university of groningen

faculty of science and engineering

Enhancing depth estimation for Transparent objects

Abhishek Ramanathapura Satyanarayana

university of groningen

faculty of science and engineering

University of Groningen

Enhancing depth estimation for Transparent objects

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Artificial Intelligence
at the University of Groningen under the supervision of
Prof. Dr. Hamidreza Kasaei (Artificial Intelligence, University of Groningen)
and
Prof. Dr. Matias Valdenegro Toro (Artificial Intelligence, University of Groningen)

Abhishek Ramanathapura Satyanarayana (s4304675)

September 30, 2023

Abstract

Transparent objects are ubiquitous both in household and industrial environments. The recognition of transparent objects in the environment is very important for the ultimate aim of grasping such objects by autonomous robotic systems. And, for this, a grasping robotic system will require high-quality RGB and depth images. However, the popular RGB-D cameras cannot provide accurate depth images for transparent objects due to their nature. In this research, we propose a novel architecture ResNet-50 + PSA model that can be used with the clearGrasp pipeline to estimate enhanced depth images from just the RGB image, especially for transparent objects. In the clearGrasp pipeline, the depth estimation task is divided into sub-tasks of transparent object segmentation, occlusion boundary segmentation, and surface normal estimation. The experiments demonstrate that the proposed ResNet-50 + PSA model is better than the DRN model as it achieved better performance on every sub-task as well as the final depth estimation task. It was also demonstrated that the clearGrasp pipeline with ResNet-50 + PSA model has better generalization on real-world novel objects.

Acknowledgments

I would like to express my sincere gratitude to my first supervisor Dr. Hamidreza Kasaei, for his guidance and support during this work. The ideas and feedback provided by him during our meetings were invaluable and helped to improve this research. I also would like to express my special thanks to my second supervisor, Dr. Matias Valdenegro Toro for devoting his valuable time and providing valuable feedback during this thesis work.

I thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high-performance computing cluster.

Finally and most importantly, I would like to thank my parents, brother, and friends for their long-standing and unconditional support, without them this journey would not have been possible.

Contents

List of Tables

List of Figures

1 Introduction

Grasping objects is an essential task that humans perform on a daily basis. These objects can range from household objects like a bottle, cup, spoon, etc. to industrial objects like test tubes, tools, etc. The task of grasping can be tedious and repetitive depending on the environment. To realize robotic grasping systems, it is necessary to automate the grasping task, that can handle a wide variety of objects. In the industry, traditional robots are usually used in factory assembly lines [4]. Such robots have a limitation in the kind of objects that they can grasp meaning they can be pre-programmed to grasp certain known objects. However, in household scenarios, the role of autonomous service robots becomes extremely important if they are able to grasp novel objects since they can encounter any kind of new object that they might have not seen before [5]. The applications of Artificial Intelligence, i.e. the machine learning and deep learning paradigms can form the core of such autonomous robotic grasping systems. The success of such grasping systems will have wider implications for the fourth industrial revolution as well as household automation through service robots. So, in recent years, there has been a growing interest in open-ended learning for robotics [6].

For any robotic grasping system, the camera sensor plays an important role in helping the system to perceive the environment. Normal cameras can only capture RGB images that can provide descriptive information about the objects. However, special cameras such as RGB-D cameras can capture both RGB and depth images. The depth images provide 3D information about the objects in the environment. Among the various types of objects that can be found in different environments, transparent objects are a unique category of objects that may be harder to grasp, when compared to normal objects. Transparent objects can be easily found in household kitchens, laboratories, restaurants, among other places. For such objects, any method that directly utilizes the RGB and depth images from a camera sensor may not suffice. This is because of the fact that the depth images for real-world transparent objects from any camera would be inaccurate when compared with that of normal objects. This is due to the nature of the transparent objects and the principle that the depth cameras work on i.e. the surfaces of all the objects would reflect light uniformly in all directions, which does not hold true for transparent objects [7]. Figure 1 shows the robotic grasping system with two transparent bottles. From the depth image in Figure 1, it can be observed that the depth values for the transparent areas of the two objects are inaccurate. The raw depth images of transparent objects from any depth camera would be invalid or may contain noise. So, there is a need for a method to estimate the depth image of transparent objects as accurately as possible, to assist a system in determining the shape of the transparent objects. The RGB images of transparent objects can give some information to a robotic system regarding the objects. Although the RGB images are useful, they alone are not sufficient to automate the task of grasping transparent objects, where perceiving the environment plays a prominent role. So, methods involving the usage of both RGB images and estimated depth images are needed for any success in developing methods for grasping transparent objects. Many researchers have proposed various grasp pose prediction models that take RGB and depth images as input, from an RGB-D camera, for predicting grasp pose maps that can then be used for the task of grasping. There has been significant progress that has been made in this regard for general objects such as GRConvNet [8], GGCNN [9] etc. Such methods' direct application for transparent objects cannot be realized since the depth images from the RGB-D camera would be noisy and inaccurate. So, it is evident that depth estimation is crucial and prominent for any method that will involve the grasping of transparent objects. With the rapid progress of artificial intelligence, machine learning, and deep learning in the last decade; large amounts of labeled datasets can be utilized to train large models that can outperform methods from traditional techniques on several tasks. A prominent example of the image classification task is the ImageNet dataset [10] on which various deep learning models have been benchmarked. Over the years a lot of progress has been made in the object recognition domain.

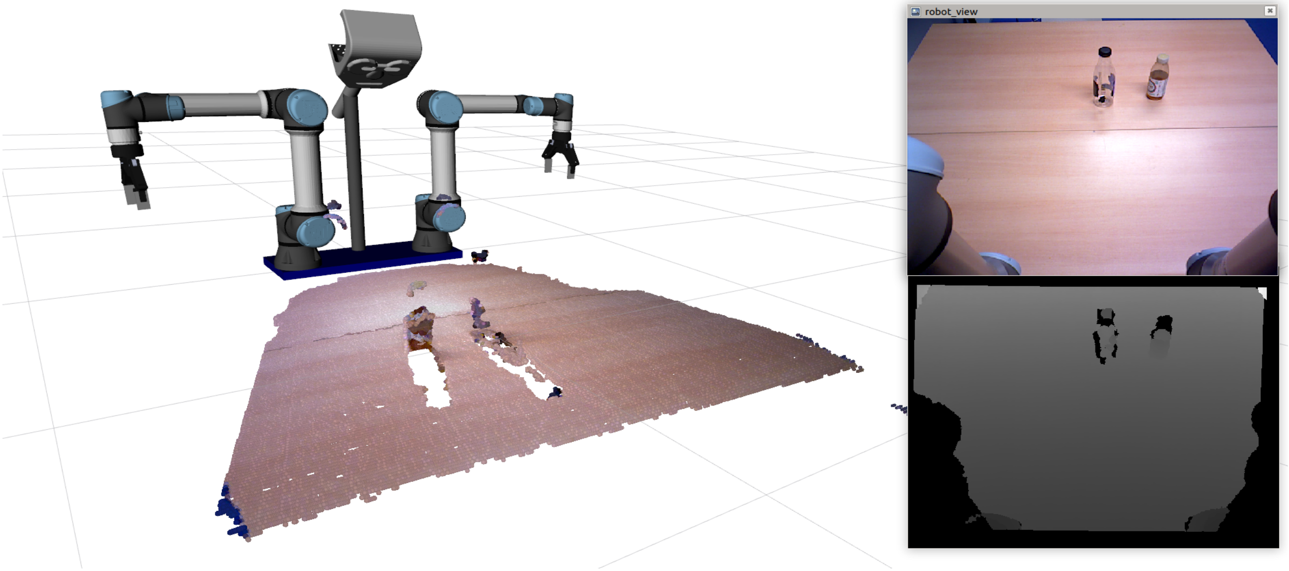


Figure 1: The figure shows a robotic grasping setup with 2 transparent bottles.

1.1 Scope of this thesis

In this research, we adopt the clearGrasp pipeline [3] (proposed by S. Sajjan et al.) to estimate the depth of transparent objects. In their pipeline, the depth estimation task is divided into the following sub-tasks — transparent object segmentation, occlusion boundary prediction, and surface normal estimation. The predictions of these three models will be used in a global optimization method to estimate the depth values of the transparent objects. Since the task of recognition of transparent objects is a difficult task, we propose a novel architecture with an attention module that has the potential to improve performance. In our research, we propose a novel encoder-decoder architecture by using the Polarized Self Attention module [11] (proposed by H. Liu et al.) with some modifications to ResNet-50 encoder [12] (proposed by K. He et al.) along with minor modifications to DeepLab_v3+ decoder [13] (proposed by L. Chen et al.). We conducted several experiments for the individual sub-tasks as well as the final task of depth estimation. In this research work, the clearGrasp dataset [3] along with some samples from the COCO dataset [14] for training the object segmentation model, some samples from the NYU_v2 dataset [15] for training the surface normal estimation model. The evaluation of our proposed model was performed and compared with the DRN model (proposed by S. Sajjan et al.) for the individual sub-tasks and the final task of depth estimation using these models in the clearGrasp pipeline using the clearGrasp dataset. In our research, for the depth estimation task, an additional small real-world test dataset called `IRL-transparent-objects-set` was collected to evaluate the depth estimation performance.

1.2 Research Questions

To summarize, this thesis focuses on the following problems:

Q1. How can transparent objects be detected and localized in the images?

Q2. Is it possible to train models to estimate refined depth values that can be used for grasp planning? How will they perform compared with the current state-of-the-art?

Q3. How can the performance of depth estimation for transparent objects pipeline be evaluated and compared and what are its main challenges?

1.3 Thesis Outline

The rest of this thesis report is further divided into different chapters. In section 2, the theoretical background related to our research work is discussed. In this section, the literature related to learning from synthetic data, object recognition, transparent object recognition, and depth estimation is summarized. In section 3, our proposed model architecture is discussed in detail along with related relevant details of the methods used in our research. In section 4, the experimental setup and a sample real-world test dataset collection details are discussed. In section 5, the results obtained with our proposed model architecture are discussed in detail. In section 6, a discussion about our conclusions from the experiments along with the shortcomings of our proposed method and possible directions for future work are presented.

2 Theoretical Background

In this chapter, a brief introduction related to the theoretical concepts related to this thesis work is discussed in section 2.1, section 2.2, and section 2.3. In addition to the theoretical concepts, a literature review of learning from synthetic data and transparent object recognition is discussed in section 2.4.1, and section 2.4.2.

2.1 Theory of artificial neural network

An artificial neural network also known as the Multi-Layer Perceptron (MLP) [16] takes input as a certain feature vector to predict a certain output feature vector for a certain task for which the multi-layer perceptron is trained. For training such a network, a differentiable loss function is minimized which measures the error in the predictions by the network. The network learns the weights of the artificial neurons in the network by a method known as gradient descent [17]. The gradient, which contains the partial derivatives of the loss with respect to the weights, is a crucial component that is required to apply any gradient descent algorithm to update the weights. As the complexity of the network increases, the gradients are complex to compute. Backpropagation algorithm [18] can be used to efficiently compute the gradients. Although the multi-layer perceptron can approximate complex functions, it can be hard to use them with images as input. A typical image is of the form $H \times W \times C$ (height, width, channels) in dimensions. If an image needs to be input to an MLP, then the image needs to be fed as a single vector of dimension $H \times W \times C$. This will cause a rapid increase in the number of parameters while dealing with high-dimensional image data. Another problem is that the spatial information in an image is lost if it is fed as a single vector to an MLP as an input. To overcome such problems that are common for high-dimensional image data, a popular type of network also known as a Convolutional Neural Network is usually used, which is discussed in section 2.2.

2.2 Convolutional Neural Network

The Convolutional Neural Network (CNN) [19] is a type of artificial neural network that can work better with images as input. Yann LeCun et al. demonstrated the ability of CNN on the handwritten digit classification task with grayscale images [19]. Although the CNN was proposed much earlier, it became extremely popular with AlexNet [20] which was the state-of-the-art model on the ImageNet [21] large-scale object recognition challenge, when it was introduced. AlexNet [20] outperformed all the other traditional machine learning models including the second-best method that used a method of averaging the predictions of several classifiers trained on Fisher Vectors [22] computed from different types of densely sampled features from SIFT algorithm [23] by a very large margin. This was when researchers began to explore the area of deep convolutional neural networks to understand their potential for computer vision tasks. CNNs are totally connected feed-forward networks that efficiently reduce the number of parameters of a model without losing out on the quality of models. The core idea behind a CNN is to learn some feature mapping of an image through learning spatial features and utilize them to represent images with the learned spatial feature mappings.

The convolution layer is the most important component of a CNN among other layers such as pooling layers, activation functions, transposed convolution, batch normalization, dropout layer, fully connected layer (or the usual dense layer of neurons same as in MLP), etc. Every layer is used for a specific functionality that is applied to the intermediate features. Such layers are usually stacked multiple times to make the network as deep as possible like in the case of a popular network named VGG

[24]. Figure 2 shows the network architecture of a popular CNN, VGG-16 network. The choice of the layers to construct a CNN depends usually on the task for which the CNN is being used. The convolution layer consists of multiple filters that are learnable parameters. During the forward pass, the convolution operation [25] is the dot product of the filter and input image pixels that will produce some intermediate features of the input image. Every filter in each convolution layer may learn to extract different image features depending on the task for which the CNN is used. This was demonstrated by Zeiler et al. in their research [26]. The pooling layers [27] are usually used to reduce the spatial dimensions of an input image feature. The activation function is usually used to introduce non-linearity since the task of solving any modern computer vision task such as image classification, object detection, semantic segmentation, etc. is highly non-linear in nature [28]. Researchers have proposed different types of activation functions and their usage depends on the application of the neural network to various data domains. A recent study was done by Dubey et al. to study the effects of using different activation functions [29]. The transposed convolution layer is usually used to increase the spatial dimensions of an input image feature. This is widely used in the semantic segmentation task such as in Fully Convolutional Networks (FCNs) [30]. In recent research, the bilinear interpolation method has replaced the transposed convolution layer. The dropout layer [31] proposed by Srivastava et al. is usually used for regularization, i.e. it can help in reducing overfitting and generalizing better on new data. The batch normalization layer [32] is an idea to reduce covariance shift during training that is usually used for speeding up the training of the network and in some cases it can also act as a regularization method.

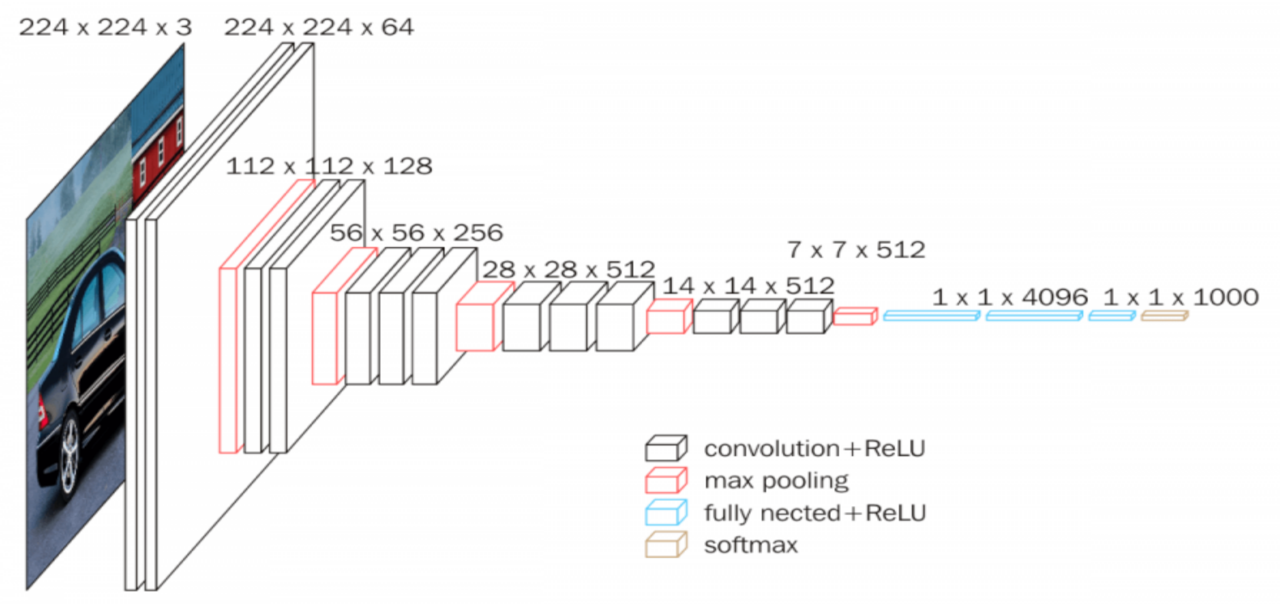


Figure 2: The network architecture of the popular CNN, VGG-16 network from the article [1].

2.3 Autoencoder

An autoencoder [33] is a model that is usually used to learn an unsupervised representation of the input data. It performs regression by predicting a reconstruction of the input data. It consists of two internal modules, the encoder taking the input (x) into a feature space ($h = f(x)$) and a decoder transforming the feature space into the reconstructed output ($\hat{x} = g(h)$) i.e. the autoencoder's output. The autoencoders can consist of fully connected layers or convolution layers depending upon the type of input data. The performance of the autoencoder is evaluated by measuring the reconstruction loss, which measures the difference between the original input and the reconstructed output. The autoencoder model is trained to minimize the reconstruction loss. For training an autoecnoder, one of the standard optimizers can be used. Figure 3 shows the network architecture of a convolutional autoencoder applied for saliency detection.

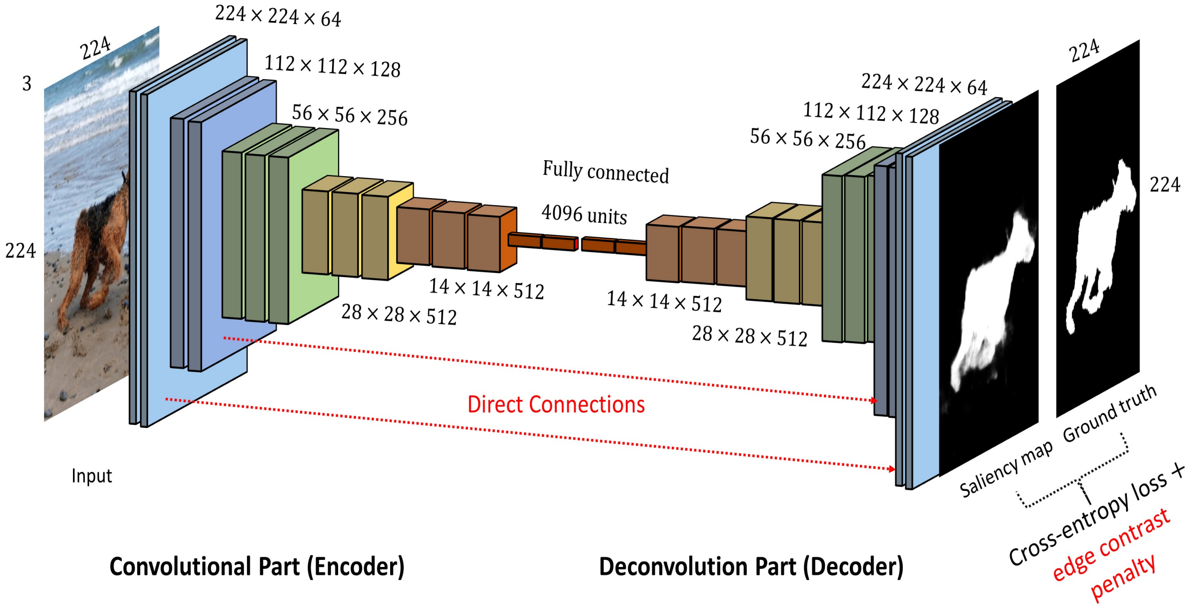


Figure 3: The network architecture of the convolutional autoencoder applied for saliency detection from the GitHub repo article [2].

2.4 Literature review

2.4.1 Learning from synthetic data

Curating large datasets is essential for the success of any deep learning algorithm. However, it is time-consuming and expensive to curate real-world datasets. A popular example of such a large dataset is the ImageNet [10] dataset with millions of images across thousands of object categories for the image classification task. A few years later, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [34] was started by expanding the tasks to include single object localization and object detection along with the already existing task of image classification. For the ILSVRC challenge, the existing ImageNet dataset was expanded with millions of images that were annotated across thousands of categories for additional tasks. For curating the datasets for this challenge, the researchers crowd-sourced the annotations which was one of the biggest challenges in creating the datasets. Another challenge that the researchers faced was the verification of the crowd-sourced annotations for which multiple workers were hired for the task of annotation verification. And, these are extremely expensive with respect to the time and resources that they consume.

An alternative is to use large synthetic datasets to train models and evaluate their performance on relatively smaller real-world datasets. Synthetic datasets are cheap to generate when compared with that of the real-world datasets. Complex computer software can be used to generate large annotated datasets. However, synthetic datasets must be used with caution as they have their own intricacies.

The synthetic datasets are not realistic in nature and can come with their own challenges of limited object models, lighting, texture among others. Previously, there have been various works where researchers have made use of synthetic datasets for various tasks with decent performance generalization on real-world data. Yuhua Chen et al. showed that domain adaptation from synthetic to real datasets for the object segmentation models performed well on the real datasets in their work [35]. In their research, they proposed two ideas to reduce the domain shift — i) on the input level, they augmented the traditional image translation network with the additional geometric information to translate synthetic images into realistic style images, and ii) on the output level, they developed a task network which simultaneously performed depth estimation and semantic segmentation on the synthetic data. Konstantinos Rematas et al. developed a CNN model [36] for depth estimation models where they trained their model on the 3D player data extracted from soccer video games. They trained and evaluated their model on both the synthetic and the real-world benchmark datasets and their model performed better than the then state-of-the-art models. Jeffrey Mahler et al. showed that their grasp planning model - Grasp Quality CNN (GQCNN) [37] trained on the synthetically generated DexNet-2.0 dataset performed well on both the known and novel rigid objects. Generating annotations for the point clouds, grasps, and analytic grasp metrics for the grasp planning task would increase the cost of time and resources required. In their research, they used a synthetically generated dataset that had 6.7 million samples and their main goal was to reduce the time to train deep learning models that require high-quality annotated large datasets. Kasaei et al. proposed MVGrasp [38], a real-time 3D object grasping method for highly cluttered environments. In their research, they trained their model on a synthetic dataset and demonstrated that it could perform well on both the synthetic and real-world datasets without further finetuning. Their method outperformed the then state-of-the-art model on both synthetic and real-world datasets in isolated and packed object scenarios.

2.4.2 Transparent object recognition

There are several standard benchmarks for general object recognition tasks. The researchers at Oxford released the Pascal VOC dataset [39] which was a large dataset with five challenges — classification, detection, segmentation, action classification, and person layout. The intended audience of the dataset was the algorithm designers, researchers who could compare their models with the state of the art by measuring the performance of their models on the VOC datasets. Another audience was the researchers who could suggest the limitations and weak points of the current generation of algorithms using the dataset as a benchmark. Later, the COCO dataset [14] was released, which has been widely used as a benchmark for common object recognition in the context of scene understanding. In this dataset, there were annotations for additional benchmark tasks such as instance segmentation, keypoint detection, and pose estimation among others. More recently, several such benchmark datasets have been released for general object recognition. This makes the lives of researchers and algorithm developers a bit easier who want to develop different models and benchmark their model performance on normal object recognition tasks.

However, the task of recognizing transparent objects is not a straightforward task like normal object recognition and it poses inherent challenges due to the nature of the transparent objects. Another reason is the lack of prominent benchmark datasets explicitly for transparent object recognition. However, there have been various research works that focused on the recognition of transparent objects in the last few years. Several methods have been proposed by various researchers for certain challenging tasks such as transparent object segmentation. Enze Xie et al. proposed the Trans10K dataset [40] consisting of tens of thousands of annotated samples of transparent objects for the segmentation task.

In their research, they proposed a boundary-aware transparent object segmentation method known as TransLab that exploits boundaries as a clue to improve the segmentation of transparent objects. However, their dataset consisted not only of household transparent objects but also glass windows, doors, and other large glass surfaces. Since it is challenging to detect glass surfaces, Haiyang Mei et al. proposed a large-scale glass detection dataset (GDD) [41]. They also developed a network called the GDNet to detect glass surfaces by learning abundant contextual features from a global perspective with a novel large-field contextual feature integration module. They demonstrated that their model outperformed other methods on the GDD test set. Haiyang Mei et al. further proposed RGBP-Glass dataset [42] and a novel architecture that could utilize trichromatic (RGB) intensities as well as trichromatic linear polarization cues from a single image for glass segmentation. They demonstrated that their network outperformed the other state-of-the-art models. Although most of these methods deal with glass surfaces, they can be adapted to other household transparent objects which are usually smaller than large glass surfaces making it even more challenging.

There have also been several methods proposed by researchers to estimate the geometry (or depth values or shape) of transparent objects. This is essential because the depth values from any RGB-D camera will be inaccurate because of the nature of the transparent objects. Estimating the depth values of transparent objects itself is a challenging task but it becomes a prerequisite for any transparent object grasp planning algorithm. Previously, several methods have been proposed for pose estimation [43, 44] with known 3D models of transparent objects. Recently, Dex-NeRF [45], a method with the neural radiance field was proposed by Jeffrey Ichnowski et al. to estimate depth values that can then be utilized to plan grasps of transparent objects. A major disadvantage of the NeRF is that it can model a single scene at a time and is very expensive to train. A major disadvantage of the previously mentioned pose estimation methods [43, 44] and the depth estimation method Dex-NeRF [45] is the generalization issue i.e. they can be easily applied to objects that the method was trained on, i.e. the objects that are already seen by the model, and they cannot be applied to novel objects, i.e. the unseen during training. Sajjan et al. proposed the clearGrasp pipeline [3] with the DRN network that can overcome some of the issues mentioned previously in estimating depth values of transparent objects. To evaluate the performance of their method, they proposed the clearGrasp dataset that consisted of large synthetic and relatively smaller real-world datasets. In their pipeline, the task of depth estimation was divided into sub-tasks — transparent object segmentation, object boundary segmentation, and surface normal estimation. Their proposed DRN network is an encoder-decoder style network to estimate the transparent object segmentation mask, object boundary segmentation, and surface normals that can then be utilized with a global optimization function for estimating the depth values for transparent objects. In their research, the DRN network was trained on the synthetic datasets and the performance was evaluated both on synthetic and real-world test datasets with a very good performance. Their method outperformed the then state-of-the-art methods. One of the main advantages of their pipeline is the generalizability on novel unseen objects, unlike the model-specific methods such as Dex-Nerf. In our research, we make use of the clearGrasp pipeline for estimating the depth values of the transparent objects. However, we propose a novel network architecture and compare its performance with the DRN network as the baseline. For training the networks, the clearGrasp dataset was used. For performance evaluation, the clearGrasp dataset and an additional real-world test set collected by us were used.

3 Methods

In this chapter, the research methods are discussed in detail. The problem statement is discussed in detail in section 3.1. A high-level overview of the datasets used in this research is discussed in section 3.2. The complete details of the network architecture used in this research are discussed in section 3.3. The details of the gradient descent-based optimization algorithms that were used in this research are discussed in section 3.4. The details of the various loss functions that were used in this research are discussed in detail in section 3.5. The details of the global optimization method that was used for the estimation of the depth values using the predictions of the individual tasks are discussed in section 3.6. Finally, the evaluation metrics used to evaluate the performance of various methods for different tasks are discussed in section 3.7.

3.1 Problem statement

The main task of depth estimation was broken down into several sub-tasks. The following were the sub-tasks — transparent object segmentation, occlusion boundary prediction, surface normal estimation, and depth estimation. The transparent object segmentation task was modeled as a pixel-wise segmentation task with just 2 classes, where each pixel belongs either to a transparent object or the background. The occlusion boundary prediction task was modeled as a pixel-wise segmentation task with 3 classes where each pixel belongs to one of the following classes — contact edge, occlusion edge, or none of those two classes. The surface normal estimation task was modeled as a pixel-wise regression task. Finally, a global optimization method i.e. the clearGrasp pipeline was used in which the predictions of the three models were used for estimating the depth values of the transparent objects. For the individual sub-tasks, a novel network architecture — ResNet-50 + PSA is proposed by us. In this research, the performance of the proposed network architecture is compared with the current state-of-the-art network DRN model for the depth estimation task for transparent objects using the clearGrasp pipeline.

3.2 Dataset

The clearGrasp dataset [3], curated by Shreeyak Sajjan et al., was the main dataset used in this research. The dataset contains labeled data for transparent objects for object segmentation, object occlusion boundaries, surface normals, and depth estimation. The dataset contains both synthetic and real-world samples. For training, only synthetic samples were used. The real-world samples were used only for the evaluation.

The following are the details of the clearGrasp dataset. The synthetic training set contained 45454 training samples belonging to 5 different objects. The synthetic validation set contained 532 validation samples belonging to the same objects as in the training set. The synthetic test set contained 408 test samples belonging to 4 novel objects unseen during training or validation. Figure 4 shows the distribution of the number of samples for the objects in the synthetic train and validation sets. Figure 5 shows the distribution of the number of samples for the objects in the synthetic train and validation sets. However, these distributions might not give the complete picture since a single sample image could contain a single object or multiple objects of the same kind. The real-world test set contained 286 test samples belonging to multiple transparent and non-transparent objects.

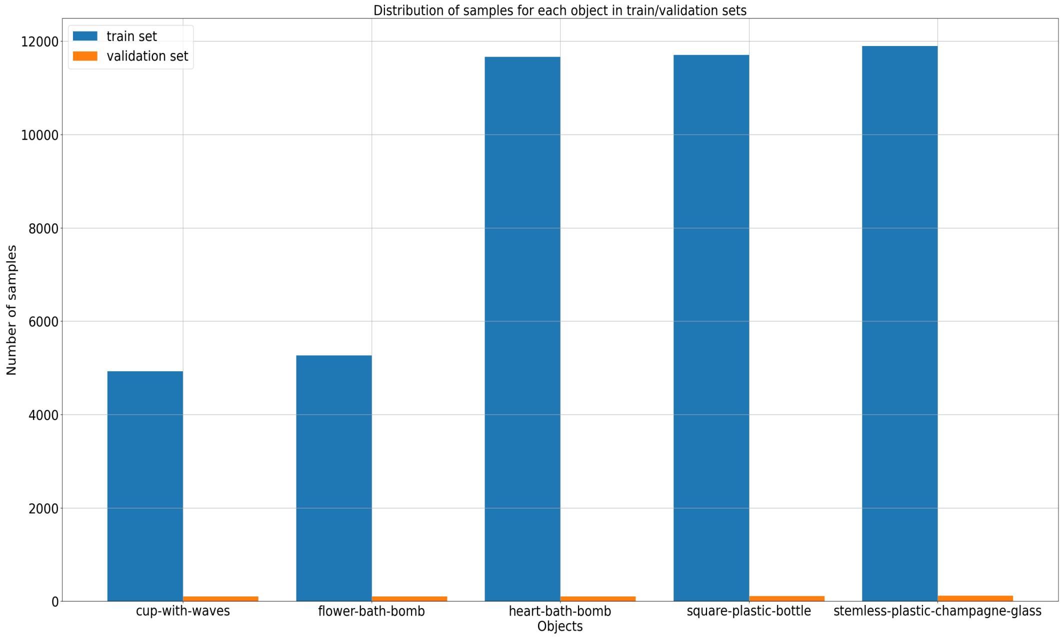


Figure 4: Distribution of the number of sample images for every object in the clearGrasp synthetic train and validation sets.

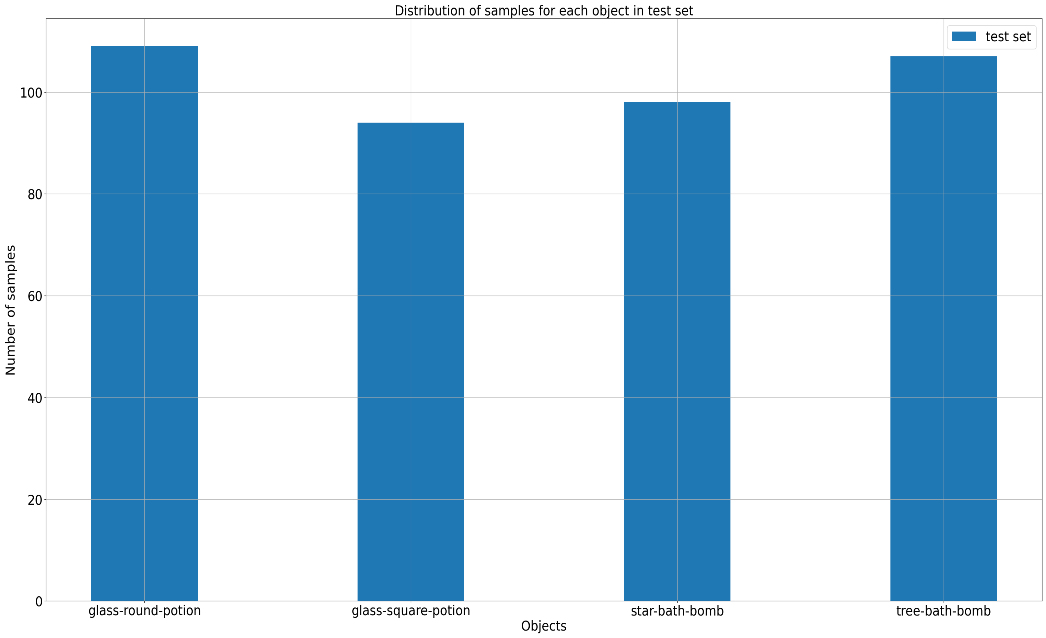


Figure 5: Distribution of the number of sample images for every object in the clearGrasp synthetic test set.

Some additional datasets were also for finetuning the models. For the transparent object segmentation task, 200 randomly selected samples from the COCO dataset [14] were used to include some non-transparent objects since the clearGrasp synthetic train set had only image samples with a high majority of transparent objects. The most prominent non-transparent object was the tray in which the transparent objects were placed. For the surface normal estimation task, 1449 samples from the NYU_v2 dataset [15] were used to include more non-transparent objects.

Additionally for this research, a small real-world test dataset called `IRL-transparent-objects-set` with 87 samples was collected by us with `Kinect-v1` RGB-D camera and will be made public. The details of the additional test dataset collection are discussed in detail in section 4.2.

3.3 Network architecture

The network is an autoencoder-style network with an encoder and a decoder. Since it is used in a supervised learning setting, i.e. to use the decoder output further for solving some other task, there might be additional layers after the decoder depending on the task for which it is being used.

For the encoder network, the ResNet style network was used with modifications. The Residual Networks (ResNets) [12], a popular CNN with residual connections, was proposed by Kaiming He et al. to overcome the vanishing gradients problem that is common in normal deep CNNs such as VGG networks [24], which was proposed by Karen Simonyan et al. For the individual tasks, the following architecture is proposed in this research. The ResNet-50 variant was used in this research. There were mainly two changes to the ResNet encoder network. The first modification was done by changing the output stride of the network. If an image is 224×224 in resolution, and is provided as the input to the normal ResNet encoder, then it is spatially reduced by a factor of 32 (also called the output stride), resulting in encoder features that are 7×7 in resolution. This output stride of the encoder was changed to 8 by changing the stride in two of the residual blocks from 2 to 1. So, if the input image is 224×224 in resolution, the resulting encoder features are spatially reduced by a factor of 8, to 28×28 in the modified ResNet. The second modification was done by adding an attention module to the main residual blocks.

The object segmentation and occlusion boundary tasks are modeled as segmentation mask prediction tasks whereas the surface normal prediction is modeled as a regression task. These tasks become difficult for the networks to learn since the networks are expected to estimate high non-linear pixel-wise semantic estimates. The main motivation for adding the attention module is to improve the capability of the encoder network's feature learning ability that might be needed for the individual tasks that can in turn result in better performance on the main task of depth estimation, especially for the transparent objects where these tasks are more difficult when compared with that of the normal objects. The attention module used for this research is the Polarized Self Attention (PSA) module [11], which was proposed by Huajun Liu et al. The authors showed that using such an attention module has certain benefits such as firstly, the Polarized filtering that helps in keeping high internal resolution in both channel and spatial attention; and secondly the Enhancement that helps in composing non-linearity that can directly fit the output distribution [11]. The authors showed that using such an attention module can boost the performance of the segmentation mask and pose estimation tasks (a regression task) [11]. Since the main task of depth estimation is divided into sub-tasks that involve both i) segmentation mask prediction for transparent object segmentation and boundary segmentation tasks and

ii) regression task for the surface normal estimation task, the PSA module was chosen as the attention module in the encoder.

For the decoder, the DeepLab-v3+ [13] style decoder network, which was proposed by Liang-Chieh Chen et al., was used. In the original research, the main idea was the use of the Atrous Spatial Pyramid Pooling (ASPP) module along with a convolution block to handle skip connections from the encoder network. In the ASPP module, the outputs of different convolution layers with different atrous (dilation) rates and the global average pooling layer were concatenated after which a final convolution layer was used. In this research, the ASPP block is unchanged and utilized as proposed in the original research. In the original research, the authors proposed a skip connection from the encoder to merge low-level features with the features from the ASPP module, which were then upsampled using bilinear interpolation to match the spatial dimension of the inputs. This is similar to the skip connection in U-Net [46] but only from one encoder layer whereas in U-Net there are skip connections from every encoder stage. In the original research, the features of the first convolution layer were used as low-level features for the skip connection. However, in this research, this was slightly modified to take the features after the first residual block with the proposed modifications as the low-level features. Another minor change that was introduced in the convolution block to handle skip connections from the encoder was to have two dropout layers [31], one with a dropout rate of 0.5 and another with 0.1.

The same network architecture was used for all the individual sub-tasks of transparent object segmentation, occlusion boundary segmentation, and surface normal estimation. However, the optimization algorithm and the loss functions used to train were different depending on the task and these details are discussed in section 3.4 and section 3.5. In this research, the performance of the proposed network architecture is compared with the current state-of-the-art network DRN model for the depth estimation task for transparent objects using the clearGrasp pipeline.

3.4 Optimization algorithm for updating weights

For training the models, the Stochastic Gradient Descent (SGD) [17] optimizer with momentum is a popular choice, i.e. to update the model's weights. The update rule for the weights using the SGD optimizer with momentum is given by Equation 1 where θ_{t+1} represents the weights of the model at time $(t+1)$, θ_t represents the weights of the model at time t, η_t is the learning rate at time t, $\nabla\mathcal{L}(\theta_t)$ represents the gradients of the loss function with respect to parameters θ_t, $\rho \in [0,1]$ is the momentum parameter, and the momentum term $(\Delta\theta_t)$ is given by Equation 2. The polynomial learning rate scheduler [47] was used along with the SGD optimizer with momentum. The learning rate update rule for the polynomial learning rate scheduler which updates the learning rate after every epoch, is given by Equation 3 where η_{t+1} is the learning rate at time $(t+1)$, η_t is the learning rate at time t, t is the current epoch, T_t is the total number of epochs, and *power* parameter controls the learning rate decay.

$$\theta_{t+1} = \theta_t - \eta_t(\nabla\mathcal{L}(\theta_t)) + \rho\Delta\theta_t \tag{1}$$

$$\Delta\theta_t = \theta_t - \theta_{t-1} = \rho\Delta\theta_{t-1} - \eta_{t-1}(\nabla\mathcal{L}(\theta_{t-1})) \tag{2}$$

$$\eta_{t+1} = \eta_t \times (1 - \frac{t}{T_t})^{power} \tag{3}$$

Another popular choice of optimizer is the Adaptive Moments (Adam) [48] optimizer proposed by Kingma and Ba et al. The exponentially weighted running averages can be biased. The Adam optimization algorithm corrects the biases in the exponentially weighted running averages. It uses both the first and the second moments of the gradients. This optimization algorithm is preferred over others for its capability to ensure stable and faster convergence in certain cases depending on the model architecture and the task where it is being applied. The update rule for the Adam optimizer is given by Equation 4 where g_t is the gradient at time t, s_t and r_t are the first and second moments respectively, β_1 and β_2 are the coefficients used in computing running averages of the gradient and the square of the gradient.

$$\begin{aligned} s_t &= \beta_1 s_{t-1} + (1-\beta_1) g_t \\ r_t &= \beta_2 r_{t-1} + (1-\beta_2) g_t \odot g_t \\ \theta_{t+1} &= \theta_t - \eta_t \frac{s_t}{\varepsilon + \sqrt{r_t}} \frac{1-\beta_2^t}{1-\beta_1^t} \end{aligned} \tag{4}$$

However, there are challenges in using the Adam optimizer along with weight decay. Loshchilov et al. proposed the AdamW [49] optimizer that decoupled the weight decay from the optimization steps taken with respect to the loss function. In this research, it was shown with empirical evidence that the proposed modification substantially improved Adam's generalization performance.

3.5 Loss functions for individual tasks

For the object segmentation task, the categorical cross-entropy loss function [50] was used. The categorical cross-entropy loss function is given by Equation 5 where $\mathcal{L}_{CE}(\theta)$ denotes the loss, c_i denotes the encoded class and p_i denotes the probability of the class i as predicted by the model, for every one of the n classes in the dataset.

$$\mathcal{L}_{CE}(\theta) = -\sum_{i=1}^{n} c_i \log(p_i) \tag{5}$$

For the occlusion boundary prediction task, the focal loss [51] was used. The focal loss is given by Equation 6 where $\alpha_F \in [0,1]$ is the focus loss weighting factor to handle class imbalance and γ_F is the focusing parameter. The focal loss was used instead of the usual categorical cross-entropy loss since the task has 3 classes with a very low percentage of boundary pixels when compared with that of non-boundary pixels. The classes were also weighted accordingly to handle their share of representation. The occlusion edge class was given a weighting of 5× whereas the contact edge was given a weighting of 3× and the non-boundary class was given a weighting of 1× in the usual cross-entropy weighting which is not to be confused with the focal loss weighting factor α_F.

$$\mathcal{L}_F(\theta) = -\alpha_F((1-\exp(-\mathcal{L}_{CE}(\theta)))^{\gamma_F}(-\mathcal{L}_{CE}(\theta)) \tag{6}$$

For the surface normal estimation task, the difference of angle in radians loss function based on cosine similarity [52] was used since the task is a regression task. The cosine similarity is given by Equation 7 where g is the groundtruth and p is the prediction. The corresponding loss function based on the cosine similarity is given by Equation 8. The ε_c was set to 1×10^{-6}.

$$CS(g,p) = \frac{g \cdot p}{\max(\|g\|_2 \cdot \|p\|_2, \varepsilon_c)} \tag{7}$$

$$\mathcal{L}_{cosine}(\theta) = \arccos CS(g, p) \tag{8}$$

3.6 Pipeline to estimate depth using global optimization

The depth estimation pipeline proposed by Zhang et al. was modified slightly by Sajjan et al. in their clearGrasp pipeline, which was used in this research [53, 3]. Figure 6 shows the clearGrasp pipeline for estimating the depth of transparent objects. In the original deep-depth estimation pipeline the depth values of the missing pixels are estimated. However, Sajjan et al. modified the pipeline slightly to include the following. Firstly, to train a network to predict masks for transparent object surfaces that can be used to remove the unreliable depth from cameras for transparent object surfaces. Secondly, to predict both the occlusion and contact edges i.e. different types of edges that can aid in accurately predicting depth discontinuity boundaries.

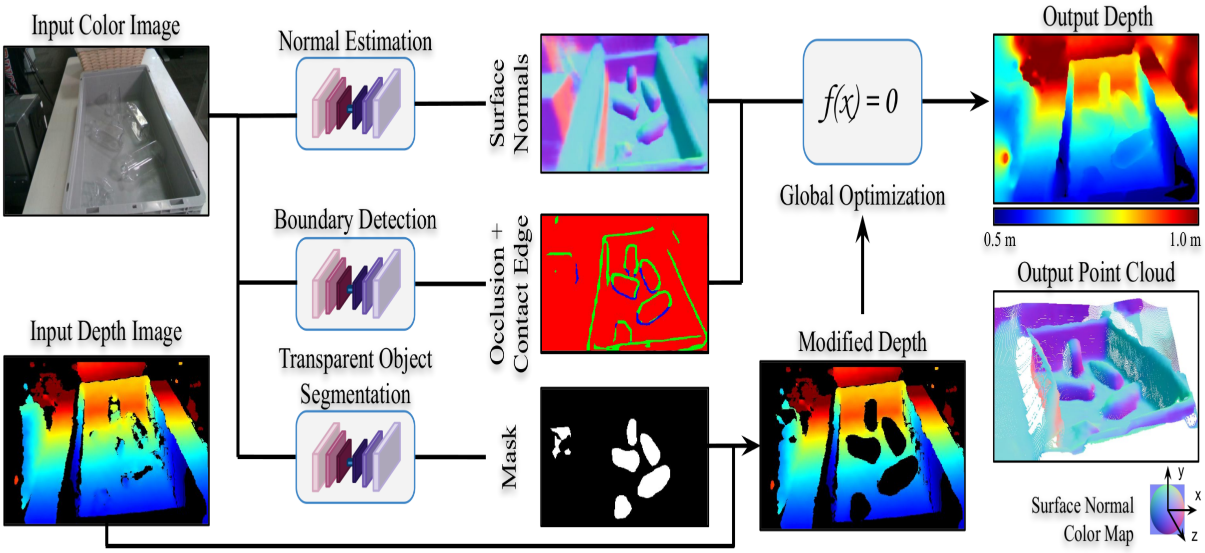


Figure 6: The clearGrasp depth estimation pipeline [3] proposed by S. Sajjan et al.

The depth prediction task will use the global optimization method proposed by Zhang et al. that will utilize the predictions from the three tasks namely — object segmentation, occlusion boundary prediction, and surface normal estimation [53]. The transparent object segmentation model's predictions are used to remove unreliable and noisy depth values of transparent objects from the RGB-D camera. The optimization algorithm estimates the depth values using the estimated surface normals to guide the shape of the reconstruction while maintaining depth discontinuities using the estimated occlusion boundaries. The optimization algorithm tries to solve a system of equations while minimizing the weighted sum of squared errors as shown in Equation 9, where E_D measures the error between estimated depth and observed raw depth, E_S measures the difference in the depths of neighboring pixels, E_N measures the consistency between estimated depth and estimated surface normal, and B is used to weight the normal terms based on the predicted probability that a pixel is on an occlusion boundary.

$$E = \lambda_D E_D + \lambda_S E_S + \lambda_N E_N B \tag{9}$$

3.7 Evaluation Metric

For the evaluation, various metrics were used for both the individual sub-tasks and the final task of depth estimation. For the object segmentation and occlusion boundary prediction tasks, the Intersection over Union (IoU) metric [54] was used. The IoU for a class is given by Equation 10 where IoU_c denotes the IoU for a class label c, TP denotes the number of true positive pixels, FP denotes the number of false positive pixels, and FN denotes the number of false negative pixels; for that given class. The m-IoU is nothing but the mean of the IoU of all the classes in any given dataset. It is given by Equation 11 for n classes.

$$IOU_c = \frac{TP}{(TP+FP+FN)} \tag{10}$$

$$m_{IOU} = \frac{1}{n}\sum_{i}^{n} IOU_{c_i} \tag{11}$$

For the surface normal estimation task, mean and median errors in degrees along with the percentage of pixels with estimated surface normals where the difference with groundtruth is less than the threshold of 11.25, 22.5, and 30 degrees were the metrics used [55, 56]. The mean error in degrees is given by Equation 12 where H is the image height, W is the image width, g_{ij} is the groundtruth for pixel at i$^{\text{th}}$ row and j$^{\text{th}}$ column, and p_{ij} is the model's prediction for that corresponding pixel. Similarly, the median error in degrees is given by Equation 13 where $i \in 1,2,...H$ and $j \in 1,2,...W$.

$$mean_{deg} = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{\arccos(CS(g_{ij},p_{ij})) \times 180}{\pi} \tag{12}$$

$$\begin{aligned} median_{deg} = median(&\frac{\arccos(CS(g_{ij},p_{ij})) \times 180}{\pi}) \\ &\forall i = 1,2,....,H; j = 1,2,....,W \end{aligned} \tag{13}$$

For the depth estimation task, the following metrics were used — root mean square error (RMSE) [57], mean error relative to depth (MeanREL) [58], mean absolute error (MAE) [57], and percentages of pixels with predicted depths within a certain interval ($\delta = |predicted - true|/true$, where $\delta \in \{1.05, 1.10, 1.25\}$). The RMSE is given by Equation 14 and the MAE is given by Equation 15 where g_i is the i$^{\text{th}}$ groundtruth and p_i is the i$^{\text{th}}$ prediction, for a total of N observations. The MeanREL is computed by computing the mean of the absolute relative errors of N observations. The MeanREL is given by the Equation 16 where g_i is the i$^{\text{th}}$ groundtruth and p_i is the i$^{\text{th}}$ prediction.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(g_i - p_i)^2} \tag{14}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|g_i - p_i| \tag{15}$$

$$MeanREL = \sum_{i=1}^{N}|\frac{g_i - p_i}{g_i}| \tag{16}$$

4 Experimental Setup

In this chapter, the details of the experimental setup are discussed in detail. For the research, programming was done in Python using the open source packages. For deep learning, the `PyTorch` package [59] was mainly used. The other standard packages include `numpy`, `scipy`, `scikit-learn`, `imageio` etc.

For training the different models, the `Hábrók` compute High-Performance Computing (HPC) cluster was used. The training was performed on powerful GPUs available in the `Hábrók` cluster. However, for the evaluation, the Dell laptop with the `Ubuntu 20.04.6 LTS` operating system (OS) having an `Nvidia GeForce GTX 1060 Mobile` GPU was used.

In section 4.1, the characteristics of the clearGrasp dataset are discussed in detail. For the additional test data collection, the `Kinect-v1` RGB-D camera was used with a tripod setup. The details of the additional test data collection procedure are discussed in section 4.2. The various preprocessing steps used in this research are discussed in detail in section 4.3. The details of the hyperparameter selection and optimization are discussed in detail in section 4.4.

4.1 ClearGrasp Dataset

As mentioned earlier, for the experiments in this research, the clearGrasp dataset was used [3]. Figure 7 shows a synthetic image sample from the clearGrasp validation set. Figure 8 shows the visualization of the label for the transparent object segmentation mask for the sample image shown in Figure 7. In the segmentation mask image, the green pixels correspond to the transparent objects and the red pixels correspond to the background i.e. any object or surface that is not transparent. Figure 9 shows the visualization of the labels for the outline segmentation mask. In the outline segmentation mask, the green pixels indicate the depth outlines (normal edges), the blue pixels indicate the occlusion outlines (contact edges), and the red pixels indicate the background. Figure 10 shows the visualization of the label for the surface normal.

For the transparent object segmentation task, one of the main disadvantages of the clearGrasp synthetic dataset was that the dataset contained only transparent objects and their groundtruth labels without any non-transparent object samples. Using only this would limit the model's learning ability and may result in more false positives. To overcome this issue, samples from the real validation set were also added to the standalone synthetic train set and the models were finetuned. The results of both these experiments are discussed in detail in section 5.2.1.

Additionally, more data samples were collected with a `Kinect-v1` RGB-D camera to create our own test set and the procedure followed for the data collection is also discussed further.

4.2 Additional test dataset collection

For this research, an additional small test dataset was collected by us with a `Kinect-v1` RGB-D camera. A total of 87 samples were collected for this test dataset called `IRL-transparent-objects-set`. The transparent object segmentation masks were annotated manually. A tripod setup was used to collect the data. There were two modes of data collection and 4 objects were considered. Firstly, for collecting the depth groundtruth values for the transparent objects. For this, the transparent objects

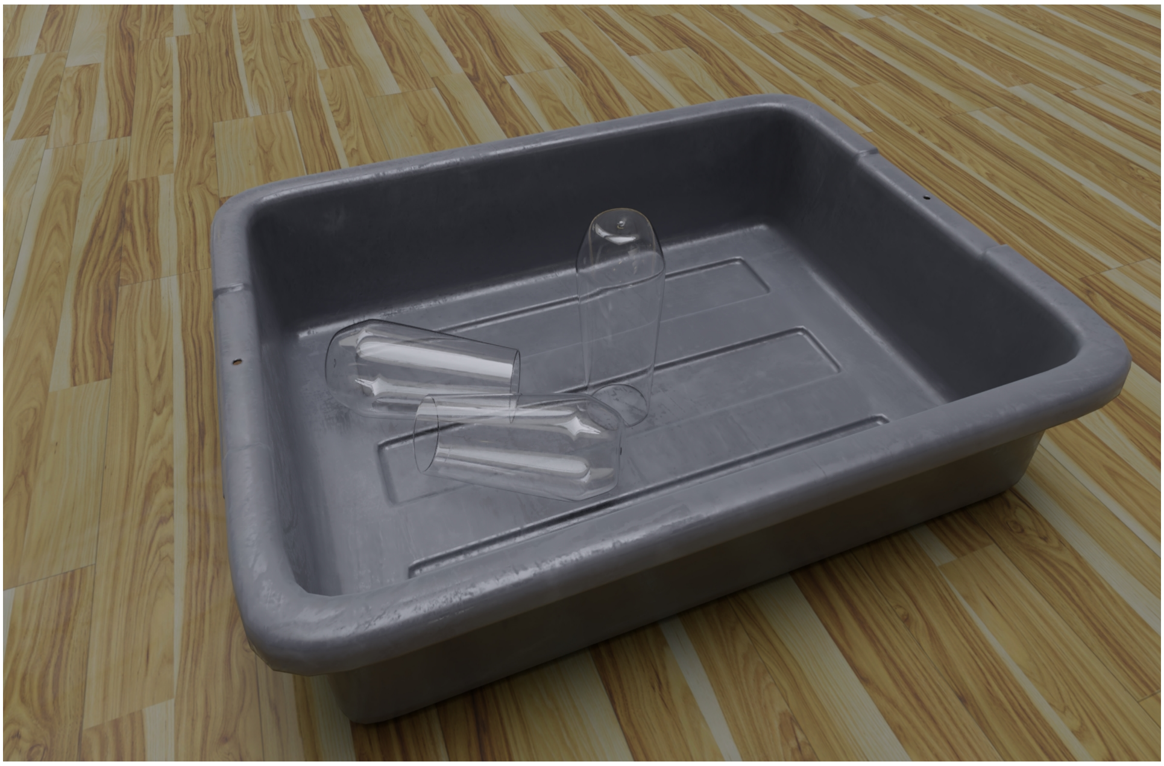


Figure 7: A synthetic RGB image sample from the clearGrasp synthetic validation set.

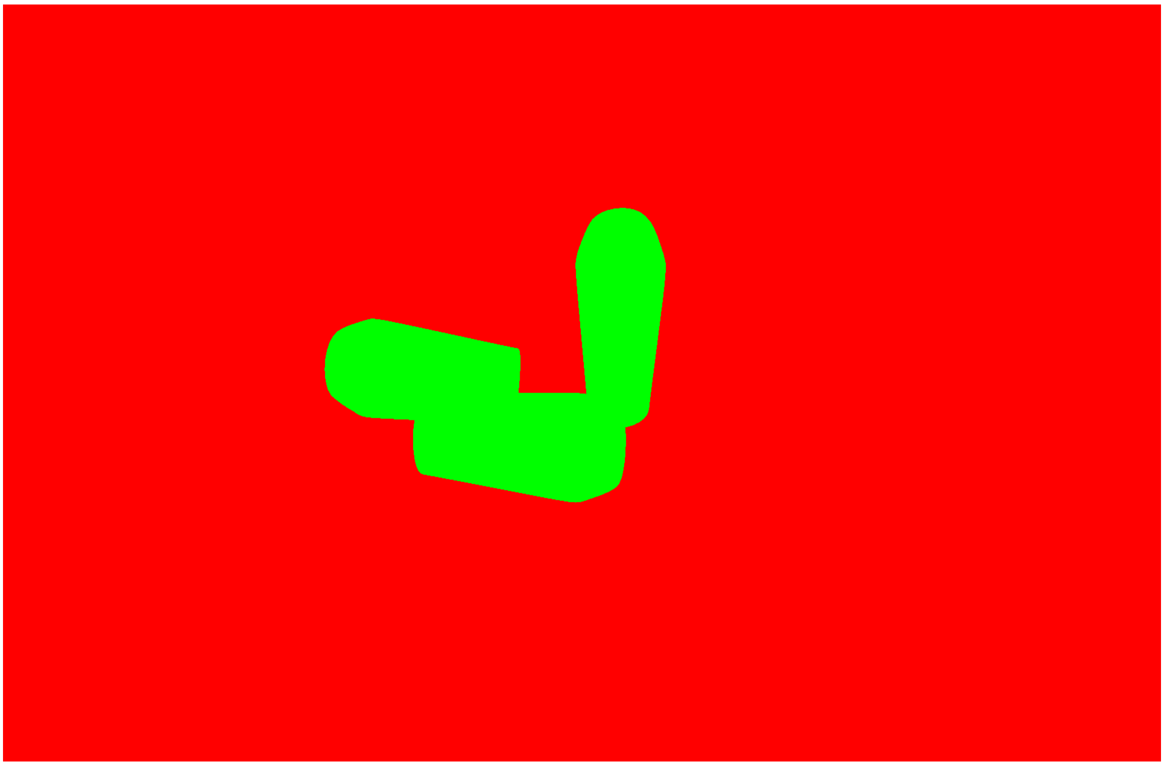


Figure 8: A sample mask visualization of the transparent object labels from the clearGrasp synthetic validation set. In the mask, the green pixels indicate the transparent objects and the red pixels indicate the background.

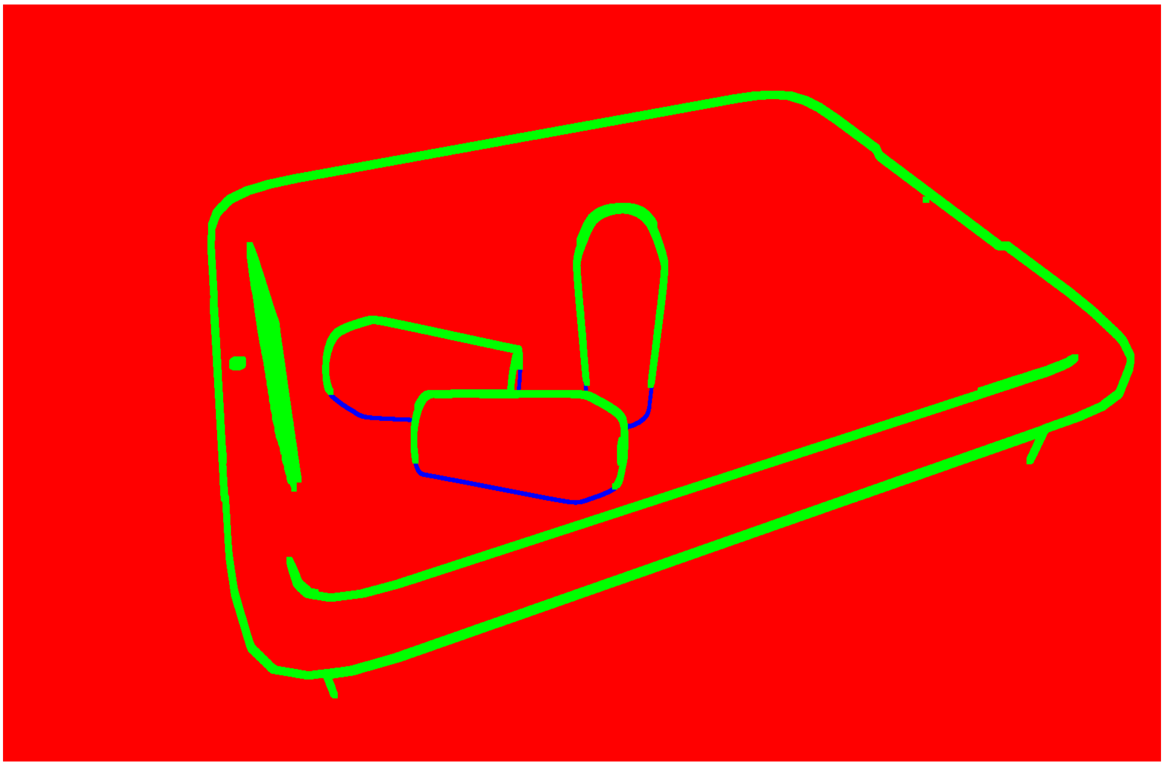


Figure 9: A sample mask visualization of the boundary segmentation labels from the clearGrasp synthetic validation set. In the mask, the green pixels indicate the normal edges, the blue pixels indicate the contact edges, and the red pixels indicate the background.

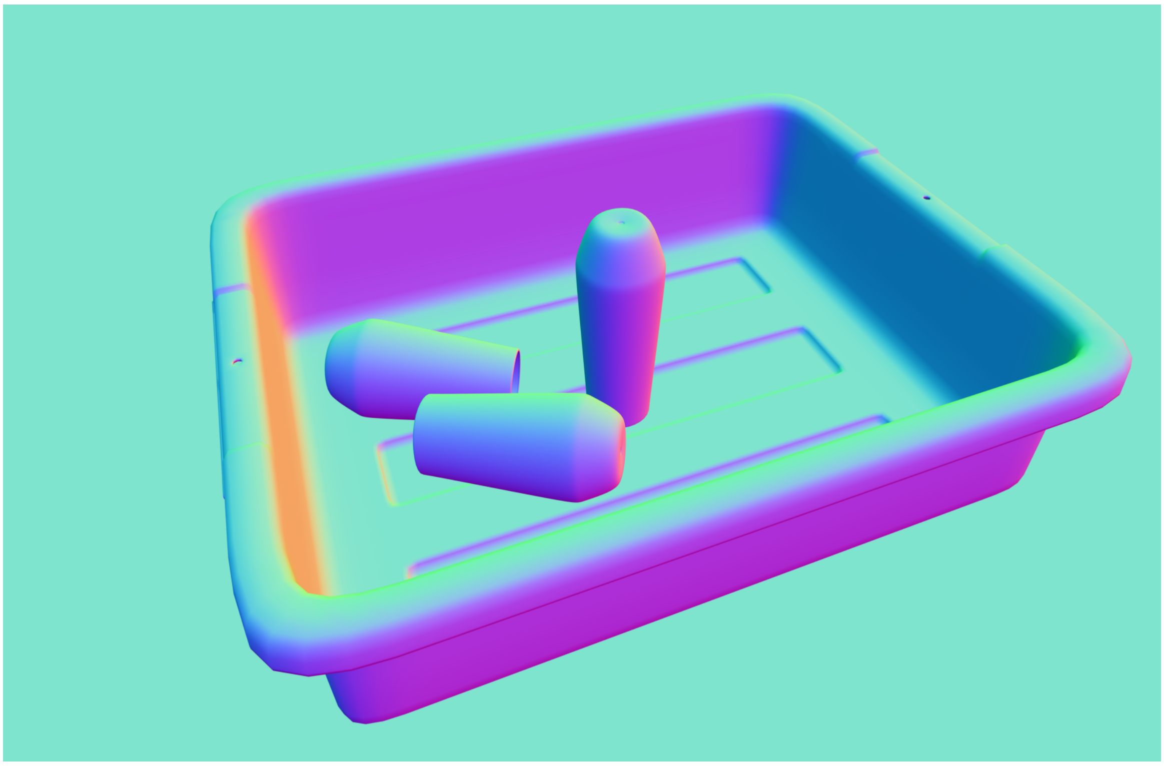


Figure 10: A sample visualization of the surface normals from the clearGrasp synthetic validation set.

were filled with various things. For collecting the test dataset, the location of the objects and the tripod were marked to make sure that the data collection was as accurate as possible. Secondly, for collecting the RGB images of the transparent objects, i.e. without filling them with anything, that will be used as the input to the models. The test dataset was collected for single-object and multi-object scenarios. Figure 11 shows the top view image of the tripod setup with the camera along with the four transparent objects. Figure 12 shows the marked location of the feet of the tripod stand. Figure 13 shows the marked location of the four transparent objects used for the test dataset collection. Figure 14 shows a sample image from the Kinect camera for the multi-object scenario with four objects. From this image, it is quite clear that the multi-object scenario was chosen in such a way that some objects are not occluding other objects whereas one of the objects is clearly occluding another object. Figure 15 shows a sample image from the Kinect camera for the single-object scenario with an object.

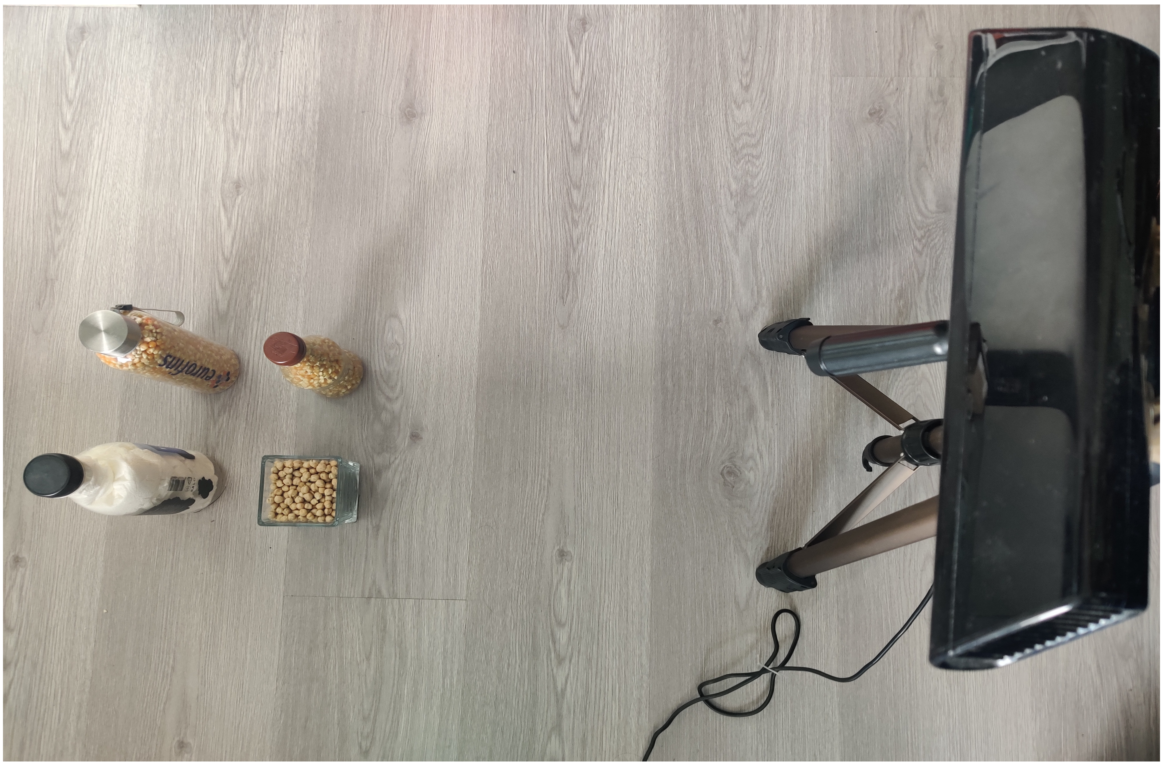


Figure 11: Top view image of the transparent objects filled with various things along with the tripod and the Kinect-v1 camera.

4.3 Preprocessing

For all the tasks, the images were converted from $[0,255]$ to $[0.0,1.0]$. Data augmentation is one of the techniques used to increase the size of the training set if it is small. Another advantage of data augmentation is that a model trained with it aids the model to generalize better than the model trained without it, on the validation and test sets. During training, various data augmentations were applied to the training set samples to increase the size of the training set. As usual, the augmentation was not applied during validation or testing. Some of the most common augmentations used in this research were horizontal and vertical flips, rotations by multiples of 90°, changing alpha (or opacity) values of images to make them brighter, applying color space modifications in the HSV color space, and

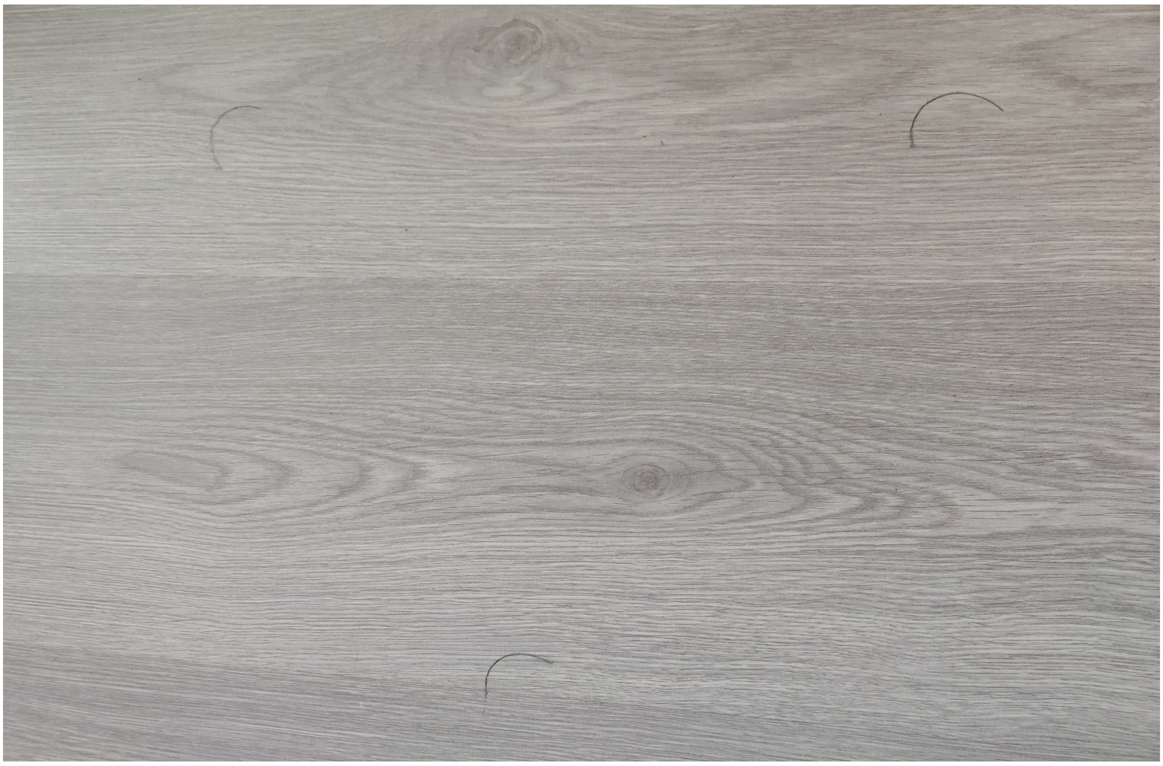


Figure 12: The location markings of the feet of the tripod stand.

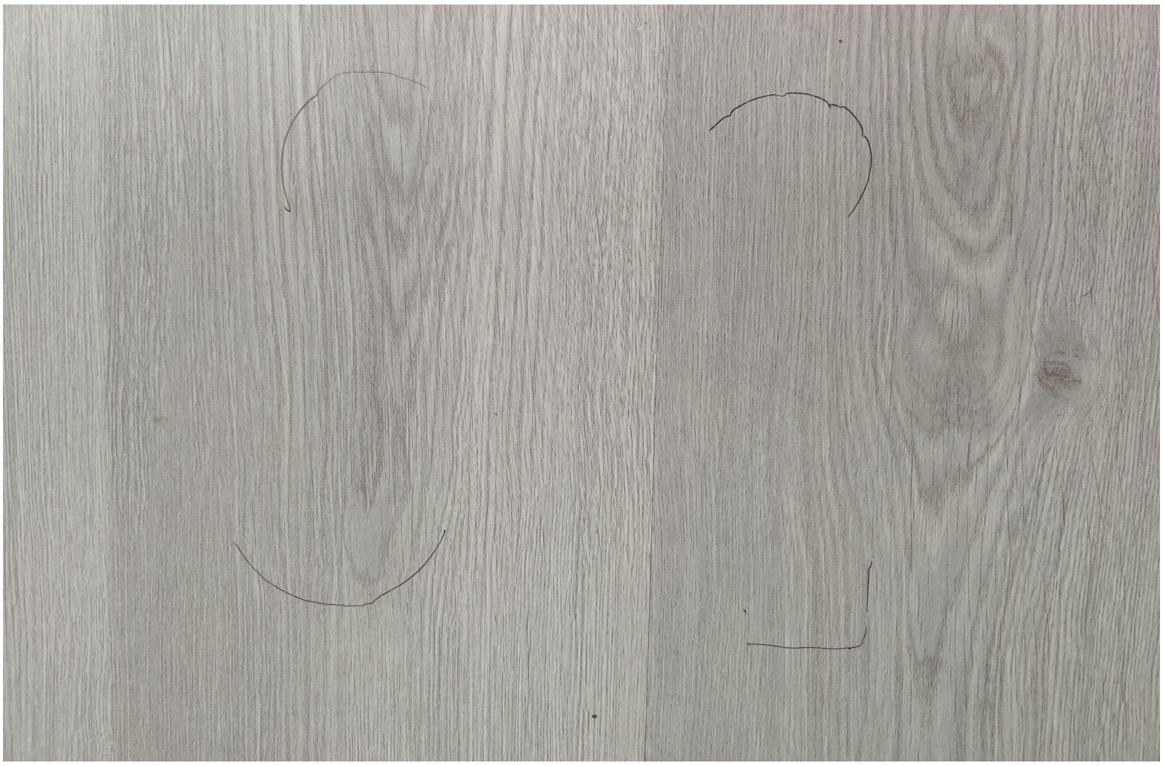


Figure 13: The location markings of the transparent objects.

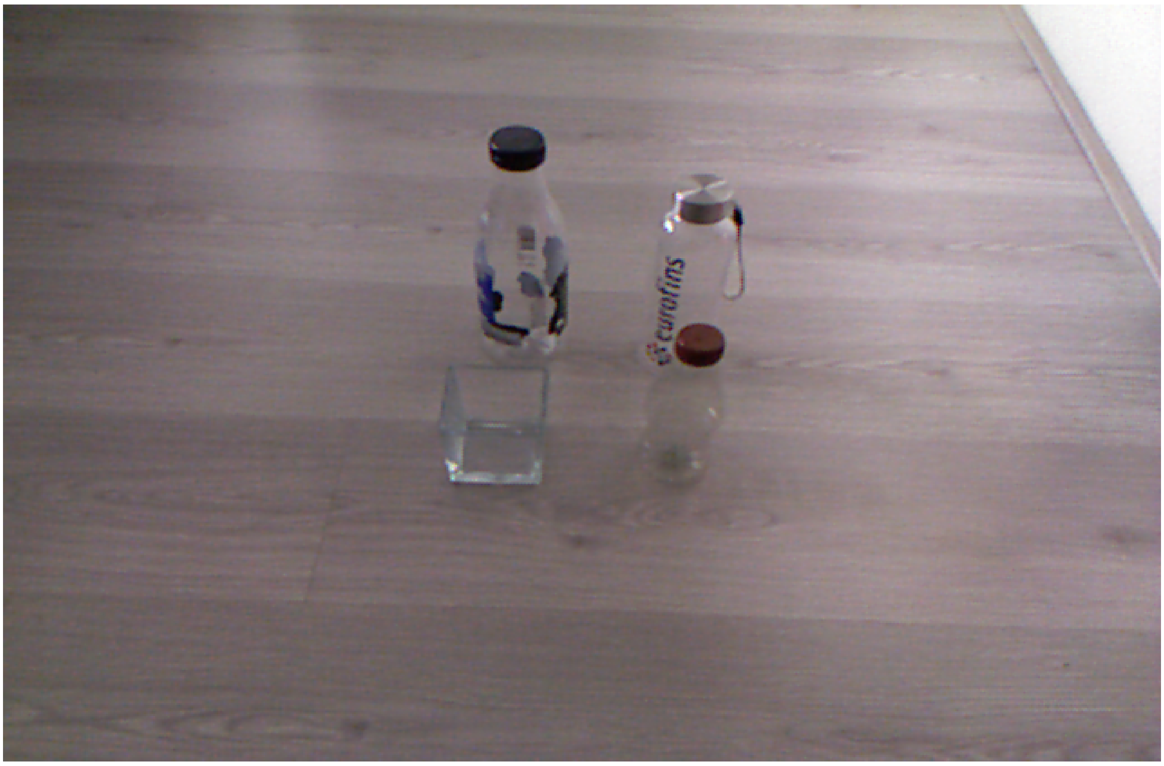


Figure 14: A sample test RGB image from Kinect camera for the multi-object scenario.

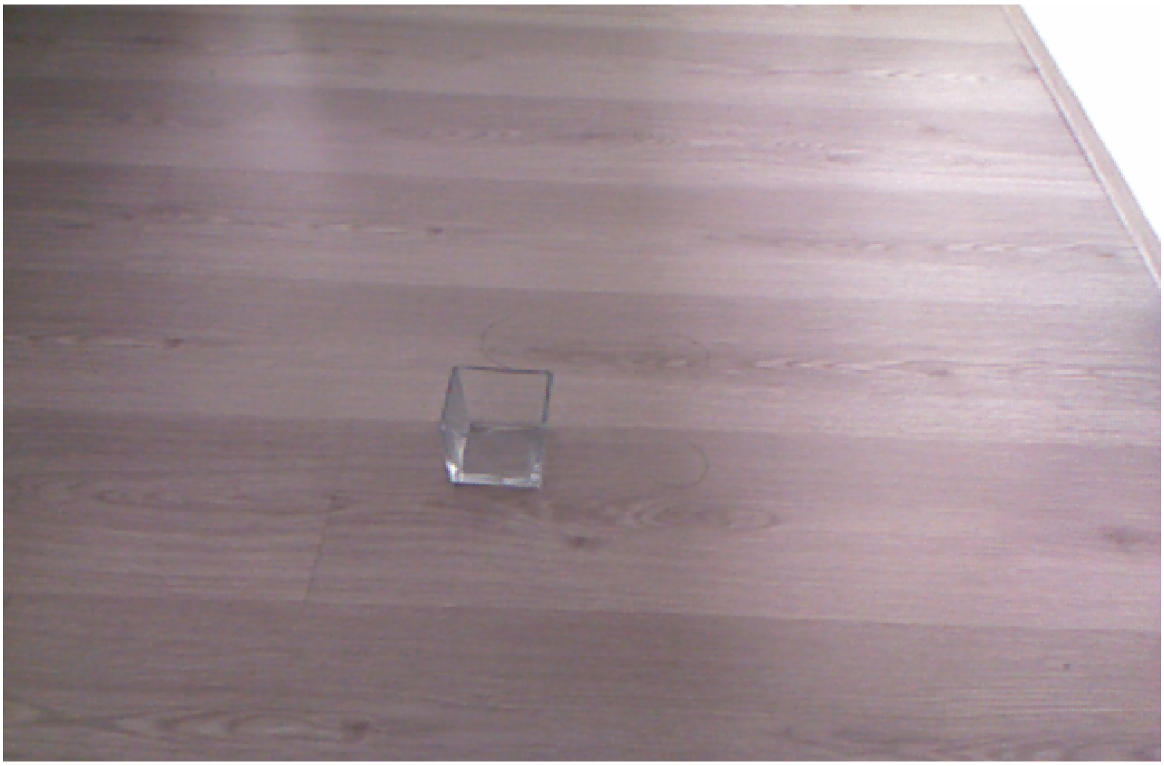


Figure 15: A sample test RGB image from Kinect camera for the single-object scenario.

applying blurring and noising.

For training the object segmentation model, all the augmentations were applied. For training the occlusion boundary prediction model, all augmentations except for vertical flips and rotations by multiples of 90° were applied. For training the surface normal estimation model, all augmentations except for the horizontal and vertical flips and rotations by multiples of 90° were applied.

4.4 Experimental configurations

As mentioned earlier in section 3.5, the categorical cross-entropy loss function was used for the transparent object segmentation task. As mentioned earlier in section 3.4, the SGD optimization algorithm with momentum was used with a polynomial learning rate scheduler for the transparent object segmentation task. From Equation 1, the momentum parameter (ρ) was set to 0.9. From Equation 3 of polynomial learning rate scheduler, the initial learning rate (η_t) was set to 1×10^{-2} and the *power* value was set to 0.25. For the weight decay, a value of 1×10^{-4} was used. Additionally, finetuning was performed for 50 epochs to study the effects of introducing negative samples, i.e. non-transparent objects from a relatively smaller real-world dataset.

The exact same set of parameters for the SGD optimization algorithm with momentum and the polynomial learning rate scheduler as mentioned above for the transparent object segmentation task was also used for the occlusion boundary prediction task. As mentioned in section 3.5, for the occlusion boundary prediction task, the focal loss was used. From Equation 6 of the focal loss, the parameters $\gamma_F = 2$ and $\alpha_F = 0.5$ were used. In the research by Tsung-Yi Lin et al., the best results for object detection were obtained with a value of $\gamma_F = 2$, so the same value was chosen for the experiments in this research [51].

For the surface normal estimation task, the radian loss function, as mentioned in section 3.5, was used. Since the surface normal estimation task is a regression task, the AdamW optimizer was preferred over the SGD optimizer with momentum as AdamW resulted in better results when compared with that of the SGD optimizer with momentum. The AdamW optimizer, from Equation 4, was combined with the polynomial learning rate scheduler, from Equation 3. The initial learning rate (η_t) was set to 1×10^{-3}, the default values of 0.9, 0.999 and 1×10^{-8} was used for β_1, β_2 and ε respectively. The weight decay was set to 1×10^{-4}. As mentioned earlier, additional samples from the real-world NYU_v2 dataset [15] were used along with the synthetic samples from the clearGrasp dataset for finetuning by including some negative samples, i.e. the non-transparent objects.

For training the models for all the individual tasks, a batch size of 32 was used. For all the tasks, input RGB images i.e., with 3 channels, with an image dimension of 256 × 256 were used. The same set of parameters was used for training both the proposed ResNet-50 + PSA model and the benchmark model, i.e. the DRN model. As discussed in section 4.3, different data augmentations were applied to different tasks.

For the depth estimation pipeline using the global optimization algorithm as given by Equation 9, different sets of values were used for λ_D, λ_S, and λ_N. The depth estimation experiments were performed with an input image dimension of 256 × 144. The experimental results for the individual tasks and the depth completion pipeline are presented in section 5.

5 Results

In this chapter, the results of the experiments are discussed in detail. Firstly, the learning curves and some observations of training and validation experiments for the individual tasks are discussed in section 5.1.1, section 5.1.2, and section 5.1.3. Next, the quantitative results of the individual tasks are discussed in section 5.2.1, section 5.2.2, and section 5.2.3. This is followed by the quantitative results of the depth estimation task in section 5.2.4. Finally, the qualitative analysis is discussed in detail for all the tasks in section 5.3.

5.1 Learning curve results

5.1.1 Object segmentation

Figure 16 shows the loss curve for the object segmentation task with the proposed ResNet-50 + PSA model. Figure 17 shows the performance curve, where the mean IoU score is tracked, for the object segmentation task with the proposed ResNet-50 + PSA model. From the loss and performance curves, it can be observed that the model is able to learn the task of transparent object segmentation with a decent performance since the loss reduces with the epochs and the mean IoU score increases with epochs, on both the training and the validation sets respectively. Except for a few places where the loss increases before reducing, it can be observed that there is a general pattern of the loss reducing with the training epochs. In general, the loss and performance curves are smoother in a majority of learning regions which is expected. In some learning regions, the loss and performance curves are not smooth but rugged which is also expected since those regions can be tricky for the model to navigate in the learning space. Another observation is that the training loss curve is higher than the validation loss curve. Similarly, the training mean IoU score curve is lower than the validation mean IoU curve. The possible reasons for this might be due to — firstly the dropout layer in the model and secondly the validation set being easier for the model to learn than the training set. It can be concluded that the proposed ResNet-50 + PSA model seems to be converging on the transparent object segmentation task.

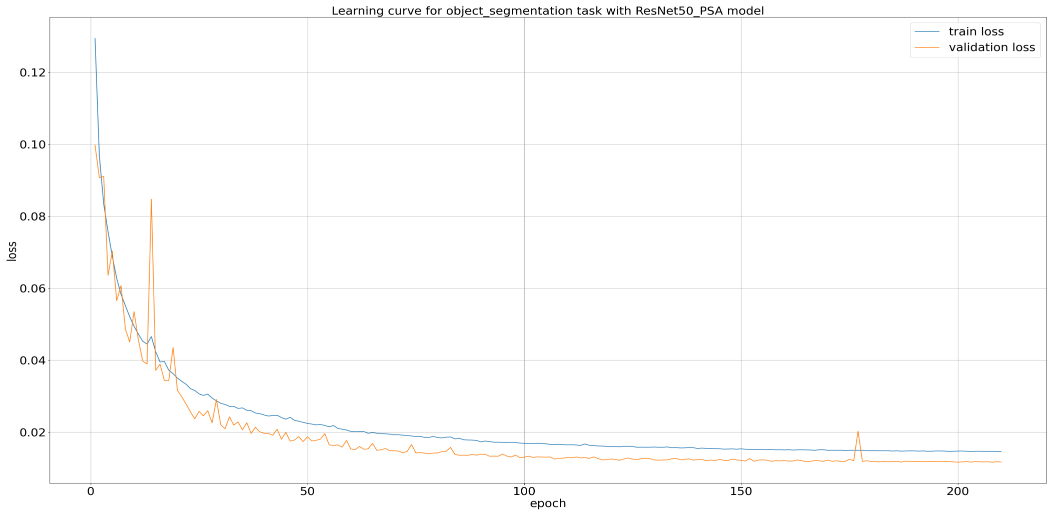


Figure 16: Loss curve of ResNet-50 + PSA model on the transparent object segmentation task.

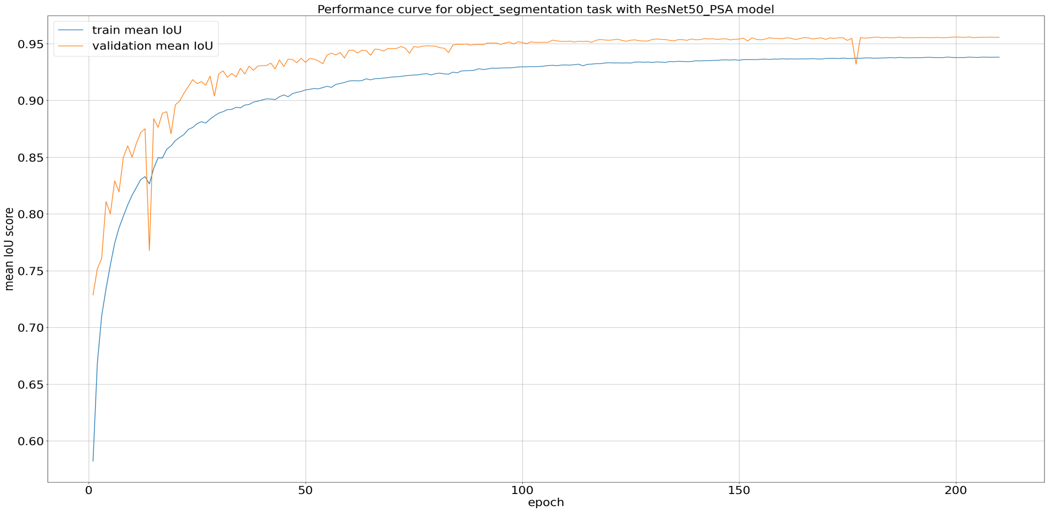


Figure 17: Performance curve of ResNet-50 + PSA model on the transparent object segmentation task.

5.1.2 Occlusion boundary detection

Figure 18 shows the loss curve for the occlusion boundary detection task with the proposed ResNet-50 + PSA model. Figure 19 shows the performance curve, where the mean IoU score is tracked, for the occlusion boundary detection task with the proposed ResNet-50 + PSA model. From the loss and performance curves, it can be observed that the model is able to learn to segment various boundaries with a decent performance since the loss reduces with the epochs and the mean IoU score increases with epochs, on both the training and the validation sets respectively. It can be clearly observed that the loss and performance curve is smoother for the training set unlike on the validation set where both the curves are not smooth but rugged. This observation is slightly different from what was observed on the transparent object segmentation task. Even though the loss curve is slightly lower on the training set when compared with that of the validation set, the performance curve is higher on the validation set when compared with that of the training set and this is a peculiar observation. This was not the observation in the case of the transparent object segmentation task. A possible reason for this could be the sparseness of the boundary segmentation task and the fact that there are three classes. Another possible reason could be that the validation set might be easier for the model to learn than the training set.

5.1.3 Surface normal estimation

Figure 20 shows the loss curve for the surface normal estimation task with the proposed ResNet-50 + PSA model with the Adam optimizer. Figure 21 shows the performance curve, where the percentage pixels with surface normals less than three selected thresholds in degrees are tracked, for the surface normal estimation task with the proposed ResNet-50 + PSA model. From the loss and performance curves, it can be observed that the model is able to learn to estimate surface normals with a decent performance since the mean and the median loss reduce with the epochs, on both the training and the validation sets respectively. This was the case when the Adam optimizer was used. However, when the SGD optimizer was used, the learning was a bit difficult and there were some oscillations in the

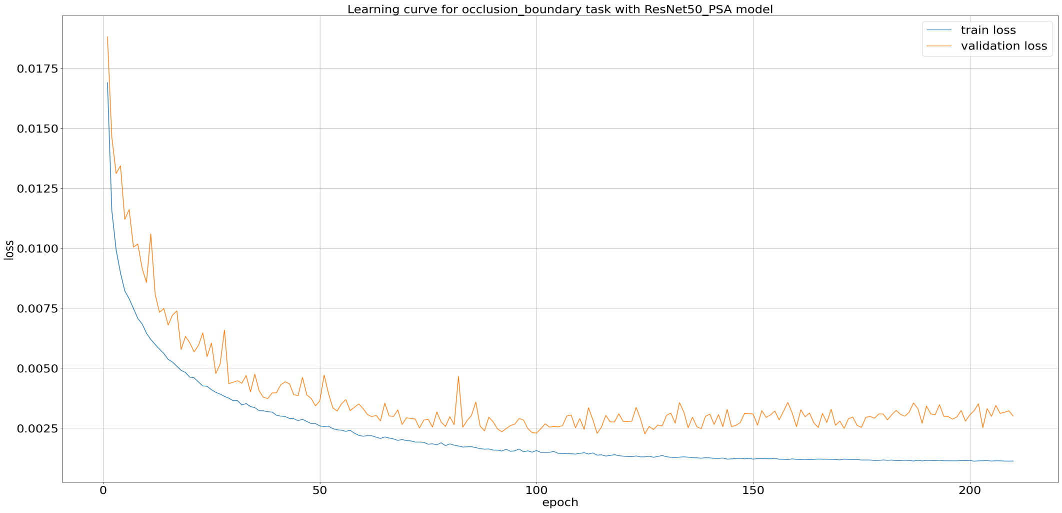


Figure 18: Loss curve of ResNet-50 + PSA model on the occlusion boundary segmentation task.

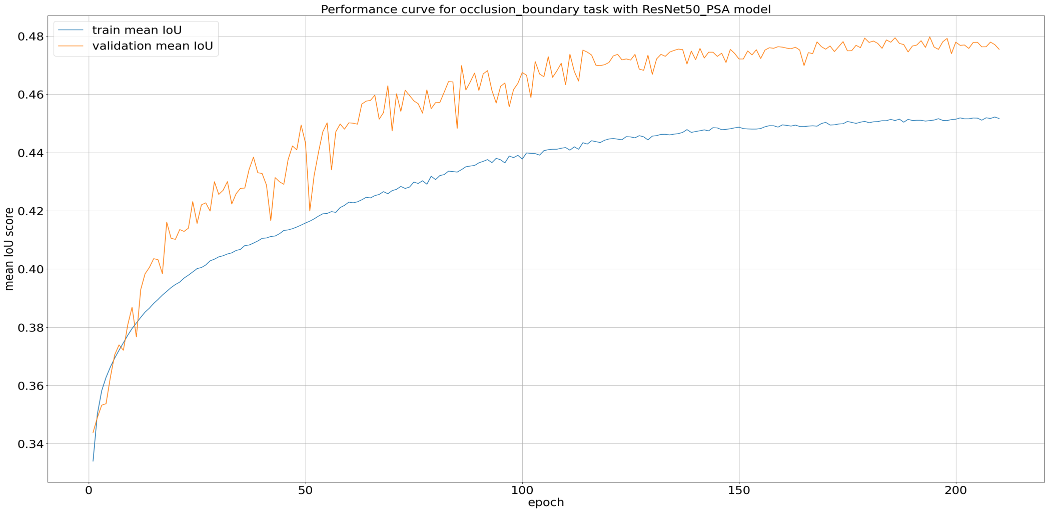


Figure 19: Performance curve of ResNet-50 + PSA model on the occlusion boundary segmentation task.

loss curve. The loss curve with the SGD optimizer can be found in section 6.4.

5.2 Quantitative Analysis

In this section, a detailed quantitative analysis is discussed for the individual tasks and the depth estimation task.

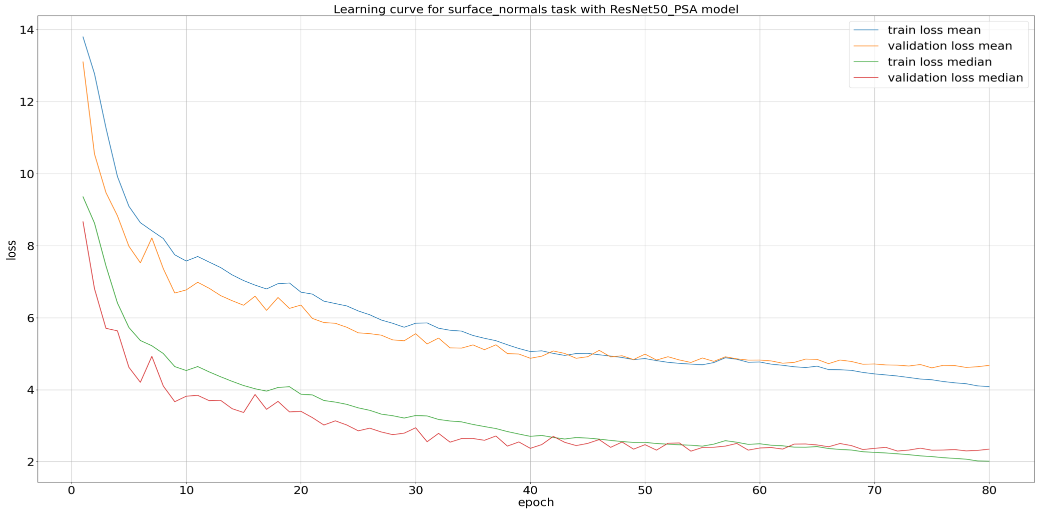


Figure 20: Loss curve of ResNet-50 + PSA model with the Adam optimizer on the surface normal estimation task.

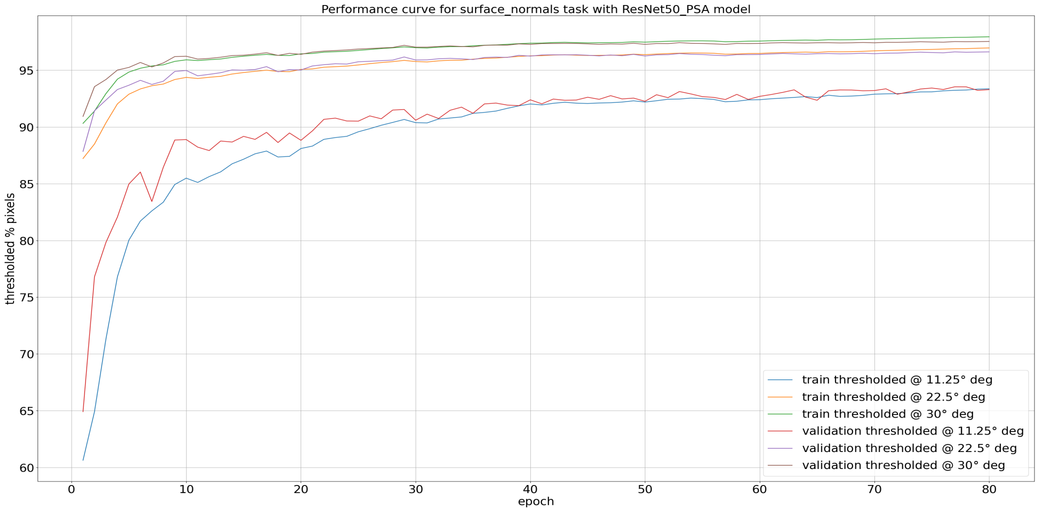


Figure 21: Performance curve of ResNet-50 + PSA model with the Adam optimizer on the surface normal estimation task.

5.2.1 Object segmentation task

The Table 1 shows the quantitative results on the transparent object segmentation task for the models trained from scratch. The groundtruth labels were available for both the synthetic and real test sets so the evaluation was done on both of them. The DRN model obtained a mean IoU score of 0.8679 with a false positive rate of 0.93% whereas the proposed ResNet-50 + PSA model obtained a mean IoU score of 0.8782 with a false positive rate of 0.71% on the synthetic test set. Furthermore, the DRN model obtained a mean IoU score of 0.4793 with a false positive rate of 5.68% whereas the proposed

ResNet-50 + PSA model obtained a mean IoU score of 0.5121 with a false positive rate of 4.97% on the real test set. It can be observed that both the models' performance dips in the case of the real test set with a very high false positive rate. This is expected since the synthetic train set did not have any representative negative samples except for the tray in which the transparent objects were placed and the model which has been largely trained on the synthetic dataset might not have learned enough feature extraction capability to the best possible extent that might be needed for the real test dataset. Although different augmentations were applied while training on the synthetic dataset, it does not seem to be enough for the model to learn so as to generalize very well on the real test dataset. The ResNet-50 + PSA model outperforms the DRN model on both the synthetic and real test sets but the difference is quite large in the case of the real test set with a difference of $+0.0328$ than the synthetic test set where the difference is only $+0.0103$. This shows that the ResNet-50 + PSA model is capable of not only achieving high performance on the synthetic test set but also being able to generalize better on the real test set when compared with that of the DRN model. To conclude, the ResNet-50 + PSA model has learned to transfer from synthetic to real dataset slightly better than the DRN model for the transparent object segmentation task.

Model	Test Set	m-IoU ↑	False Positive (%) ↓
DRN	Synthetic test	0.8679	0.93
	Real test	0.4793	5.68
ResNet-50 + PSA	Synthetic test	**0.8782**	**0.71**
	Real test	**0.5121**	**4.97**

Table 1: Quantitative results of the models trained from scratch with the synthetic dataset for the transparent object segmentation task.

Additional experiments were performed to study the effects of introducing negative samples from the real dataset along with the synthetic train set. Table 2 shows the quantitative results of the transparent object segmentation task finetuned with both the synthetic train set and a relatively smaller real dataset with negative object samples. The finetuned ResNet-50 + PSA model outperformed its respective model which was trained from scratch without the negative samples, by obtaining a higher m-IoU score (0.7636 vs. 0.5121) and lower false positive rate (0.94 vs. 4.97) on the real test set. Similarly, the finetuned DRN model outperformed its respective model which was trained from scratch without the negative samples, by obtaining a higher m-IoU score (0.6921 vs. 0.4793) and lower false positive rate (1.69 vs. 5.68) on the real test set. The finetuned ResNet-50 + PSA model outperformed the finetuned DRN model on the real test set with a very high difference in m-IoU (0.7636 vs. 0.6921). The ResNet-50 + PSA model also obtained a much lower false positive rate when compared with that of the DRN model (0.94 vs. 1.69) on the real test set. To conclude, the finetuning of the model by adding some negative samples, i.e. non-transparent objects improved the overall performance of the model on the real test set including a drastic reduction in the false positive rate.

5.2.2 Occlusion boundary segmentation task

The Table 3 shows the quantitative results on the boundary outline segmentation task for the models trained from scratch. Since the groundtruth labels were unavailable for the boundary segmentation task for the real test set, the evaluation was performed only on the synthetic test set. The DRN model obtained a mean IoU score of 0.4592 whereas the proposed ResNet-50 + PSA model obtained a mean

Model	Test Set	m-IoU ↑	False Positive (%) ↓
DRN	Real test	0.6921	1.69
ResNet-50 + PSA	Real test	**0.7636**	**0.94**

Table 2: Quantitative results of the models finetuned with the synthetic and a small real-world dataset for the transparent object segmentation task.

IoU score of 0.4734 on the synthetic test set. Since the task of boundary segmentation task is sparse and involves three classes, it can be observed that both DRN and ResNet-50 + PSA models obtained lower mean IoU scores when compared with that of the transparent object segmentation task which is less sparse and involves only two classes, on the synthetic test set and this is as expected due to the nature of these tasks. The proposed ResNet-50 + PSA model outperforms the DRN model on the synthetic test set by a fair amount of $+0.0142$.

Model	Test set	m-IoU ↑
DRN	Synthetic	0.4592
ResNet-50 + PSA	Synthetic	**0.4734**

Table 3: Quantitative results of the models trained from scratch with the synthetic dataset for the boundary segmentation task.

5.2.3 Surface normal estimation task

The Table 4 shows the quantitative results of the surface normal estimation task for the models trained and finetuned with both the clearGrasp synthetic train dataset and the real-world NYU_v2 dataset. The finetuned ResNet-50 + PSA model obtained a mean error of 7.1170 and a median error of 2.8968 whereas the DRN model obtained a mean error of 9.3725 and a median error of 4.9174 on the synthetic test set. The finetuned ResNet-50 + PSA model obtained percentage scores of 87.08, 92.66, and 94.20 whereas the DRN model obtained percentage scores of 78.12, 89.31, and 93.04 at thresholds of 11.25°, 22.5°, and 30° on the synthetic test set. The finetuned ResNet-50 + PSA model outperformed the DRN model on all the metrics by a significant margin on the clearGrasp synthetic test set. The finetuned ResNet-50 + PSA model obtained a mean error of 32.7944 and a median error of 15.4948 whereas the DRN model obtained a mean error of 32.1225 and a median error of 17.5736 on the real-world test set. The finetuned ResNet-50 + PSA model obtained percentage scores of 46.45, 61.86, and 65.07 whereas the DRN model obtained percentage scores of 36.17, 59.12, and 64.38 at thresholds of 11.25°, 22.5°, and 30° on the real-world test set. Although the finetuned ResNet-50 + PSA model outperformed the DRN model on all the metrics except for the mean error on the real-world test set, both models' performance dipped significantly when compared with the performance on the synthetic test set.

5.2.4 Depth estimation task

For the depth estimation task, the best results were obtained with the following parameters — $\lambda_D = 500$, $\lambda_S = 0.001$, and $\lambda_N = 5$. The Table 5 shows the quantitative results of the depth estimation task using the global optimization method with the best-performing models for both ResNet-50 + PSA and DRN models for each of the individual tasks. The ResNet-50 + PSA model obtained RMSE of

Model	Test set	mean ↓ error (in deg)	median ↓ error (in deg)	%@11.25° ↑	%@22.5° ↑	%@30° ↑
DRN	Synthetic	9.3725	4.9174	78.12	89.31	93.04
	Real	**32.1225**	17.5736	36.17	59.12	64.38
ResNet-50 + PSA	Synthetic	**7.1170**	**2.8968**	**87.08**	**92.66**	**94.20**
	Real	32.7944	**15.4948**	**46.45**	**61.86**	**65.07**

Table 4: Quantitative results of the models trained from scratch for the surface normal estimation task.

0.04655, MeanREL of 0.08147, and MAE of 0.04139 whereas the DRN model obtained RMSE of 0.05052, MeanREL of 0.08931, and MAE of 0.04568 on the synthetic test set. The ResNet-50 + PSA model also obtained better scores than the DRN model for all the delta metrics on the synthetic test set. The ResNet-50 + PSA model obtained RMSE of 0.02814, MeanREL of 0.04425, and MAE of 0.02276 whereas the DRN model obtained RMSE of 0.03097, MeanREL of 0.05144, and MAE of 0.02657 on the real-world test set. The ResNet-50 + PSA model also obtained better scores than the DRN model for all the delta metrics except for the $\delta_{1.25}$ score on the real-world test set. The ResNet-50 + PSA model clearly outperforms the DRN model for both the synthetic test set and the real-world test set. Another important thing to note here is that the models obtained better scores on the real-world test set when compared with that of the synthetic test set which is expected since the real-world test set is relatively smaller when compared with the synthetic test set.

Model	Test set	RMSE ↓	MeanREL ↓	MAE ↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
DRN	Synthetic	0.05052	0.08931	0.04568	31.74	68.33	96.14
	Real	0.03097	0.05144	0.02657	58.91	88.16	**99.41**
ResNet-50 + PSA	Synthetic	**0.04655**	**0.08147**	**0.04139**	**37.97**	**72.31**	**97.23**
	Real	**0.02814**	**0.04425**	**0.02276**	**68.90**	**90.57**	99.19

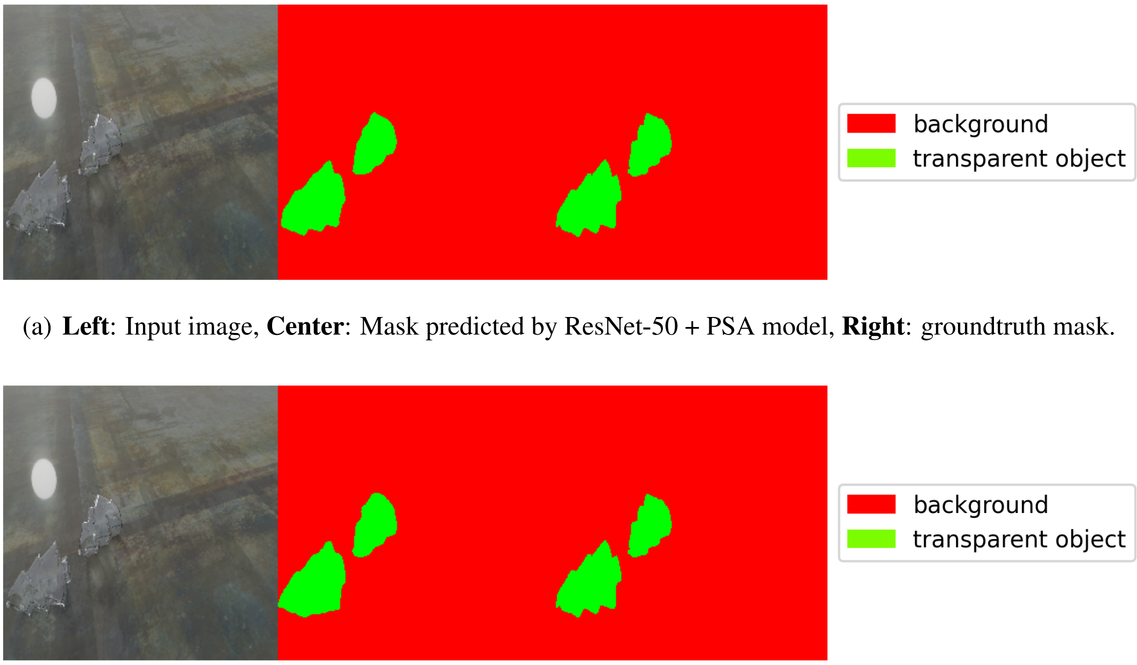
Table 5: Quantitative results for the depth estimation task using the global optimization method with the best-performing models for each of the individual tasks.

5.3 Qualitative analysis

In this section, a qualitative analysis of predictions of the models on various tasks is presented only on some of the selected samples from the test set which were intriguing enough when compared to normal observations in some of the other samples. The qualitative analysis is presented for the transparent object segmentation trained on the synthetic train set in section 5.3.1, the transparent object segmentation finetuned on the synthetic train set and some samples from the real dataset in section 5.3.2, the boundary segmentation task in section 5.3.3, the surface normal estimation task in section 5.3.4 and the depth estimation task in section 5.3.5.

5.3.1 Transparent object segmentation trained on synthetic train set

Figure 22 shows the visualization of the masks predicted by both the ResNet-50 + PSA and DRN models, trained on the synthetic train set, for a sample from the synthetic test set. The tree bath bomb object in the input image was present neither in the training nor validation sets. Figure 22(a) shows the prediction by the ResNet-50 + PSA model where the left image is the input image, the center image is the mask predicted by the model, and the right image is the groundtruth mask. Figure 22(b) shows the prediction by the DRN model with a similar convention. From the predicted masks for both models, it can be clearly observed that both models have learned fairly well to segment the transparent objects even though there is a presence of some kind of bright light reflection on the surface. However, when carefully observed, the ResNet-50 + PSA model prediction is slightly better for this particular example as the model has learned to segment finer details of the object such as the sharp edges when compared with that of the prediction by the DRN model.



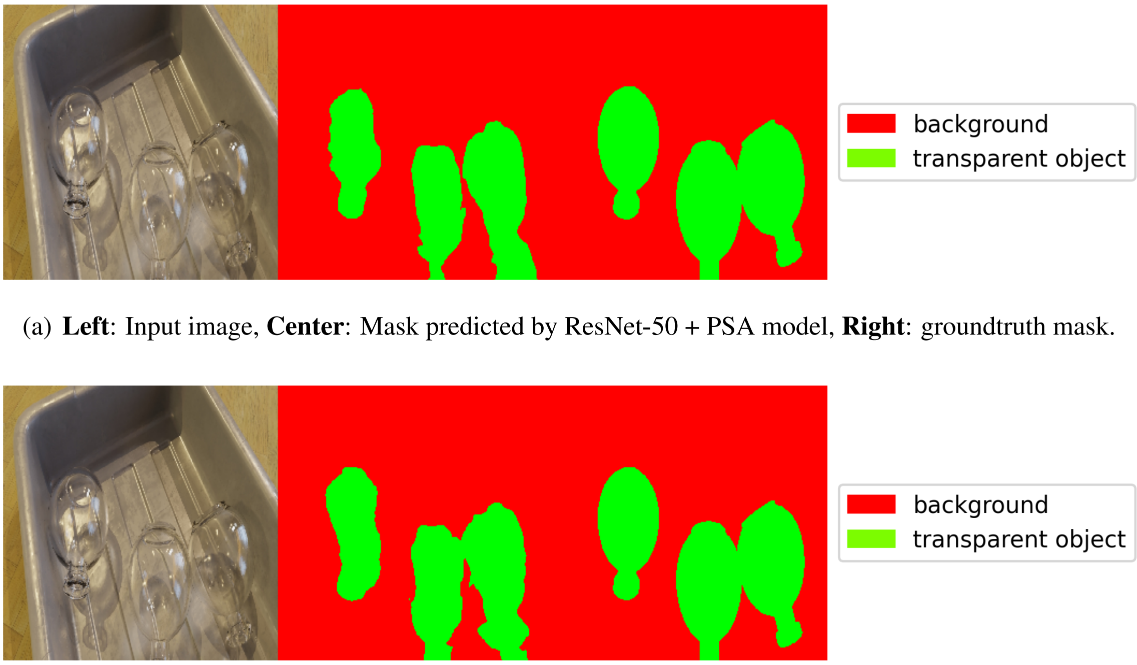
(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

Figure 22: Visualization showing the comparison of the transparent object segmentation prediction masks, for the tree bath bomb object from the synthetic test set, between ResNet-50 + PSA and DRN models.

Figure 23 shows the visualization of the masks predicted by both the models, trained on the synthetic train set, for yet another sample from the synthetic test set. This glass round potion object was neither present in the train nor validation sets. In both Figure 23(a) and Figure 23(b), it can be observed that both the models had some difficulties in segmenting the transparent glass round potion object. Here, the transparent object shadows might have confused the model which might have resulted in the bad predictions.

Figure 24 shows the visualization of the masks by both the models, trained on the synthetic train set, for the star bath bomb object from the synthetic test set that was present in neither the train nor validation sets. In the input image, it can be observed that there is a clear texture that has the potential to confuse the model. From Figure 24(a) and Figure 24(b), it can be observed that although both models have learned to segment in spite of the presence of the texture, the ResNet-50 + PSA model slightly



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

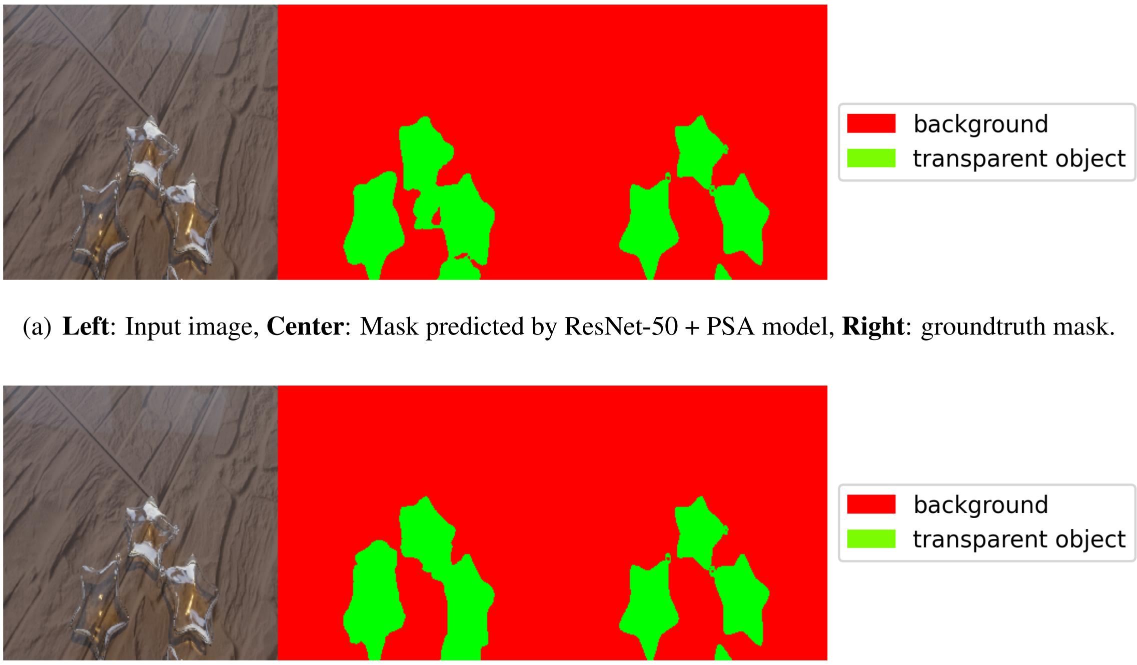
(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

Figure 23: Visualization showing the comparison of the transparent object segmentation prediction masks, for the glass round potion object from the synthetic test set, between ResNet-50 + PSA and DRN models.

performs better in segmenting star shaped transparent object when compared with the DRN model. However, both models have segmented some pixels as belonging to some transparent object but are in fact belonging to the background class. When the false positives are taken into consideration for this test sample, the ResNet-50 + PSA model has predicted lower false positives when compared with that of the DRN model.

Figure 25 shows the visualization of the masks predicted by both the models, trained on the synthetic train set, for a sample image from the real-world test set with a non-transparent object. From Figure 25(a) and Figure 25(b), it can be observed that ResNet-50 + PSA model has not segmented the yellow capsicum as a transparent object whereas the DRN model has segmented the yellow capsicum as a transparent object. Both models have correct predictions for the black object. Both models have incorrect predictions for the gray object. In this particular example, it is clear that the ResNet-50 + PSA model is better in terms of avoiding false positives when compared with that of the DRN model. However, both models have correct predictions for the tree bath bomb transparent object from the real-world test set even though the models were trained on the synthetic train set.

Figure 26 shows the visualization of the masks predicted by both the models, trained on the synthetic train set, for a sample image from the real-world test set with multiple transparent and non-transparent objects. From Figure 26(a) and Figure 26(b), it is clear that the transparent objects and the black square box have been segmented correctly by both the models. However, both models have incorrectly segmented some pixels belonging to the gray object and the two animal toys as transparent objects. A possible reason for this might be the lack of negative samples in the synthetic train set.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

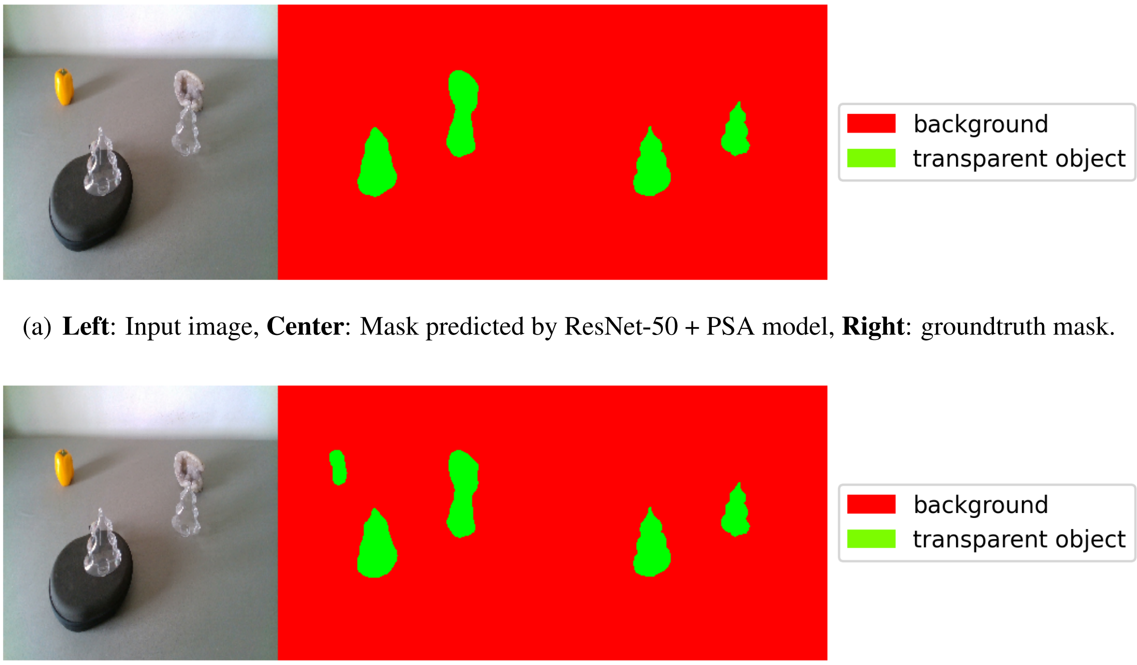
Figure 24: Visualization showing the comparison of the transparent object segmentation prediction masks, for the star bath bomb object from the synthetic test set, between ResNet-50 + PSA and DRN models.

However, it is clear, that the transparent objects are being segmented correctly even in the images from the real-world test dataset by both models even though they have been trained on the synthetic train set. The generalization from the synthetic to the real-world dataset has been learned with a good enough performance.

5.3.2 Transparent object segmentation finetuned on synthetic train set and real set

Since the model trained only on the synthetic train set had a lot of false positives on the real-world test set, additional finetuning experiments were performed with both the synthetic train set and negative samples, i.e. with non-transparent objects, from the real-world dataset. In this section, visualizations are shown only for the real-world test set for both the ResNet-50 + PSA and the DRN finetuned models.

Figure 27 shows the visualization of the masks predicted by both the finetuned models, which were trained on the synthetic train set and a small real-world dataset with non-transparent objects, for a real-world test set sample image with multiple transparent and non-transparent objects. From Figure 27(a) and Figure 27(b), both models have predicted correct masks for the yellow capsicum and incorrect masks for the gray object while maintaining correct masks for the two transparent objects. The wrong mask predictions by both models for the gray object are expected since the gray object appears on a gray-colored table which can confuse the model and make it believe that the gray object indeed is a transparent object.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

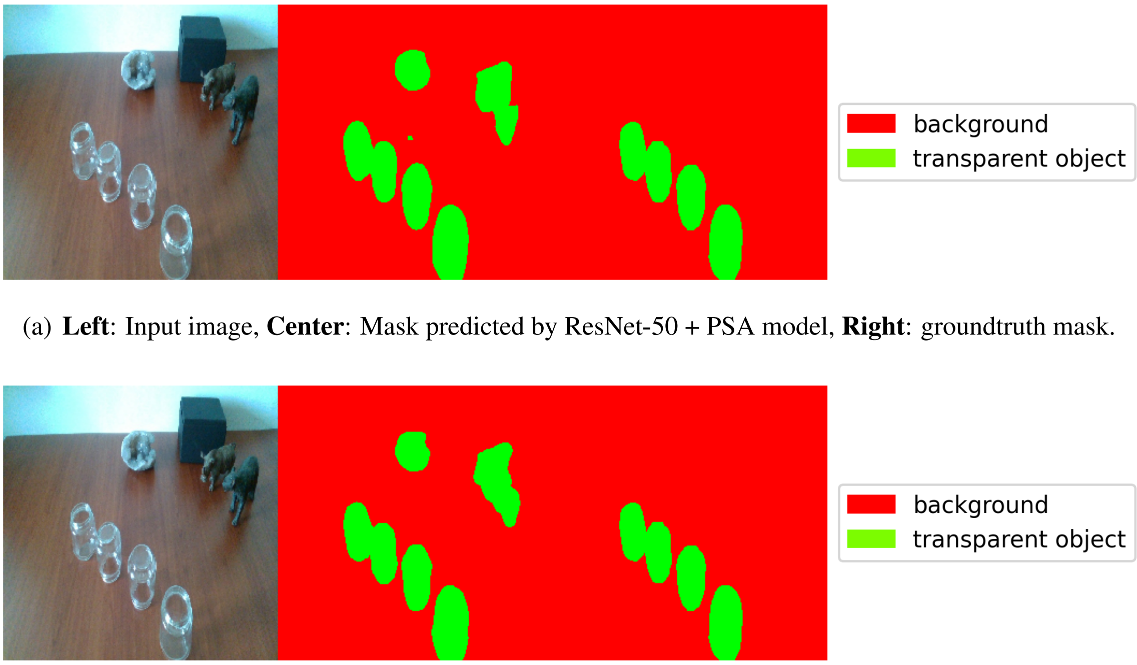
(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

Figure 25: Visualization showing the comparison of the transparent object segmentation prediction masks, for a real-world test set sample with two transparent objects and three other objects, between ResNet-50 + PSA and DRN models.

Figure 28 shows the visualization of the masks predicted by both the finetuned models, which were trained on the synthetic train set and a small real-world dataset with non-transparent objects, for a real-world test set sample image with multiple transparent and non-transparent objects. From Figure 28(a), it can be observed that a part of the object for one of the animal toys and the gray object has been predicted as a transparent object by the finetuned ResNet-50 + PSA model. From Figure 28(b), it can be observed that the mask prediction is much better by the finetuned DRN model for this particular example.

Figure 29 shows the visualization of the masks predicted by both the finetuned models, which were trained on the synthetic train set and a small real-world dataset with non-transparent objects, for a real-world test set sample image with multiple transparent and non-transparent objects. From Figure 29(a), it can be clearly observed that the mask prediction is pretty good by the finetuned ResNet-50 + PSA model. From Figure 29(b), it can be clearly observed that the mask prediction for the gray object is wrong even though the object is not on a gray-colored table. Also, some background pixels belonging to the table have been predicted with the wrong mask. Another important observation in the mask predictions by both models is that the mask edges are not sharp enough for the star-shaped objects when compared with the groundtruth.

To summarize, the false positives have reduced significantly on the real-world test set after finetuning both models with both synthetic transparent and real-world non-transparent object samples.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

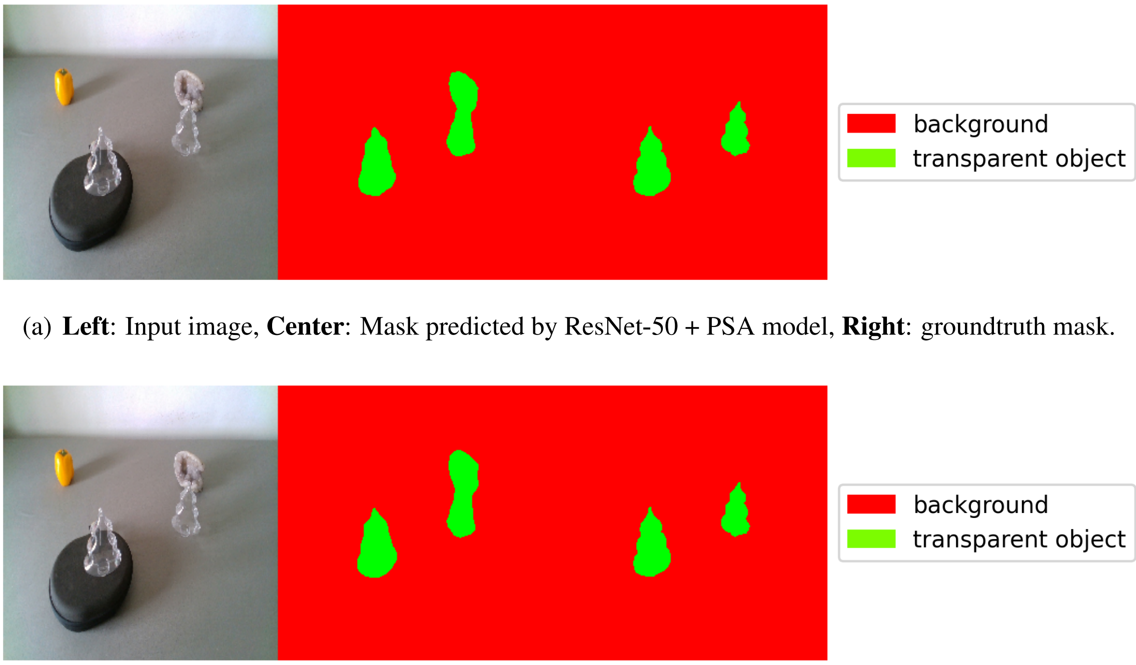
Figure 26: Visualization showing the comparison of the transparent object segmentation prediction masks, for a real-world test set sample with four transparent objects and four other objects, between ResNet-50 + PSA and DRN models.

5.3.3 Boundary segmentation

Figure 30 shows the visualization of the boundary segmentation prediction masks with ResNet-50 + PSA and DRN models for the glass round potion object from the synthetic test set. From Figure 30(a) and Figure 30(b), it can be clearly observed that the contact edges (in blue) are accurately segmented by both the models for two objects. The normal edges (in green) are almost accurately segmented by both the models for all four objects except for the fact that some of them are thicker than the groundtruth. For one of the objects, there are some false predictions in the normal edges. In this particular test sample, both models have learned to segment both types of edges for this particular novel object.

Figure 31 shows the visualization of the boundary segmentation prediction masks with ResNet-50 + PSA and DRN models for the glass square potion object from the synthetic test set. From Figure 31(a) and Figure 31(b), it can be clearly observed that the contact edges (in blue) are not accurately segmented by both the models for these objects. The normal edges (in green) are also not accurately segmented by both the models for all these objects as they are much thicker than the expected groundtruth.

Figure 32 shows the visualization of the boundary segmentation prediction masks with ResNet-50 + PSA and DRN models for the star bath bomb object from the synthetic test set. From Figure 32(a) and Figure 32(b), it can be observed that the normal edges, in green, are almost accurately segmented by ResNet-50 + PSA model whereas there are some false positives in the DRN model prediction. When it comes to contact edges, in blue, are not segmented accurately by both models.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

Figure 27: Visualization showing the comparison of transparent object segmentation prediction masks, for a real-world test set sample with two transparent objects and three other objects, between finetuned ResNet-50 + PSA and finetuned DRN models.

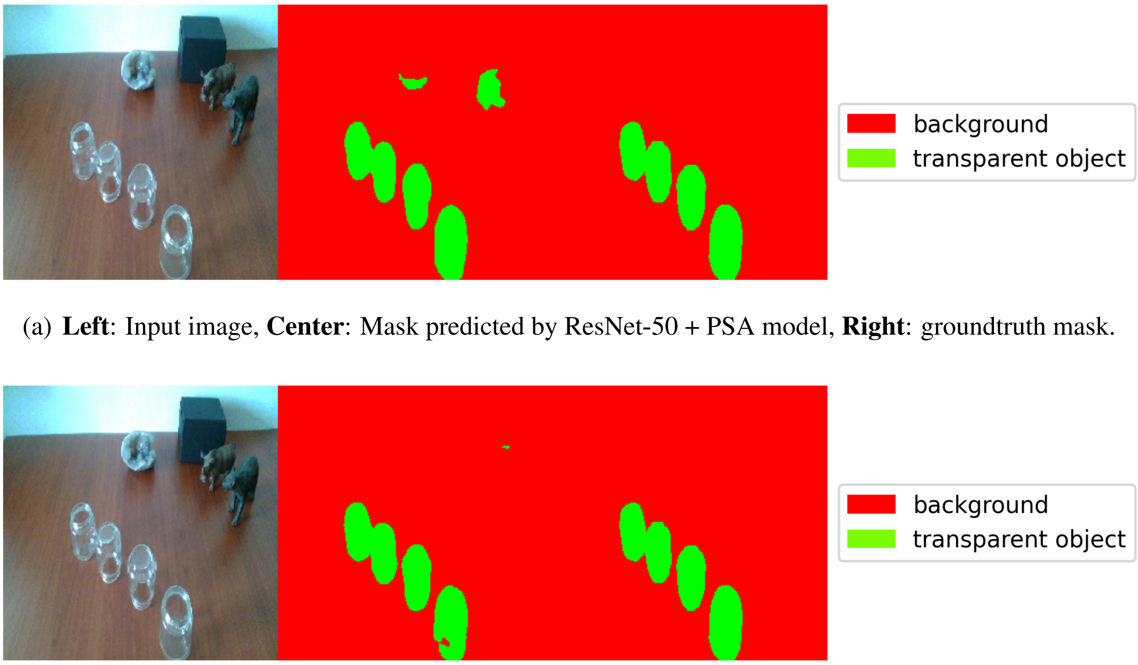
Figure 33 shows the visualization of the boundary segmentation prediction masks with ResNet-50 + PSA and DRN models for the tree bath bomb object from the synthetic test set. From Figure 33(a) and Figure 33(b), it can be observed that the contact edges, in blue, are almost accurately segmented by both models. The normal edges, in green, are segmented more accurately by the ResNet-50 + PSA model when compared with that of the DRN model prediction which has predicted thicker segmentation for the normal edges.

From the visualizations of the boundary segmentation task, it was observed that in a majority of the cases, the segmented boundaries were either thicker or not as accurate as one would expect them to be. Since the task of boundary segmentation is much more sparser than the transparent object segmentation, the results are as expected.

5.3.4 Surface normal estimation

Figure 34 shows the visualization of the estimated surface normals by ResNet-50 + PSA and DRN models for the glass round potion object from the synthetic test set. From Figure 34(a) and Figure 34(b), it can be clearly observed that the normals estimated by the ResNet-50 + PSA model are much better than normals estimated by the DRN model. For one of the objects, normals are non-existent in the prediction by the DRN model.

Figure 35 shows the visualization of the estimated surface normals by ResNet-50 + PSA and DRN



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

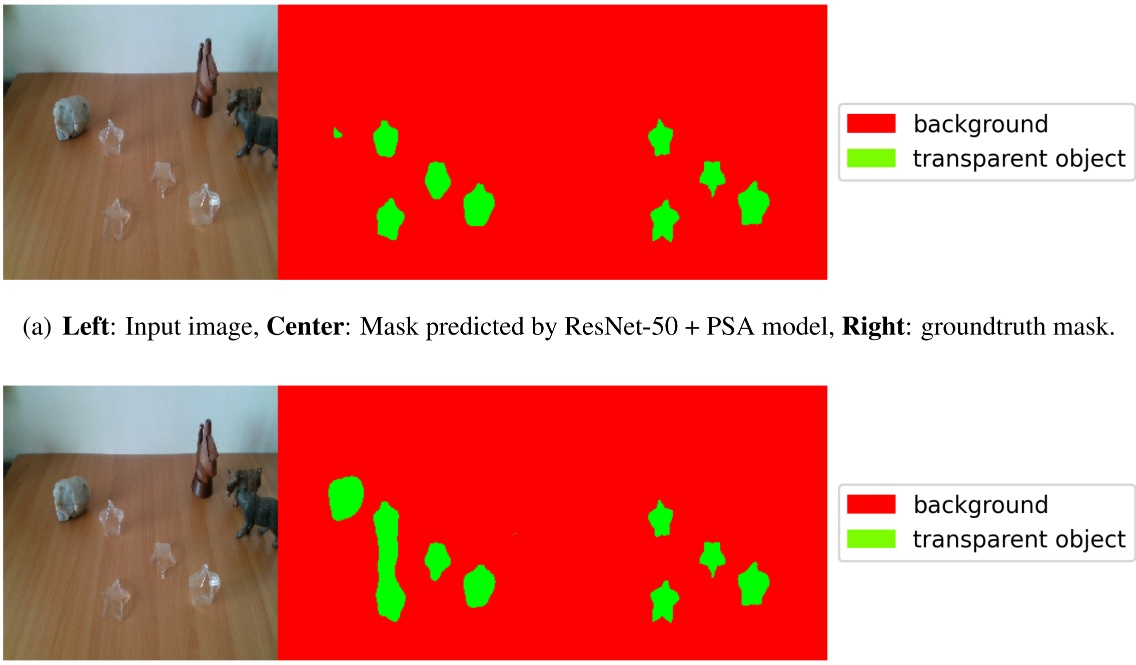
(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

Figure 28: Visualization showing the comparison of the transparent object segmentation prediction masks, for a real-world test set sample with four transparent objects and four other objects, between finetuned ResNet-50 + PSA and finetuned DRN models.

models for the glass square potion object from the synthetic test set. From Figure 35(a) and Figure 35(b), it can be clearly observed that the normals estimated by both models are equally bad. For some of the objects, normals are non-existent in the prediction by both models. However, the normals estimated for the tray is better by both models.

Figure 36 shows the visualization of the estimated surface normals by ResNet-50 + PSA and DRN models for a sample image with single transparent and multiple non-transparent objects from the real-world test set. From Figure 36(a) and Figure 36(b), it can be clearly observed that the normals estimated by both models are equally bad for most of the objects. For the black object, normals are non-existent in the prediction by the DRN model. However, the normals estimated for the transparent object slightly is better by the ResNet-50 + PSA model when compared with that of the DRN model.

Figure 37 shows the visualization of the estimated surface normals by ResNet-50 + PSA and DRN models for a sample image with multiple transparent and non-transparent objects from the real-world test set. From Figure 37(a) and Figure 37(b), it can be clearly observed that the normals estimated by the ResNet-50 + PSA model are better for some objects when compared with that of the DRN model. For the two transparent objects i.e. star-shaped and the small bottle left of it, normals estimated by the ResNet-50 + PSA model are better than that of the DRN model. For the transparent object in the extreme left of the image, a portion of the normals estimated by the ResNet-50 + PSA model are better than that of the DRN model. However, for the other two transparent objects and the rest of the non-transparent objects, the normals estimated are bad by both models.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

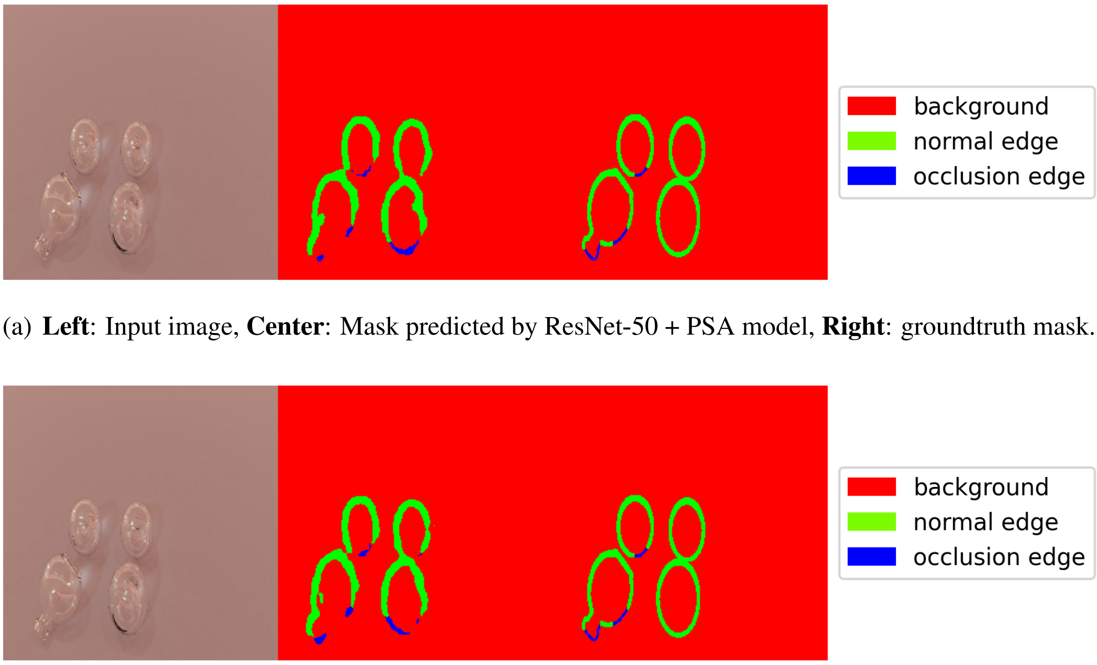
Figure 29: Visualization showing the comparison of the transparent object segmentation prediction masks, for a real-world test set sample with four transparent objects and four other objects, between finetuned ResNet-50 + PSA vs finetuned DRN models.

5.3.5 Depth estimation

Figure 38 shows the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image with single transparent and multiple non-transparent objects from the real-world test set. From Figure 38(a) and Figure 38(b), it can be clearly observed that both the estimated depth maps are good for the transparent object. However, if we closely look at the sharp edges of the tree bath bomb transparent object, the estimated depth map with the ResNet-50 + PSA model has captured the sharp edge shape for the object better than the DRN model.

Figure 39 shows the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image with multiple transparent and non-transparent objects from the real-world test set. From Figure 39(a) and Figure 39(b), it can be clearly observed that the estimated depth values with the DRN model are better than that of the ResNet-50 + PSA model for the right bottom transparent object. However, for the other three objects, upon closer inspection, it can be observed that the estimated depth values are better with the ResNet-50 + PSA model when compared with that of the DRN model.

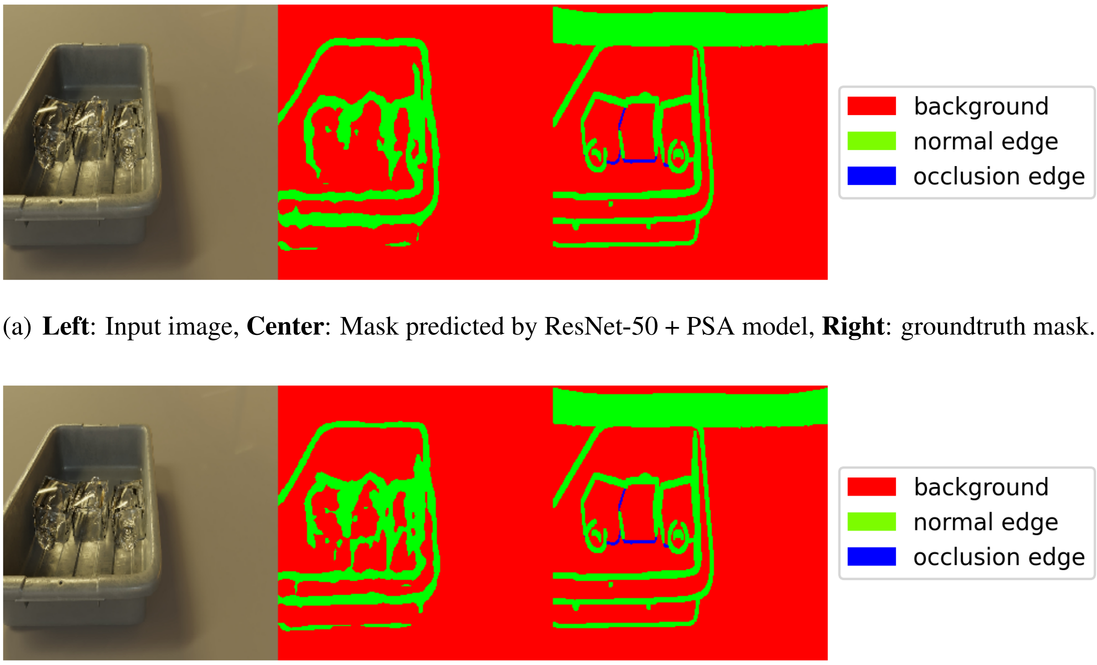
Figure 40 shows the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image with multiple transparent and non-transparent objects from the real-world test set. From Figure 40(a) and Figure 40(b), it can be clearly observed that the estimated depth values with the DRN model are better than that of the ResNet-50 + PSA model for the right bottom transparent object. However, for the other three objects, upon closer inspection, it can be observed that the estimated depth values are better with the ResNet-50 + PSA model when compared with that



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

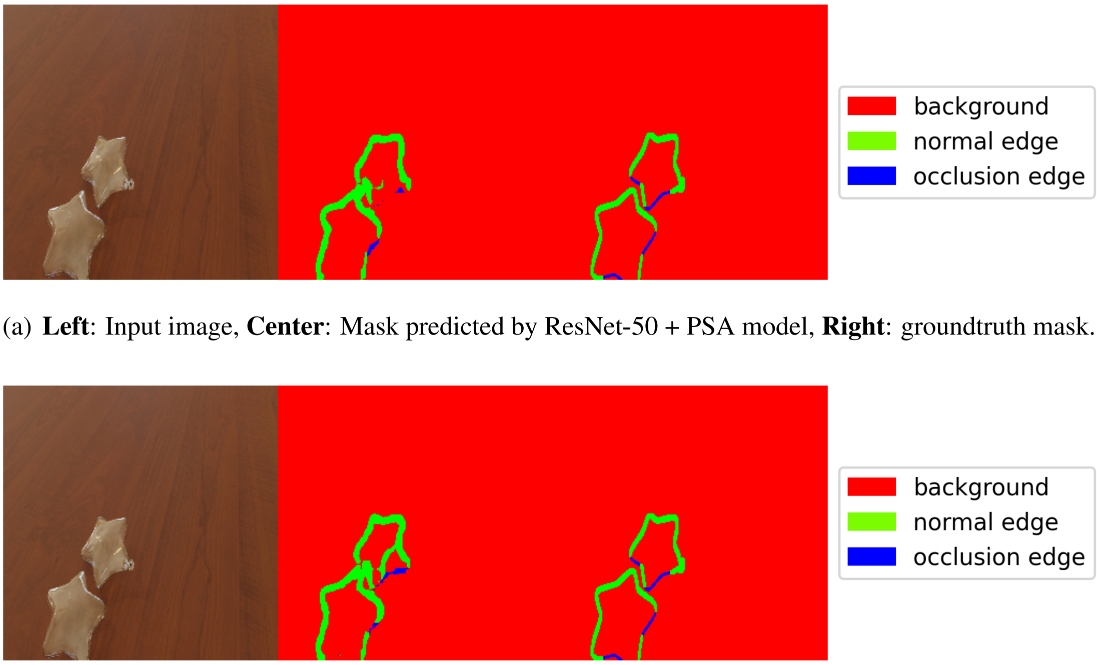
Figure 30: Visualization showing the comparison of the boundary segmentation prediction masks between ResNet-50 + PSA and DRN models for glass round potion object from the synthetic test set.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

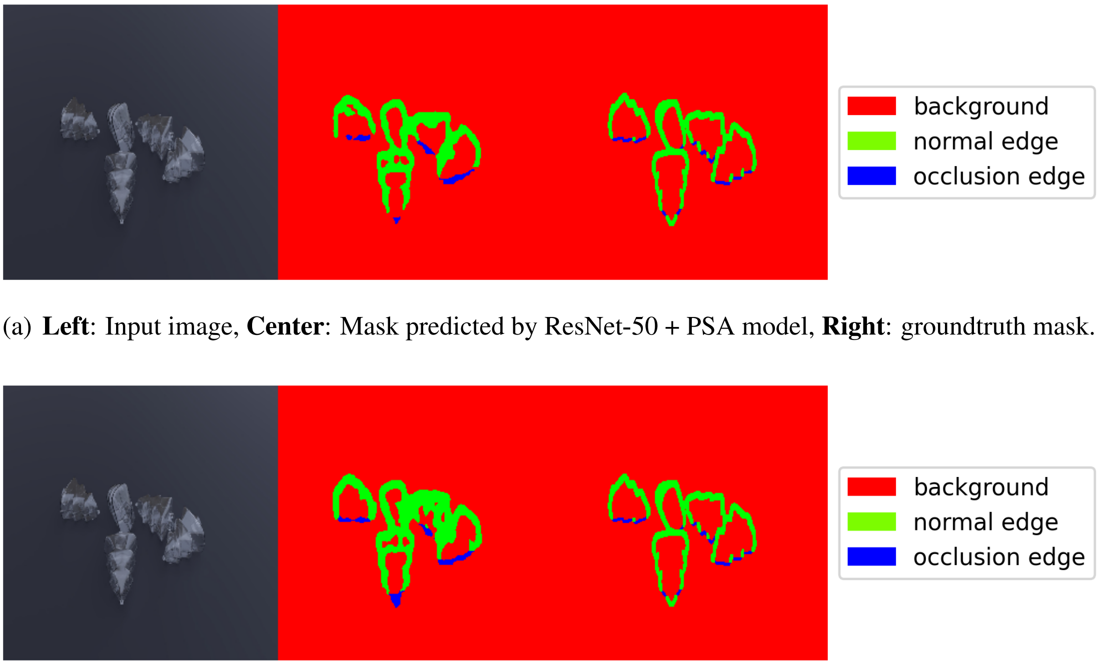
Figure 31: Visualization showing the comparison of the boundary segmentation prediction masks between ResNet-50 + PSA and DRN models for glass square potion object from the synthetic test set.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

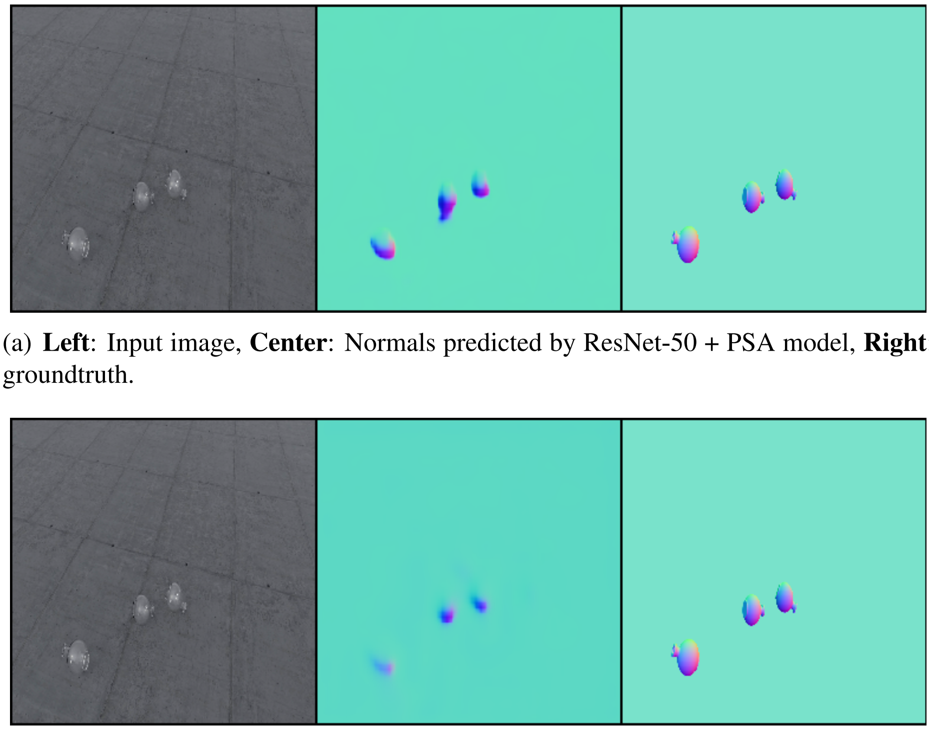
Figure 32: Visualization showing the comparison of the boundary segmentation prediction masks between ResNet-50 + PSA and DRN models for star bath bomb object from the synthetic test set.



(a) **Left**: Input image, **Center**: Mask predicted by ResNet-50 + PSA model, **Right**: groundtruth mask.

(b) **Left**: Input image, **Center**: Mask predicted by DRN model, **Right**: groundtruth mask.

Figure 33: Visualization showing the comparison of the boundary segmentation prediction masks between ResNet-50 + PSA and DRN models for tree bath bomb object from the synthetic test set.



(a) **Left**: Input image, **Center**: Normals predicted by ResNet-50 + PSA model, **Right**: groundtruth.

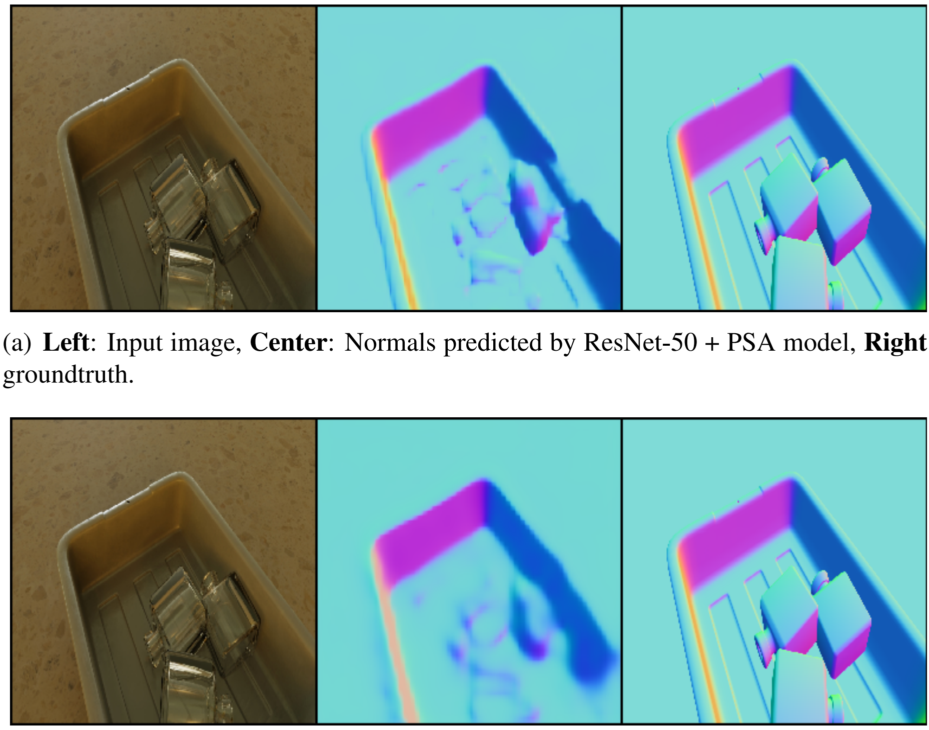
(b) **Left**: Input image, **Center**: Normals predicted by DRN model, **Right**: groundtruth.

Figure 34: Visualization showing the comparison of the estimated surface normal between ResNet-50 + PSA and DRN models for glass round potion object from the synthetic test set.

of the DRN model.

Figure 41 shows the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image with multiple transparent and non-transparent objects from the real-world test set. From Figure 41(a) and Figure 41(b), it can be clearly observed that the estimated depth values with the DRN model are better than that of the ResNet-50 + PSA model for the right bottom transparent object. However, for the other three transparent objects, it can be observed that the estimated depth values are equally good with both models. There were some other important observations in this test sample. On the right side of the right bottom transparent object the distinction in the depth values between the object and tray has been better estimated with the DRN model when compared with that of the ResNet50 - PSA model. On the top side of the top transparent object, the distinction in the depth values between the object and tray has been better estimated with the DRN model when compared with that of the ResNet50 - PSA model.

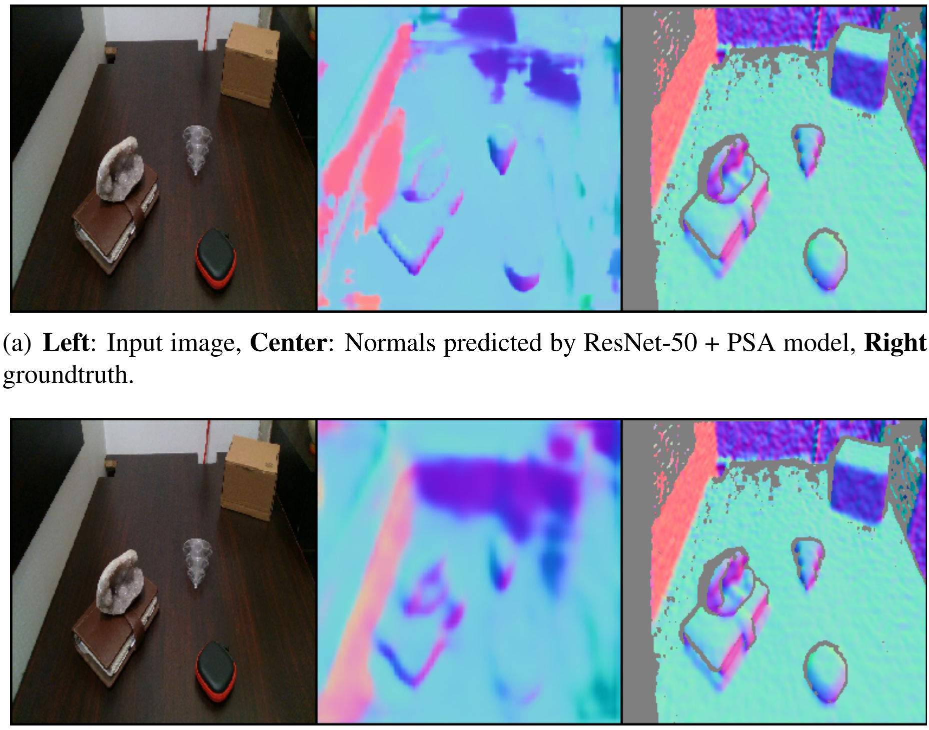
A few more visualizations of the estimated depth maps for some test samples for the real-world dataset called the `IRL-transparent-objects-set` are included in section 6.4.



(a) **Left**: Input image, **Center**: Normals predicted by ResNet-50 + PSA model, **Right**: groundtruth.

(b) **Left**: Input image, **Center**: Normals predicted by DRN model, **Right**: groundtruth.

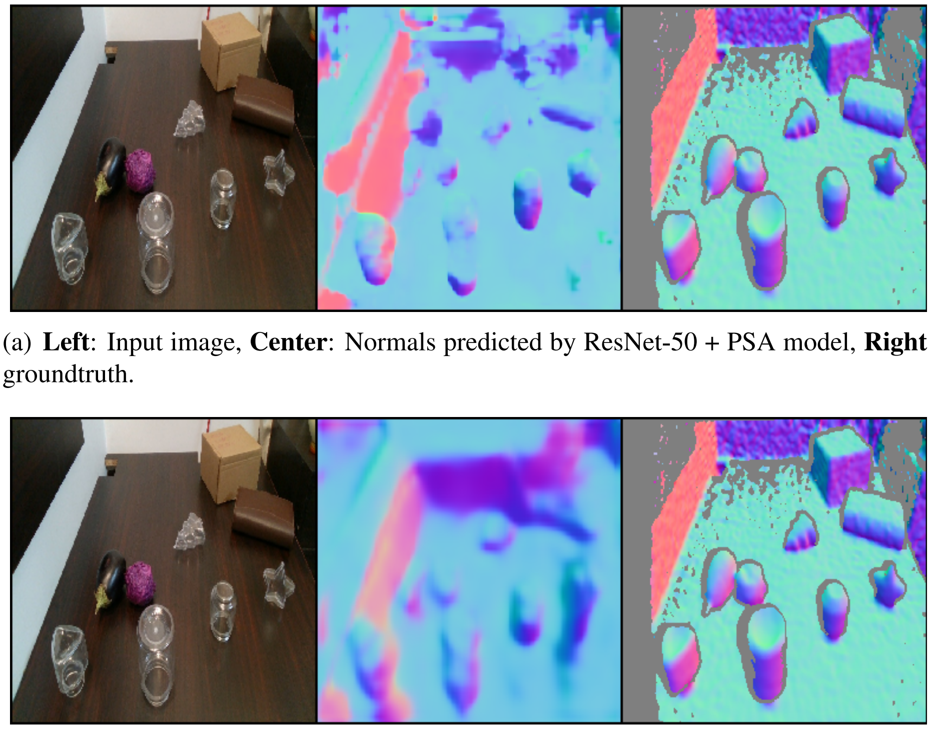
Figure 35: Visualization showing the comparison of the estimated surface normal between ResNet-50 + PSA and DRN models for glass square potion object from the synthetic test set.



(a) **Left**: Input image, **Center**: Normals predicted by ResNet-50 + PSA model, **Right**: groundtruth.

(b) **Left**: Input image, **Center**: Normals predicted by DRN model, **Right**: groundtruth.

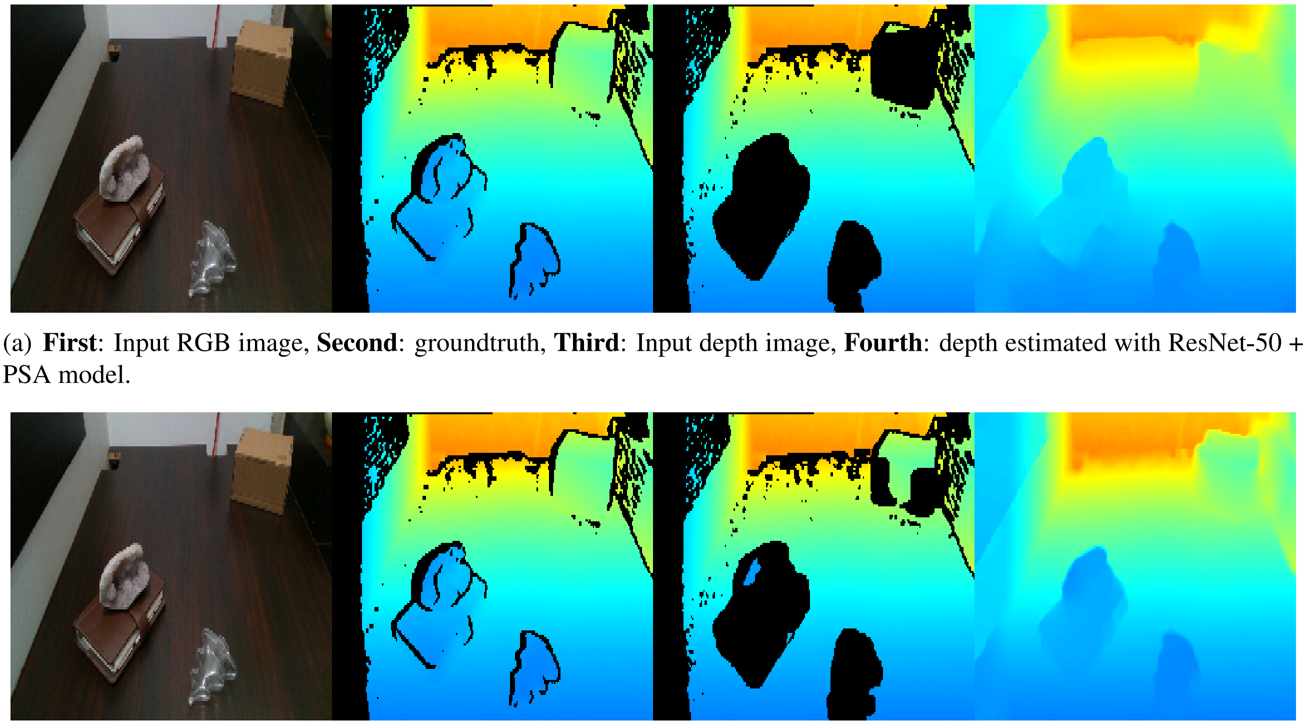
Figure 36: Visualization showing the comparison of the estimated surface normal between ResNet-50 + PSA and DRN models for a sample image from the real-world test set with multiple objects.



(a) **Left**: Input image, **Center**: Normals predicted by ResNet-50 + PSA model, **Right**: groundtruth.

(b) **Left**: Input image, **Center**: Normals predicted by DRN model, **Right**: groundtruth.

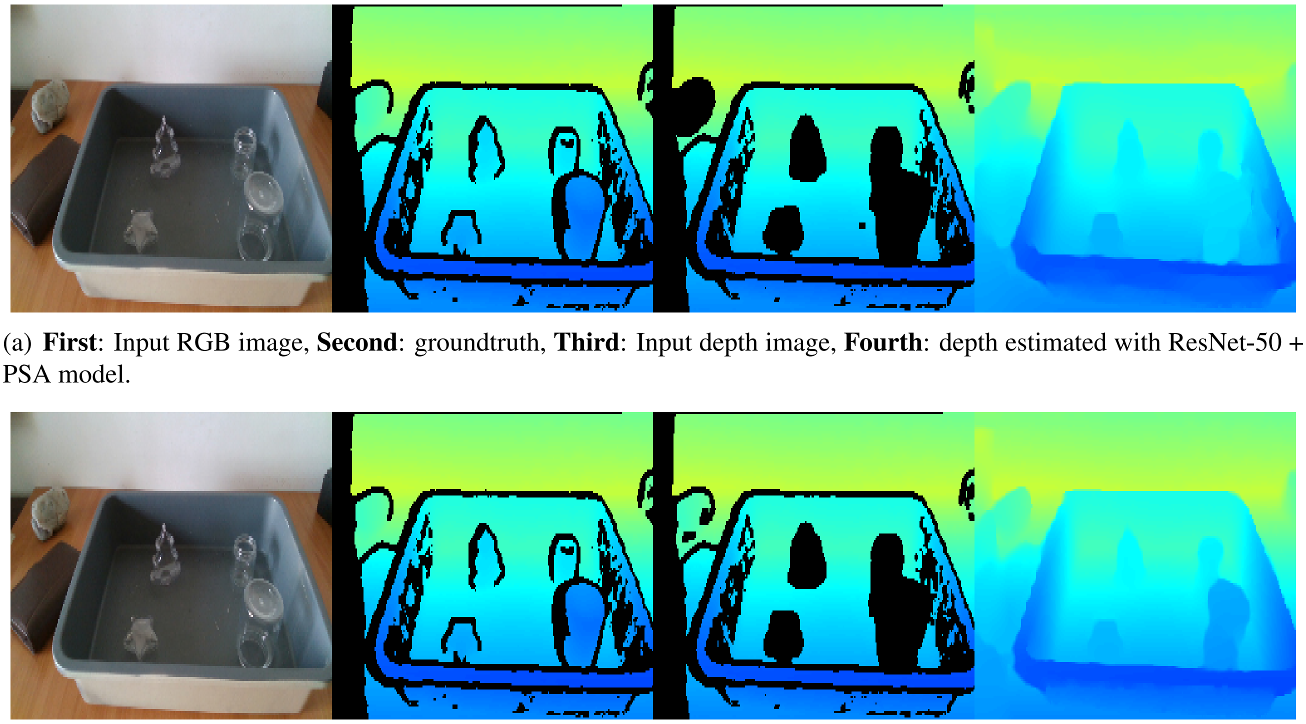
Figure 37: Visualization showing the comparison of the estimated surface normal between ResNet-50 + PSA and DRN models for a sample image from the real-world test set with multiple objects.



(a) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with ResNet-50 + PSA model.

(b) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with DRN model.

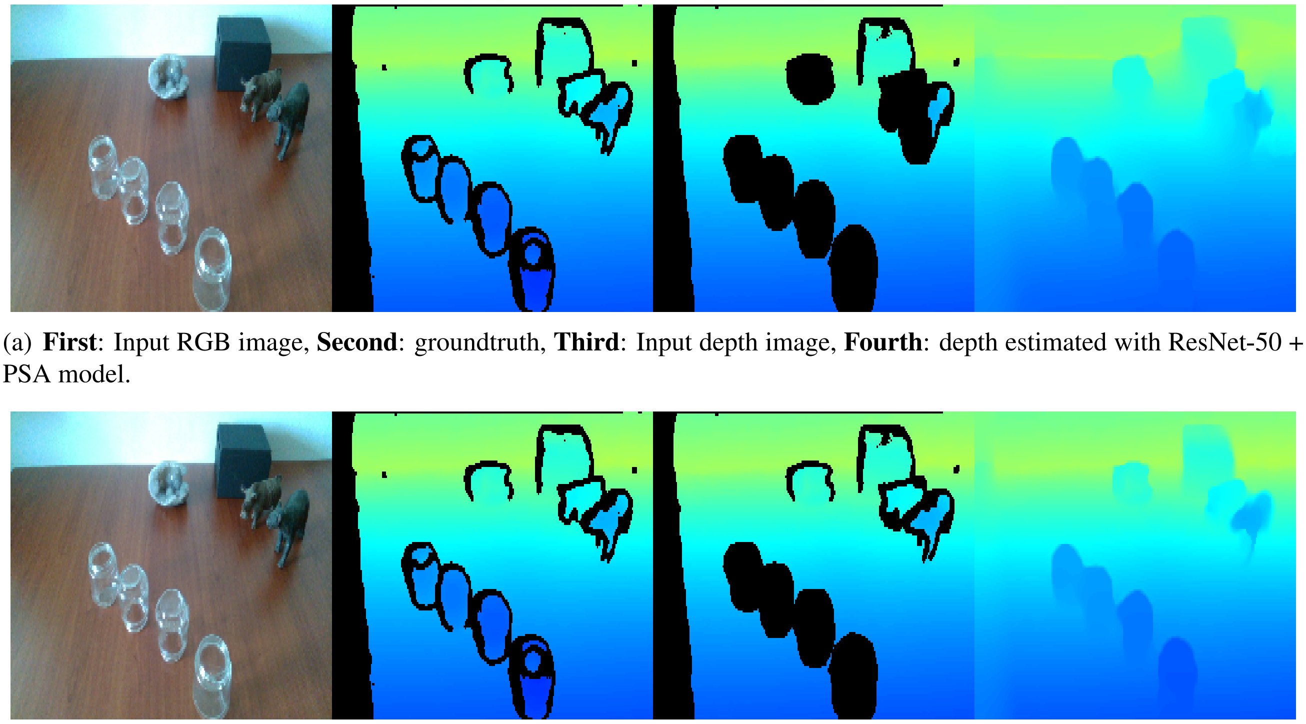
Figure 38: Visualization showing the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image from the real-world test set with multiple objects.



(a) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with ResNet-50 + PSA model.

(b) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with DRN model.

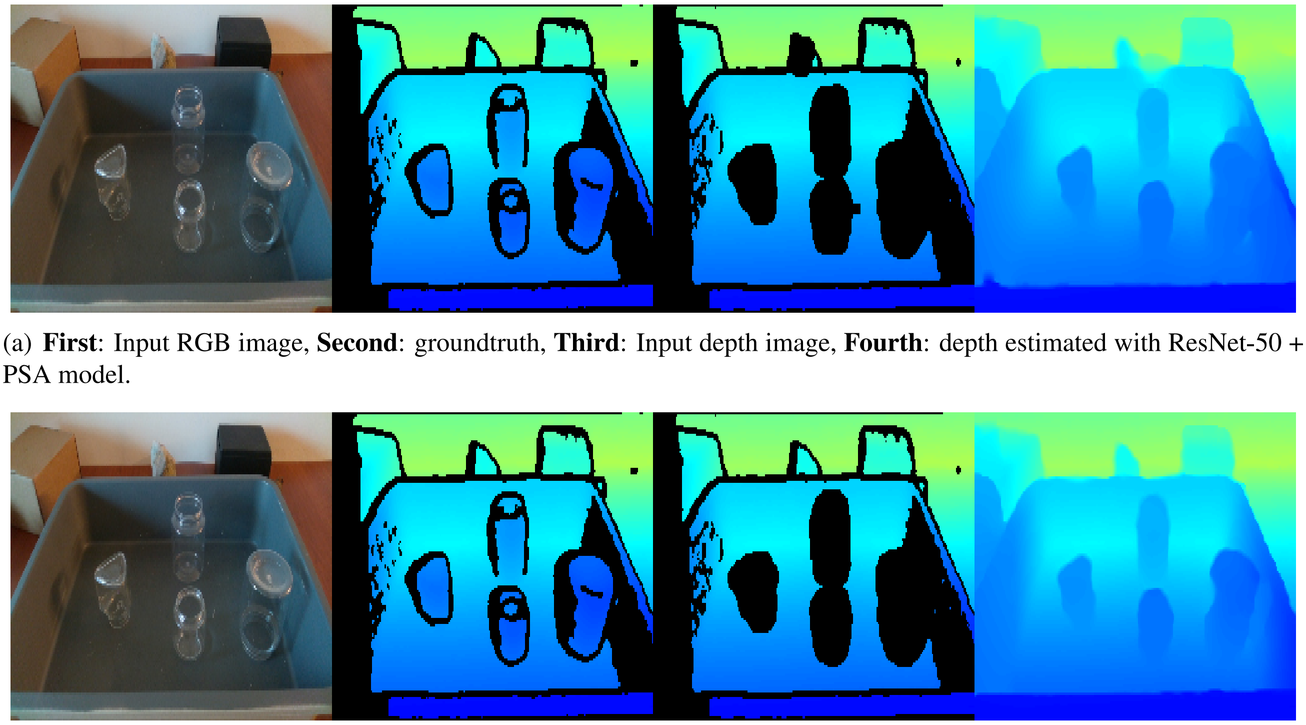
Figure 39: Visualization showing the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image from the real-world test set with multiple objects.



(a) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with ResNet-50 + PSA model.

(b) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with DRN model.

Figure 40: Visualization showing the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image from the real-world test set with multiple objects.



(a) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with ResNet-50 + PSA model.

(b) **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: depth estimated with DRN model.

Figure 41: Visualization showing the comparison of the estimated depth maps between ResNet-50 + PSA and DRN models for a sample test image from the real-world test set with multiple objects.

6 Discussion and Conclusion

In this chapter, a brief discussion of this research is discussed in section 6.1, the main research questions are answered in section 6.2, the summary of the main contributions of this research is discussed in section 6.3 and a direction towards the future work is presented in section 6.4.

6.1 Discussion

The depth estimation task for the transparent objects is a hard task. With the clearGrasp pipeline, the depth estimation task for transparent objects is further divided into sub-tasks such as transparent object segmentation, boundary segmentation, and surface normal estimation. The predictions of these models are used for the final depth estimation task. From the experimental results, it is clear that the proposed ResNet-50 + PSA model outperforms the DRN model on the synthetic and real-world datasets for every sub-task as reported in section 5.2.1, section 5.2.2, and section 5.2.3.

From the quantitative and qualitative analysis reported in section 5.2.1 and section 5.3.1 and section 5.3.2, it was clear that the models for the transparent object segmentation task had some difficulties with the false positive rate on the real-world test set since the synthetic clearGrasp train set had only transparent objects. To, overcome this issue, a small set of real-world samples from the COCO dataset was added to the synthetic train set, and the models were finetuned. This resulted in a boost in the performance on the real-world test set as reported in section 5.2.1 and section 5.3.2. Similarly, for the surface normal estimation task, the clearGrasp synthetic train dataset was combined with a small set of real-world samples from the NYU_v2 dataset.

Furthermore, the ResNet-50 + PSA model when used in the clearGrasp pipeline clearly outperforms the clearGrasp pipeline with DRN model, for the transparent objects' depth estimation task as reported in section 5.2.4. However, a major challenge might be with the application since the clearGrasp pipeline takes the predictions from three models trained for different sub-tasks to estimate the depth values for the transparent objects and this might be a bit time expensive. Although there are constant improvements in the hardware with better and more powerful GPUs being released, the method if it has to be used in any environment in a realistic robotic system with embedded hardware, needs to be fast enough. This completely depends on the environment where such a robotic system might be finally deployed.

From the qualitative analysis reported in section 5.3, it was clear that in some cases, the ResNet-50 + PSA model had poor performance when compared with the DRN model on some samples. On some other samples, the observation was vice versa. Overall, it was observed that both models generalized on the real-world samples although trained with a majority of synthetic samples for transparent object segmentation and surface normal estimation tasks as reported in section 5.3.1, section 5.3.2, and section 5.3.4. From the reported analysis in section 5.3.3, it was clear that the models had difficulty in segmenting the boundaries accurately in some cases due to the sparse nature of the task. Finally, from the reported analysis in section 5.3.5, it was clear that the ResNet-50 + PSA model produced better depth values for transparent objects when compared with the DRN model when used in the clearGrasp pipeline. Overall, the depth values estimated for the transparent objects in the real-world samples generalized better with the ResNet-50 + PSA model when compared with that of the DRN model.

In this research, a relatively small real-world test dataset called `IRL-transparent-objects-set` with 87 samples across 4 transparent objects was collected with `Kinect-v1` RGB-D camera. In this dataset, data was collected for both isolated and multi-object scenarios. The procedure of the data collection experiment is explained in detail in section 4.2.

6.2 Answers to research questions

How can transparent objects be detected and localized in the images?

The robots can effectively detect and localize transparent objects by using models trained for the transparent object segmentation task. Because, if anything has to be done first, it is to identify the pixels in an image that belong to transparent objects since recognition of transparent objects is a hard task when compared with that of normal objects. This was evident from the experimental results reported in section 5.2.1, section 5.3.1, and section 5.3.2. This information can be supplemented by using models trained for boundary segmentation task and surface normal estimation task that give extra information about the contact and occlusion edges along with the nature of the surfaces where transparent objects can be found. This was evident from the results reported in section 5.2.2 and section 5.2.3.

Is it possible to train models to estimate refined depth values that can be used for grasp planning? How will they perform compared with the current state-of-the-art?

From the experimental results reported in section 5.2.4 and section 5.3.5, it is clear that encoder-decoder models can be trained for transparent object segmentation, boundary segmentation, and surface normal estimation sub-tasks and used with the clearGrasp depth estimation pipeline to estimate depth values for the transparent objects which can be further used for grasp planning of transparent objects. From the experimental results, it was clear that the proposed ResNet-50 + PSA model outperformed the DRN model for the depth estimation task on the synthetic and real-world test sets with known and novel objects.

How can the performance of depth estimation for transparent objects pipeline be evaluated and compared and what are its main challenges?

From the evaluation metrics discussed in section 3.7 and the quantitative results discussed in section 5.2.4, it is clear that RMSE, MAE, MeanREL along with the $\delta_{1.05}$, $\delta_{1.10}$, and $\delta_{1.25}$ can be used to evaluate the depth estimation pipeline for transparent objects. However, there are certain challenges. One of them is the need of groundtruth depth data for evaluation of the performance, since the depth values from an RGB-D camera would be inaccurate. For this, two separate data collection procedures had to be performed one where raw transparent objects are used to collect input RGB data and another where the transparent objects have to be filled with some things and the groundtruth depth data can be collected. Such a methodology has two challenges, first one is that the objects and the camera position need to be fixed for both the data collection procedures and the second one is that even though the transparent objects are filled with something to collect the groundtruth depth data, this might not be completely accurate since the thickness of the transparent object body might not be taken into account meaning only the portion of the transparent object which can hold whatever is used to fill them is the portion for which the groundtruth depth data is collected. This might not be completely accurate but can be used along with the thickness of the transparent object body information to estimate accurate groundtruth depth data.

6.3 Summary of Main Contributions

In this section, the main contributions of this research are presented.

- The main contribution of this research is the proposed ResNet-50 + PSA model that outperformed the DRN model on all the individual sub-tasks i.e. the transparent object segmentation, boundary segmentation, and the surface normal estimation tasks.
- The proposed ResNet-50 + PSA model outperforms the DRN model when used in the clearGrasp pipeline for the transparent object depth estimation task.
- From some of the observations from this research, it can also be argued that the clearGrasp synthetic dataset without many non-transparent general objects increases the false positive rates in the case of the transparent object segmentation task. So, there is a need for a synthetic dataset with the presence of both transparent and non-transparent objects in a balanced way. Otherwise, additional samples from a real-world dataset might have to be used for finetuning.
- By using a very small amount of real-world samples with non-transparent objects when compared with that of a large synthetic dataset with transparent objects, a performance boost was achieved for the transparent object segmentation task on the real-world test sets.
- A small additional real-world test set, called `IRL-transparent-objects-set`, with 87 samples specifically for the depth estimation task for transparent objects with isolated and multi-object scenarios.

6.4 Future Work

In this section, some of the possible future research directions are discussed.
The raw depth values directly from any RGB-D camera for transparent objects are usually inaccurate and noisy due to the nature of the transparent objects. From this research, it was clear that the proposed ResNet-50 + PSA model outperformed the DRN model for the transparent object depth estimation task with the clearGrasp pipeline using RGB images as input. The depth estimation method from this research can be used with a grasp planning algorithm to further evaluate the performance of different grasp planning methods especially for the transparent objects since one of the main limitations is the inaccurate and noisy raw depth values from any RGB-D camera.

Another possible direction of research is to introduce the concept of uncertainty for individual subtasks, especially for transparent objects. For the surface normal estimation task for general scenes, there is a recent research work [56] by Bae et al. for estimating and exploiting the aleatoric uncertainty that is inherent to the data mostly due to camera sensor noise. This idea can be applied to the surface normal estimation task specifically for transparent objects. This can be further extended for estimating epistemic uncertainty that may be produced by the model due to lack of training data, model misspecification, and so on. These ideas can also be applied to the other sub-tasks in this depth estimation pipeline i.e. the transparent object segmentation task and the occlusion boundary segmentation task.

Bibliography

[1] M. ul Hassan, “Vgg16 – convolutional network for classification and detection,” 2018.

[2] A. Meyer, “Saliency detection convolutional autoencoder,” 2017.

[3] S. S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, “Cleargrasp: 3d shape estimation of transparent objects for manipulation,” *CoRR*, vol. abs/1910.02550, 2019.

[4] Y. Li, L. Schomaker, and S. H. Kasaei, “Learning to grasp 3d objects using deep residual u-nets,” *CoRR*, vol. abs/2002.03892, 2020.

[5] S. H. Kasaei, N. Shafii, L. S. Lopes, and A. M. Tomé, “Interactive open-ended object, affordance and grasp learning for robotic manipulation,” *CoRR*, vol. abs/1904.02530, 2019.

[6] S. Kasaei, J. Sock, L. Seabra Lopes, A. Maria Tome, and T.-K. Kim, “Perceiving, learning, and recognizing 3d objects: An approach to cognitive service robots,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, April 2018.

[7] J. Jiang, G. Cao, J. Deng, T.-T. Do, and S. Luo, “Robotic perception of transparent objects: A review,” *ArXiv*, vol. abs/2304.00157, 2023.

[8] S. Kumra, S. Joshi, and F. Sahin, “Antipodal robotic grasping using generative residual convolutional neural network,” *CoRR*, vol. abs/1909.04810, 2019.

[9] D. Morrison, P. Corke, and J. Leitner, “Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach,” *CoRR*, vol. abs/1804.05172, 2018.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[11] H. Liu, F. Liu, X. Fan, and D. Huang, “Polarized self-attention: Towards high-quality pixel-wise regression,” *CoRR*, vol. abs/2107.00782, 2021.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.

[13] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *CoRR*, vol. abs/1802.02611, 2018.

[14] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014.

[15] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.

[16] D. E. Rumelhart and J. L. McClelland, *Learning Internal Representations by Error Propagation*, pp. 318–362. 1987.

[17] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.

[18] Hecht-Nielsen, “Theory of the backpropagation neural network,” in *International 1989 Joint Conference on Neural Networks*, pp. 593–605 vol.1, 1989.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[22] J. Sanchez, F. Perronnin, T. E. J. Mensink, and J. Verbeek, “Image classification with the fisher vector: Theory and practice,” *International Journal of Computer Vision*, 2013.

[23] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

[24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[25] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” 2018.

[26] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *CoRR*, vol. abs/1311.2901, 2013.

[27] H. Gholamalinezhad and H. Khosravi, “Pooling methods in deep neural networks, a review,” *CoRR*, vol. abs/2009.07485, 2020.

[28] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, “Non-linear convolution filters for cnn-based learning,” *CoRR*, vol. abs/1708.07038, 2017.

[29] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, “A comprehensive survey and performance analysis of activation functions in deep learning,” *CoRR*, vol. abs/2109.14545, 2021.

[30] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *CoRR*, vol. abs/1411.4038, 2014.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[32] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015.

[33] D. Bank, N. Koenigstein, and R. Giryes, “Autoencoders,” *ArXiv*, vol. abs/2003.05991, 2020.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *CoRR*, vol. abs/1409.0575, 2014.

[35] Y. Chen, W. Li, X. Chen, and L. V. Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," *CoRR*, vol. abs/1812.05040, 2018.

[36] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. M. Seitz, "Soccer on your tabletop," *CoRR*, vol. abs/1806.00890, 2018.

[37] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *CoRR*, vol. abs/1703.09312, 2017.

[38] S. H. Kasaei and M. Kasaei, "Mvgrasp: Real-time multi-view 3d object grasping in highly cluttered environments," *CoRR*, vol. abs/2103.10997, 2021.

[39] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, Jan. 2015.

[40] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," *CoRR*, vol. abs/2003.13948, 2020.

[41] H. Mei, X. Yang, Y. Wang, Y. Liu, S. He, Q. Zhang, X. Wei, and R. W. Lau, "Don't hit me! glass detection in real-world scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] H. Mei, B. Dong, W. Dong, J. Yang, S.-H. Baek, F. Heide, P. Peers, X. Wei, and X. Yang, "Glass segmentation using intensity and spectral polarization cues," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12612–12621, 2022.

[43] I. Lysenkov and V. Rabaud, "Pose estimation of rigid transparent objects in transparent clutter," in *2013 IEEE International Conference on Robotics and Automation*, pp. 162–169, 2013.

[44] C. Phillips, M. Lecce, and K. Daniilidis, "Seeing glassware: from edge detection to pose estimation and shape recovery," 06 2016.

[45] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *CoRR*, vol. abs/2110.14217, 2021.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[47] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[49] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017.

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[51] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017.

[52] T. Chanwimalueang and D. P. Mandic, "Cosine similarity entropy: Self-correlation-based complexity analysis of dynamical systems," *Entropy*, vol. 19, no. 12, 2017.

[53] Y. Zhang and T. A. Funkhouser, "Deep depth completion of a single RGB-D image," *CoRR*, vol. abs/1803.09326, 2018.

[54] M. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," vol. 10072, pp. 234–244, 12 2016.

[55] S. Hickson, K. Raveendran, A. Fathi, K. Murphy, and I. A. Essa, "Floors are flat: Leveraging semantics for real-time surface normal prediction," *CoRR*, vol. abs/1906.06792, 2019.

[56] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," *CoRR*, vol. abs/2109.09881, 2021.

[57] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[58] R. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

Appendix

The small real-world test dataset with transparent objects collected with the `Kinect-v1` called the `IRL-transparent-objects-set` can be found in the following GitHub repository `https://github.com/AbhishekRS4/irl_kinect_transparent_objects_set`. The code used in this research can be found in the GitHub repository `https://github.com/AbhishekRS4/clearGraspPlus`.

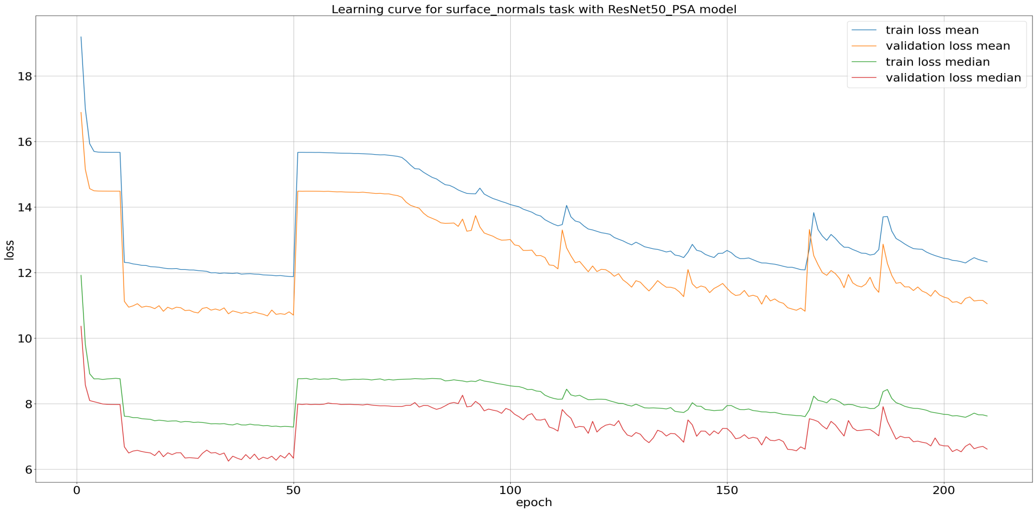


Figure 42: Loss curve of ResNet-50 + PSA model with SGD optimizer on the surface normal estimation task.

Figure 42 shows the loss curve for the surface normal estimation task with the proposed ResNet-50 + PSA model with the SGD optimizer. It can be clearly observed that there are oscillations at multiple learning regions. The task of learning to estimate surface normals, which is a regression task, is a relatively harder task when compared to other segmentation tasks. So, the oscillations at quite a few learning regions are expected for the surface normal estimation task.

Figure 43, Figure 44, Figure 45, Figure 46, and Figure 47 shows the estimated depth maps with ResNet-50 + PSA model for sample test images with different transparent objects from the real-world test set `IRL-transparent-objects-set`. The estimated depth maps are good for some transparent objects but not so good for the rest of them.

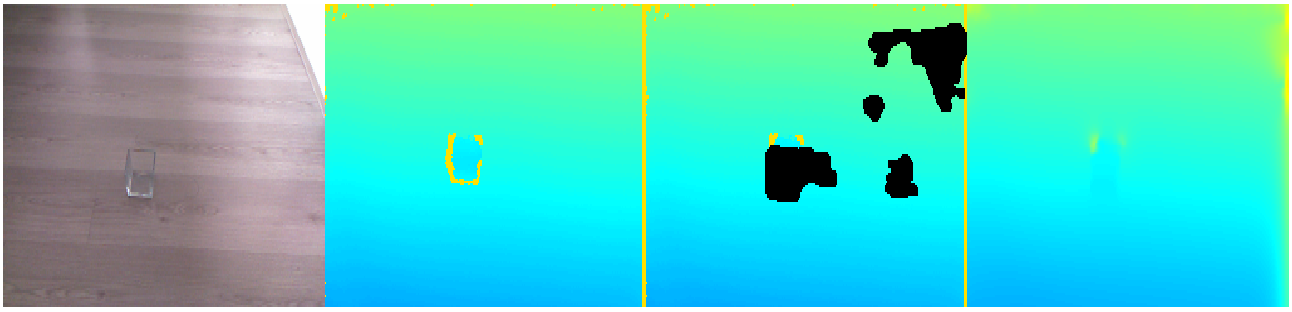


Figure 43: Visualization showing the estimated depth map with ResNet-50 + PSA model for a sample test image from the real-world test set `IRL-transparent-objects-set`. **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: estimated depth.

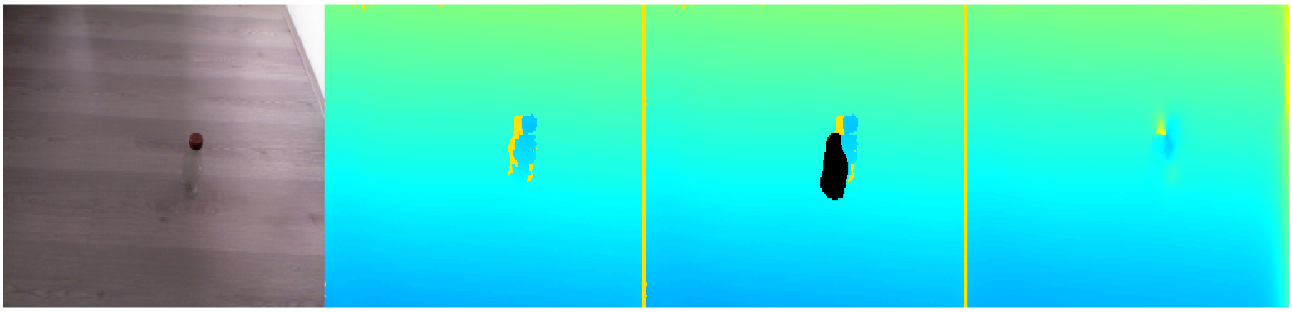


Figure 44: Visualization showing the estimated depth map with ResNet-50 + PSA model for a sample test image from the real-world test set `IRL-transparent-objects-set`. **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: estimated depth.

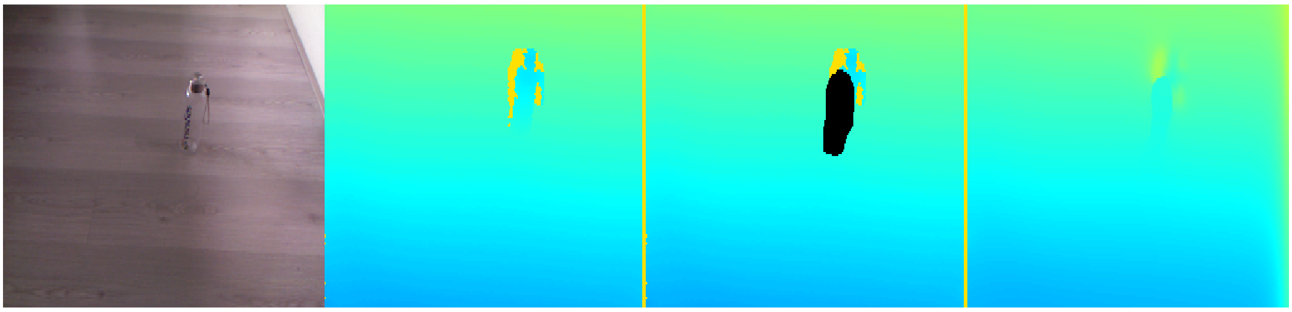


Figure 45: Visualization showing the estimated depth map with ResNet-50 + PSA model for a sample test image from the real-world test set `IRL-transparent-objects-set`. **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: estimated depth.

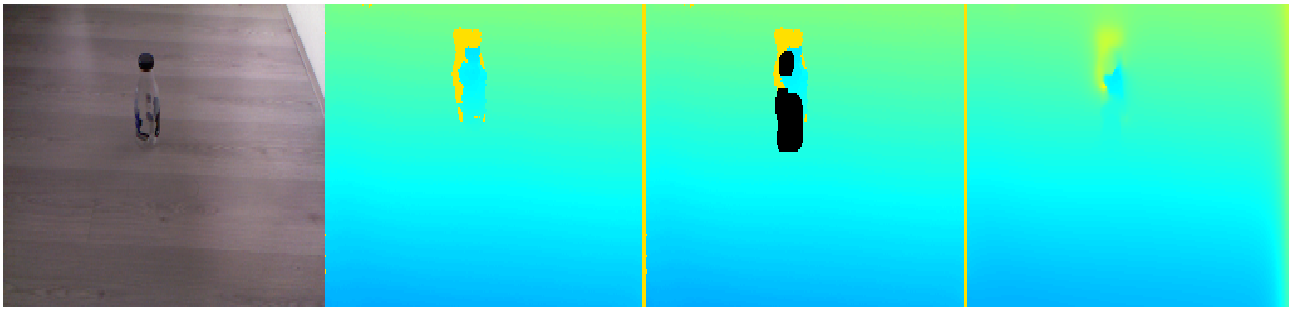


Figure 46: Visualization showing the estimated depth map with ResNet-50 + PSA model for a sample test image from the real-world test set `IRL-transparent-objects-set`. **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: estimated depth.

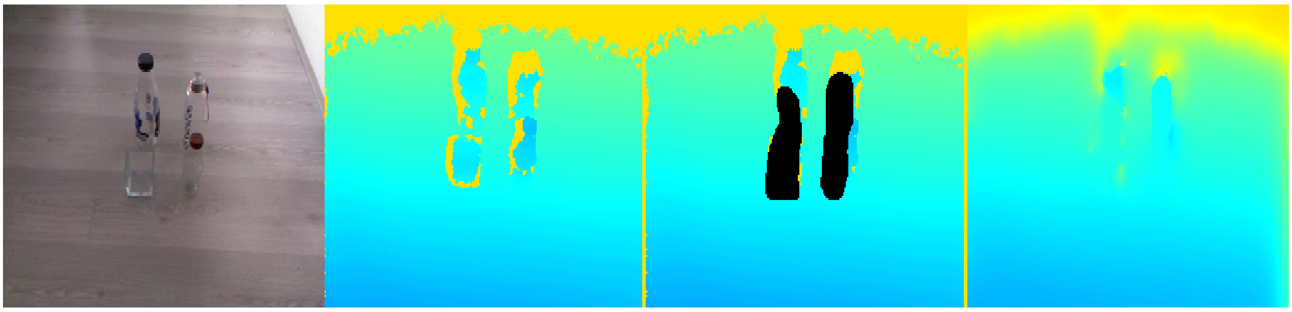


Figure 47: Visualization showing the estimated depth map with ResNet-50 + PSA model for a sample test image from the real-world test set `IRL-transparent-objects-set`. **First**: Input RGB image, **Second**: groundtruth, **Third**: Input depth image, **Fourth**: estimated depth.