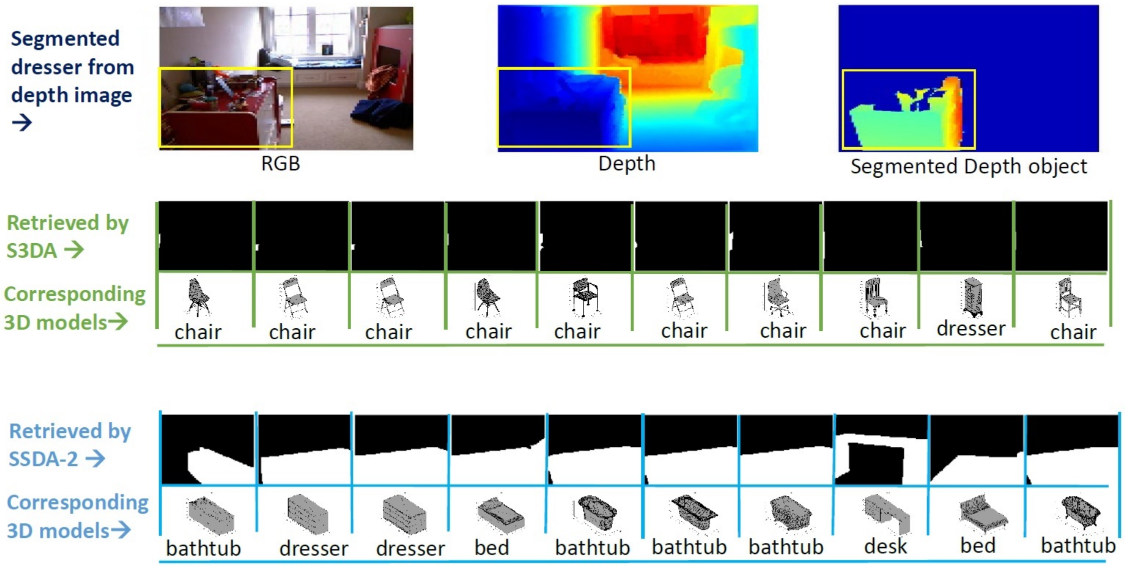


Figure 5.8: Retrieval example: dresser(first ten retrieved dresser models).

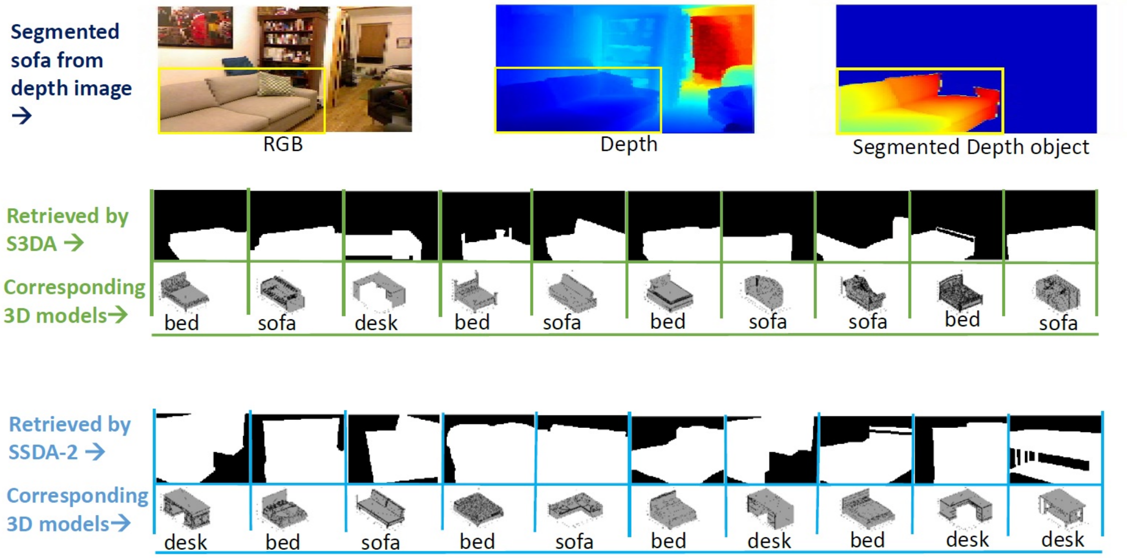


Figure 5.9: Retrieval example: sofa (first ten retrieved sofa models).

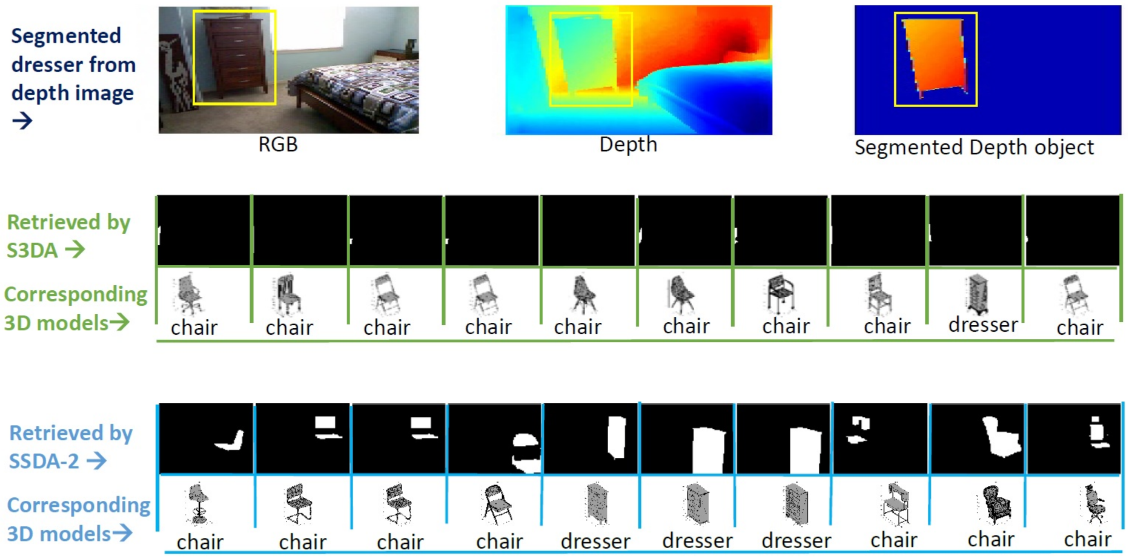


Figure 5.10: Retrieval example: dresser (first ten retrieved dresser models).

observe from this retrieval is that the sofa models retrieved by S3DA have more shape similarity to the segmented bed from the depth image than SSDA-2.

In Figure 5.10 we see that SSDA-2 has confused dresser with chair mostly and S3DA has confused dresser with chair. The reason for this bad performance could be the ambiguity present in the dresser due to occlusion. In this case of dresser retrieval, SSDA-2 has given comparably better performance than S3DA by retrieving three dressers on the fifth, sixth and seventh rank positions. S3DA has retrieved only one dresser and the rank of position of that dresser model is eighth. One other thing, we observe from this retrieval is that the dresser models retrieved by SSDA-2 have more shape similarity to the segmented dresser from the depth image than S3DA.

Figure 5.11 is basically a failure case by SSDA-2 and S3DA models. Both the approaches have confused monitor with chair mostly. The reason for this bad performance could be the ambiguity present in the monitor due to shape similarity between the upper part of a chair and a monitor. In this case of monitor retrieval, SSDA-2 has given comparably better performance than S3DA by retrieving two monitors on

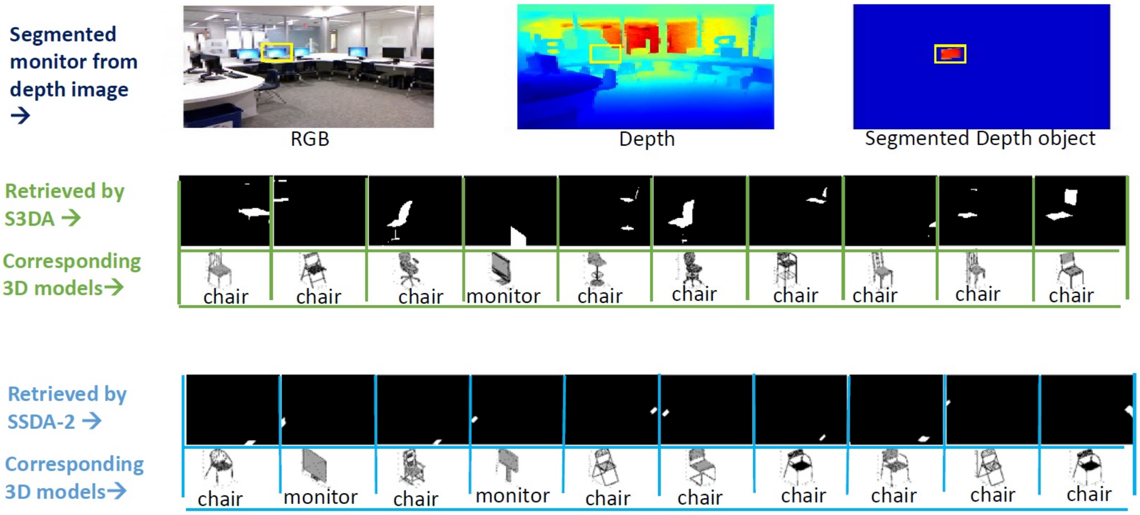


Figure 5.11: Retrieval example: monitor (first ten retrieved monitor models).

the second, and fourth rank positions. S3DA has retrieved only one monitor and the rank of position of that monitor model is fourth. One other thing, we observe from this retrieval is that the models retrieved by S3DA have more shape similarity to the segmented monitor from the depth image than SSDA-2.

In Figure 5.12, we see that both S3DA and SSDA-2 have given similar performance. S3DA has retrieved three sofa models among the first ten retrieved models. On the other hand, SSDA-2 has also retrieved three sofa models. S3DA has retrieved sofa models on the eighth, ninth and tenth ranked positions and SSDA-2 has retrieved sofa models on the fifth, seventh and eighth positions. Though SSDA-2 has performed better than S3DA rankwise, we observe from this retrieval that the sofa models retrieved by S3DA have more shape similarity to the segmented bed from the depth image than SSDA-2.

In Figure 5.13 we see that the retrieval case is a failure for S3DA model. SSDA-2 has retrieved a lot of correct dresser models on the first ten retrievals. SSDA-2 has confused dresser mostly with chair. In this case of dresser retrieval, SSDA-2 has given comparably better performance than S3DA by retrieving five dressers on the second,

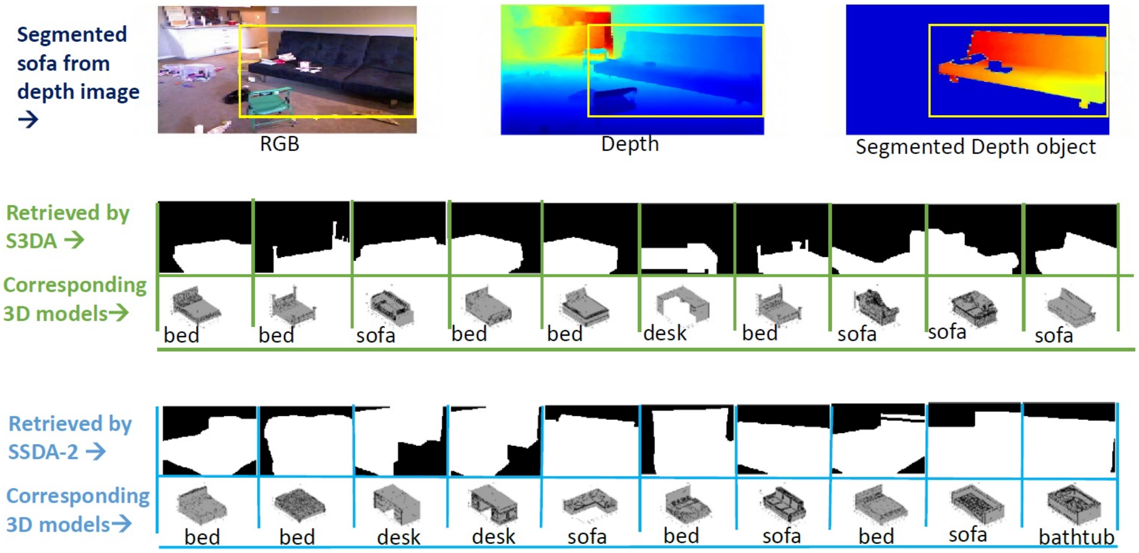


Figure 5.12: Retrieval example: sofa (first ten retrieved sofa models).

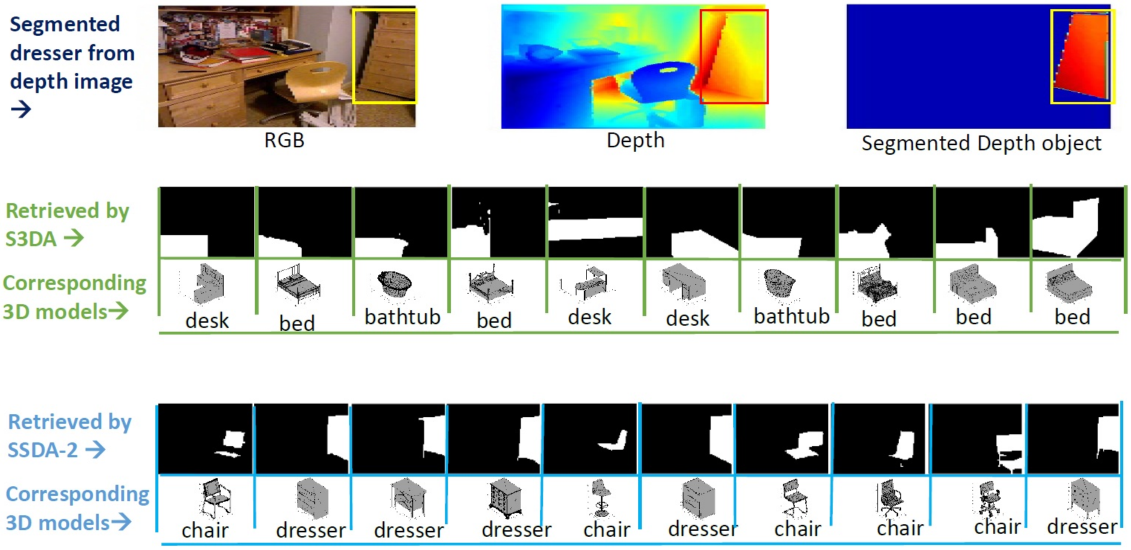


Figure 5.13: Retrieval example: dresser (first ten retrieved dresser models).

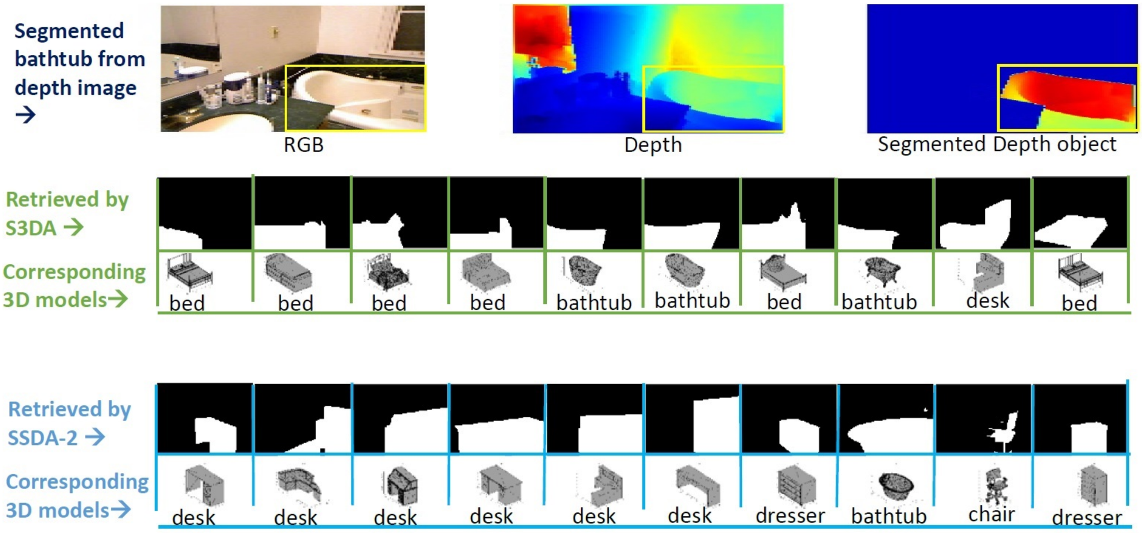


Figure 5.14: Retrieval example: bathtub (first ten retrieved bathtub models).

thrid, fourth, sixth and tenth rank positions. S3DA has retrieved no dresser model at all.

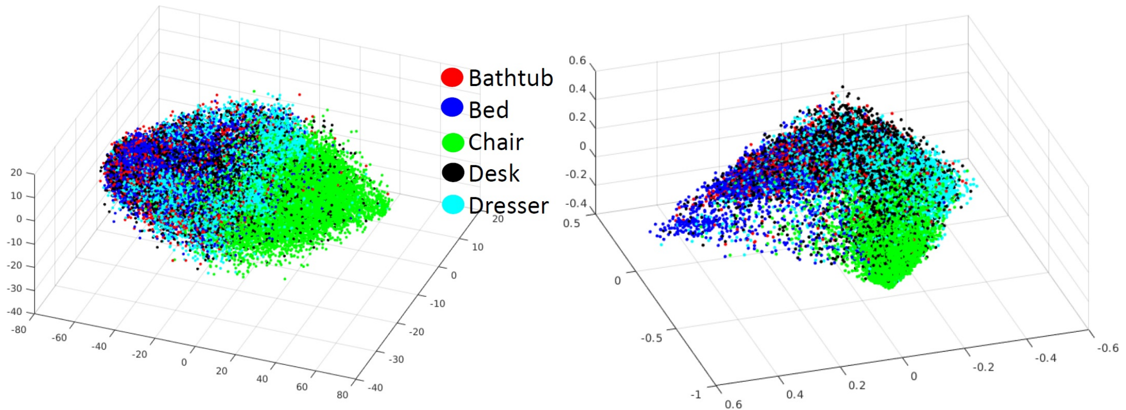
In Figure 5.14, we see that both S3DA and SSDA-2 have not performed very well yet S3DA has given better performance than SSDA-2 comparably by retrieving relevant 3D models. S3DA has retrieved three bathtub models among the first ten retrieved models. On the other hand, SSDA-2 has retrieved only one bathtub model. S3DA has retrieved bathtub models on the fifth, sixth and eighth ranked positions and SSDA-2 has retrieved bathtub models on the eighth position only. One other thing, we observe from this retrieval is that the bathtub models retrieved by S3DA have more shape similarity to the segmented bathtub from the depth image than SSDA-2.

5.3.4 Analysis and discussion

1. Computational cost

The training time is measured on a PC with 1.2GHz CPU and GTX 750 GPU. We have applied semantic modeling on SSDA-2 and SSDAA. So the training time of S3DA and S3DAA involve the training time of SSDA-2 and SSDAA. SSDAA training time is less than SSDA-2 because of less parameters. So evidently the total training time of S3DAA is less than S3DA. We chose the learning rate of 0.01 to avoid overfitting with 50% dropout ratio that further reduces the risk of overfitting.

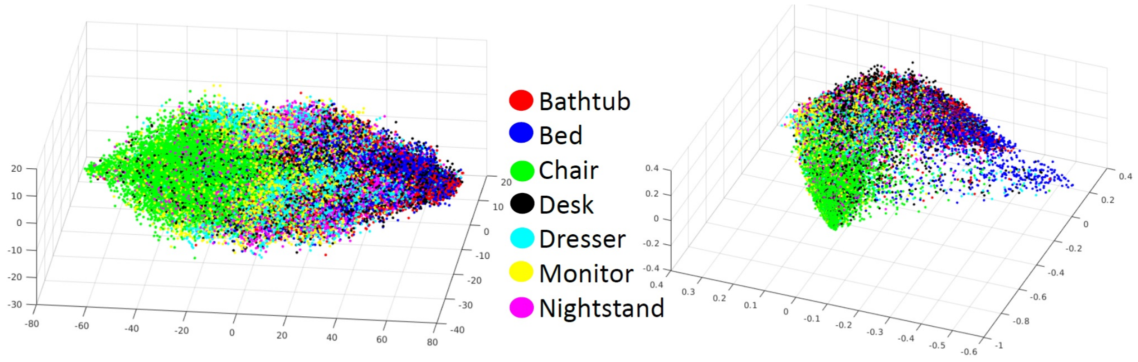
2. Feature space comparison



(a) Supervised 5-category cluster (SSDA-2)

(b) Supervised SM 5-category cluster (S3DA)

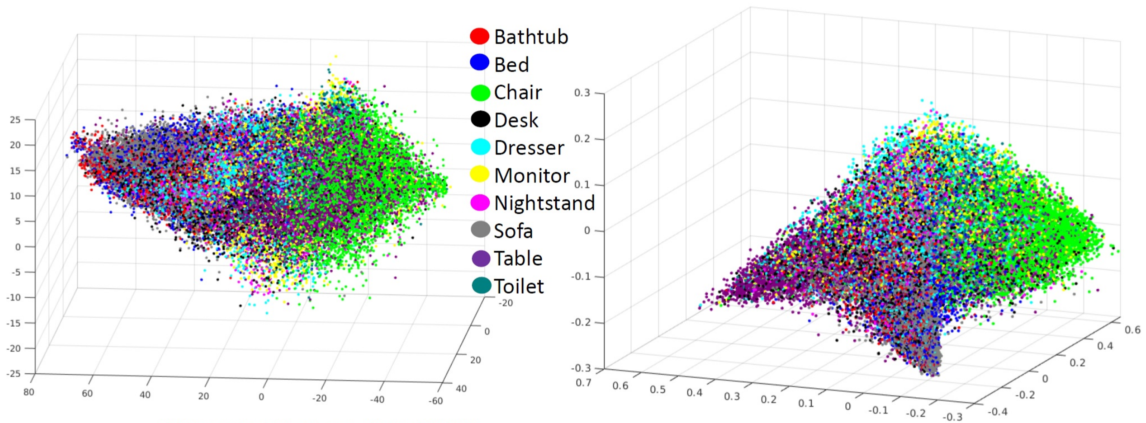
Figure 5.15: Feature space visualization for 5 category: 3D PCA on 30D SSDA-2 feature and 30D S3DA feature. The SVM classification accuracy for 3D features of S3DA is 46% and 39% for SSDA-2, around 1.18 times greater than SSDA-2.



(a) Supervised 7-category cluster (SSDA-2)

(b) Supervised SM 7-category cluster (S3DA)

Figure 5.16: Feature space visualization for 7 category experiment: 3D PCA on 30D SSDA-2 feature and 30D S3DA feature.



(a) Supervised 10-category cluster (SSDA-2)

(b) Supervised SM 10-category cluster (S3DA)

Figure 5.17: Feature space visualization for 10 category experiment: 3D PCA on 30D SSDA-2 feature and 30D S3DA feature.

To visualize the supervised and semantics-enhanced supervised autoencoder features in a low dimensional space, we have applied 3D PCA on 30D autoencoder features. In the Figures 5.15, 5.16 and 5.17 we show the low dimensional feature comparison between supervised (SSDA-2) and semantic enhanced supervised autoencoder (S3DA) for 5, 7 and 10 categories respectively. From the figures we observe that S3DA features space is pointier than SSDA-2. In S3DA every corner is roughly dedicated for an object category. Obviously S3DA feature space has feature overlaps due to shape similarity among different object categories yet S3DA feature space has less overlap than SSDA-2. We see that the feature space overlap increases from five category feature space to seven category features space and seven category feature space to ten category feature space. Improvement in retrieval accuracy in S3DA suggest that this sharp corner feature space is a better criteria for 3D model retrieval.

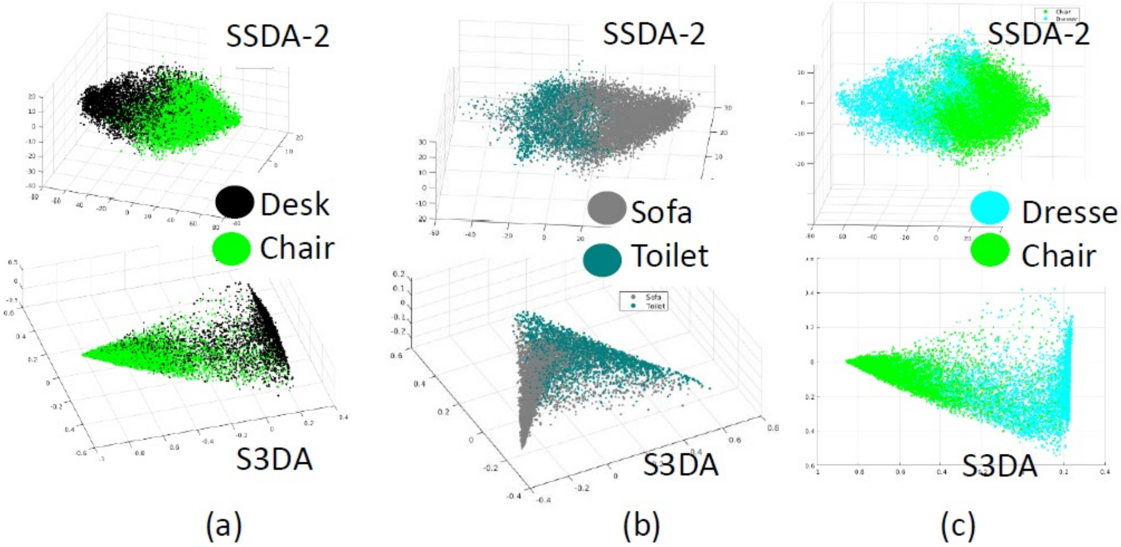


Figure 5.18: Pairwise feature space comparison: 3D PCA of 30D S3DA features are sharper than SSDA-2 feature which indicates less overlap.

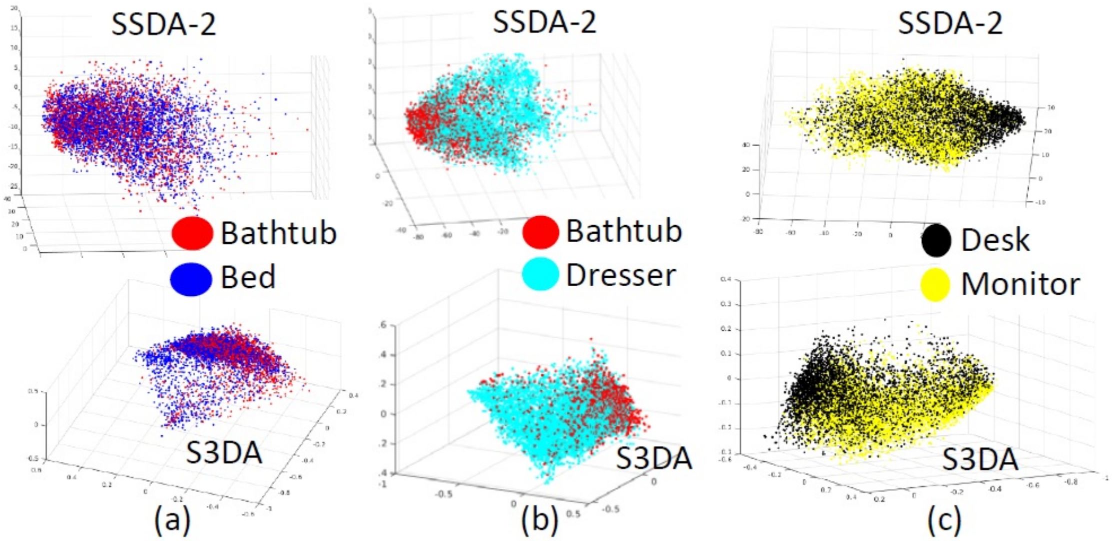


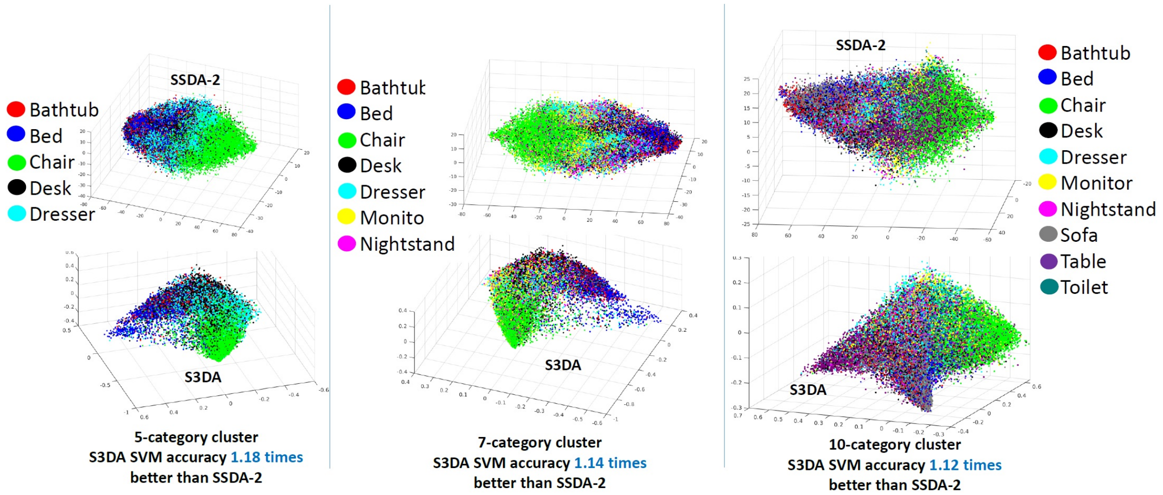
Figure 5.19: Pairwise feature space comparison.

In Figure 5.18(a)-(c), we show the clusters for desk-chair, sofa-toilet and dresser-chair pair comparisons between supervised (SSDA-2) and supervised-SM (S3DA). In Figure 5.19(a)-(c), we show the clusters for bathtub-bed, bathtub-dresser and desk-monitor. In Figure 5.18(a)-(c), the sharper nature of the S3DA feature space suggests less overlap whereas the SSDA-2 feature space overlap more for desk-chair, sofa-toilet and dresser-chair pairs. In Figure 5.19(a), bathtub and bed clusters overlap badly for both the cases. In Figure 5.19(b), S3DA feature space is slightly sharper than S3DA for the bathtub-dresser pair. In Figure 5.19(c), S3DA desk-monitor feature space suggests better pairing than SSDA-2 since in S3DA feature space the lower part is mostly monitor dominated and upper left part is mostly desk dominated. For SSDA-2, the desk-monitor feature space is more ambiguous than S3DA.

3. Advantage of semantic modeling over supervised autoencoder

To visualize the supervised autoencoder and semantics-enhanced supervised autoencoder features in a low dimensional space, we have applied 3D PCA for dimensionality reduction of 30D autoencoder features. To compare SSDA-2 and S3DA, we visualize their respective features in the case of 5, 7 and 10 categories. To verify cluster separability among different categories, we apply a simple SVM method for PCA classification. The higher the classification accuracy, the lower the cluster overlap.

In Figure 5.20, we show the low dimensional feature comparison between supervised (SSDA-2) and semantic enhanced supervised autoencoder (S3DA) approaches. Obviously both of the cases have overlaps present in the clusters for different categories due to object similarity and incompleteness. But the SVM classification accuracy using 3D PCA of S3DA is 1.18 times greater than 3D PCA of SSDA-2. For visualization clarity we have displayed two clusters at a time. In Figure 5.21, we show the clusters for desk-chair, sofa-toilet and dresser-chair pair comparisons. S3DA clusters seem to form a well defined sharper cluster which suggest less overlap region between two pairs.



5-category cluster
S3DA SVM accuracy 1.18 times
better than SSDA-2

7-category cluster
S3DA SVM accuracy 1.14 times
better than SSDA-2

10-category cluster
S3DA SVM accuracy 1.12 times
better than SSDA-2

Figure 5.20: Feature space visualization for 5, 7 and 10 category experiment: 3D PCA on 30D SSDA-2 feature and 30D S3DA feature. The SVM classification accuracy for 3D features of S3DA is around 1.18, 1.14 and 1.12 times greater than SSDA-2 for 5, 7 and 10 category clusters respectively.

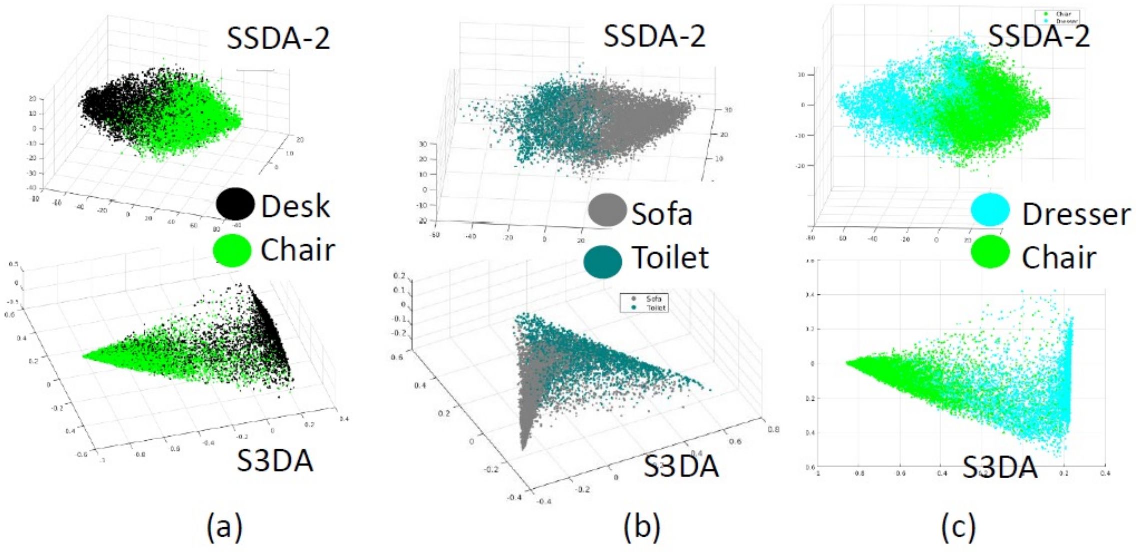


Figure 5.21: Pairwise feature space comparison: 3D PCA of 30D S3DA feature are sharper than SSDA-2 feature which suggests less overlap.

In conclusion of the retrieval by semantics-enhanced supervised deep autoencoder (S3DA) approach, we can say that adding semantic modeling on top of supervised autoencoder SSDA-2 features, we can obviously achieve better retrieval performance. At the same time, some examples presented above suggest that the supervised autoencoder approach alone has some advantages for successful retrieval.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this research we dealt with a cross-modal retrieval problem of depth image based 3D model retrieval. Depth image based 3D model retrieval has the challenges of occlusion, noise, and view variability present in depth data. A huge amount of ambiguity exists due to shape similarity among different objects. All the existing 3D model retrieval approaches use a pairwise network or separate network to train each domain. We argue that using single deep network is convenient than pairwise networks since this process is computationally inexpensive. The synthetic and real depth images are trained together in our models which increases the possibility to bring two different domains in a single feature space. We have proposed an unsupervised and three novel supervised models for cross-modal 3D model retrieval using one single deep autoencoder. In the proposed supervised deep autoencoder models we have shown the strength of supervision on autoencoder by doing classification aware reconstruction. We have studied the retrieval of ten indoor objects (bathtub, bed, chair, desk, dresser, night stand, table, toilet, monitor and sofa). We also argue that generating different views increases the possibility of successful retrieval. Compared to 1000D dimensional features, 30D features work meaningfully in our models which is a significantly faster retrieval approach. The overhead of computing features is decreased in our models since each domain since we work with depth images directly.

6.1 Deep autoencoder structures

We investigate both supervised and unsupervised approaches using an autoencoder to show that supervision on the autoencoder can capture details present in the data and restructure the features to improve the retrieval performance. We also propose an asymmetric supervised deep autoencoder by investigating the relation between the number of encoders and decoders. We claim that asymmetric structure of a supervised deep autoencoder learns more robust and effective unified embedding to bridge the gap between cross-domains if those different domains are trained together. We have compared supervised symmetric and supervised asymmetric structure of a deep autoencoder for depth image based 3D model retrieval and shown that asymmetric structure leads to stabler features to improve 3D model retrieval accuracy.

6.2 Semantic modeling of autoencoder features

Semantic modeling on the supervised features offers the next level of abstraction of ambiguity of the depth images in our research. In a nutshell, two-level abstraction improves 3D model retrieval accuracy by minimizing incompleteness and view variability present in the real depth images. Retrieving 3D model based on depth images can be considered a transfer learning approach since real depth and rendered depth have different depth quality. We performed 5, 7 and 10-category experiments to show the scalability of our algorithms. We have measured the effectiveness of our model on NYUD2 depth image dataset and ModelNet10 3D models of the same category. We outperformed all the state of the art methods for cross-modal retrieval. The proposed supervised method outperforms the recent approaches for cross modal 3D model retrieval based on depth images.

6.3 Future work

The future plan of this research would be proposing a new model combining the advantages of the proposed supervised deep autoencoders and semantic modeling for retrieval. Having an autoencoder for each object category could be another potential model for retrieval. A novel future work would be proposing a deep autoencoder having two different encoders in a deep autoencoder to evaluate two different dimensional features, the encoder providing lower dimensional feature such as two or three dimensional feature space will be used to increase inter object cluster difference to handle ambiguity more precisely.

REFERENCES

[1] F. Zhuang, D. Luo, X. Jin, H. Xiong, P. Luo, and Q. He, "Representation learning via semi-supervised autoencoder for multi-task learning," in *Proc. ICDM*, 2015.

[2] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification.," in *AAAI*, 2017.

[3] A. K. Dhaka and G. Salvi, "Semi-supervised learning with sparse autoencoders in phone classification," *arXiv preprint arXiv:1610.00520*, 2016.

[4] A. Gogna and A. Majumdar, "Semi supervised autoencoder," in *Proc. NIPS*, 2016.

[5] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. EMNLP*, 2011.

[6] A. K. Dhaka, "Semi-supervised learning with sparse autoencoders in automatic speech recognition," 2016.

[7] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3d shape retrieval," *Neurocomputing*, vol. 204, pp. 41–50, 2016.

[8] Z.-M. Liu, Y.-Y. Chen, S. Hidayati, S.-C. Chien, F.-C. Chang, and K.-L. Hua, "3d model retrieval based on deep autoencoder neural networks," in *Proc. IC-SigSys*, 2017.

[9] F. Wang, L. Kang, and Y. Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *Proc. CVPR*, 2015.

[10] J. Zhu, F. Zhu, E. K. Wong, and Y. Fang, "Learning pairwise neural network encoder for depth image-based 3d model retrieval," in *Proc. ACM Multimedia*, 2015.

[11] T. Du and L. Liao, "Deep neural networks with parallel autoencoders for learning pairwise relations: Handwritten digits subtraction," in *Proc. ICMLA*, 2015.

[12] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proc. ACM Multimedia*, 2010.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," *ECCV*, pp. 746–760, 2012.

[14] 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*, 2015.

[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, 2010.

[16] C. Diaz, M. Walker, D. A. Szafir, and D. Szafir, "Designing for depth perceptions in augmented reality," in *Mixed and Augmented Reality (ISMAR), 2017 IEEE International Symposium on*, pp. 111–122, IEEE, 2017.

[17] RealityTechnologies, "The ultimate guide to understanding augmented reality (ar) technology." `https://www.realitytechnologies.com/augmented-reality/`, 2019. [Online; accessed 22-April-2019].

[18] iCanDesign, "Room planner: Interior and floorplan design for ikea." https://apkpure.com/room-planner-interior-floorplan-design-for-ikea/com.icandesignapp.all, 2018. [Online; accessed 23-April-2019].

[19] T. Shao, A. Monszpart, Y. Zheng, B. Koo, W. Xu, K. Zhou, and N. J. Mitra, "Imagining the unseen: Stability-based cuboid arrangements for scene understanding," *ACM Transactions on Graphics*, vol. 33, no. 6, 2014.

[20] E. Guizzo, "How google's self-driving car works." https://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works, 2011. [Online; accessed 15-December-2018].

[21] W. Xu, Q. Zhu, Z. Du, and Y. Zhang, "Design and implementation of 3d model database for general-purpose 3d gis," *Geo-spatial Information Science*, vol. 13, no. 3, pp. 210–215, 2010.

[22] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. ICML*, 2009.

[23] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. ICML*, 2011.

[24] R. Huang, C. Liu, G. Li, and J. Zhou, "Adaptive deep supervised autoencoder based image reconstruction for face recognition," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[25] M. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *Proc. ICML*, ACM, 2008.

[26] T. Darom and Y. Keller, “Scale-invariant features for 3-d mesh models,” *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2758–2769, 2012.

[27] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *Proc. CVPR*, 2009.

[28] J. Zhu, J.-R. Rizzo, and Y. Fang, “Learning domain-invariant feature for robust depth-image-based 3d shape retrieval,” *Pattern Recognition Letters*, 2017.

[29] P. Vincent, “A connection between score matching and denoising autoencoders,” *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[30] V. Turchenko, E. Chalmers, and A. Luczak, “A deep convolutional auto-encoder with pooling-unpooling layers in caffe,” *arXiv preprint arXiv:1701.04949*, 2017.

[31] S. Jenni and P. Favaro, “Self-supervised feature learning by learning to spot artifacts,” *arXiv preprint arXiv:1806.05024*, 2018.

[32] R. Liu and J. Jia, “Reducing boundary artifacts in image deconvolution,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 505–508, IEEE, 2008.

[33] A. Majumdar and A. Tripathi, “Asymmetric stacked autoencoder,” *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 911–918, 2017.

[34] N. Bosch and L. Paquette, “Unsupervised deep autoencoders for feature extraction with educational data,” in *Deep Learning with Educational Data Workshop at the 10th International Conference on Educational Data Mining*, 2017.

[35] Y. Sun, H. Mao, Q. Guo, and Z. Yi, “Learning a good representation with unsymmetrical auto-encoder,” *Neural Computing and Applications*, vol. 27, no. 5, pp. 1361–1367, 2016.

[36] H. Lee, E. Yang, and S. J. Hwang, “Deep asymmetric multi-task feature learning,” *CoRR*, vol. abs/1708.00260, 2017.

[37] J. Liang and R. Liu, “Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network,” in *Proc. CISP*, 2015.

[38] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.

[39] S. Song and J. Xiao, “Sliding shapes for 3d object detection in depth images,” in *Proc. ECCV*, 2014.

[40] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” in *Proc. Shape modeling applications*, 2004.

[41] K. Järvelin and J. Kekäläinen, “Ir evaluation methods for retrieving highly relevant documents,” in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.

[42] A. Siddiqua and G. Fan, “Supervised deep-autoencoder for depth image-based 3d model retrieval,” in *PROC. WACV*, pp. 939–946, IEEE, 2018.

[43] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos, “Bridging the gap: Query by semantic example,” *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 923–938, 2007.

VITA

Ayesha Siddiqua

Candidate for the Degree of

Doctor of Philosophy

Dissertation: DEEP AUTOENCODERS FOR CROSS-MODAL RETRIEVAL

Major Field: Electrical Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2019.

Completed the requirements for the Master of Science in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in 2011.

Completed the requirements for the Bachelor of Science in Computer Science & Engineering at Shahjalal University of Science and Technology, Sylhet, Bangladesh in 2006.

Experience:

Graduate Research Assistant in Visual Computing and Image Processing Lab (VCIPL), School of Electrical and Computer Engineering, Oklahoma State University, January 2009 - May 2019

Professional Memberships:

IEEE Student member