

Deep Learning Based Object Recognition from RGB-D Images

Yandong Deng

A thesis submitted to the Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2019

School of Engineering, Computer and Mathematical Sciences

Abstract

In the field of computer vision, image recognition has been developing for a long time. Text recognition, license plate recognition, etc. are already very mature technologies. In recent years, with the development of deep neural networks, the technology of object recognition has been further developed. Starting with RCNN(Region-CNN), although the accuracy of recognition is constantly improving. But at the same time, object recognition faces many challenges. These challenges include lighting conditions, similar colors for object and backgrounds, and more.

To meet these challenges, this thesis proposes a method to improve the accuracy of object recognition model by using depth information. This method uses the Grab-cut algorithm segmentation the depth information and uses the depth map after the segmentation to complete the segmentation of the target object. This method avoids the impact of complex scenes on object recognition. Using this method also reduces the effects of illumination, shadows, colors, etc. on recognition accuracy. The effectiveness of our proposed method is demonstrated by testing the depth map database we collected. As a result of the experiment, the average accuracy of the method can be improved by 5% to 10%.

Table of Contents

Deep Learning Based Object Recognition from RGB-D Images.....	1
Abstract.....	i
Table of Contents.....	ii
List of Figures.....	iv
List of Tables	vi
Attestation of Authorship.....	vii
Acknowledgment.....	viii
Chapter 1 Introduction.....	1
1.1 Background and Motivation	1
1.2 Objectives	4
1.3 Structure of the Thesis	5
Chapter 2 Literature Review.....	6
2.1 Review of Object Recognition	6
2.1.1 Object Recognition/Detection based on Machine learning	7
2.1.2 Object Recognition based on Deep Learning	9
2.2 RGB-D Images	13
2.2.1 Applications	14
2.2.2 Object Recognition Using RGB-D Images	16
2.2.3 Image Segmentation Based on Depth Map	17
2.2.4 Depth Image Datasets	19
Chapter 3 Methodology	21
3.1 Grab-cut Algorithm.....	21
3.1.1 Color model	22
3.1.2 Iterative Energy Minimization Segmentation.....	24
3.2 YOLO	24
3.2.1 YOLOv1	25
3.2.2 YOLO9000 & YOLOv3	26
3.3 Database Collection and Preprocessing.....	28
3.3.1 Database Collection.....	28
3.3.2 Preprocessing.....	31
3.4 Research Design	33
3.4.1 Experimental Plan.....	33
3.4.2 Training.....	34
3.4.3 Evaluation Method	34
Chapter 4 Experimental results and Analysis.....	36
4.1 Experimental results	36

4.1.1 Result of the first recognition	36
4.1.2 Segmentation of Depth Map.....	41
4.1.3 Results of the Second Experiment.....	44
4.2 Analyses and Observations.....	49
4.2.1 Comparison of Results.....	49
4.2.2 Other Observations	52
Chapter 5 Conclusions and Future Works	58
5.1 Conclusions	58
5.2 Future Works	59
References	61

List of Figures

Figure 3.1 Example of using the Grab-cut algorithm.....	22
Figure 3.2 The structure of the classifier Darknet-53 used by YOLOV3.....	27
Figure 3.3 Chair database example.....	30
Figure 3.4 Signal processing flow of homomorphic filtering.....	31
Figure 3.5 Example of originally acquired depth maps of the chair image.....	31
Figure 3.6 Example of a Full Format depth map after preprocessing.....	32
Figure 3.7 Example of a RAW Format depth map after preprocessing.....	32
Figure 3.8 Example of original chair RGB images acquired.....	33
Figure 3.9 Example of RGB image after preprocessing.....	33
Figure 4.1 Examples of recognition in the chair database.....	37
Figure 4.2 Examples of recognition in the chair database.....	38
Figure 4.3 Confidence score vs image perspective for the chair dataset.....	39
Figure 4.4 Examples of recognition in the suitcase database.....	40
Figure 4.5 Examples of recognition in the suitcase database.....	40
Figure 4.6 Suitcase database line chart.....	41
Figure 4.7 The workflow of the segmentation process.....	42
Figure 4.8 Example of depth maps in Raw and Full formats.....	43
Figure 4.9 Segmentation results of suitcase depth map.....	43
Figure 4.10 Segmentation results of chair depth map.....	44
Figure 4.11 Examples of recognition in the chair database.....	45
Figure 4.12 Examples of recognition in the chair database.....	45
Figure 4.13 Confidence score for chair images with different orientations.....	46
Figure 4.14 Examples of recognition in the suitcase database.....	47
Figure 4.15 Examples of recognition in the suitcase database.....	48
Figure 4.16 Confidence score for chair images with different orientations.....	68
Figure 4.17 Confidence scores for Chair dataset in Experiments 1 and 2.....	49
Figure 4.18 The results of Figure 4.17 in bar chart form.....	50

Figure 4.19 Confidence scores for Suitcase dataset in Experiments 1 and 2.....	51
Figure 4.20 Results in Figure 4.19 expressed in bar chart form.....	52
Figure 4.21 Example of undetected object.....	53
Figure 4.22 Chair database recognized example.....	54
Figure 4.23 Suitcase database recognized example	54
Figure 4.24 Comparison of different lighting conditions.....	56
Figure 4.25 RAW format depth map obtained under different lighting conditions.....	57

List of Tables

Table 4.1 Chair database detection results.....	37
Table 4.2 Suitcase database detection results.....	39
Table 4.3 Chair database detection confidence scores.....	45
Table 4.4 Suitcase database statistics.....	47

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature:  Date: 4 April 2019

Acknowledgment

My deepest thanks are to my primary supervisor Professor Edmund Lai who has provided me with much technical guidance and support. I believe that I could not have been able to achieve my master's degree without his invaluable help and supervision. Also, I would like to express my sincere gratitude to my friends for their help in life. And, I would like to deeply thank my parents for their financial support during my entire time of academic study in Auckland.

Yandong Deng

Auckland, New Zealand

April 2019

Chapter 1 Introduction

1.1 Background and Motivation

Identifying objects visually is an extremely easy task for humans, but not so for a computer. Image recognition refers to the process by which a computer processes an image and identifies objects that exist in that image. In general, image recognition is a two-step process. The first step is image segmentation [76,77,78] where the image is divided into some meaningful regions. In the second step, the features of each region are extracted and a classifier determines which kind of object that region of the image belongs to.

Research in image recognition started with text recognition [79, 80] in the 1950s. The objects to be identified were letters, numbers, and symbols. A popular application of text recognition is car license plate recognition [81, 82]. This technology is now very mature. The processing of digital images [83] began in the 1960s. Digital images have great advantages such as easy storage, convenient and compressible transmission, and not easy to be distorted. Image recognition has since been a very active area of research. Many different techniques have been invented for various purposes. But the features that are extracted from the images are typically human engineered which are quite specific to an application. With the development of deep neural networks [84, 85], research on image recognition has progressed very rapidly. For the PASCAL VOC dataset [86], the accuracy of recognition has increased from the initial 30% to the current 90%. In the 2012

ImageNet Large-Scale Recognition Challenge (ILSVRC) [88] competition, the Alex Net model achieved an accuracy of 85%, an improvement of over 10% compared with the best traditional model. The advantage of using neural networks is that the network learns relevant feature representations from the training images automatically, eliminating the need for feature engineering.

This is a turning point in the history of image recognition. These achievements shift the focus from traditional image recognition methods to methods using deep neural networks. In the ILSVRC 2013 competition, all participants use solutions and algorithms based on deep learning techniques. In the 2015 ILSVRC competition, Convolutional Neural Network (CNN) based algorithms achieved recognition accuracy exceeding 95%, higher than the recognition rate of humans. From 2017 to 2018, 29 out of 38 participants in the competition provided solutions exceeding 95% recognition accuracy, the highest being 97.3%. This illustrates the great potential of deep learning in image recognition.

Although deep learning has so far achieved great success in the field of image recognition, there are still many challenges that we need to face. The first challenge is how to improve the generalization capabilities of the model; how to make the existing model achieve good performance in the type of images that the network has not been trained on. Untrained scenes can create many problems for the network. Usually, a simple solution to this problem is to increase the training dataset. However, it is not always possible to have all scenes in the training dataset. The second challenge is how to make better use of small data sets. Identifying objects that have only been seen once is relatively easy for humans. However, it is very difficult for a computer. With existing technology,

if a large data set is used, the accuracy of recognition can be easily increased. But with the small data sets, the results are not as good as we wish. The third challenge is how to make computers understand the relationship of objects in the three-dimensional world. This is the focus of the research presented in this thesis.

The advent of depth cameras has enabled the 3D scene to be captured digitally. However, the cost of these cameras tends to be very high. More recently, low-cost depth cameras have become commercially available. Examples include Kinect [93] and ZED [94].

The availability of depth information has opened up a lot of possibilities. Two-dimensional face recognition technology has been developed for decades. However, it is still difficult to achieve high-accuracy recognition with face images taken from all kinds of angles, lighting conditions, and facial occlusion (e.g. by eyeglasses). Depth information makes it possible to improve accuracy [90]. Depth information can also be used to generate 3D emoji, recognize the facial activity, and perform sight-line correction [95]. Another area where depth information is widely used is intelligent human-computer interaction. For example, the Kinect camera was originally designed for human-computer interaction with Xbox users [91] to play computer games. Other intelligent human-computer interaction includes human skeleton extraction and tracking, and gesture recognition and tracking. In robot vision, 3D information is used for 3D Simultaneous Localization and Mapping (SLAM) that allows the robot to know where it is located relative to surrounding objects and the environment [96,97]. Similar technology is also used in autonomous driving, augmented reality (AR) and virtual reality (VR) [92, 98, 99].

One of the difficulties with object recognition is the lighting and the angle at which an image is taken. It is possible that the use of depth information can provide extra information that is needed to overcome this problem. Furthermore, using depth information can better preserve the advantages of geometric features of objects and therefore aid us in constructing models that can achieve high recognition accuracies in unfamiliar scenes.

1.2 Objectives

Lighting conditions have always been a challenge to the recognition model. Even the most advanced object recognition systems cannot overcome these challenges. Since the depth information is less interfered by the light source, the main research objective of this project is to use the depth information to overcome the influence of the complex light source on the object recognition, so as to improve the recognition accuracy. In order to achieve a more complex lighting environment, the project will collect and use its own database. This project selected YOLO as the main recognition model for comparative experiments. The Grab-cut algorithm is used for segmenting the depth map. Details of YOLO and Grab-cut are described in Chapter 3. The two main research questions are stated below.

Q1: Could the grab-cut algorithm be effectively used to segment objects from their backgrounds from images with depth information?

Q2: What is the change in the recognition accuracy using the YOLO network

with and without using the depth information to segment the image?

1.3 Structure of the Thesis

A literature review is presented in Chapter 2. This review is divided into two parts. The first part is a review of the development and status of object recognition. In the second part, the development of image processing and recognition techniques that make use of the depth information is reviewed. Currently available datasets with depth map are also reviewed.

In Chapter 3, the methodology used in this research and the design of the computer experiments will be introduced. The YOLO architecture and the Grab-cut algorithm, which are used in this research, will be described in detail. Also included are the details of the dataset that is collected for this project.

The results of the experiments are presented and analyzed in Chapter 4. These results will help answer the research questions that are presented in this chapter.

Finally, in Chapter 5, conclusions are drawn. Furthermore, the limitations and potential future works are discussed.

Chapter 2 Literature Review

2.1 Review of Object Recognition

As an important branch of computer vision, object recognition has a wide range of applications in many areas of daily life. With the development of intelligent hardware devices, there are an increasing amount of images and video information which consolidate the increasingly important role played by computer vision technology in human life. Object detection and identification have also become active research directions. Detection and identification technologies have the following applications in real life such as object tracking, video surveillance, information security, autopilot, image retrieval, medical image analysis, network data mining, drone navigation, remote sensing image analysis, defense systems, etc.

Object detection and recognition refer to finding an object from a scene (picture), including the two processes of detection (where) and identification (what). The difficulty of the task lies in the extraction and recognition of the area to be detected. Therefore, the main framework of the task is to firstly establish a model for extracting candidate regions from the scene and next to identify the classification model of the candidate area. Finally fine-tuning the parameters of the classification model and the location of the effective candidate frame.

Techniques for object detection and identification could be broadly classified into two categories:

- i. Object detection and recognition method based on traditional image processing and machine learning algorithm;
- ii. Object detection and recognition method based on deep learning.

In the following two subsections, I will discuss and review the above two categories of methods.

2.1.1 Object Recognition/Detection based on Machine learning

The most significant object recognition algorithms that belong to the traditional algorithm category include Cascade, SVM, DPM, and their variants.

Research on object recognition in the 1980s has used tree search to match how well the features in the model match the features in the image [1]. However, when the number of image features and model features is increased, the difficulty of recognition increases accordingly. This is true especially when the background of the picture is very complex and when there are many noises in the picture [2]. In order to solve this problem, researchers at the Robotics Institute of Carnegie Mellon University are the first to use the concept of cascading in the field of image recognition at the beginning of the 21st century [3]. With cascading, each classifier calculates more object edge features than the previous region. Other researchers also proposed a machine learning method with cascading ideas at around the same time. For example, in [4], Adaboost was used to speed up recognition. Adaboost constructs a classifier by concentrating on only a small number of important features. This reduces the total number of features which is similar to the approach by Haar [100, 101] which speeds up recognition processing.

Support Vector Machine (SVM) was proposed in 1995. Compared with traditional statistical machine learning, SVM avoids the problem of resulting variations due to individual differences [5]. SVM finds the best solution between the complexity of the model and the learning ability based on limited sample information to obtain the best results. SVM has a large advantage in cases where there are a small sample size and a non-linear distribution of samples. It is widely used in the use of machine learning. In the field of computer vision, changes in illumination are one of the factors that affect recognition accuracy. Sabri et. al. used SVM for image recognition as early as 2004 [6]. They recognized that if the target object remains unchanged, then only the lighting conditions need to be changed. The geometric position of several information points in the slice remains unchanged. This makes it possible to use SVM.

Computer vision has its applications in many areas of identification. In [7], the author uses HOG (direction gradient histogram rendering method) and SVM for vehicle brand recognition. In order to use HOG, the author simulated the shape and appearance of the vehicles and the HOG/SVM architecture is parameterized. As a statistical method, the use of SVM is very extensive. SVM has also been used with neural networks to achieve better target recognition accuracies. Zhang et. al [8] proposed a new recognition method based on SVM and KNN. They used SVM to run on the kernel matrix without reference to perform a rough analysis of the object. For example, in a photo of a cat, the SVM only needs to determine whether there is any animal in the picture. This is followed by fine identification of animal species by a K-Nearest Neighbour (KNN) algorithm.

Finally, a traditional algorithm called DPM is has been used very successfully for

target detection. Before the deep learning-based algorithm was not available, the DPM algorithm was the VoC (Visual object Class) Test champion for five consecutive years [9, 14]. DPM can be regarded as a derivative of HOG, and its general idea is consistent with HOG. The first step is to first calculate the gradient histogram, then use the SVM to train and get a new gradient model of the object. Finally, use the new gradient model and target matching. Unlike the HOG algorithm, DPM has made many improvements in the generated gradient model. DPM has great advantages when dealing with large data sets. It also has certain advantages when dealing with situations where the appearance of the object has changed dramatically. Because of this, the DPM algorithm has been applied to many fields including face detection [10, 11], and pedestrian detection on the road [12, 13]. DPM achieved good accuracy for both of these applications. However, in the field of image recognition, the response speed of the system and the accuracy of recognition are equally important. What DPM algorithm really needs to improve on is its recognition speed. Cascading has been used to accelerate the recognition speed [102]. An alternative method makes use of the Fast Fourier Transform (FFT) [103]. In [14], the author speeds up the recognition process by constraining the level of the root filter.

2.1.2 Object Recognition based on Deep Learning

Deep learning allows neural networks consisting of multiple processing layers to learn data with multiple abstract features [15]. Deep learning is now widely used in speech recognition, visual object recognition, object detection and pharmacology [104,105,106].

Deep learning algorithms currently used in the field of object recognition can be

divided into three categories. The first category is based on regional recommendations for object detection and recognition such as R-CNN, Fast-R-CNN, and Faster-R-CNN. The second category is regression-based object detection and recognition algorithms such as YOLO and SSD. The third category is search-based object detection and recognition algorithms such as AttentionNet which is based on visual attention. In the next section, I will review several mainstream neural networks in the field of target recognition. The advantages and disadvantages of these neural networks are compared.

a. R-CNN and Variants

In 2013, Ross Girshick and his team applied the convolutional neural network (CNN) to object detection [16]. Their method, known as R-CNN, adopted the approach of generating a candidate region on the image and then recognize the target in this region. This method is much faster than traditional target recognition algorithms. It also increases the accuracy by about 20%.

However, R-CNN has two main issues. R-CNN adopts a method of convolving a region into a candidate region. With this method, the computer performs repeated convolution work when the candidate regions overlap. For every candidate region, additional storage space is required, which slows down the recognition speed. The second issue with R-CNN is that the picture is likely to be deformed after R-CNN processing [16].

In [17], SPP-NET is developed to overcome these two problems. First, it used a method to convolve the entire image before generating candidate regions. The benefit of

this is that it saves storage space. Secondly, a pooling layer is used to adapt to the size of the input image before the FC Layer, breaking the constraint that R-CNN needs fixed size images.

Further improvements to R-CNN are subsequently proposed in [18]. The resulting the Fast R-CNN network. Higher accuracies and lower storage demand are achieved. With R-CNN, the candidate frames are classified to determine whether there are any objects, and if there are objects, a bounding box is computed. Fast R-CNN, on the other hand, computing the bounding box and classifying the candidate boxes are performed at the same time.

In 2015, another variant of R-CNN, known as Faster R-CNN, is proposed [19]. It makes use of a new concept called Region Proposal Networks (RPN). It can process a single picture in 10 milliseconds, compared with around 2 seconds for R-CNN. Faster R-CNN relies on external candidate region methods, such as selective search. Faster R-CNN changed the fast R-CNN candidate area method to an internal deep network which is more efficient in generating regions of interest (ROI). Since the neural network itself is generating candidate regions, it can learn more advanced and abstract features. The location of each window in the feature map generates K anchors, and then determines the position of each anchor (foreground or background). At the same time, the precise location of the bounding box is returned to make the prediction of the bounding box more accurate.

b. Yolo and SSD

The principle of the object recognition system based on R-CNN can be divided into

two steps. The first step is to generate a Region Proposal, then CNN is used to extract features of the region. The second step is to classify the feature maps in the CNN and correct the position of the region in the image. The approach of YOLO and SSD is quite different. YOLO and SSD make use of the regression method to output the border and category of the target. When analyzing a picture, first they give a rough range for classification, and finally iterate over the range to refine the position. YOLO stands for “You Only Look Once” and SSD stands for “Supervised Salient Object Detection”.

YOLO imposes a strong spatial constraint on the boundary, which limits the number of objects the model can predict. The first generation of YOLO used coarse features to predict the bounding box, so there was a loss of function handling errors. The main source of error for YOLO is positioning error [23].

SSD aims to improve on YOLO. SSD adds the anchor concept of Faster R-CNN to YOLO. Moreover, SSD fuses the characteristics of different volume base layers to make predictions. To increase the accuracy of SSD, feature maps of different scales are generated and differentiated the predictions based on the aspect ratio. These features make end-to-end training easier. Even with low-resolution images, SSDs can still achieve good accuracy [24]. In [24], SSD is compared with YOLO and Faster R-CNN using Pascal VOC [20] and MS COCO [21]. The results show that SSD is superior to the other two in most cases with these datasets.

In August 2018, a new version of YOLO was proposed [25]. This version of YOLO, denoted as YOLOv3, is much faster than the previous algorithm. It is 3.8 times faster than

SSD with the same accuracy. YOLOv3 uses a backbone network to extract features from the picture. DarkNet-53 consists of 3×3 and 1×1 convolution kernels and skip connections like ResNet. Compared to ResNet-152, DarkNet has lower BFLOP (billion floating point arithmetic) but is up 2 times faster while achieving the same accuracy. FPN has also been added to YOLOv3 to better detect small objects. The use of FPN in YOLOv3 replaces the feature extractor in Faster R-CNN, SSD and YOLOv2, achieving better recognition. The FPN consists of top-down paths that are common networks for feature extraction. Compared to SSD, YOLOv3's ability to detect small objects is significantly enhanced. This is because the SSD performs object detection using only the upper layer of the neural network, and therefore the detection performance for small objects is poor.

YOLOv3 is the fastest deep learning algorithm for object recognition so far. In the standard database, the difference in recognition accuracy with SSD is small. But the recognition speed of YOLOv3 is higher [63]. However, applying YOLOv3 to color images face many challenging problems, mostly related to illumination, shadow projection, and colour camouflage due to the foreground and background-like colors [62]. The objective of this thesis is to investigate if the use of depth information could help overcome some of these challenges.

2.2 RGB-D Images

The depth image refers to an image in which the pixel values are related to the distance between that point in the image and the camera plane. It directly reflects the geometric

features and shape of the object. The depth image can be calculated by coordinate transformation from point cloud data captured by an RGB-D camera. In November 2010, a low-cost depth camera called Kinect is released by Microsoft Inc. for its XBOX platform [26]. Since then, several inexpensive RGB-D cameras became commercially available, opening up new possibilities for capturing depth images.

Depth cameras can be divided into two types, depending on how depth information is obtained. a TOF structure [48]. By continuously transmitting a light pulse to the target, and then receiving the light returned from the object with a sensor, the depth information of the target is obtained from the time it takes for a pulse to travel to the target and back [27]. The second type of depth cameras makes use of stereoscopic vision [49]. Depth is computed through triangulation by matching left and right images. Compared to the TOF cameras, the stereoscopic cameras has the advantages of high resolution and low power consumption. An example of which is the ZED camera [107].

2.2.1 Applications

In recent years, depth cameras have been applied to object pose recognition, camera tracking, scene reconstruction target tracking, and recognition, face recognition and other fields. They have also been applied to 3D scanning. Previously, 3D scanning has been limited by the instrument because 3D scanners are large and expensive devices. The emergence of handheld RGB-D cameras has changed this landscape. For example, KinectFusion [28] is a 3D reconstruction project, which is based on the Kinect camera, that enables 3D modeling of the target. In the field of robotics, RGB-D cameras have also

been used for Simultaneous Localization and Mapping (SLAM). A featureless SLAM algorithm was proposed in [30]. This algorithm is able to construct a 3D scene in a large-scale environment [31]. Thomas et al. [32] used RGB-D cameras to achieve high-quality surface reconstruction. They used the GPU's 3D loop buffering technique to effectively process the depth map. Furthermore, they overcame closed-loop constraints due to pose estimation and lighting factors.

RGB-D images can also be used to perform more reliable face recognition, and there are already commercial products, such as the iPhoneX by Apple Inc. In [33], a method to performing face recognition using a depth camera is described. Different lighting conditions, the camouflage of the face (such as wearing sunglasses), posture problems of the face, etc. are all challenges encountered in face recognition [34]. The most reliable way to solve such problems is to use 3D information since the effects of illumination on the depth map is limited. More stable features could also be extracted using the depth map.

Human body pose recognition is another area of applications for RGB-D cameras. In order to improve the quality of life of the elderly, as early as 2007, Jansen et al. [50] proposed a model for automatic detection of the behavior of the elderly. The model they proposed uses depth information. In [51], a method for human gesture recognition that requires only a single depth image was proposed. This method does not require time information. The inspiration comes from the idea of object recognition. They use depth information to relabel parts of the human body.

Hand gesture recognition is also a direction of depth image application. The geometry of the hand is very complicated and the hand is small, which makes the identification difficult [52]. Athitsos et al. [53] created a large database of hand postures in complex environments. In [54], hand features are extracted from depth images for training [54]. Liu and Fujimura [55] used threshold processing to detect depth data from the hand and measure the shape similarity by using the chamfer distance.

2.2.2 Object Recognition Using RGB-D Images

In recent years, as RGB-D cameras have become faster, and high-quality depth information has become available. Research on object recognition based on depth information is also increasing. In [35], the HMP concept was introduced. HMP uses sparse coding to learn hierarchical features from raw RGB-D data in an unsupervised fashion. The R-CNN network was also applied to depth images for object recognition [36]. The training parameters of the R-CNN network is finetuned by initializing the learning rate to 0.001 and reducing the number of iterations by 10 times every 20k iterations. The results show that these fine adjustments are effective, with an average accuracy of 37.3%, which is an 56% improvement on the existing methods. In [37], a semi-supervised learning framework for RGB-D object recognition was proposed and achieved good results. Based on the ideas stated in the Yahua's paper, we know that most object recognition project is based on RGB images, and the geometric features provided by RGB images are often unreliable. Illumination factors and backgrounds are too cluttered to affect the results of the recognition, and the indistinct distinction between the

background and foreground colors also affects the accuracy of the recognition. Depth images provide us with reliable geometric features and shape cues. Kevin et al. [38] used depth information to segment the images. They used an adaptive Gaussian mixture model [39] to overcome the depth noise problem. In order to extract low-level image features more efficiently, special kernel descriptors are designed a hierarchical application of them are used [40]. They applied this method to an RGB-D dataset [41]. They processed the depth information into a grayscale image. This is very similar to the idea used in this research. Although this method achieved good results, the results were still affected by the fact that there was no noise and void processing in the depth image. In [42] the descriptor method is used to link the color information with the depth information. The idea is novel, but there is still no way to avoid the influence of noise in the depth map.

2.2.3 Image Segmentation Based on Depth Map

Image segmentation is an important part of indoor scene analysis, object pose recognition, target recognition, etc. Nathan et al. used depth information to propose a main surface for interpreting indoor scenes [57]. The database used in this work consists of images taken in a messy indoor environment. The algorithm first calculates the normal of the surface and then use RANSAC to fit the plane to the point and segment according to the depth information. In [58] a real-time plane segmentation method was proposed and collected a database of indoor scenes.

Using an RGB-D camera is not only helps to segment the various planes in a scene, but we can also use depth information to segment the object we want to identify. Most of

the perceptual work on RGB-D pictures is focused on semantic segmentation, the task of assigning category labels to each pixel [36]. In semantic segmentation of indoor scenes, super pixels are classified into 40 main object categories in NYUD2 [59]. Xiaofeng et al. combined the kernel descriptor and super pixel method to segment the object [60]. In [61], SLAM is used to form multiple views, and 3D pixels are marked with absolute positions in each scene. The object is segmented by the information of the pixel. Since the depth map is often very noisy, Massimo et al. [62] combined the colour features of the image and the depth features of the image while performing the segmentation task. In [64], RGB-D segmentation is performed through unsupervised learning. Depth cues are used to better protect the boundaries of objects and constrain the smoothness and consistency of the surface of objects.

The Grab-cut algorithm can effectively extract the foreground in a picture or segment the picture [65]. It splits the image by comparing the color information and contrast information of the image. In [66], the implementation of this algorithm was described in detail. There are also subsequent efforts to optimized it, but the basic idea of this algorithm has not been changed. Shoudong et al. [67] used the MSNST algorithm to extract the color information of the original image and then segmented using the Grab-cut function. Most Grab-cut algorithms can only extract features from images. In [68, 69], a fully automated Grab-cut algorithm, which is called Vabcut, is proposed. They used it for the segmentation of human behavior in the video. Further improvements in convergence speed and segmentation accuracy of the Grab-cut algorithm, the region of interest of the user is used to constrain the existing Grab-cut algorithm in [70].

The Grab-cut algorithm has also been applied to segment RGB-D images. In [71], it was used in combination with the Kinect camera. The authors successfully segmented the contours of the human body. However, due to excessive noise, it could not effectively use the depth information. The author of [72] used the Grab-cut algorithm to segment the depth grayscale image, but the author did not explore the effects of noise in the depth map.

2.2.4 Depth Image Datasets

Dataset collection is an important part of every field of RGB-D camera applications. Several different RGB-D image datasets are available. They are collected for specific application areas of research, such as gesture recognition, human actions, and pose estimation. Since this research is in target detection and recognition, only object datasets are reviewed here.

In [38], a total of 51 categories of images with 300 different objects are collected. Also included in the dataset are images with multiple angles of the objects. The objects are common ones found in homes and offices. These objects are rotated on a fixed turntable to extract their multiple angles, and the pose of the object remains consistent. In the dataset used in [43], the depth maps produced by some existing RGB-D cameras are collected. A dataset that is derived from video clips is reported in [44]. The scenes in this dataset are very diverse, with a total of 2347 pictures. In order to make this dataset better suited for robot vision, the objects in the images are manually annotated, providing a label covering for each pixel in the image. Since the environment in the real world is much more complicated than the environment in the laboratory (such as lighting factors,

occlusion, etc.), more and more scholars are collecting RGB-D datasets based on real-world settings [45]. Currently, a large database based on the real-world is reported in [46]. This dataset contains images of more than 10,000 objects in real-world settings. Another dataset with more than 10,000 objects [47]. 3D models of every object in the dataset are created.

Chapter 3 Methodology

This research is inspired by the approach described in [38] that make use the depth grayscale to directly implement the segmentation process. Based on the literature review presented in Chapter 2, the idea is to use the Grab-cut algorithm [65] for segmenting objects in an RGB-D image and use YOLOv3 for target recognition. It has been observed that the grayscale depth image is filled with large patches of information including the shape of the object. The Grab-cut algorithm could effectively segment large areas of grayscale in the depth map that contain object geometry. This idea is simple and efficient, requiring demand on hardware, and can split any object in the picture.

In this chapter, I will elaborate on the details of these two algorithms. Following that, the experimental plan and explain the evaluation methods used will be described. In order to have full control of the experimental conditions, I have collected a dataset of RGB-D images that will be used in this research. A full description of this dataset will also be presented.

3.1 Grab-cut Algorithm

The Grab-cut algorithm can effectively extract the foreground in a picture [65]. It segments an image by comparing the color and contrast of the various parts of the image. A detailed description of its implementation can be found in [66]. This algorithm requires a small amount of user interaction for it to work. A user specifies the area of interest with a rectangular border to complete the segmentation of the foreground. Figure 3.1 is an

example of RGB image segmentation using the Grab-cut algorithm.



Figure 3.1 Example of using the Grab-cut algorithm

The Grab-cut algorithm was developed from the Graph-cut algorithm [108]. With Graph-cut, the target and background models is a grayscale histogram. In Grab-cut, it is replaced by an RGB three-channel mixed Gaussian model. The segmentation process of Grab-cut is an interactive one involving interactive segmentation estimation and model parameter learning. These two parts of the process are explained in the subsections below.

3.1.1 Color model

As mentioned above, the Grab-cut algorithm uses a three-channel mixed Gaussian model. A full covariance Gaussian mixed model (GMM) $K = \{k_1, k_2, \dots, k_N\}$ of N Gaussian components, where k_i is the Gaussian component corresponding to the i^{th} pixel, is used to model the target and background. Each pixel must come from a certain Gaussian component of the target GMM or a certain Gaussian component of the background GMM.

The Gibbs energy of the entire image is given by

$$\mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z})$$

Here, U is the region term, given by

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n),$$

which is the negative logarithm of the probability that a pixel belongs to the target or the background.

$$D(\alpha_n, k_n, \underline{\theta}, z_n) = -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log \det \Sigma(\alpha_n, k_n) \\ + \frac{1}{2} [z_n - \mu(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)].$$

This is obtained by taking the negative logarithm of the mixed Gaussian density model:

$$D(x) = \sum_{i=1}^K \pi_i g_i(x; \mu_i, \Sigma_i), \quad \sum_{i=1}^K \pi_i = 1, \quad g(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left[-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]$$

The parameter θ of the GMM has three elements. They are the weight π of each Gaussian component, the mean vector μ of each Gaussian component, and the 3×3 covariance matrix Σ . Thus,

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, k = 1 \dots K\}$$

This is the parameter that the algorithm needs to estimate. Once it is determined, by substituting the RGB color value of a pixel into the target and background GMM, the probability that this pixel belongs to the target/background can be obtained. After this, the value of U of the Gibbs energy formula can be determined by us. Then the value of U could be computed.

The second term in the Gibbs energy formula above is the boundary energy term V , given by

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in \mathbf{C}} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2.$$

Grab-cut uses a two-norm method to measure the similarity of two adjacent pixels.

The parameter β here is determined by the contrast of the image. If the contrast of the image is low, then the difference $\|z_m - z_n\|$ between two pixels m and n is small. In this case, it needs to be multiplied by a relatively large β to amplify this difference. For images with high contrast, this difference is large, and so it needs to be multiplied by a small β to reduce the difference. The initial value of the constant γ is usually 50, with a better value obtained after training. With both U and V (and thus E) computed, the image foreground could be determined.

3.1.2 Iterative Energy Minimization Segmentation

The segmentation process of Grab-cut is iterative. Users need to manually tell the computer about the general range of targets. After passing the user's markup, we get some pixels that may belong to the target ($A_n = 1$) and some pixels that may belong to the background ($A_n = 0$). The pixels are classified by the K -means algorithm and are recorded as K_n . By using the resulting set of pixel samples, the parameter θ of the GMM, as described in the previous section, is computed.

At this stage, a rough segmentation can be obtained. If the user is not satisfied with the result of this segmentation, manual marking is performed. After each manual marking, the segmentation result is recalculated. The process of this iterative is to gradually reduce the range of pixels where $A_n = 1$. Since the iterative process is a process with a decreasing range, the iterative process must be convergent.

3.2 YOLO

3.2.1 YOLOv1

YOLO [23] is a deep learning architecture proposed to speed up object detection and recognition after R-CNN, fast-R-CNN and faster-R-CNN. YOLO treats object detection as a regression problem. The input to YOLO is entire image. The position of the bounding box of an object is regressed and its associated target category is obtained directly at the output layer. This is in contrast to other object detection techniques which are divided into two separate steps.

YOLO divides the input image into a grid of $S \times S$ cells. If the coordinates of the center position of an object falls within a certain cell, then this cell is responsible for detecting the object. At the same time, each cell also needs to predict the value of the bounding boxes. A value known as “confidence” is predicted for each bounding box. This confidence value is computed by:

$$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

If an object falls in a grid cell, then $\Pr(\text{object})=1$. Otherwise it is 0. The second term represents intersection over union (IOU) which is a value between the predicted bounding box and the actual ground truth value.

Apart from confidence, each bounding box predicts 5 values (x, y, w, h), and a category, which is denoted as class C . With an $S \times S$ grid that predicts B bounding boxes and C categories. The output is a tensor of $S \times S \times (5B + C)$.

In a test, the class information for each grid prediction is multiplied by the confidence information predicted by the bounding box. The result is the class-specific

confidence score for each bounding box, given by

$$\Pr(Class_i | Object) * \Pr(Object) * IOU_{pred}^{truth} = \Pr(Class_j) * IOU_{pred}^{truth}$$

The first item on the left side of the equation is the category information for each grid prediction, and the second and third items are the confidence of each bounding box prediction. This product encodes the probability that the predicted box belongs to a class.

A threshold is set in order to filter out the boxes with low confidence scores and perform NMS (Non-Maximum Suppression) processing on the reserved boxes to get the final test results.

3.2.2 YOLO9000 & YOLOv3

In YOLO, it is difficult to detect objects that are very close in distance to each other. The reason is that only two boxes are predicted in a grid and belong to only one category. YOLO9000 was proposed to overcome this problem [109]. It uses a Word Tree to mix and detect data sets and identify data sets. It can detect more than 9,000 classes of objects.

The coordinate values of the bounding box are predicted by the fully connected layer in YOLO. In YOLO9000, the fully connected layer is removed and Anchor Boxes are used to predict the Bounding Boxes. This is because the prediction offset is simpler than the predicted coordinate value. A pooling layer has also been removed. This allowed the output of the convolutional layer to have a higher resolution. Using the Anchor Box will reduce the accuracy slightly but using it will allow YOLO9000 to predict more than a thousand boxes, with a recall of 88% and an mAP (mean Average Precision) of 69.2%.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3.2 The structure of the classifier Darknet-53 used by YOLOv3

A completely different approach is used in YOLOv3 compared to other object detection methods [63]. It applies a single neural network to the entire image. This neural network divides the image into different regions, predicting the bounding box and probability of each region. It also uses a better classification network and a better classifier called Darknet-53, instead of using SoftMax to classify each box. The main reason is that Softmax is not suitable for datasets with multiple label classifications.

Moreover, Softmax can be replaced by multiple independent logistic classifiers without compromising accuracy.

3.3 Database Collection and Preprocessing

A dataset of RGB-D images has been collected for this research. This section describes the content and the collection process of this dataset. The images in this dataset have also been preprocessed using homomorphic filtering algorithm. This preprocessing is also discussed.

3.3.1 Database Collection

The RGB-D image database that has been collected for this research included color and depth images of a total of six objects. Each object in the database includes at least fifty pictures taken from different angles. The objects that have been chosen are common objects in everyday life, such as trash cans, chairs, suitcases, etc. The images are taken using a ZED camera [107]. The ZED is a stereoscopic camera that can capture 1080p HD video at 100FPS and WFGA at 30FPS for clear images. The acquisition depth is 0.5-20 meters. Still images have a maximum resolution of 4416×1242 pixels. Before collecting the images in this database, the camera is calibrated using the SDK provided by the manufacturer.

In the course of our literature review, it has been found that lighting condition is a main factor that affects the accuracy of object recognition. The intensity of the light

causes the color characteristics of the object as well as the changes in the texture characteristics. This poses a great challenge to the accuracy of object recognition. At the same time, angled lighting can cause shadows. The appearance of large shadows in a darker environment is similar to occlusion, which also greatly affects the accuracy of object recognition. In order to explore the performance of our method under these conditions, a strong light source is used.

When taking the images, the camera is 1.5 meters away from the object. Our process of collecting the images is similar to that used by [38] where the objects are placed on a turntable to record images of objects as the table is turned. For this database, the position of the camera is manually rotated through 360° around the object. However, the position and angle of the light source remain unchanged throughout this process. Figure 3.3 shows the RGB images of a chair in this database. In order to save space, images of only 15 different angles are shown in this Figure.

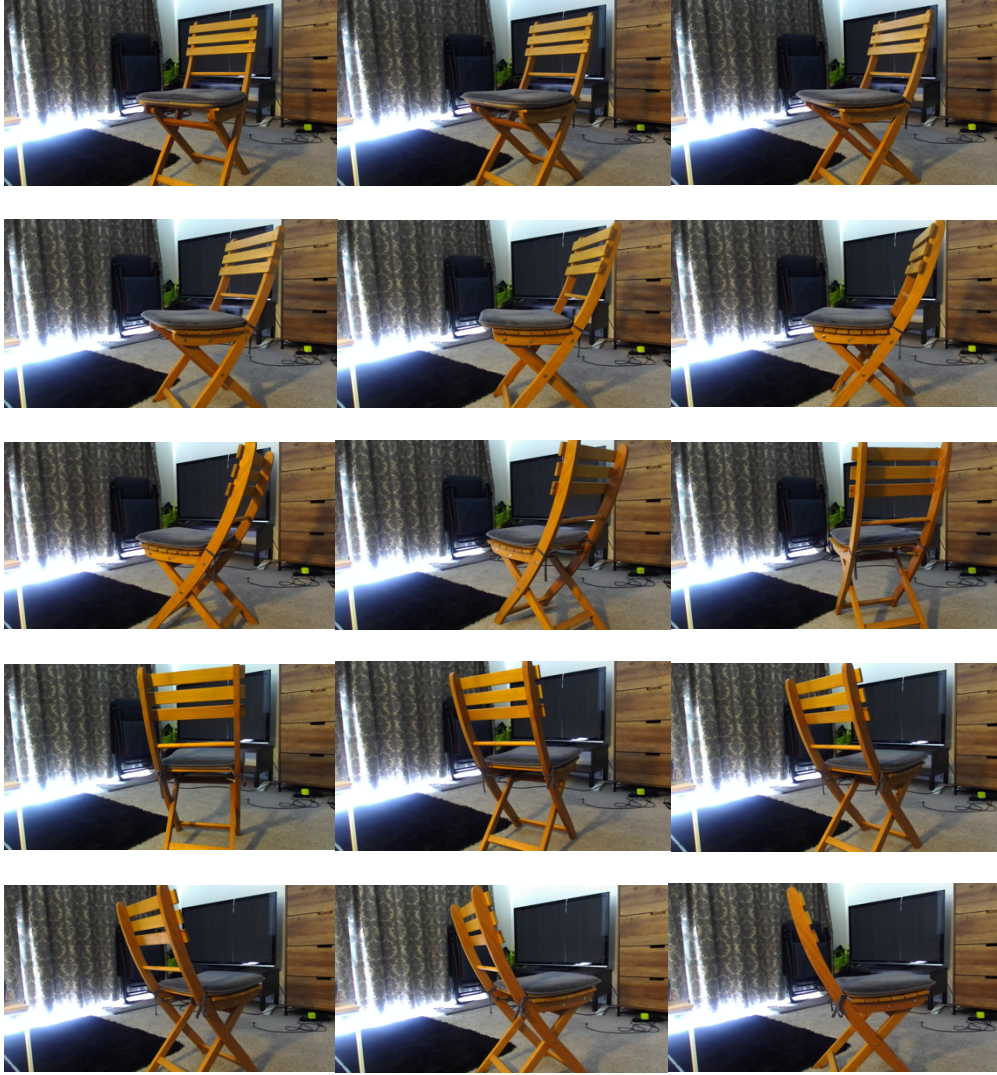


Figure 3.3 Chair database example

The depth map of the images in the database has been recorded in two formats: FULL and RAW. The depth map in RAW format is similar to the depth map collected in [38]. The FULL format is closer to a disparity map [73-75]. Since the RAW format depth map is noisier, the FULL format depth maps are used in the experiment. But the depth map in the RAW format contains a lot of information. Hence these depth maps are also kept in the database.

3.3.2 Preprocessing

Some basic preprocessing of the images could improve the recognition accuracies. Since the images used in this experiment consist of both RGB and depth images, different preprocessing methods are applied to these two kinds of images.

For the images that have been acquired, the depth grayscale images tend to be too dark. This phenomenon also exists in other image datasets mentioned in the previous section. So the brightness of the depth grayscale image is adjusted. For both the RGB and depth images, homomorphic filtering is used to increase their quality.

The basic flow of homomorphic filtering is shown in Figure 3.4.

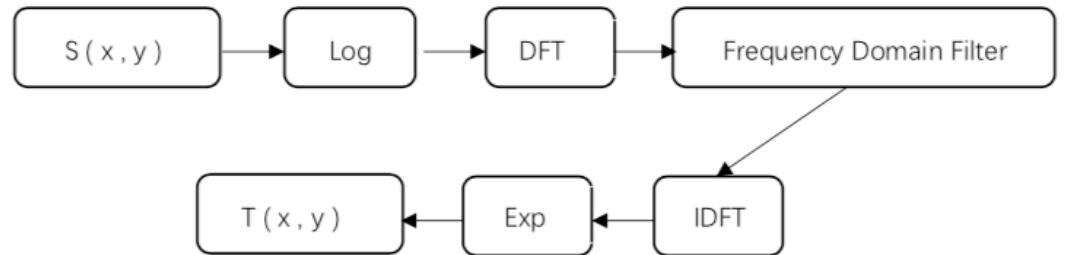


Figure 3.4 Signal processing flow of homomorphic filtering



Figure 3.5 Example of originally acquired depth maps of the chair image



Figure 3.6 Example of Full Format depth maps after preprocessing

The effects of homomorphic filtering are illustrated in Figures 3.5 and 3.6. Figure 3.5 shows three full-format depth maps of the chair images that have not been pre-processed. It is obvious that they are too dark. The image below is the Full format depth map after our preprocessing. After increasing the brightness and using homomorphic filtering to improve the contrast of the image, we obtained depth maps as shown in Figure 3.6 that are more useful. The same pre-processing could also be used for depth maps in the RAW format. Figure 3.7 is the result of preprocessing the depth maps in RAW format.

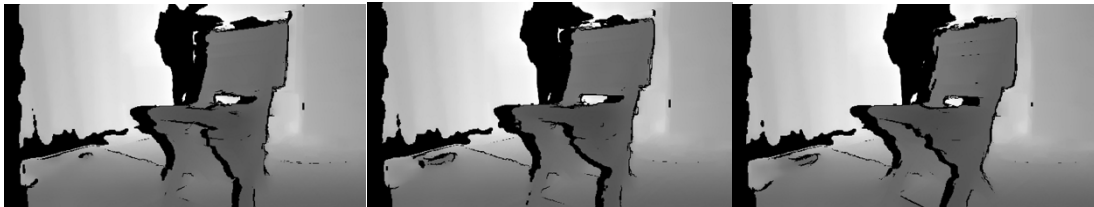


Figure 3.7 Example of a RAW Format depth map after preprocessing

For the RGB images, the brightness is generally acceptable. However, their contrast could be improved using homomorphic filtering. Figures 3.8 and 3.9 show the chair images before and after filtering respectively. Compared to the depth map, the RGB images do not change much visually before and after preprocessing. The preprocessing of the depth maps has a greater impact on the experimental results.



Figure 3.8 Example of original chair RGB images acquired



Figure 3.9 Example of chair RGB image after preprocessing

3.4 Research Design

Computer experiments are planned to answer the research questions presented in Chapter 1. All experiments are conducted using the database of RGB-D images that have been acquired for the purpose of this research as discussed in the previous section.

3.4.1 Experimental Plan

In order to determine the effects of image segmentation using Grab-cut on the object recognition accuracies by YOLO, a method of comparative experiments is adopted. The first experiment is designed to establish a baseline for comparison. YOLOv3 is used to identify all the objects in the database images. The code for YOLO is based on Python using the TensorFlow library.

For the second experiment, the depth maps are segmented using the Grab-cut algorithm. A contour map belonging to the recognition object is obtained. Finally, Photoshop is used to manually merge the contour map and the RGB image. The depth images are pre-processed as discussed in Section 3.3.2. Both the Full and Raw formats of the depth grayscale image are processed. During segmentation, the one that provides the best segmentation results is used. YOLOv3 is then applied to identify the objects after image segmentation.

3.4.2 Training


The YOLOv3 network needs to be a trained process for target detection. Supervised training with relevant data is important as it greatly affects accuracy. A YOLOv3 network has already been trained in [25, 109] and the pre-trained model has been made available. This model has been trained using datasets ImageNet [87], Coco [20], and VOC [21]. Since it already contains the categories we use to detect objects (chairs and suitcases). So, in this project, we used the YOLOv3 pre-trained model provided by darknet [110].

3.4.3 Evaluation Method

In the YOLO series of object detection algorithms, an important step is to use the non-maximum suppression algorithm (NMS) to obtain the target area. Non-Maximum Suppression needs to find the bounding box with high confidence based on the coordinate information of the score matrix and the region. For forecast boxes that are overlapping, only the one with the highest score is retained. The confidence score is given by

$$\text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

In this mathematical expression, the value of IOU (intersection-over-union) intuitively determines the class-specific confidence score. It is defined by

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


It is a ratio of on the area where the two bounding boxes overlap. The NMS processes can only work with one category at a time. These two bounding boxes are those of the detection results and the Ground Truth. This algorithm can only work with one category at a time. If there are N categories, NMS needs to be executed N times.

Chapter 4 Experimental results and Analysis

4.1 Experimental results

The aim of this research is to object detection how the accuracy of YOLOv3 could be improved by depth image segmentation using Grab-cut. The computer experiment results are shown in this Section. There are three parts of the results. The first part will show the detection results using the RGB images. In the second part, the results of segmenting the depth map using the Grab-cut algorithm are shown. In the third part, the experimental results using the segmented image test are shown.

Images of two objects have been selected for these experiments. The objects are a chair and a suitcase. Each dataset consists of 52 sets of images. Each set of images includes an RGB image, a depth map in Raw format, and a depth map in Full format. As described in Section 3.3, these images are captured at different angles with a light source that is placed in such a way that there will be shadows of objects in the image.

4.1.1 Result of the first recognition

The first experiment establishes the “ground truth” by which comparisons are made. It involves only the RGB images of the datasets. In this work, we use the value of IOU after NMS screening as the most important value for evaluation (class-specific confidence score) as described in Section 3.4.3. In this and subsequent experiments, a threshold of 0.5 is used. This means that if the IOU value is higher than 50%, then the detection is considered successful. A value below 50% will be recorded as a detection failure.

Chair dataset

The results of the chair dataset is shown in Table 4.1. Using a threshold of 50%, 4 out of 52 images fail to be detected, equivalent to 7.7%. But most of the confidence scores of successful detection are between 60% and 80%. In order to avoid bias due to individual differences when the average accuracy is calculated, the two extreme values -- the lowest and highest confidence scores are removed. In this case, the average accuracy is 72.65%.

confidence score	90%	80%	70%	60%	50%	Not recognized
No. of images	4	14	16	11	3	4

Table 4.1 Chair database detection results.



Figure 4.1 Examples of recognition in the chair database.

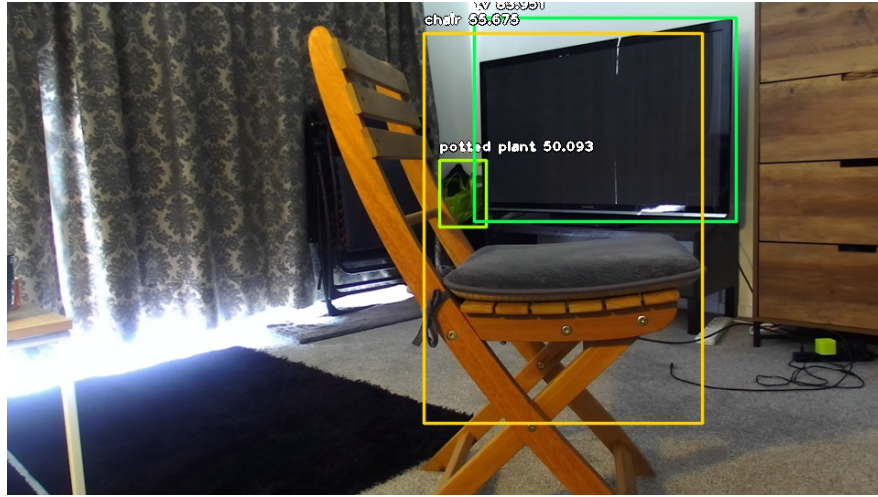


Figure 4.2 Examples of recognition in the chair database.

Figures 4.1 and 4.2 are examples of detection with high and low confidence scores respectively. It is interesting to note that a higher score is obtained when the front of the chair is facing the camera. But when the side of the chair is facing the camera, the accuracy is much lower. Incidentally, Figure 4.2 shows the case where the score is lowest.

Confidence score as a function of the angle of rotation is shown in the graph in Figure 4.3. In this graph the accuracy is divided into ten groups after the two extreme values have been removed. Thus each group has five samples and the average value of these five samples is displayed. Each group of objects has a similar angle and perspective. In this figure, the vertical axis is the magnitude of the confidence score, and the horizontal axis are the ten groups of images. It can be observed that the lowest average is the fourth group. The images in the fourth group are those where the side of the chair faces the camera.

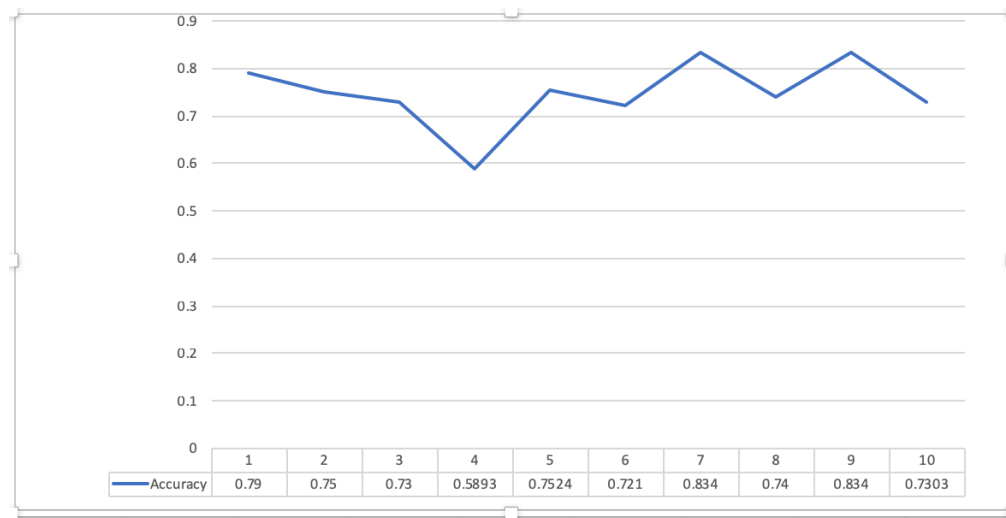


Figure 4.3 Confidence score vs image perspective for the chair dataset.

Suitcase Dataset

A similar experiment is performed on the suitcase dataset. The results are shown in Table 4.2. The highest score is 96.7%, and the lowest is 64.5%. There is no undetected image. The average score, with the highest and lowest values removed, is 87.62%. This is substantially higher than that for the chair dataset, by about 15%. This could be due to greater contrast between the color of the suitcase and the background color.

confidence score	90%	80%	70%	60%	50%	Not recognized
No. of images	14	30	2	6	0	0

Table 4.2 Suitcase database detection results

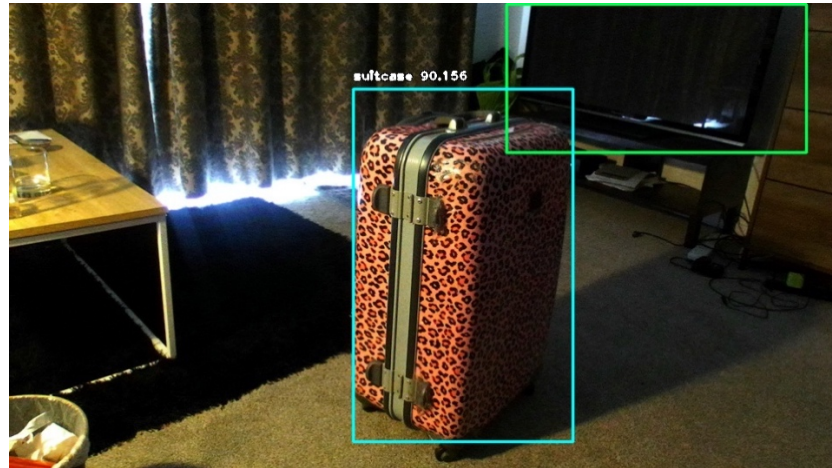


Figure 4.4 Examples of recognition in the suitcase database.

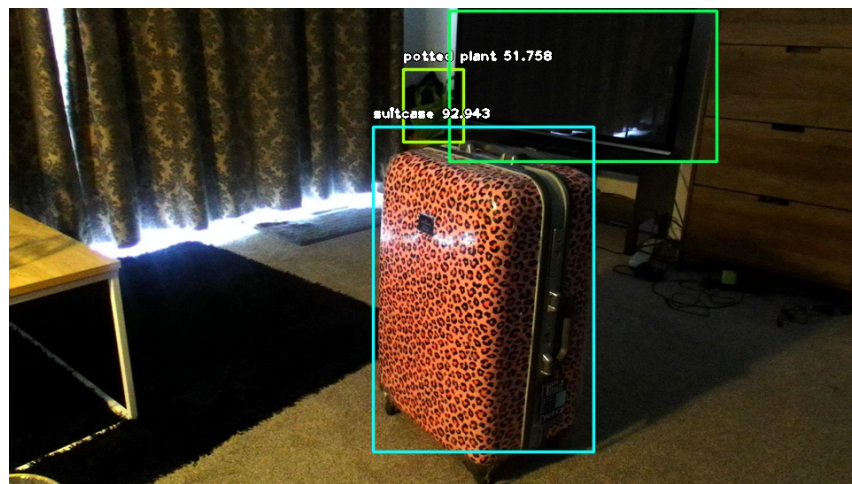


Figure 4.5 Examples of recognition in the suitcase database.

Figures 4.4 and 4.5 show two examples of the results with high and low scores respectively. As with the chair dataset, the results vary depending on the angle of the suitcase in the image; when the suitcase is facing sideways, the accuracy is usually lower. A similar graph to Figure 4.3 is produced for this dataset and is shown in Figure 4.6. Group 4 consists of images where the suitcase is facing sideways.

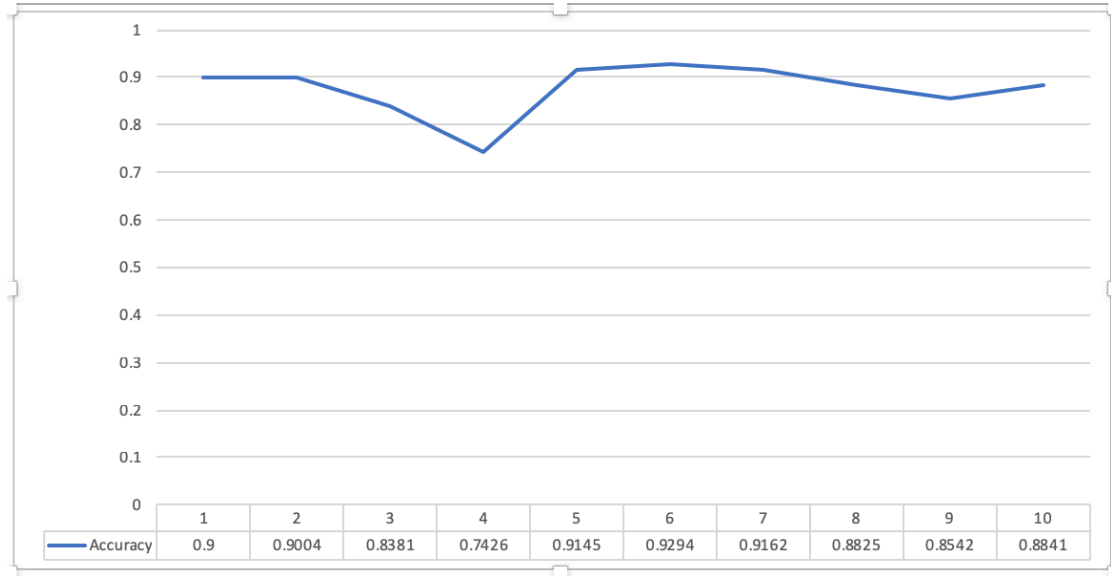


Figure 4.6 Suitcase database line chart

4.1.2 Segmentation of Depth Map

The main goal of this part of the experiment is to use the depth information provided by the depth camera to perform background separation of the original image. Because the principle of the Grab-cut algorithm is based on image color information and contrast, and since the depth images we collect have strong contrast, it is suitable for this purpose. The flow of this algorithm is as follows. After completing the cutting of the depth image, the contrast brightness is adjusted. A contour map based on the depth information is obtained. Then Photoshop is used to combine the contour map with the RGB image to separate object from the background. This process is illustrated in Figure 4.7.

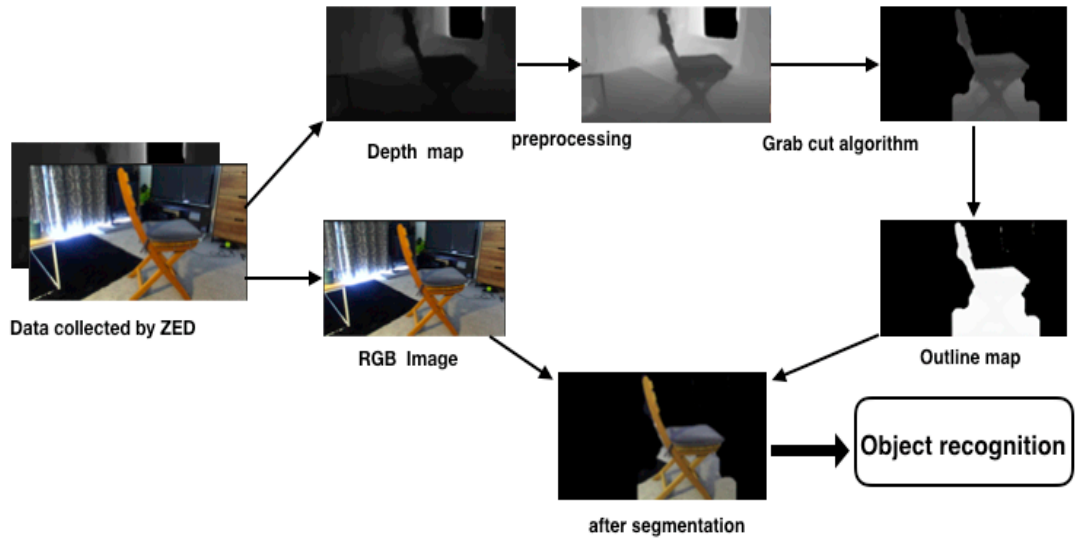


Figure 4.7 The workflow of the segmentation process.

In the process of collecting the datasets, depth maps are obtained in two different formats. The Raw format holds the most primitive depth information while the Full format is calculated from the Raw format. Each of these two depth images has its own advantages. The depth map in the Raw format has more large-area patches, which in some cases is more conducive to the process of cutting. However, these depth maps also tend to contain a lot of noise and “holes” in the image. They may be related to the light and angle of the shot, which we will discuss in more detail in Section 4.2. For this reason, depth maps in the Full format have been used.

Figure 4.8 shows examples of depth maps of different formats of the chair images. The two pictures on the left are depth maps in Raw format while those on the right are in Full format. The images in the first row are from the image of the chair in the same position and those in the second row are in a different position from the first.

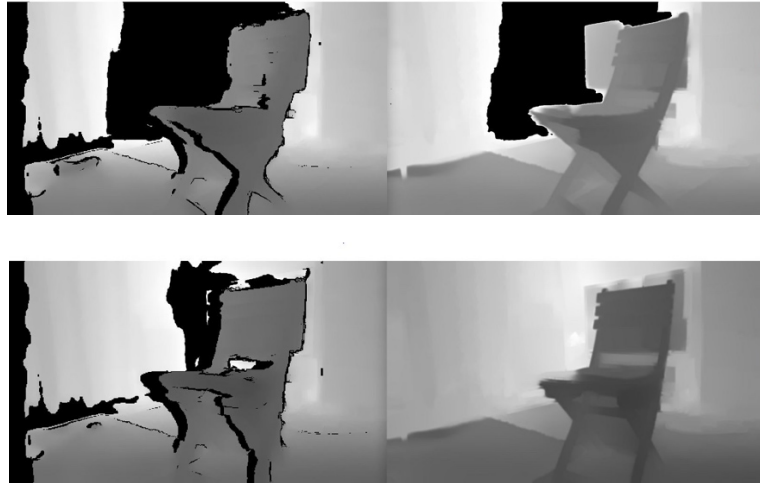


Figure 4.8 Example of depth maps in Raw and Full formats.



Figure 4.9 Segmentation result of a suitcase depth map.

An example of the depth map and the result of the segmentation process is shown in Figure 4.9. The image on the left is the depth map where the lighter the color, the closer it is to the camera, and the darker the color, the farther from the camera. In this case, a good separation is achieved.



Figure 4.10 Segmentation result of a chair depth map

Figure 4.10 shows the depth map of a chair image. In this case, the lighter the color in the depth map, the farther the object is from the camera. Interference can be observed due to view angle and lighting factors. It is not easy to distinguish between the object and the ground using the depth map. As a result, the segmentation is not as successful as in Figure 4.9.

Examples of the segmented RGB images can be found in Figures 4.11, 4.12, 4.14 and 4.15.

4.1.3 Results of the Second Experiment

This experiment is conducted in a similar way to the first experiment. The same datasets are used. The only difference is that in this experiment, the segmented images produced using the process described in the previous section are used for object detection.

Chair Dataset

The results are shown in Table 4.3. The number of unrecognized images and inaccurate positioning has not improved much. However, the confidence score of most

images has increased by about 10%, and has even reached 15%. After removing the highest and lowest scores, the minimum and maximum scores are 62.83% and 95.53% respectively.

confidence score	90%	80%	70%	60%	50%	Not recognized
No. of images	12	19	15	2	1	3

Table 4.3 Chair database detection confidence scores.



Figure 4.11 Example of detection in the chair dataset.



Figure 4.12 Examples of detection in the chair dataset.

Figures 4.11 and 4.12 show the bounding boxes of two example images. They are the same sample images as in Figures 4.1 and 4.2. The score of the first image increased from 90.96% to 95.40% and that of the second picture increased from 55.67% to 68.94%.

The confidence score still varies with the angle of rotation of the chair. When the front of the chair faces the camera, the score is higher. When the back or side of the chair is facing the camera the score is lower as shown in Figure 4.13. However, the lowest average score has increased from around 60% to 70%.

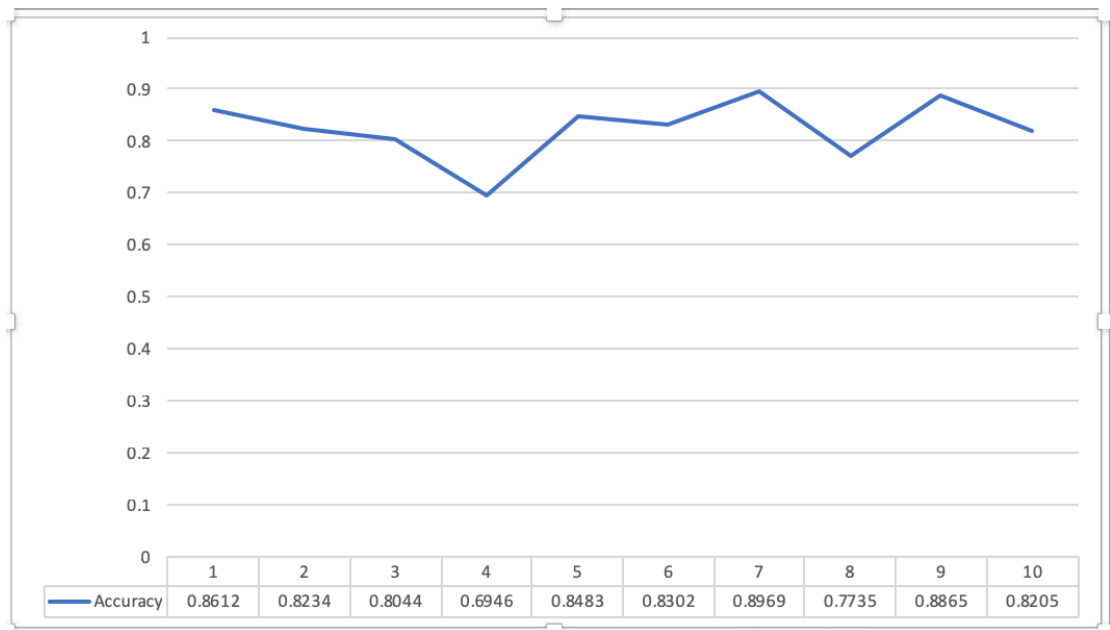


Figure 4.13 Confidence score for chair images with different orientations

Suitcase Dataset

Table 4.4 shows the results for the suitcase dataset. The confidence scores are very

good as expected. The lowest score is 72.02%, and the highest is 98.86%. Moreover, more than 60% of the scores are higher than 90%. The average score is 92.29%. Compared with the first experiment, there is a 5.2% increase.

confidence score	90%	80%	70%	60%	50%	Not recognized
No. of images	28	16	8	0	0	0

Table 4.4 Suitcase database statistics

The segmented images of Figures 4.4 and 4.5 are shown in Figures 4.14 and 4.15 respectively. While the score changes with the angle of the suitcase, the amount of change is not very large. The scores for the two images have improved to varying degrees. The score for the first image has increased from 90.12% to 95.41%, and that for the second one has increased from 92.94% to 98.16%.

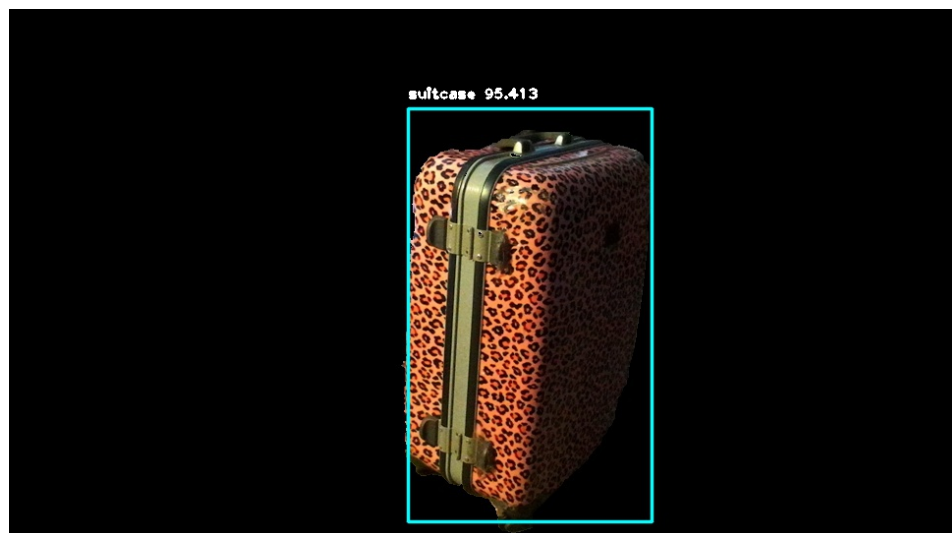


Figure 4.14 Examples of recognition in the suitcase database.

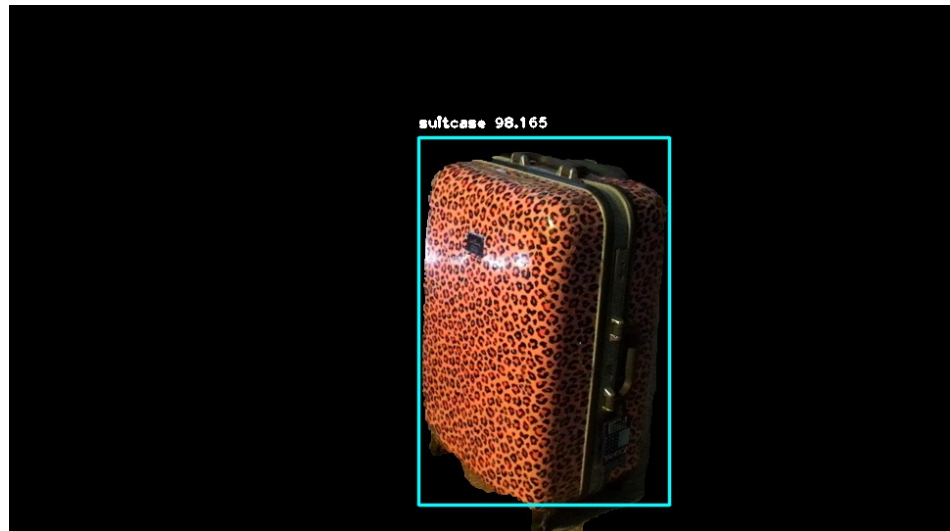


Figure 4.15 Examples of recognition in the suitcase database.

The effect of the orientation on the confidence score is shown in Figure 4.16. The scores fluctuate between 80% and 100%.

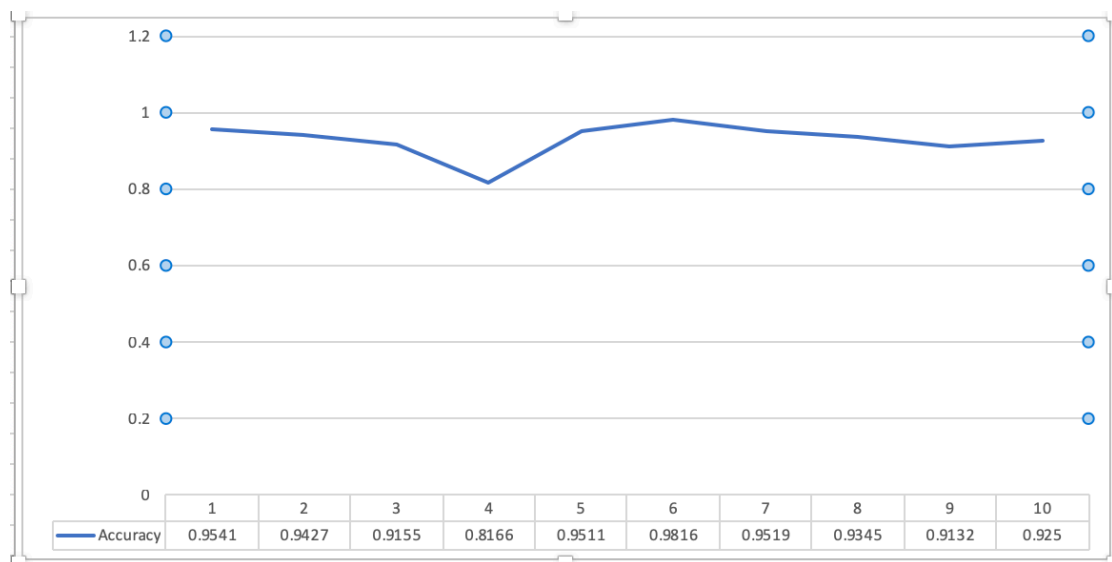


Figure 4.16 Confidence score for chair images with different orientations

4.2 Analyses and Observations

4.2.1 Comparison of Results

Chair Dataset

Figures 4.3 and 4.13 are combined into one graph in Figure 4.17. The blue line represents the result of the first experiment and the red line represents that of the second experiment. From the figure, one can see that in the fourth group, the average confidence scores are the lowest. The scores drop in the sixth and eighth groups, with the later group dropping more significantly. The overall trend of score variations between groups in the two experiments is approximately the same. For the first experiment, the highest and the lowest average scores differ by 24.47%. In the second experiment, this difference is 20.23%.

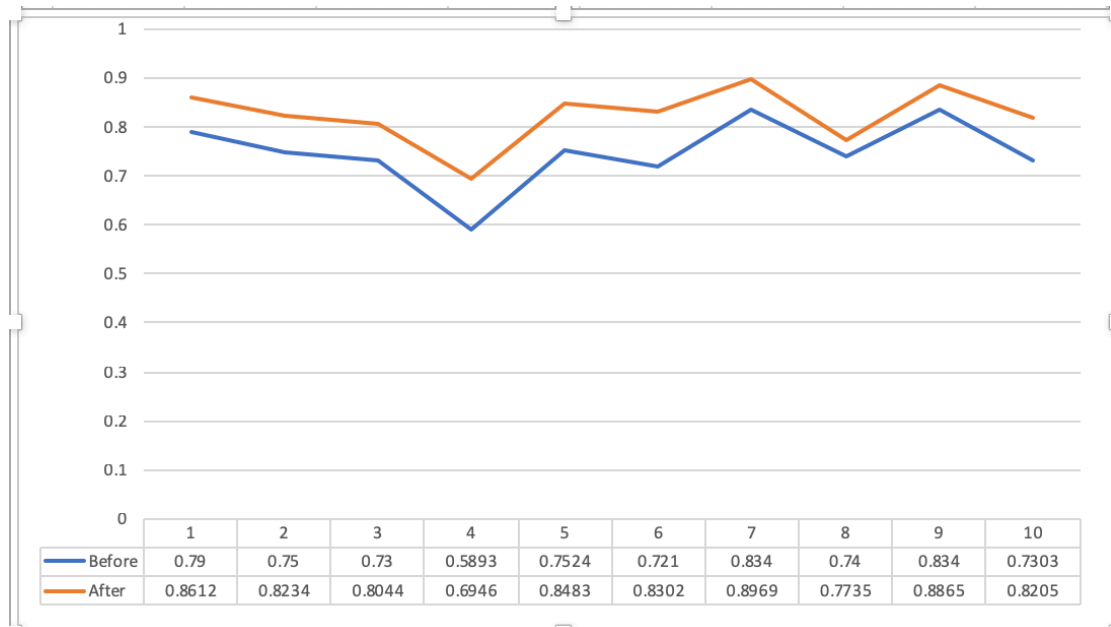


Figure 4.17 Confidence scores for Chair dataset in Experiments 1 and 2.

In Figure 4.18, the graph in Figure 4.17 is expressed as a bar graph for comparison. The blue bars represent the results of the first experiment while the red bars are for the second experiment. The highest increase in score (10.53%) is in the fourth group. The lowest increase was in group 8, an increase of 3.35%. Thus the segmentation has an impact on the images with the lowest original confidence scores.

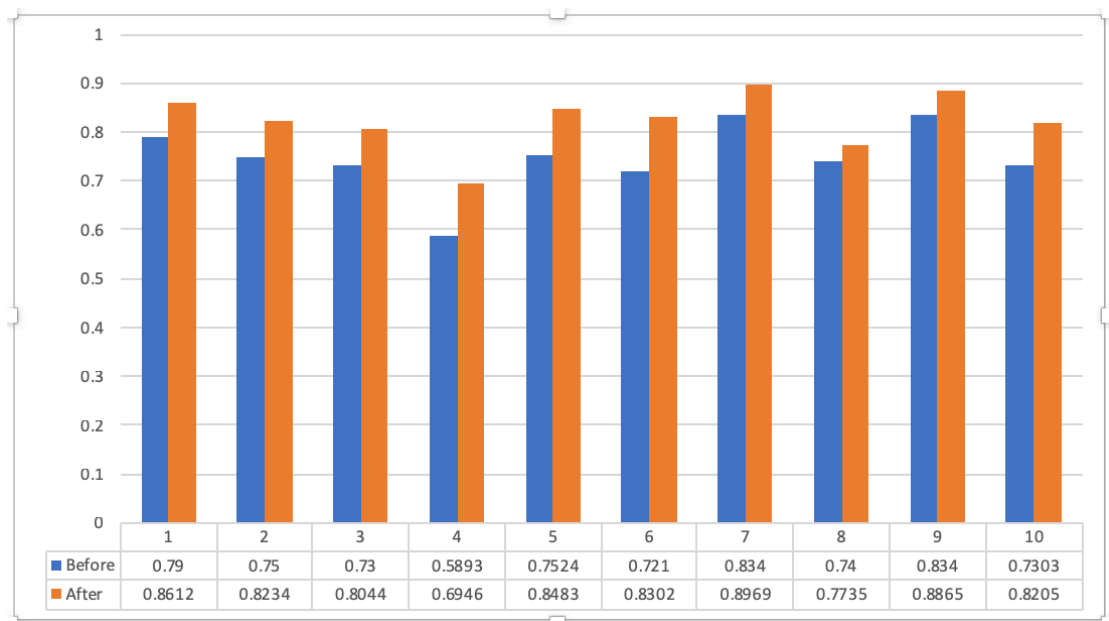


Figure 4.18 The results of Figure 4.17 in bar chart form.

The average accuracy of the first experiment is 72.65%, and that of the second experiment is 82.40%. The above experimental data shows that the accuracy of the recognition changes as the chair rotates. After segmenting the original RGB image using the depth map, the results are more stable, and the accuracy is improved. The average improvement is about 10%.

Suitcase Dataset

A comparison of the confidence scores of the 10 groups of images in experiment 1 and 2 is shown in Figure 4.19. Again, the trends are similar in both cases. This graph is expressed in a bar chart form in Figure 4.20. It can be observed that the confidence scores in the second experiment are higher than in the first. The third group is the one with the greatest difference in scores. For this group, the average score is 83.81% in the first experiment and 91.55% in the second experiment, giving a difference of 7.74%. On the whole, the improvement is relatively small.

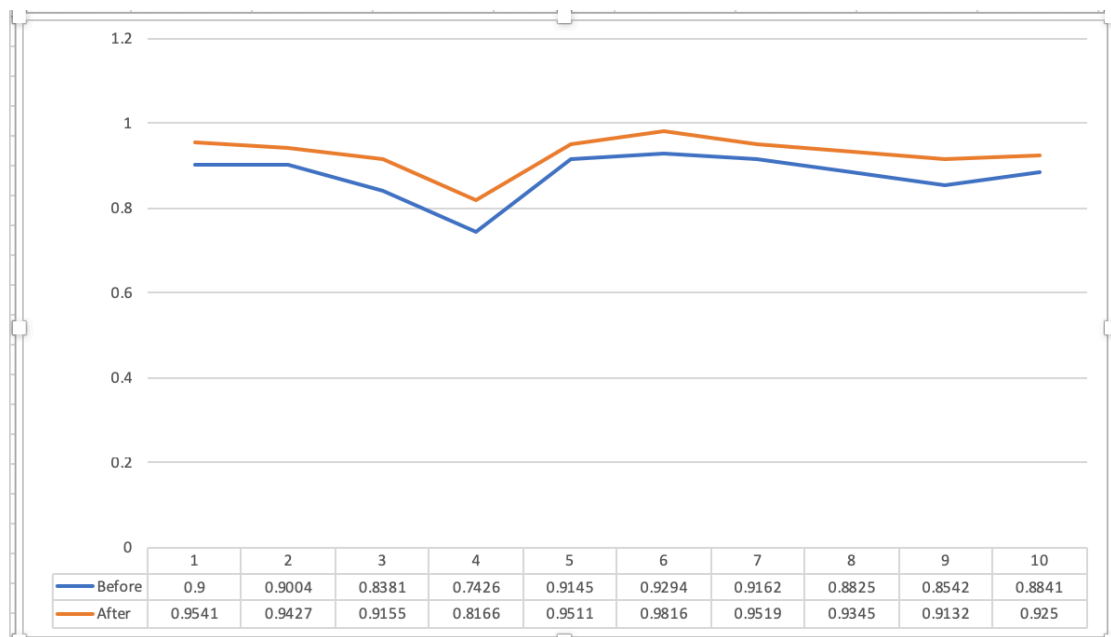


Figure 4.19 Confidence scores for Suitcase dataset in Experiments 1 and 2

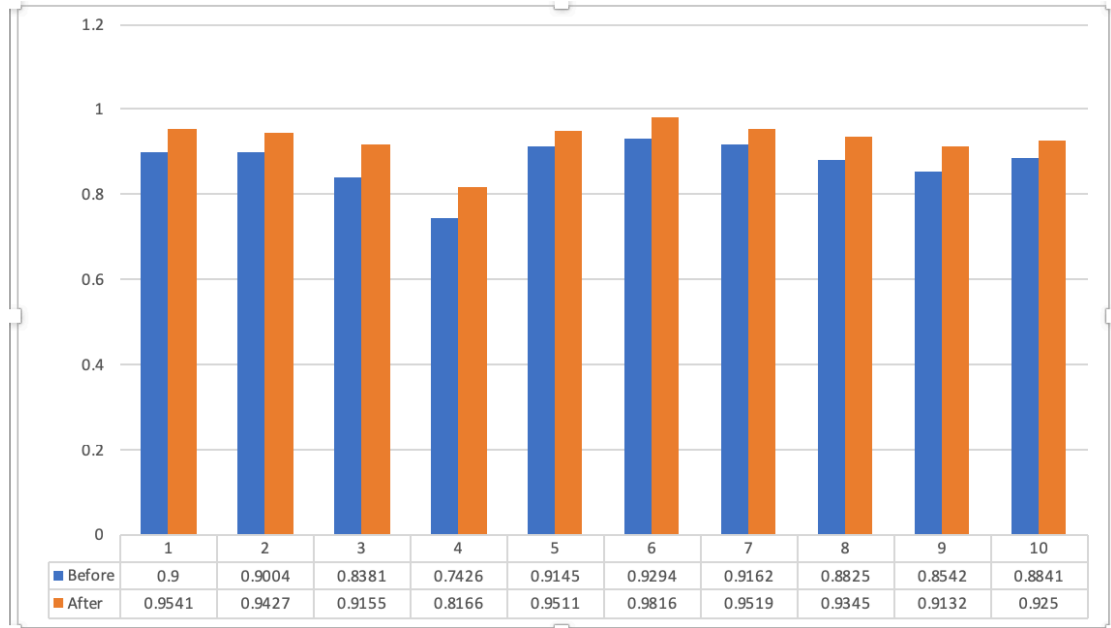


Figure 4.20 Results in Figure 4.19 expressed in bar chart form.

The average score of the first experiment is 87.62%. This is compared with 92.29% for the second experiment – an increase of 5.2%. Compared to the 10% improvement in the chair database, the 5.2% increase in the suitcase dataset is much smaller. But the score of the suitcase dataset is much higher than the chair dataset. So, this is still a good performance.

4.2.2 Other Observations

Apart from the confidence scores, in the course of the experiment, three other phenomena have been observed.

Effect of Angle of View

It has been observed that the confidence score varies as the angle of view is changed.

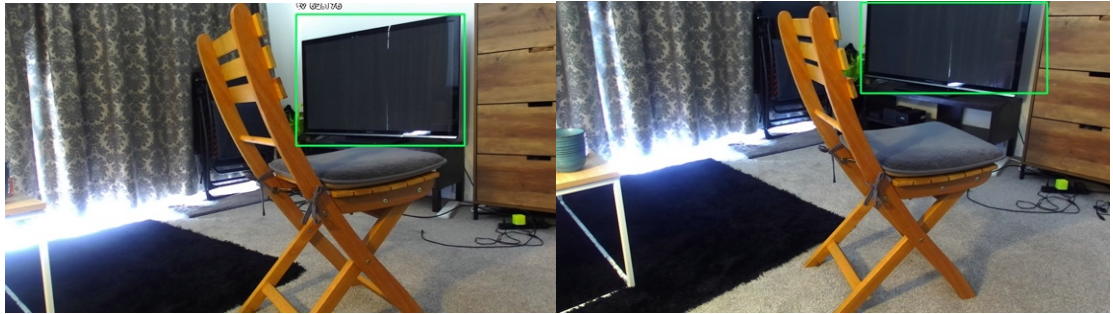


Figure 4.21 Example of undetected object

Low scores or even undetected objects usually occur when the object is viewed side-on. Figure 4.21 shows two images where the chair is not detected by YOLOv3. A plausible reason is that the images are not properly pre-processed. In the traditional object recognition process, the color distribution, overall brightness, and size of the image are usually adjusted to provide the best possible results. However, the RGB images of the chair and suitcase datasets have not been preprocessed. Preprocessing is only performed on the depth images. Some images are dark due to lighting factors. Darker pictures tend to lose image details and have a negative impact on the confidence score. For the above reasons, the darkness of the RGB image affects the accuracy at different view angles.

With object recognition, feature extraction is the first step and an important part of the recognition method. A second plausible reason is that the YOLOv3 model training is not sufficient to extract sufficient features of the object when it is viewed from the side. Increased learning of the objects side-on would possibly improve accuracy.



Figure 4.22 Chair database recognized example

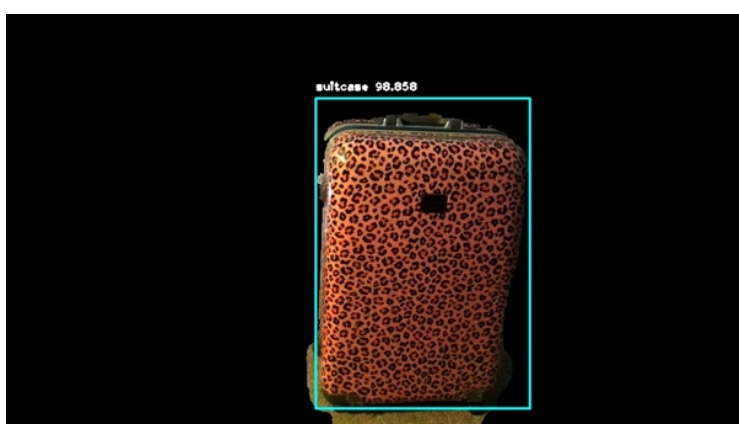


Figure 4.23 Suitcase database recognized example

The Suitcase Dataset Produces Better Results

Figures 4.22 and 4.23 show a chair and a suitcase image that is in the dataset used in the experiments. Both images have been segmented using the depth map. Also, these two images have the object viewed at the same angle. Even though the conditions are roughly the same, the confidence score of the suitcase is significantly higher than that of the chair. It has been discussed in the previous section that the scores of the suitcase images are always higher than the chair images. This could be due to the following two reasons. The first one is that the color of the chair is close to the color of the light. This makes the

chairs more affected by illumination. The small color difference between the foreground and background in the chair images thus results in low scores. Thus, in this case, depth information helps to segment the chair from the background.

The second reason is that the chair is more complicated in structure compared with the suitcase. This makes the chair a lot more difficult to model. Given a set of features, extracting the same points and distinguishing the different points becomes a problem to be solved by the model. Moreover, the shape of the chair also results in more shadows that could interfere with the recognition process.

Noise problem in the RAW format depth map

From experience, in most cases, the Full format of the depth maps has less noise. While this is the case, depth maps in RAW format with large same-color patches are more suitable for cutting. If the noise in these depth maps could be reduced, then Raw format depth maps could produce more accurate segmentation.

The ZED camera is a stereoscopic camera. A stereoscopic camera calculates depth information by matching pixels belonging to the same target in two images. The method used by the stereoscopic camera is called the epipolar constraint. Epipolar constraint refers to the fact that when the same spatial point is imaged separately on the two images, the left projection point p_1 is known, then the corresponding projection point p_2 must be on the polar line relative to p_1 , which can reduce the matching range. When the two cameras of a stereoscopic camera are not in the same horizontal plane, the camera uses

image correction technology to level the two cameras by matrixing the two images.

The key to the method of computing the depth map is the matching of pixel points. To get a depth map with high precision and less noise, the original image cannot lose too much detail. Since lighting conditions affect the quality of the picture, it is possible that if the lighting conditions are good, the noise of the depth map will be reduced.



Figure 4.24 Comparison of different lighting conditions

In order to verify this speculation, depth maps are acquired in different lighting conditions in RAW format and compared. The two images shown in Figure 4.24 are under different lighting contrasts, with the one on the right under brighter lighting. Figure 4.25 shows the corresponding depth maps of these two images. By increasing the brightness of the image, the depth image has less “holes” in them and thus makes it easier to segment the chair. Course this is just an unscientific verification and is not comprehensive nor conclusive. This could be a direction of future research.

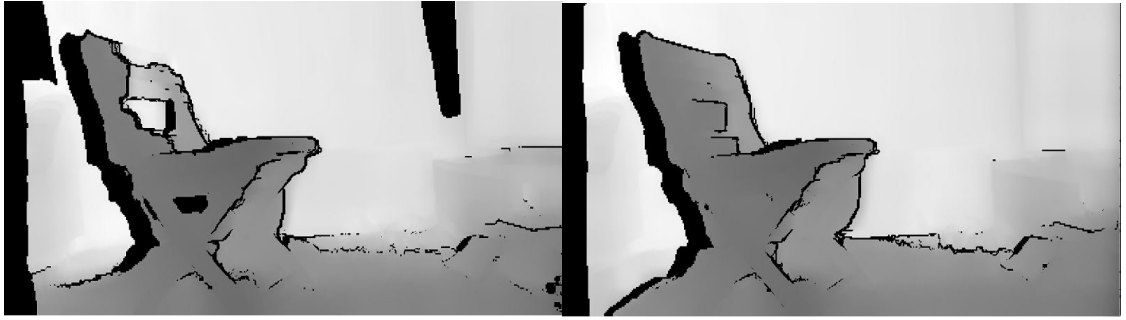


Figure 4.25 RAW format depth map obtained under different lighting conditions.

Chapter 5 Conclusions and Future Works

5.1 Conclusions

Illumination conditions have always been one of the important factors affecting the accuracy of object recognition. Although the performance of the existing object recognition system is already excellent, there is still room for improvement in the case of a complicated lighting environment. With the advent of depth cameras, further improvements in the field of image recognition have become possible. This thesis proposes a method to improve the accuracy of YOLO object recognition model. Our proposed method uses the depth information of segment the target. One of the benefits is that it can reduce the impact of complex environments on recognition. In order to get the environment needed for the experiment, we collected our own depth image database. And we preprocessed the image for the characteristics of the depth map. We collected this database in the absence of sufficient lighting conditions. In the process of collecting the database, we used a strong yellow light source. This practice changes the color characteristics of the object and creates a lot of shadows that can challenge existing recognition systems. After reviewing the literature, we have chosen the YOLO model. In parallel, we chose the Grab-cut algorithm when cutting the depth map.

In order to test the performance of our proposed method, we conducted three experiments in this project. In the first experiment, we used the YOLO model to identify the color images in the database, and the results were not very satisfactory. In the second experiment, we used Grab-cut on the data. The depth map in the middle was cut. After

merging the segmented contour map with the original color map, we have completed the extraction of the target object. In the third experiment, we used YOLO to identify the segmented image. Compare the results of the first and second experiments, after the treatment increased by 5.2% to 10%. This proves that the method we proposed is effective. We also made our own analysis of several phenomena that appeared in the experiment. These phenomena included different angles lead to different accuracy; the accuracy of the suitcase database is higher than the accuracy of the chair database. At the same time, we also analyzed how to reduce the RAW format depth map noise.

5.2 Future Works

Regardless of the great efforts, we contributed to this project, there are still considerable limitations and factors hinder us from betterment. We intend to make improvements in our future work.

1. The database we collected in this project includes about 400 images, and in our database, only one scenario is included. We plan to continue to maintain the collection of the database in the future. And we plan to collect the database in more scenarios.
2. When collecting the depth database, we found that the distance from the ground to the camera is similar to the distance from the object to the camera. So, both have

similar colors and shades in the depth map. This will challenge us to the process of segmentation. In future work, we will begin to work on the use of depth information to identify ground.

3. We used YOLO as our object recognition model in our experiments. In future work, we will try to use other object recognition models and compare their performance.
4. In the process of analysis, to reduce the noise in the depth map. We have proposed ways to increase illumination. In future work, we will investigate how to reduce the noise in the depth map.

References

- [1] W.E.L. Grimson., Object recognition by computer : the role of geometric constraints.
MIT Press, 1990.
- [2] W.E.L. Grimson and D. Huttenlocher., “On the sensitivity of geometric hashing.” . In
Proceedings Third International Conference on Computer Vision, 1990, pages
334–338.
- [3] Carmichael, O. T., & Hebert, M., “Object Recognition by a Cascade of Edge Probes.”
In BMVC, 2002, pp. 1-10.
- [4] Viola, P., & Jones, M., “Rapid object detection using a boosted cascade of simple
features.” IEEE Computer Society Conference on Computer Vision and Pattern
Recognition, 2001, pp. I-I.
- [5] Joachims, T., “Making large-scale SVM learning practical” Technical Report, SFB
475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität
Dortmund, 1998
- [6] Boughorbel, S., Tarel, J. P., & Fleuret, F., “Non-Mercer Kernels for SVM Object
Recognition.” In BMVC , 2004, pp. 1-10.
- [7] Llorca, D. F., Arroyo, R., & Sotelo, M. A., “Vehicle logo recognition in traffic images
using HOG features and SVM.” International IEEE Conference on Intelligent
Transportation Systems-(ITSC), 2013, pp. 2229-2234.
- [8] Zhang, H., Berg, A. C., Maire, M., & Malik, J., “SVM-KNN: Discriminative nearest
neighbor classification for visual category recognition.” IEEE Computer Society
Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2126-

2136.

- [9] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. “The pascal visual object classes (voc) challenge.” *International journal of computer vision*, Vol. 88, No. 2, 2010, pp.303-338.
- [13] Yan, J., Lei, Z., Yi, D., & Li, S. Z., “Multi-pedestrian detection in crowded scenes: A global view.” *IEEE Conference on Computer Vision and Pattern Recognition* , 2012, pp. 3124-3129
- [12] Yan, J., Zhang, X., Lei, Z., Liao, S., & Li, S. Z., “Robust multi-resolution pedestrian detection in traffic scenes.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3033-3040.
- [10] Yan, J., Zhang, X., Lei, Z., Yi, D., & Li, S. Z., “Structural models for face detection.” *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013, pp. 1-6.
- [11] Zhu, X., & Ramanan, D., “Face detection, pose estimation, and landmark localization in the wild.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2879-2886.
- [14] Yan, J., Lei, Z., Wen, L., & Li, S. Z., “The fastest deformable part model for object detection.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2497-2504.
- [15] LeCun, Y., Bengio, Y., & Hinton, G. , “Deep learning.” *nature*, Vol. 521, 2015, pp.436
- [16] Girshick, R., Donahue, J., Darrell, T., & Malik, J., “Rich feature hierarchies for

- accurate object detection and semantic segmentation.” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.
- [17] He, K., Zhang, X., Ren, S., & Sun, J., “ Spatial pyramid pooling in deep convolutional networks for visual recognition.” In European conference on computer vision, Vol 37, No. 9, 2014, pp. 346-361
- [18] Girshick, R., “Fast r-cnn.” In Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440-1448.
- [19] Ren, S., He, K., Girshick, R., & Sun, J., “Faster r-cnn: Towards real-time object detection with region proposal networks.” In Advances in neural information processing systems, 2015, pp. 91-99.
- [20] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A., “The pascal visual object classes (voc) challenge.” International journal of computer vision, Vol. 88, No. 2, 2010, pp.303-338.
- [21] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L., “Microsoft coco: Common objects in context.” In European conference on computer vision, 2014, pp. 740-755.
- [22] Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z., “Single-shot refinement neural network for object detection.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4203-4212.
- [23] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., “You only look once: Unified, real-time object detection.” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

- [24] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C.,
 “Ssd: Single shot multibox detector.” In European conference on computer vision,
 2016, pp. 21-37.
- [25] Redmon, J., & Farhadi, A., “YOLOv3: An incremental improvement.” arXiv
 preprint arXiv:1804.02767, 2018
- [26] Litomisky, K., “Consumer rgb-d cameras and their applications.” Rapport technique,
 University of California, No.20, 2012
- [27] Horaud, R., Hansard, M., Evangelidis, G., & M  nier, C., “An overview of depth
 cameras and range scanners based on time-of-flight technologies.” Machine
 vision and applications, Vol. 27, No.7, 2016, pp.1005-1020.
- [28] Izadi, S., Newcombe, R. A., Kim, D., Hilliges, O., Molyneaux, D., Hodges, S., ... &
 Fitzgibbon, A., “Kinectfusion: real-time dynamic 3d surface reconstruction and
 interaction.” In ACM SIGGRAPH 2011 Talks, 2011, p. 23
- [29] Engelhard, N., Endres, F., Hess, J., Sturm, J., & Burgard, W., “Real-time 3D visual
 SLAM with a hand-held RGB-D camera.” In Proc. of the RGB-D Workshop on
 3D Perception in Robotics at the European Robotics Forum, Vol. 180, April, 2011,
 pp. 1-15
- [30] Kerl, C., Sturm, J., & Cremers, D., “Dense visual SLAM for RGB-D cameras,”
 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),
 2013, pp. 2100-2106.
- [31] Engel, J., Schops, T., & Cremers, D., “LSD-SLAM: Large-scale direct monocular
 SLAM.” In European Conference on Computer Vision, September, 2014, pp.

- [32] Whelan, T., Kaess, M., Johannsson, H., Fallon, M., Leonard, J. J., & McDonald, J.,
 “Real-time large-scale dense RGB-D SLAM with volumetric fusion.” *The International Journal of Robotics Research*, Vol. 34, No. 4-5, 2015, pp. 598-626.
- [33] Goswami, G., Bharadwaj, S., Vatsa, M., & Singh, R., “On RGB-D face recognition using Kinect.” In *Biometrics: Theory, Applications and Systems (BTAS)*, 2013, pp. 1-6.
- [34] Li, B. Y., Mian, A. S., Liu, W., & Krishna, A., “Using kinect for face recognition under varying poses, expressions, illumination and disguise.” *IEEE Workshop on Applications of Computer Vision (WACV)*, 2013, pp. 186-192
- [35] Bo, L., Ren, X., & Fox, D., “Unsupervised feature learning for RGB-D based object recognition.” In *Experimental Robotics*, 2013, pp. 387-402.
- [36] Gupta, S., Girshick, R., Arbeláez, P., & Malik, J., “Learning rich features from RGB-D images for object detection and segmentation.” In *European Conference on Computer Vision*, 2014, pp. 345-360
- [37] Cheng, Y., Zhao, X., Huang, K., & Tan, T., “Semi-supervised learning for rgb-d object recognition.” *22nd International Conference on In Pattern Recognition (ICPR)*, 2014, pp. 2377-2382.
- [38] Lai, K., Bo, L., Ren, X., & Fox, D., “A large-scale hierarchical multi-view rgb-d object dataset.” In *Robotics and Automation (ICRA)*, 2011, pp. 1817-1824.
- [39] KaewTraKulPong, P., & Bowden, R. “An improved adaptive background mixture model for real-time tracking with shadow detection.” In *Video-based surveillance*

systems, 2002, pp. 135-144

- [40] Bo, L., Ren, X., & Fox, D., “Kernel descriptors for visual recognition.” In Advances in neural information processing systems, 2010, pp. 244-252.
- [41] Bo, L., Lai, K., Ren, X., & Fox, D., “Object recognition with hierarchical kernel descriptors.” In Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1729-1736.
- [42] Blum, M., Springenberg, J. T., Wülfing, J., & Riedmiller, M. A., “A learned feature descriptor for object recognition in rgb-d data.” In ICRA, 2012, pp. 1298-1303.
- [43] Song, S., Lichtenberg, S. P., & Xiao, J., “Sun rgb-d: A rgb-d scene understanding benchmark suite.” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 567-576.
- [44] Silberman, N., & Fergus, R. , “Indoor scene segmentation using a structured light sensor.” IEEE International Conference on Computer Vision Workshops (ICCV Workshops), November, 2011, pp. 601-608.
- [45] Firman, M., “RGB-D datasets: Past, present and future.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 19-31.
- [46] Dib, A., & Charpillet, F., “Pose estimation for a partially observable human body from RGB-D cameras.” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 4915-4922.
- [47] Choi, S., Zhou, Q. Y., Miller, S., & Koltun, V., “A large dataset of object scans.” arXiv preprint arXiv:1602.02481, 2016

- [48] Park, J., Kim, H., Tai, Y. W., Brown, M. S., & Kweon, "High quality depth map upsampling for 3d-tof cameras." IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1623-1630
- [49] Hauck, A., Ruttinger, J., Sorg, M., & Farber, G., "Visual determination of 3D grasping points on unknown objects with a binocular camera system." IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, No. 99CH36289, 1999, pp. 272-278
- [50] Jansen, B., Temmermans, F., & Deklerck, R., "3D human pose recognition for home monitoring of elderly." 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 4049-4051
- [51] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., ... & Blake, A., "Real-time human pose recognition in parts from single depth images." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2011, pp. 1297-1304
- [52] Keskin, C., Kırac, F., Kara, Y. E., & Akarun, L., "Real time hand pose estimation using depth sensors." In Consumer depth cameras for computer vision, 2013, pp. 119-137
- [53] Athitsos, V., & Sclaroff, S., "Estimating 3D hand pose from a cluttered image." IEEE Computer Society Conference on Computer Vision and Pattern Recognition, , 2003, pp. II-432
- [54] Malassiotis, S., & Strintzis, M. G., "Real-time hand posture recognition using range data." Image and Vision Computing, Vol. 26, No. 7, 2008, pp.1027-1037.

- [55] Liu, X., & Fujimura, K., “Hand gesture recognition using depth data.”, In Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 529-534
- [56] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R., “Indoor segmentation and support inference from RGB-D images.” In European Conference on Computer Vision, Berlin, Heidelberg, 2012, pp. 746-760.
- [57] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R., “Indoor segmentation and support inference from RGB-D images.” In European Conference on Computer Vision, Berlin, Heidelberg, 2012, pp. 746-760.
- [58] Holz, D., Holzer, S., Rusu, R. B., & Behnke, S., “Real-time plane segmentation using RGB-D cameras.” In Robot Soccer World Cup, Berlin, Heidelberg., 2011, pp. 306-317.
- [59] Gupta, S., Arbelaez, P., & Malik, J., “Perceptual organization and recognition of indoor scenes from RGB-D images.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 564-571.
- [60] Ren, X., Bo, L., & Fox, D., “Rgb-(d) scene labeling: Features and algorithms.” 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2759-2766
- [61] Koppula, H. S., Anand, A., Joachims, T., & Saxena, A., “ Semantic labeling of 3d point clouds for indoor scenes.” In Advances in neural information processing systems, 2011, pp. 244-252.
- [62] Camplani, M., & Salgado, L., “Background foreground segmentation with RGB-D

- Kinect data: An efficient combination of classifiers.” *Journal of Visual Communication and Image Representation*, Vol. 25, No.1, 2014, pp.122-136.
- [63] Redmon, J., & Farhadi, A., “YOLOv3: An incremental improvement.” *arXiv preprint arXiv:1804.02767*, 2018
- [64] Mishra, A. K., Shrivastava, A., & Aloimonos, Y., "Segmenting “simple” objects using RGB-D.” *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4406-4413
- [65] Rother, C., Kolmogorov, V., & Blake, A., “Grabcut: Interactive foreground extraction using iterated graph cuts.” In *ACM transactions on graphics (TOG)*, Vol.23, No. 23, 2004, pp. 309-314
- [66] Talbot, J. F., & Xu, X., “Implementing grabcut.” *Brigham Young University*, 2006
- [67] Han, S., Tao, W., Wang, D., Tai, X. C., & Wu, X., “Image segmentation based on GrabCut framework integrating multiscale nonlinear structure tensor.” *IEEE transactions on image processing*, Vol. 18, No. 10, 2009, pp.2289-2302.
- [68] Hernández-Vela, A., Reyes, M., Ponce, V., & Escalera, S., “Grabcut-based human segmentation in video sequences.” *Sensors*, vol 12, no.11, 2012, pp.15376-15393.
- [69] Poullot, S., & Satoh, S. I., “VabCut: a video extension of GrabCut for unsupervised video foreground object segmentation.” *International Conference on Computer Vision Theory and Applications (VISAPP)* , Vol.2, 2014, pp. 362-371.
- [70] Hua, S., & Shi, P., “GrabCut color image segmentation based on region of interest.” *International Congress on Image and Signal Processing (CISP)*, 2014, pp. 392-396.

- [71] Gulshan, V., Lempitsky, V., & Zisserman, A., "Humanising grabcut: Learning to segment humans using the kinect." IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1127-1133.
- [72] He, H., McKinnon, D., Warren, M., & Upcroft, B., "Graphcut-based interactive segmentation using colour and depth cues." 2010
- [73] Seki, A., & Pollefeys, M., "Patch Based Confidence Prediction for Dense Disparity Map." In BMVC , Vol. 2, No. 3, 2016, pp. 4
- [74] Hamzah, R. A., & Ibrahim, H., "Literature survey on stereo vision disparity map algorithms." Journal of Sensors, 2016
- [75] Mühlmann, K., Maier, D., Hesser, J., & Männer, R., "Calculating dense disparity maps from color stereo images, an efficient implementation." International Journal of Computer Vision, Vol. 47, 2002, pp.79-88.
- [76] Shi, J., & Malik, J., "Normalized cuts and image segmentation." Departmental Papers (CIS), 2000, pp.107.
- [77] Felzenszwalb, P. F., & Huttenlocher, D. P., "Efficient graph-based image segmentation." International journal of computer vision, Vol.59, No.2, 2004, pp.167-181.
- [78] Pal, N. R., & Pal, S. K., "A review on image segmentation techniques." Pattern recognition, Vol. 26, No.9, 1993, pp.1277-1294.
- [79] Hull, J. J., "A database for handwritten text recognition research." IEEE Transactions on pattern analysis and machine intelligence, Vol. 16, No. 5, 1994, pp.550-554.
- [80] Kim, G., Govindaraju, V., & Srihari, S. N., "An architecture for handwritten text

- recognition systems.” *International Journal on Document Analysis and Recognition*, Vol.2, No.1, 1999, pp.37-44.
- [81] Chang, S. L., Chen, L. S., Chung, Y. C., & Chen, S. W., “Automatic license plate recognition.” *IEEE transactions on intelligent transportation systems*, Vol.5, No.1, 2004, pp.42-53.
- [82] Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Loumos, V., & Kayafas, E., “A license plate-recognition algorithm for intelligent transportation system applications.” *IEEE Transactions on Intelligent transportation systems*, Vol. 7, No.3, 2006, pp.377-392.
- [83] Jensen, J. R., & Lulla, K., “Introductory digital image processing: a remote sensing perspective.”, 1987
- [84] LeCun, Y., Bengio, Y., & Hinton, G., “Deep learning.” *nature*, Vol.521, No.7553, 2015, pp.436,
- [85] Schmidhuber, J., “Deep learning in neural networks: An overview.” *Neural networks*, Vol.61, 2015, pp.85-117
- [86] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A., “The pascal visual object classes challenge: A retrospective.” *International journal of computer vision*, Vol. 111, No.1, 2015, pp.98-136.
- [87] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L., “Imagenet: A large-scale hierarchical image database.” *IEEE conference on computer vision and pattern recognition*, 2009, pp. 248-255
- [88] Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., & Fei-Fei, L., “ILSVRC-2012.”,

2012

- [89] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Hasan, M., Van Esesn, B. C., ... & Asari, V. K., "The history began from alexnet: A comprehensive survey on deep learning approaches." arXiv preprint arXiv:1803.01164, 2018
- [90] Pan, G., Han, S., Wu, Z., & Wang, Y., "3D face recognition using mapped depth images." IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 175-175
- [91] Leyvand, T., Meekhof, C., Wei, Y. C., Sun, J., & Guo, B., "Kinect identity: Technology and experience." Computer, Vol.44, No. 4, 2011, pp.94-96
- [92] Azuma, R. T., "A survey of augmented reality." Presence: Teleoperators & Virtual Environments, Vol. 6, No.4, 1997, pp.355-385
- [93] Smisek, J., Jancosek, M., & Pajdla, T., "3D with Kinect." In Consumer depth cameras for computer vision, 2013, pp. 3-25
- [94] Gupta, T., & Li, H. "Indoor mapping for smart cities—An affordable approach: Using Kinect Sensor and ZED stereo camera." International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2017, pp. 1-8
- [95] Carr, P., Sheikh, Y., & Matthews, I., "Monocular object detection using 3d geometric primitives." In European Conference on Computer Vision, 2012, pp. 864-878
- [96] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., & Burgard, W., "An evaluation of the RGB-D SLAM system." In Icra, Vol. 3, No. c, May 2012, pp. 1691-1696
- [97] Kerl, C., Sturm, J., & Cremers, D., "Dense visual SLAM for RGB-D cameras." In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2100-2106.
- [98] Heun, V., Kasahara, S., & Maes, P., "Smarter objects: using AR technology to program physical objects and their interactions." In CHI'13 Extended Abstracts

on Human Factors in Computing Systems, 2013, pp. 961-966

- [99] Jayaram, S., Vance, J. M., Gandh, R., Jayaram, U., & Srinivasan, H., “Assessment of VR technology and its applications to engineering problems.” *Journal of Computing and Information Science in Engineering*, Vol. 1, No.1, 2001, pp.72.
- [100] Lienhart, R., & Maydt, J., “An extended set of haar-like features for rapid object detection.” In *Proceedings. International Conference on Image Processing*, Vol. 1, 2002, pp. I-I
- [101] Mita, T., Kaneko, T., & Hori, O., “Joint haar-like features for face detection.” In *Tenth IEEE International Conference on Computer Vision*, Vol. 2, October 2005, pp. 1619-1626.
- [102] Alpaydin, E., & Kaynak, C., “Cascading classifiers.” *Kybernetika*, Vol.34, No.4, 1998, pp.369-374.
- [103] Van Loan, C., *Computational frameworks for the fast Fourier transform*, Vol. 10, 1992
- [104] Deng, L., & Yu, D., “Deep learning: methods and applications.” *Foundations and Trends® in Signal Processing*, Vol.7, 2014, pp.197-387.
- [105] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I., “A survey on deep learning in medical image analysis”, *Medical image analysis*, Vol.42, 2017, pp.60-88.
- [106] Deng, L., Li, J., Huang, J. T., Yao, K., Yu, D., Seide, F., ... & Gong, Y., “Recent advances in deep learning for speech research at Microsoft.” *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8604-8608
- [107] "ZED Stereo Camera - Stereolabs", *Stereolabs.com*, 2019. [Online]. Available: <https://www.stereolabs.com/zed/>.
- [108] Kwatra, V., Schödl, A., Essa, I., Turk, G., & Bobick, A., “Graphcut textures: image and video synthesis using graph cuts.” *ACM Transactions on Graphics (ToG)*, Vol.22, No.3, 2003, pp.277-286.
- [109] Redmon, J., & Farhadi, A., “YOLO9000: better, faster, stronger.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp.

7263-7271.

- [110] “Darknet: Open source neural networks”, Redmon, J, 2013–2016. [Online].
Available: <http://pjreddie.com/darknet/>,