Optimization of object classification and recognition for e-commerce logistics

Ren, Meixuan

2018

Ren, M. (2018). Optimization of object classification and recognition for e-commerce logistics. Master's thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/75867>

<https://doi.org/10.32657/10356/75867>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**OPTIMIZATION OF OBJECT
CLASSIFICATION AND RECOGNITION
FOR E-COMMERCE LOGISTICS**

REN MEIXUAN

**SCHOOL OF MECHANICAL AND
AEROSPACE ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY**

2018

**OPTIMIZATION OF OBJECT
CLASSIFICATION AND RECOGNITION
FOR E-COMMERCE LOGISTICS**

Submitted by

REN MEIXUAN

REN MEIXUAN

Robotics Research Centre

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering (Mechanical Engineering)

2018

ABSTRACT

E-commerce, an online transaction in the information-based society, draws on various technologies to achieve automated order picking process for the fulfillment of supply chain's productivity. Robotic systems like Amazon Kiva are applied in logistics warehouses for low labor cost and high efficiency. Amazon Robotic Challenge (ARC) in 2017 aimed to explore a solution to bin picking problem in cluttered environment which is a common situation in logistics warehouses. Since the perception strategy is a key factor to picking performance, this thesis proposes a robust vision-based approach to object recognition for the robotic system of Team Nangyang in ARC.

In this thesis, traditional methods and deep learning methods for object recognition are reviewed and verified. Five perception approaches based on GMS (Grid-based Motion Statistics), CNN (convolutional neural network) and image differencing are proposed to achieve the order picking. First the experiments of GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window are designed and conducted. Then two hybrid methods which combine CNN + dynamic sliding window with GMS and image differencing are proposed and tested to get a more accurate suction point. Finally, after comparing all the experimental results, a conclusion is drawn that CNN + dynamic sliding window + image differencing is a robust perception method to realize the object recognition in unstructured workspace in logistics warehouses.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to **Prof. Chen I-Ming** for his valuable encouragement advice and guidance.

My special thanks to people who helped me during the project, which includes my project leader **Dr. Albert Causo** and my research group members such as **Ms. Pang Wee Ching, Mr. Chong Zheng Hao, Mr. Ramamoorthy Luxman, Mr. Kok Yuan Yik, Mr. Weng Ching-Yeng, Mr. Zhao Yi, Mr. Hendra Suratno Tju.** Without their help and rich experience, my research could not proceed so smoothly.

Finally, I wish to acknowledge the help provided by my family and friends for the support and inspiring discussion received.

TABLE OF CONTENTS

ABSTRACT.....	I
ACKNOWLEDGEMENTS.....	II
LIST OF FIGURES.....	VI
LIST OF TABLES.....	VIII
CHAPTER1 Introduction.....	1
1.1 Background and Motivation.....	1
1.2 Objectives.....	3
1.3 Organizations.....	4
CHAPTER2 Literature Review.....	6
2.1 Traditional Methods.....	7
Determination of feature.....	7
Feature detector.....	8
Feature descriptor.....	10
Feature matcher.....	14
2.2 Deep Learning Methods.....	15
LeNet.....	16
AlexNet.....	16
ZFNet.....	17

VGGNet.....	17
Inception (GoogLeNet).....	17
ResNet.....	18
MaskRCNN.....	18
CHAPTER3 Problem Statement.....	20
3.1 Problem Statement.....	20
3.2 Introduction of ARC.....	23
3.3 Decision of Methodology.....	25
CHAPTER4 Recognition Approaches.....	27
4.1 Introduction of GMS and CNN.....	28
GMS.....	28
CNN.....	29
4.2 GMS + Fixed Sliding Window.....	32
4.3 CNN + Fixed Sliding Window.....	34
4.4 CNN + Dynamic Sliding Window.....	37
4.5 CNN + Dynamic Sliding Window + GMS.....	38
4.6 CNN + Dynamic Sliding Window + Image Differencing.....	40
CHAPTER5 Experiment and Results.....	42
5.1 Experimental Setup.....	42

5.2 Data Acquisition and Training.....	43
Database setup.....	43
Training process.....	45
Training results.....	48
5.3 Experimental Results.....	50
Accuracy calculation.....	50
GMS + fixed sliding window.....	50
CNN + fixed sliding window.....	53
CNN + dynamic sliding window.....	55
CNN + dynamic sliding window + GMS.....	57
CNN + dynamic sliding window + image differencing.....	58
5.4 Discussion.....	60
Improvement process.....	60
Results comparison.....	63
CHAPTER6 Conclusions and Perspectives.....	70
6.1 Conclusions.....	70
6.2 Perspectives.....	74
REFERENCE.....	75

LIST OF FIGURES

Figure 1 General process for the vision system.....	7
Figure 2 Neighborhood feature points for GMS matcher.....	15
Figure 3 Architecture of LeNet (CNN) [35].....	16
Figure 4 Structure for the storage system.....	21
Figure 5 Sample image acquired by the robot system.....	21
Figure 6 Amazon Robotic Challenge dataset in 2017 [45].....	22
Figure 7 Methodologies summary for ARC 2015.....	23
Figure 8 Robotic system of Delft in ARC 2016 [48].....	24
Figure 9 Overview flowchart of the exploration process.....	27
Figure 10 Architecture for GoogLeNet Inceptionv3 [42].....	30
Figure 11 Introduction of the layers in GoogLeNet Inceptionv3.....	30
Figure 12 Neuron structure of the convolution layer [49].....	31
Figure 13 Recognition process of GMS + fixed sliding window.....	33
Figure 14 Recognition process of CNN + fixed sliding window.....	35
Figure 15 Recognition process of CNN + dynamic sliding window + GMS.....	39
Figure 16 GMS matching results of the sliding window image.....	39
Figure 17 Recognition process of CNN + dynamic sliding window + image differencing.....	41
Figure 18 Part of the database for GMS.....	44
Figure 19 Part of the database for CNN.....	45
Figure 20 Original dataset example used to train the model.....	48

Figure 21 Accuracy result of the CNN training process.....	49
Figure 22 Cross entropy result of the CNN training process.....	49
Figure 23 Other results for the CNN training process.....	49
Figure 24 Recognition and matching result by SIFT.....	50
Figure 25 Recognition and matching result by GMS.....	51
Figure 26 Recognition accuracy of GMS + fixed sliding window.....	52
Figure 27 Recognition accuracy of CNN + fixed sliding window.....	55
Figure 28 Recognition accuracy of CNN + dynamic sliding window.....	56
Figure 29 Recognition accuracy of CNN + dynamic sliding window + GMS.....	58
Figure 30 Recognition accuracy of CNN + dynamic sliding window + image differencing.....	59
Figure 31 GMS + fixed sliding window result for the item with feature.....	61
Figure 32 GMS + fixed sliding window result for the item without feature.....	61
Figure 33 Problem of the wrong suction point location.....	62
Figure 34 Result comparison of GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window.....	65
Figure 35 Advantages and disadvantages of GMS and CNN methods.....	66
Figure 36 Result comparison of CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing....	68
Figure 37 Comparison of the two hybrid recognition methods.....	69

LIST OF TABLES

Table 1 Accuracy of different deep learning models [41, 42].....	18
Table 2 Dynamic sliding window sizes for different items.....	37
Table 3 Image number for the training process.....	47
Table 4 Recognition accuracy of GMS + fixed sliding window.....	51
Table 5 Tuning the parameters of window size and step size for CNN + fixed sliding window.....	53
Table 6 Recognition accuracy of CNN + fixed sliding window.....	54
Table 7 Recognition accuracy of CNN + dynamic sliding window.....	55
Table 8 Recognition accuracy of CNN + dynamic sliding window + GMS.....	57
Table 9 Recognition accuracy of CNN + dynamic sliding window + image differencing	58
Table 10 Result comparison of GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window.....	64
Table 11 Result comparison of CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing....	67
Table 12 Result comparison of all the recognition methods.....	72

CHAPTER1 Introduction

1.1 Background and Motivation

E-commerce, an aggregation of commercial transaction, is in a boom with the growth of economy and information technology. Considering its characteristics of high efficiency and convenience, business enterprises are now transferring into an e-commerce age. Logistics activities play a significant role in online transaction by supporting the supply and sale chain. In order to increase the efficiency and decrease the labor cost, it is necessary to transform the logistics activities into full automated ones. Under this occasion, robots become a good tool to meet the commercial requirements and achieve the goal of industrial automation [1, 2].

Logistics robots for the warehouses soared and became popular since Amazon purchased Kiva Systems, which consisted of Autonomous Mobile Robots (AMRs) in 2012. After rebranding Kiva as Amazon Robotic and testing AMRs in other companies, Amazon keeps trying to figure out an optimal way to design, manufacture and deploy them [3]. Amazon robotic system is testified to perform well in items of accuracy and efficiency in logistics activities. With the encouragement of some successful examples, increasing number of e-commerce companies start concentrating on setting up automated robotic logistics systems.

Order picking, one of the logistics activities in warehouses, is taking and collecting products in a specified quantity before shipment according to customers'

orders. Because of its significance on ensuring high productivity of supply chain, the realization of the order picking by robots has become a popular research area. However, in some warehouses, a common occasion is that products may not be stored orderly and tidily. Thus, commercially viable automated picking according to a specific order in unstructured workspace is a challenging research topic. Amazon Robotic Challenge (ARC) is held to explore a solution to the item picking problem in unstructured workspace in order to set up a connection between industrial automation and academic research. According to the rules for ARC in 2017, our dual-arm robotic system is set up to accomplish a completely automated order picking process.

A picking robotic system usually consists of a mechanical system, a vision system and a program manager. The program manager is used to get the order information and send it to the vision system for recognition. After computing and getting the item location, motion planning is conducted to grip the item to the right place in the storage system. To acquire a robust and accurate systematic performance, the key point is that the robotic system needs a strong recognition and classification algorithm and strategy. Therefore, this thesis focuses on the perception method of the robotic system and tries to improve it to realize high-quality recognition. The research of the perception method is based on the robotic system and the order picking process in ARC 2017.

1.2 Objectives

Nowadays, intelligent automatic robots in warehouses are worth studying to realize high efficiency and accuracy. Accurate recognition methods and strategies are needed to ensure the order no matter in messy or common workspace. The research objective for the thesis is to find out a robust and feasible perception method to realize order picking in unstructured environment. In this thesis, the problem is defined under the same condition of 2017 Amazon Robotic Challenge. Therefore, the main task is proposing and implementing a robust perception strategy to realize object recognition in Amazon Robotic Challenges. After summarizing the recognition methodologies, an exploration process of the methods which can be used for object recognition in unstructured bins is explained. CNN + dynamic sliding window + image differencing which is a hybrid method of deep learning method and traditional method is proposed to achieve a high recognition rate up to about 95%.

The main objectives include:

1. Implementing the robotic system to accomplish the perception strategy of the order picking task under the settings of Amazon Robotic Challenge.
2. Proposing possible perception methods by literature review and setting up the experiments to test the object recognition results.
3. Evaluating the recognition accuracy of the proposed strategies and analyzing the systematic performance.
4. Making a conclusion of a perception method of object recognition in messy bins for the intelligent automatic picking robots in warehouses.

1.3 Organizations

The report is divided into six chapters, which are briefly summarized as following:

Chapter 1 briefly introduces the background of the picking robots for e-commerce logistics and explains the motivation of researching object recognition in cluttered workspace. The objectives and organizations of the thesis are also introduced.

Chapter 2 reviews and discusses the theories and characteristics of the traditional methods and deep learning methods for the perception strategy of the robotic system.

Chapter 3 shows some cases of actual robotic picking systems in Amazon Robotic Challenge. After the statement of rules and the problem, methodologies of the recognition system in our study are analyzed and decided according to the literature review results and robot structure.

Chapter 4 introduces the recognition methods used in our robot system. This chapter contains the theory and algorithms for GMS (Grid-based Motion Statistics) and CNN (convolutional neural network), GMS + fixed sliding window, CNN + fixed sliding window, CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing.

Chapter 5 explains the overall experimental design and database setup. The relevant results are presented which include the devices, data collection and training, and the recognition accuracy of the five methods. This chapter also explains the complete improvement process of the recognition strategy and makes a comparison among the five approaches.

Chapter 6 draws the conclusion from comparing and summarizing the perception methods. The hybrid method of CNN with dynamic sliding window and image differencing gives the best performance with a recognition accuracy of 95%.

CHAPTER2 Literature Review

Considering the working process of the e-commerce warehouses, bin-picking robots are required to classify the ordered objects and accomplish the picking tasks. Basically, the solution contains six steps: initial data acquisition, object detection, pose estimation, path planning, object grasping and object placement. However, industrial automation for order picking still remains a research topic because of the complex working conditions and the accuracy. Therefore, the accuracy and stability of the logistics robots need to be improved [4].

Apparently, the vision system is a key factor of picking robots in warehouses to realize and ensure high accuracy. A robust object recognition method is needed to achieve an automated picking process in cluttered environment. In this chapter, methods for object recognition are presented and discussed for further use and application.

The general process for the object recognition is shown in Figure 1. Based on the principles, object recognition can be divided into traditional methods and deep learning methods. Traditional methods mostly concentrate on the operations of features. They realize the recognition by finding and matching the features between target images and reference databases. As for deep learning methods, convolutional neural network is used to train the classification model and implement the recognition process. Both of the methods will be discussed in the following literature review.

Figure 1 General process for the vision system

2.1 Traditional Methods

In traditional computer vision, feature contains the information of an image such as some key points and color regions. To achieve the recognition, feature detectors are first applied to find and extract the feature information. Then feature descriptors are usually used to express the feature by different mathematical methods. Finally, recognition will be accomplished by applying proper feature matchers to get correct match points and their locations. Generally, four significant steps of recognition are: Determination of feature; Choice of feature detectors; Choice of feature descriptors; Choice of feature matchers to realize the recognition [5].

Determination of feature

In image processing, feature contains the information of an image such as edges, points, corners, interest regions and ridges [6]. Recognition of a target item is just a process to find and locate the features of it in an image. Features of images consist of

global features and local features. Global feature uses one multidimensional feature vector to describe the information of the whole image, such as color, texture and shape. etc. As for the local features, the detection process is to separate some interest regions and collect the information from them. An example of local features can be multiple interest points or some other features in a specific part of an image [7].

Feature detector

Feature detector is an algorithm used to find the features in an image. Some detectors can detect only one kind of features, while others can detect more. Different feature detectors are needed to be chosen according to the characteristic of the images and the target items. This section introduces common feature detectors for different features such as edges, corners and points.

Steps of edge detection are filtering, enhancement, detection and localization. For 1-D edge detection, Roberts operator, Sobel operator and Prewitt operator are commonly used [8]. The Roberts operator does an approximation to the gradient magnitude of the image and applies 2×2 convolution masks to calculate the entire image horizontally and vertically. Sobel operator and Prewitt operator are discrete differentiation operators which compute the approximate derivative by two 3×3 masks. 2-D edge detector includes Laplacian operator and Second directional derivative, which are for the cases where there are too many detected edge points. In this situation, choosing the points that have local maximums in gradient values can be a good approach by finding the points that have a peak in the first derivative and a zero

crossing in the second derivative [9]. Another famous edge detector is Canny edge detector, whose target is to realize a good-qualified performance by finding a single edge. John F. Canny used the calculus of variations and optimized the function. It summed the four exponential terms together and could also be expressed as first derivative of a Gaussian approximately [10].

For corners and interest points, there are many different methods like Shi and Tomasi, Hessian feature strength measures and Level curve curvature. Harris operator, a common corner detector, calculates the differential of the corner score which is proportional to the direction. Another operator called SUSAN tries to acquire the corner by comparing brightness difference in the defined circle which leads to the robustness to noise [11-14]. Features from Accelerated Segment Test (FAST), presented by Edward Rosten and Tom Drummond, is a corner feature detector [15]. FAST accomplishes detecting a corner by comparing the intensity around a possible point. The basic process is listed.

1. Choose a point P;
2. Set up a threshold;
3. Decide 16-pixel contiguous points set in the Bresenham;
4. Compare the pixels and regard P as a corner if several continuous points on the circle are brighter or darker than P.

Blob detection aims to get the feature information of a region or global features like color and texture. Determinant of Hessian (DoH), Difference of Gaussians (DoG), Laplacian of Gaussian (LoG), MSER, PCBR, Grey-level blobs are some common

methodologies [16, 17]. DoH, DoG and LoG achieve the detection by applying the convolution to the image, getting the derivatives of the Gaussian function and obtaining the jumping edges of the zero-crossing for the second derivatives.

There are also other famous detectors, such as Hough transform and affine invariant feature detection. Hough transform is commonly used to detect the shape of the target item [18, 19]. Affine shape adaptation, Harris affine and Hessian affine are used to accomplish the affine invariant feature detection [20, 21].

Feature descriptor

Feature descriptor is the mathematical description in different formats of the features. For global features, the descriptors for color include Dominant Color Descriptor (DCD), Scalable Color Descriptor (SCD), Color Structure Descriptor (CSD), Color Layout Descriptor (CLD) and Group of frame (GoF) or Group-of-pictures (GoP). Homogeneous Texture Descriptor (HTD), Texture Browsing Descriptor (TBD) and Edge Histogram Descriptor (EHD) are used to describe the textures. For another feature, shape, Region-based Shape Descriptor (RSD), Contour-based Shape Descriptor (CSD) and 3-D Shape Descriptor (3-D SD) are always used. Finally Motion Activity Descriptor (MAD), Camera Motion Descriptor (CMD), Motion Trajectory Descriptor (MTD) and Warping and Parametric Motion Descriptor (WMD and PMD) [22, 23] are available to decide the motion.

As we can see, there are a large number of global feature descriptors. However, in recognition application, local features are more commonly used than the global

features. In this case, the orientation and the scale for the features need to be considered for real practice. Therefore, a feature descriptor with good performance not only needs to align the information of location, scale, orientation and their image structure, but also needs to have good discriminate matching performance and sensitivity to local feature. With the effort of the scientists, a lot of local feature descriptors are proposed and improved like SIFT, SURF, GLOH and LBP [24]. Generally, SIFT makes a mathematical expression of the feature detected by Difference of Gaussians (DoG). As for SURF, which increases the speed of SIFT, uses the idea for Determinant of Hessian (DoH) to realize the description. Some commonly used feature descriptors are introduced here to provide an idea of their working processes.

Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) whose author is Lowe is the most famous feature descriptor. Since scale-invariant coordinates are relative to local features of the image, SIFT transforms the image data to mathematical expression through four steps: detecting scale-space extrema, localizing key points, assigning the orientation and describing the key points in a math expression [25].

To detect the scale-space extrema, Lowe used the LoG function to identify feature points. If the points were invariant to scale and orientation, they would be recorded as the interest points. After that, Lowe compared each point to its neighbors to decide and localize the useful points. Once a key point was found, Lowe tried to exclude the wrong points and increase the accuracy by applying Taylor expansion of

the scale-space function and comparing to the nearby data for location, scale, and ratio of principal curvatures. Then a consistent orientation would be assigned to the points according to the local feature information by using an orientation histogram method. With all the operations, the local feature is expressed in a math format which is easier for calculation.

Speeded-Up Robust Features Descriptor (SURF)

As introduced previously, SURF approximates the points detected by Hessian blob detector integrally. Then it extracts a square region of the points by adding the Haar wavelet response together. The center and orientation of the region is decided by the selected points. Finally, Gaussian is used to weigh the responses and lead to the robustness [26].

Gradient Location-Orientation Histogram (GLOH) & Local Binary Pattern (LBP)

Gradient Location and Orientation Histogram increases the performance of SIFT by using more spatial histogram regions for calculation [27]. As for Local binary patterns, a binary descriptor, performs particularly well to find a feature for texture classification. It can also be combined with the Histogram of oriented gradients (HOG) descriptor to give a better result [28, 29].

Oriented FAST and Rotated BRIEF (ORB)

ORB, a binary descriptor, makes an improvement on the hybrid method of Features from Accelerated Segment Test (FAST) and Binary Robust Independent Elementary Features (BRIEF). Compared to SIFT and SURF, ORB has a faster performance. FAST is a feature detector for corners and key points detection. Binary Robust Independent Elementary Features is a binary descriptor by comparing Hamming distances of the chosen point pairs [30, 31].

After finding the interest points by FAST, the image is first smoothed. Then a patch of points around the feature points are chosen and defined. The intensity of the two points in the patch will be compared to get 1 or 0. Then several pairs of points will be chosen from the image and form a binary string uniquely. Five ways are tested for BRIEF to decide the point pairs and the one with best performance is $X, Y \sim \text{Gaussian} \left(0, \frac{1}{25} S^2\right)$. Finally, hamming distances of all the features in target images and reference images are calculated. The two features with the shortest distance will be regarded as the same ones.

ORB combines the two methods to a new feature descriptor by making three improvements [32].

1. Adding a component of accurate orientation to FAST.
2. Calculating oriented BRIEF features in a more efficient way.
3. Improving oriented BRIEF features through variance and correlation.

Feature matcher

After extracting the features and applying the descriptors, a common approach to image matching is comparing the descriptors of the original images and the target images through the distance between two descriptors. In this part, how to reject the wrong points to ensure the accuracy and speeding up the algorithm become two decisive factors. For now, Brute-Force Matcher, the fast library for approximate nearest neighbors (FLANN) and the randomized k-d forest are the most common methods [33]. Brute-Force Matcher is just a basic algorithm to compute all the distances between different descriptors and acquire the matching result by comparing the distances.

GMS is a feature matching method with ORB applied as a feature descriptor. It is a statistical methodology presented by JiaWang Bian and Wen-Yan Lin which increases the accuracy of the match points by comparing the points around the features and rejecting the wrong feature points [34]. JiaWang Bian proposed an assumption that motion smoothness causes a neighborhood around a true match to view the same 3D location. Likewise, the neighborhood around a false match views geometrically different 3D locations [34]. As shown in Figure 2, the neighborhood of match x_i is defined as $\{a,b\}$ a pair of small support regions around the respective features. According to the assumption, true match neighborhoods in the yellow circle will have many more supporting matches than false match neighborhoods in the red circle. By calculating the supporting matches and comparing with the threshold, GMS is able to reject the wrong match points and increase the matching performance.

Figure 2 Neighborhood feature points for GMS matcher

2.2 Deep Learning Methods

Deep learning, which typically consists of supervised learning, unsupervised learning and reinforcement, is widely applied in image processing for computer vision. It was developed sharply after the presence of convolutional neural network because CNN accelerated the computing speed by replacing the fully connected layers with convolutional layers.

The overall training process of the Convolutional Neural Network is:

1. Choose proper initial values for all training parameters and weights according to the model evaluation.
2. Input training images and conduct forward propagation to obtain the output vector of probabilities for all classes.
3. Calculate the total error at the output layer.

4. Compute the gradients of the error according to the preset weights by back propagation.
5. Apply gradient descent and update all the parameters and weights to decrease the output error.

Considering the whole history of the development of convolutional neural network, the milestones are: LeNet - AlexNet – ZFNet – VGGNet – Inception (GoogLeNet) – ResNet – MaskRCNN.

LeNet

LeNet developed by Yann LeCun in 1990's was the first successful Convolutional Networks model [35]. This model consists of 7 layers as Figure 3 shows.

Figure 3 Architecture of LeNet (CNN) [35]

AlexNet

Alex Krizhevsky, Ilya Sutskever and Geoff Hinton presented AlexNet by establishing a deeper and more featured CNN architecture of which the layers stacked

on top of each other. AlexNet won ImageNet ILSVRC challenge in 2012 with extraordinary performance. After this, CNN became more popular in computer vision [36].

ZFNet

Matthew Zeiler and Rob Fergus, the winner for ILSVRC 2013, tweaked the hyperparameters of the architecture for AlexNet to make an improvement. They also expanded the middle convolutional layers and decreased the size of the stride and filter to get ZFNet [37].

VGGNet

As for ILSVRC 2014, Karen Simonyan and Andrew Zisserman presented VGGNet and found that one of the critical components for good performance of a CNN model was the depth of the network. However, VGGNet is a little bit more expensive to evaluate because of more memory and parameters (140M) in fully connected layer [38].

Inception (GoogLeNet)

This method was also presented in 2014 and won the ILSVRC. GoogLeNet is a convolutional network presented by Szegedy et al. in Google. It mainly contains 4 versions until now. In inception v1, inception architecture is presented which has deep

layers. In this version, one of the most important contribution is that the full connected layers are canceled which help cut the unnecessary parameters and increase the speed sharply [39]. In the second version, Batch Normalization (BN) was proposed which allowed the backward calculation [40]. As for inception v3 and v4, more contributions about the structure are made and increase the accuracy [41, 42]. Table 1 shows the accuracy of different deep learning models and the inception model has the highest accuracy.

Table 1 Accuracy of different deep learning models [41, 42]

Network	Top5-error	Top1-error
GoogLeNet	29%	9.20%
VGG	24.40%	6.80%
PReLU	21.59%	5.71%
BN-Inception	22%	5.82%
Inception-v3	18.77%	4.20%

ResNet

Kaiming He et al. won ILSVRC 2015 with Residual Network whose error is only 3.6%. For better accuracy, special skip connections and batch normalization are contained and appeared frequently in this model [43].

MaskRCNN

Mask R-CNN, an intuitive extension of Faster R-CNN, is a latest method to realize the recognition by constructing the mask branch properly to get good result. In this model, a simple quantization-free layer called RoIAlign was presented to fix the misalignment and preserves exact spatial locations [44].

From all the introductions of the development of the deep learning history for computer vision, it is found that the structure of the models become deeper and more complex by replacing the full connected layers by other functional layers. Parameters are also optimized to make the speed as fast as possible. Considering the difficulties of our problem and the characteristics of our robotic system, GoogLeNet inception v3 can be good enough to use.

CHAPTER3 Problem Statement

3.1 Problem Statement

To promote the application of picking robots for e-commerce, Amazon organized Amazon Robotic Challenge (ARC) which aimed to set up a connection between industrial and academic communities through robots. In 2017, the problem is unstructured automation such as commercially automated picking in a cluttered bin. In this thesis, the problem is defined as the recognition method for Amazon Robotic Challenge (ARC) 2017 based on the robotic system of Team Nanyang [45].

The challenge consists of three tasks:

1. A pick task to remove target items from storage system and place them into boxes.
2. A stow task to take target items from totes and place them into storage system.
3. A Final Round task where all items are first stowed and then selected items are picked into boxes.

According to the rules, recognition is needed in the picking process. Based on our system, the storage system is the initial location for all the items which is shown in Figure 4. The task is to use the cameras over the storage system to acquire RGB images and find the target item correctly by giving the right suction point. Then the point cloud is calculated by the stereo cameras to get the depth information for the gripper. Figure 5

minutes before the run to enhance the difficulty. In this thesis, only the 40 known items would be considered as the target items for recognition and localization.

Figure 6 Amazon Robotic Challenge dataset in 2017 [45]

3.2 Introduction of ARC

Amazon organizes the Amazon Robotic Challenge (ARC) in order to research the picking robots applied in warehouses and promote the automation of the e-commerce systems. Amazon made a great success to achieve highly automated e-commerce logistics system in its warehouses. However, the automation of order picking especially in cluttered workspace is still challenging to be accomplished which is also the target for the competition in 2017. In this section, the 3-year competition is introduced to provide a frame of the logistics picking problem.

Figure 7 Methodologies summary for ARC 2015

During the first Amazon Picking Challenge, held in Seattle Washington, the winners are RBO from the Technical University of Berlin. Their robot consisted of a Barrett arm, a Nomadic Technologies mobile platform and a commercial vacuum

cleaner with a suction cup. Their strategy for the vision system is to give the position of the items through multiple features and 3D bounding box and to find the suction point by 2D feature matching. The methodologies of the top-3 teams are summarized in Figure 7 [46, 47]. It shows that most of the team applied traditional feature matching methods as the perception strategies in the picking robots in 2015.

The second challenge was held at RoboCup 2016 in Leipzig, Germany. All the teams were required to move 12 specific items from an Amazon Robotic shelf into a tote, and put the 12 items into a partially full shelf. The winner team Delft's system contained the robot arm, gripper, tote camera (Ensenso N35), rail, shelf and tote as shown in Figure 8 [48]. In their vision system, the whole process could be divided into object recognition and pose estimation. For object detection, a pre-trained deep neural network based on Faster R-CNN and bounding boxes were applied. Then Super 4PCS were used to accomplish pose estimation by matching the items with their CAD models.

Figure 8 Robotic system of Delft in ARC 2016 [48]

For ARC 2017 in Japan, which is introduced previously and concentrates on picking in cluttered workspace, the difficulties increase a lot due to the number of the items in storage system, the items which are hard to detect (e.g. transparent ones), the limited training time for the unknown items, the different and arbitrary light conditions and so on. Therefore, this thesis will deal with the method for the recognition in cluttered environment within limited time, which can be a problem in e-commerce systems.

3.3 Decision of Methodology

After analyzing the problem and our system, the strategy is decided to use RGB images to proceed the classification and get the 2D location because of the poor quality of the 3D point cloud models of target items caused by noise. While dealing with the 2D images, the key point is to pick correctly according to the order list for e-commerce. Therefore, accuracy becomes the most significant factor.

According to the analysis of the dataset about the different structures, shapes and characteristics, the challenges for recognition are summarized:

1. Some featureless items are really difficult to be recognized by traditional methods.
2. The messy environment increases the probability of mismatching and wrong recognition.

3. Due to the different sizes of the items and the crowded storage system, overlaps occur in the unstructured environment. Some small items may even be covered by the big items.
4. All the items have different structures, shapes and characteristics. So the suction points for some items are needed to be given properly.
5. Among all the items, some of them can be sharply influenced by environmental conditions like light which may cause big changes and mistakes in the image.

Considering all these problems, we decided to choose proper traditional feature matching methods and deep learning methods to conduct the experiments. For a simple procedure and short preparation time, ORB + GMS is chosen as the traditional feature matching method because of its high matching quality and accuracy. Then deep learning methods are explored to recognize more kinds of the items with different features. GoogLeNet inception v3 method is used because of its higher accuracy compared to other methods. During the experiments, we tune the parameters of the two methods to get the best performance. After thinking about the advantages and disadvantages of traditional methods and deep learning methods, we decide to combine the deep learning method with some other traditional operations including GMS and image differencing to acquire more accurate suction points for the system. As a summary, 5 methods are proposed and tested to solve the problem of bin picking in cluttered workspace. The principle of the five methods as well as the mathematical theory are introduced and explained in chapter 4.

CHAPTER4 Recognition Approaches

In this thesis, we attempt to develop a robust object recognition technique for item picking in a cluttered bin. The developing process has been divided into five stages, where we proposed five methods and tested them progressively in order to achieve a better accuracy. Figure 9 illustrates a overview of the developing process, where we implement and carry out the tests to analyze and improve the perception till we achieve high accuracy for object recognition.

Figure 9 Overview flowchart of the exploration process

This developing process includes five methods. The theory and process of the five approaches will be explained in this chapter.

1. Grid-based Motion Statistics (GMS) + fixed sliding window,
2. Convolutional neural network (CNN) + fixed sliding window,
3. Convolutional neural network (CNN) + dynamic sliding window,
4. CNN + dynamic sliding window + GMS,
5. CNN + dynamic sliding window + image differencing

4.1 Introduction of GMS and CNN

Since our perception strategy uses two vision methods, GMS (Grid-based Motion Statistics) and CNN (Convolutional neural network) will be introduced first for further usage.

GMS

GMS is a statistical feature matching presented by JiaWang Bian and Wen-Yan Lin [34]. They increased the matching accuracy by rejecting the wrong feature points. The theory for GMS method is:

1. Acquire the original image $\mathbf{O}$ and the target image $\mathbf{T}$ and divide them into two sets of $m \times n$ non-overlapping cells respectively.

$$\mathbf{O}_{ij} \in \mathbf{O}; \mathbf{T}_{ij} \in \mathbf{T} \quad (1)$$

2. Use ORB feature descriptor to find the match points mp_{ij} in every set of $\mathbf{O}_{ij}$ and $\mathbf{T}_{ij}$.
3. Build a function φ_{ij} to represent the number of the match points mp_{ij} .
4. Calculate the score of the cell pair with considering the neighbor cells as σ_{ij} .
Cells on the margin are not considered.

$$\sigma_{ij} = \sum_{i=1}^{i+1} \sum_{j=1}^{j+1} \varphi_{kl} \quad (2)$$

$$1 < i < m - 1, 1 < j < n - 1$$

5. Set the threshold τ and compare the scores for all cell pairs with it. Abandon the match points mp_{ij} in the cell if $\sigma_{ij} < \tau$.

As so far, the correct match points between the original image $\mathbf{O}$ and the target image $\mathbf{T}$ are obtained.

CNN

Considering the requirement for accuracy and speed, GoogLeNet Inceptionv3 is used in our work [42]. Figure 10 shows the full architecture of the deep learning model. This model is made up several kinds of training layers. Figure 11 explains the function of the layers.

Figure 10 Architecture for GoogLeNet Inceptionv3 [42]

Figure 11 Introduction of the layers in GoogLeNet Inceptionv3

In this deep learning model, fully convolutional inception networks are changed to other ways of factorizing convolutions in order to increase the computational efficiency and speed. Inception v3 makes factorization into smaller convolutions and

also optimizes the inception modules which have 3 different sizes: 35×35 , 17×17 and 8×8 . In a word, all of these structures lead to achieving higher accuracy and efficiency.

Convolution layers are the most important parts of CNN, whose primary purpose is to extract and summarize the features of input dataset. Basic mathematical expression for the convolution layer is shown as follows.

$$S_{ij} = X_{ij} * W_{ij} = \sum_m \sum_n X_{i+m, j+n} W_{m, n} \quad (3)$$

The neuron structure for the convolution layer is shown in Figure 12.

Figure 12 Neuron structure of the convolution layer [49]

In order to minimize the preparation time, we decide to use the existing model and apply transfer learning [50] to train the last fully connected layer. After getting the result of a label list and a trained model, classification process is realized by passing an image into the trained model to obtain a score vector of the items in the model. If the score for one item in the label list is pretty high, there is a high possibility that the item is in the image.

4.2 GMS + Fixed Sliding Window

While designing our perception strategy, traditional feature methods are first taken into consideration because they don't spend too much time on preparation activities like a training process. After the literature review, Grid-based Motion Statistics (GMS) based on Oriented FAST and Rotated BRIEF (ORB) is decided because its large number of accurate match points.

According the GMS method introduced previously, we can get a set of match points after applying it to an original image and a target image. Considering our challenging workspace, we decided to combine GMS with sliding window because of three reasons:

1. The items are difficult to match due to the difference of sizes and features;
2. Matching results between the image of the whole storage system and the reference image may not be good since the environment is too cluttered to find correct match points;
3. The cluttered environment is difficult to achieve segmentation and the segmentation results may not be able to contain complete information for target items.

A sliding window of a quarter of the image size is chosen because it tends to include the complete information of part of the big items or the whole entities of the small items. This method excludes the useless features caused by the cluttered environment. Figure 13 shows the process of GMS + sliding window method. The steps and principles for GMS + fixed sliding window are:

1. Capture the target image in the scene of the storage system $\mathbf{I} \in \mathbf{N}^{m \times n \times 3}$;
2. Acquire 9 images from $\mathbf{I}$ by chopping it with a sliding window of $m/2 \times n/2$ and step size of $m/4 \times n/4$.

$$\mathbf{I}_\alpha \in \mathbf{I}, \alpha = \{1, 2, \dots, 9\} \quad (4)$$

The center location of the 9 image slices is:

$$\left(\frac{km}{4}, \frac{ln}{2}\right), \text{where } 1 \leq k, l \leq 3, k, l \in \mathbf{N} \quad (5)$$

3. Calculate the number of the match points N_α between $\mathbf{I}_\alpha$ and the reference images from the database;
4. Locate the target item in $\mathbf{I}_\alpha$ by finding the maximum of N_α ;

$$\hat{\alpha} = \max_{\alpha} N_{\alpha} \quad (6)$$

5. Set the threshold N_{thresh} . If $N_{\hat{\alpha}} > N_{thresh}$, the item is considered in $\mathbf{I}_{\hat{\alpha}}$ and the location is calculated as the center of all the match points in $\mathbf{I}_{\hat{\alpha}}$.

Figure 13 Recognition process of GMS + fixed sliding window

4.3 CNN + Fixed Sliding Window

Considering the different types and characters, some items like the transparent plastic wine glass and some items with uniform colour may meet some difficulty in recognition by feature matching. Convolutional Neural Networks (CNN), made up of neurons with weights and biases, is then applied in the image recognition because of its high accuracy and generalization.

We choose GoogLeNet inception v3 (convolution neural network) and use transfer learning to train the final layer and shorten the training time. For CNN method, the target image will be compared with the pre-trained model and a score of probability will be provided for all the labels. The higher the score is, the more possibilities that the image has the same item as the label. With the inspiration of GMS + fixed sliding window, we also decide to apply the sliding window method to CNN to deal with the cluttered environment as shown in Figure 14. Every window image will be passed into the pre-trained CNN model. After the calculation, a vector of the 40 scores for all training items in the window image is produced to show the possibility that every item is in the image or not.

Figure 14 Recognition process of CNN + fixed sliding window

While applying the CNN method, the most significant thing is to get the score matrix S based on the sliding window. The steps for the process of CNN + fixed sliding window are listed here.

1. Use a sliding window of $a \times a$ pixels and step size of b pixels to slice the images into q squared windows. I is a set of all the sliced window images;

$$I_{\beta} \in I, \beta = \{1, 2, \dots, q\} \quad (7)$$

2. Pass I_{β} into the trained model to get a score vector for m items, p is the item sequence number in the name label list of the items.

$$S_{\beta} = [S_{1\beta} \quad \dots \quad S_{p\beta}]^T \quad (8)$$

3. After acquiring and saving the score vectors for all the image slices, a score matrix S is obtained to give out the probability for every item present in an image slice.

$$\mathbf{S} = \begin{bmatrix} S_1 \\ \vdots \\ S_\beta \end{bmatrix} \begin{bmatrix} S_{11} & \cdots & S_{1q} \\ \vdots & \ddots & \vdots \\ S_{p1} & \cdots & S_{pq} \end{bmatrix} \quad (9)$$

4. After getting the score matrix of all the items for all the windows, the scores can be compared and operated to localize the target item.

Considering the θ^{th} ($0 < \theta \leq m$) item in the label list, the score vector of it is:

$$\mathbf{S}_\theta = [S_{\theta 1} \quad \cdots \quad S_{\theta q}] \quad (10)$$

5. Find the highest score S_{max} in the vector which demonstrate that the corresponding window has the highest probability to contain the target item.

$$S_{max} = \|\mathbf{S}_\theta\|_\infty \quad (11)$$

6. If $S_{\theta k} = S_{max}$, then the center of the k^{th} window will be regarded as the item location.

$$x = qk(\text{mod } \left\lfloor \frac{b-p}{q+1} \right\rfloor) + \frac{p}{2} \quad (12)$$

$$y = q \left\lfloor \frac{k}{\left\lfloor \frac{b-p}{q+1} \right\rfloor} \right\rfloor + \frac{p}{2} \quad (13)$$

(x, y) is the coordinate of the center of the window with the highest score,

p is the window size, q is the step size, b is the width of the whole image.

$\lfloor \cdot \rfloor$ refers to the max integer less than $\cdot$.

To improve the performance of the CNN + fixed sliding window, many tests are done to tune the parameters of the window size and step size. The results are shown in chapter 5.

4.4 CNN + Dynamic Sliding Window

After some tests with the previous methods, a problem is found that some small items like flashlight and scissors are difficult to be discovered. The possible reason for this problem is because the size of the sliding window for small items is too big. The background occupies more area of the window which causes a lower score for the window and difficulties for recognition. Therefore a dynamic sliding window depends on item sizes is applied for the recognition. The operation steps for this methods is similar except that every item is assigned a specific window size according to its dimension. As shown in Table 2, for the big items like avery binder whose pixel size is larger than 160 pixels, we all choose 80×80 square as the window size. As for the small items whose pixel size is smaller than 160 pixels, the window size is set as about half of the item's length to contain enough information.

Table 2 Dynamic sliding window sizes for different items

Item name	Window size	Item name	Window size
Avery binder	100×100	Laugh out loud jokes	80×80
Balloons	60×60	Marbles	60×60
Band aid tape	60×60	Measuring spoons	60×60
Bath sponge	60×60	Mesh cup	80×80
Black fashion gloves	60×60	Mouse traps	80×80
Burts bees baby wipes	80×80	Pie plates	100×100
Colgate toothbrush 4pk	80×80	Plastic wine glass	60×60
Composition book	80×80	Poland spring water	80×80

Crayons	60×60	Reynolds wraps	80×80
Duct tape	60×60	Robots DVD	80×80
Epsom salts	80×80	Robots everywhere	100×100
Expo eraser	60×60	Scotch sponges	80×80
Fiskars scissors	60×60	Speed stick	60×60
Flashlight	50×50	Table cloth	100×100
Glue sticks	80×80	Tennis ball container	80×80
Hand weight	60×60	Ticonderoga pencils	80×80
Hanes socks	100×100	Tissue box	60×60
Hinged ruled index cards	60×60	Toilet brush	80×80
Ice cube tray	100×100	White facecloth	80×80
Irish spring soap	60×60	Windex	80×80

Since the high accuracy and the generalization for all kinds of items, CNN + dynamic sliding window is decided to be definitely applied during the recognition and classification. For CNN + dynamic sliding window, the actual result is a window which has the highest probability of containing the target item or part of it. This method may lead to a problem of wrong picking if the center of the window is not located on the target item accurately. In this case, the target becomes to find the actual item part in the sliding window.

4.5 CNN + Dynamic Sliding Window + GMS

In order to find the actual item part in the sliding window as an accurate suction point, GMS is decided to be applied to the extract the information in the window and find the location of the item. It can locate the item more accurately by extracting and matching the features. The whole process is illustrated in Figure 15. The center of the sliding with the highest score is not located on the item accurately. Therefore, GMS is applied to the window image to find the item location.

Figure 15 Recognition process of CNN + dynamic sliding window + GMS

When the window which contains the part of the target item is extracted, the image is first compared with the database including the whole item. However the result is not so good and the match points may not be found or be found wrongly. The analysis is because that the extracted window is usually incomplete and only contains part of the features. As shown in the left image in Figure 16, the half part of the hand weight is not able to match with the whole hand weight since the information is not equal. After discovering this, the sliding window itself is duplicated and compared to itself to find and match the features in the window. In this case, the match points for the item part in the image are able to be found and the suction point located on the item can be decided by find the center of the match points.

Figure 16 GMS matching results of the sliding window image

4.6 CNN + Dynamic Sliding Window + Image Differencing

Based on the fixed position of the whole system, it is possible to find the location of the target item in the extracted sliding window by removing the background. Since the background is black, removing the background by the color condition may be a good method. But after analyzing the whole situation, this method is not proper because there are black items like black fashion gloves and mesh cup. In this case, another method is added to fix this problem by comparing the image of the extracted window with the original background of the shelf.

The basic idea of image differencing will be introduced here. First, an image of original background and an image of an item in the background are prepared. $I_{original}$, $I_{background}$, whose sizes are both $(m \times n)$ pixels. Then the two images will be compared pixel by pixel and the pixels will be labeled as a white point if the color information of the two pixels in two images is different and the difference is large than the threshold.

$$|I_{original}(x,y) - I_{background}(x,y)| > I_{threshold} \quad (14)$$

$I_{original}(x,y)$, $I_{background}(x,y)$ and $I_{threshold}$ present the intensity for the specific pixel.

If the below condition is reached, the pixel will be labeled as a white region. Then the location of the item will be given by finding the center of the largest white region by applying the boundary box.

Considering the whole system and the classification, the whole process is shown in Figure 17. The problem is the center of the window with the highest score is

not located on the item accurately. Therefore, after extracting the sliding window which contains the target item, the location of the window will be recorded and extract the same window in the prepared background image. Then image differencing method is applied to compare the extracted window with its background. Finally, the background in the extracted window will be detected and removed to decide the location of the item. This method improves the cases in which the GMS cannot work and become suitable for all kinds of items.

Figure 17 Recognition process of CNN + dynamic sliding window + image differencing

CHAPTER5 Experiment and Results

After deciding the methods, experiments are designed and conducted which include acquiring the image data and using the perception methods to achieve the recognition. In this chapter, the setup of the database and training results are first presented. Then the whole experiment process and recognition results of the five methodologies are researched and discussed.

5.1 Experimental Setup

According to the structure of the robotic system and the strategy for the picking process, experimental setup has several important components.

1. Zed Camera
2. Scanner system
3. Storage system (shelf)
4. Zotac ZOBX (CPU: intel core i7-6700, GPU: NVIDIA GeForce GTX 1080 8 GB) with Ubuntu 16.04 and CUDA 8 installed

While doing the experiment, the zed cameras over the storage system are first used to take images for the shelf box with items placing randomly. Then the image will be processed and calculated by GPU. The output is expected to be the pixel coordinate location of the target item in the image which is necessary for the next step to get the depth information for the suction.

5.2 Data Acquirement and Training

To accomplish the recognition, a standard database is needed to be used as a reference to compare with the experiment image. After reviewing the articles of object recognition, a result is found that setting up the database and getting the experiment images by the same cameras usually lead to a better result because the same resolution and the same responses to the light conditions. Therefore, based on our strategy, we decide to establish our own database by zed cameras instead of using the data from amazon.

Database setup

Since the experiment images are all from the top view of the shelf and GMS has the capability to deal with the slightly different view with same features, the database for GMS method will be the images for the items taken by the zed camera from the same height as the ones on the shelves. The database includes the views of different sides of the item especially the sides with clear features in a black background. Figure 18 gives an example of the database of the hanes socks. The database is set like this because GMS is based on features of the item to realize the recognition. What's more, the purpose of the black background is to prevent the reflection as much as possible.

Figure 18 Part of the database for GMS

CNN requires sufficient image data to train the model. Considering the possibility that the postures of the items may be different, images from different views and rotations will all be taken in order to get as many cases for training. In order to achieve this, a scanner which includes one rotation platform and two cameras are manufactured. The platform is rotated by a motor with adjustable speed. The items will be put on the center of the scanner with black background and images for the item will be taken and saved during the rotation. In this process, different sides of the item also need to be captured to give more possible views. To use the image capture system, the pixel coordinates of the rotation center in the captured image will be measured and set up first. Then after taking the images, we'll use a square based on the size of the items to cut the images and remain the pure item in the images for training. Figure 19 gives several example images of hanes socks captured by the scanning system.

Figure 19 Part of the database for CNN

Training process

After acquiring all the images for CNN, training the model is a significant step for the whole experiment. As we all know, a disadvantage of the deep learning method is the long training time for deep layers. Therefore, instead of training all the layers, transfer learning method is used which only the top layer will be retrained and fewer images will be used. Due to the similarity of the low-layer structure, the layers are reused without any change to support further calculation. All the layers except for the old top layer will be kept in the process of transfer learning [50]. During the training, the old top layer will be removed and a new one based on the photos acquired previous will be trained and added. By doing this, training time will be decreased sharply. This transfer learning method is easy to use and also has high efficiency which is possible to be used in the real e-commerce system.

Once the training starts, all the images will be analyzed and calculated to get the bottleneck values for them. Bottlenecks are the informal terms used before the final layer. The pre-trained penultimate layer provided by ImageNet includes 1,000 classes. It contains enough information and proper parameters which make it ready for the accomplishment of classifier by adding a very patch of training images.

After obtaining the bottlenecks, forward propagation is conducted and 4,000 training steps by default are conducted to train the top layer with the output of training accuracy, validation accuracy and the cross entropy. Ten images are picked as a small patch randomly for every step of the training process to find the bottlenecks information and predict the output of the final layer. As soon as forward propagation is finished, the back propagation process is done by comparing the output results with the actual labels to update the weights. Finally, the model will be automated tested with the test dataset to judge if the full-trained model can works well or not.

There are three main factors in the training process: training accuracy, validation accuracy and cross entropy. The training accuracy shows the percentage of the images that are labeled correctly. The validation accuracy is used to measure and represent the random level of the training dataset. Cross entropy is a loss function to check if the training works well. To acquire good training performance, cross entropy needs to be controlled as small as possible.

Considering the cluttered environment whereby only part of the items are visible, sliding window method is selected to separate the items and keep complete information as much as possible. The objective for the recognition and training process is to increase the accuracy of recognition while comparing the sliding window images

with the model. When the model was first trained with images with a complete item in black background, the sliding window images with only part of the items may not get a high score because of the incomplete features. In order to improve this, a program is written to chop training images in to pieces by sliding window. More possible cases are considered and trained by doing this, and it is more possible to realize the recognition in a cluttered storage system even based on part of the items. A sliding window of (80×80) pixels and a 30-pixel step size are applied to chop the image and set up the training dataset. The total number of the training dataset is 208520. Table 3 gives a summary of the number of the training dataset for all the items. Figure 20 gives an example as part of the images in the training dataset.

Table 3 Image number for the training process

Item name	Original image number	Chopped image number
Avery binder	300	11100
Balloons	600	6000
Band aid tape	360	1800
Bath sponge	420	4200
Black fashion gloves	360	3600
Burts bees baby wipes	340	5780
Colgate toothbrush 4pk	360	6120
Composition book	240	6240
Crayons	420	2100
Duct tape	360	3600
Epsom salts	360	6120
Expo eraser	540	5400
Fiskars scissors	480	960
Flashlight	300	600
Glue sticks	300	5100
Hand weight	600	6000
Hanes socks	360	9360
Hinged ruled index cards	480	3600
Ice cube tray	300	7800
Irish spring soap	600	3000
Laugh out loud jokes	240	2400
Marbles	360	1800

Measuring spoons	360	3600
Mesh cup	360	3600
Mouse traps	360	3600
Pie plates	240	6240
Plastic wine glass	240	1200
Poland spring water	420	4200
Reynolds wrap	360	9360
Robots DVD	240	2400
Robots everywhere	240	4080
Scotch sponges	540	5400
Speed stick	360	3600
Table cloth	660	13620
Tennis ball container	360	9360
Ticonderoga pencils	300	5100
Tissue box	480	8160
Toilet brush	360	9360
White facecloth	360	3600
Windex	360	9360

Figure 20 Original dataset example used to train the model

Training results

Training the 200,000 image dataset roughly takes 75 minutes to train with the setup. The final test accuracy based on the training dataset is 92.1%. And the training results are shown here. Figure 21 shows the training accuracy, Figure 22 shows the

training cross entropy and Figure 23 shows some other training parameters. As the training continues, the identification accuracy increases but the cross entropy decreases.

Figure 21 Accuracy result of the CNN training process

Figure 22 Cross entropy result of the CNN training process

Figure 23 Other results for the CNN training process

5.3 Experimental Results

Accuracy calculation

To proceed the experiments, we use cameras to take the images of the storage system and make sure that every items show up 50 times. Then the images are tested with the 5 methods to check and record the times for the appearance of every item. If the successful identification times is T , we can calculate the accuracy of the item by:

$$R = T/50 \quad (15)$$

This section gives the related results for the 5 methods especially the accuracy results.

GMS + fixed sliding window

We first applied GMS to identify the item in the whole image. As shown in Figure 24 and Figure 25, the matching result of SIFT is more messy than the one of GMS since GMS excludes the wrong match points. However, the target item is still difficult to be found because of the similar features in the cluttered bin.

Figure 24 Recognition and matching result by SIFT

Figure 25 Recognition and matching result by GMS

In order to reduce the influence of the cluttered environment, sliding window is used to chop the image to small pieces. By doing this, the number of the items in the sliding window is limited which will decrease the chance of mismatching. The recognition accuracy of the 40 items are shown in Table 4 and Figure 26.

Table 4 Recognition accuracy of GMS + fixed sliding window

Item name	Identify accuracy	Item name	Identify accuracy
Measuring spoons	0.06	Crayons	0.76
Table cloth	0.38	Tissue box	0.36
Epsom salts	0.84	Laugh out loud jokes	0.7
Burts bees baby wipes	0.44	Ticonderoga pencils	0.94
Plastic wine glass	0.12	Composition book	0.92
Speed stick	0.64	Mesh cup	0.06
Expo eraser	0.52	Band aid tape	0.66
Hinged ruled index cards	0.52	Hand weight	0.32

Glue sticks	0.48	Avery binder	0.24
Poland spring water	0.42	Bath sponge	0.16
Ice cube tray	0.04	Marbles	0.28
flashlight	0.02	Robots DVD	0.88
Black fashion gloves	0.36	Fiskars scissors	0.34
Colgate toothbrush 4pk	0.82	Reynolds wrap	0.74
Duct tape	0.14	White facecloth	0.16
Tennis ball container	0.72	Toilet brush	0.36
Windex	0.82	Robots everywhere	0.98
Balloons	0.32	Hanes socks	0.76
Irish spring soap	0.74	Mouse traps	0.66
Scotch sponges	0.32	Pie plates	0.7

Figure 26 Recognition accuracy of GMS + fixed sliding window

CNN + fixed sliding window

During the test, it is found that some small items are more difficult to be found because in a sliding window with small items in it, the area of the background is more likely to be larger than the one with big items which will lead to a lower score for the window. Another case is that it may be more difficult to see if the sliding window cut the item into pieces and doesn't step on the one containing the majority. Therefore, two parameters can be adjusted to increase the success rate of the recognition: sliding window size and step size.

1. Sliding window size is the size of the sliding window which the best choice should make the most area of the sliding window contains only one or part of the item in order get a higher score and more accurate results.
2. Step size is the main effector for the time because it will decide the total number of the sliding window needed to be calculated.

In order to tune the size, test of one image for different sliding window size is summarized. Table 5 shows that the result is the best when the sliding window size is (80×80) pixels and the step size is 30 pixels.

Table 5 Tuning the parameters of window size and step size for CNN + fixed sliding window

Sliding window	Step size	Item number	Recognition number
40×40	20	17	9
60×60	20	17	10
80×80	20	17	9
100×100	20	17	8
120×120	20	17	8
60×60	30	17	10

80×80	30	17	11
60×60	40	17	8
80×80	40	17	9

Based on the previous result with optimal sliding window size of (80×80) pixels and step size of 30 pixels, the result for CNN + fixed sliding window is in Table 6 and Figure 27.

Table 6 Recognition accuracy of CNN + fixed sliding window

Item name	Identify accuracy	Item name	Identify accuracy
Measuring spoons	0.64	Crayons	0.6
Table cloth	0.96	Tissue box	0.88
Epsom salts	0.96	Laugh out loud jokes	0.94
Burts bees baby wipes	0.96	Ticonderoga pencils	0.84
Plastic wine glass	0.68	Composition book	0.96
Speed stick	0.7	Mesh cup	0.82
Expo eraser	0.66	Band aid tape	0.8
Hinged ruled index cards	0.86	Hand weight	0.62
Glue sticks	0.86	Avery binder	0.96
Poland spring water	0.78	Bath sponge	1
Ice cube tray	0.96	Marbles	0.8
Flashlight	0.28	Robots DVD	0.94
Black fashion gloves	0.92	Fiskars scissors	0.5
Colgate toothbrush 4pk	0.9	Reynolds wrap	0.74
Duct tape	0.5	White facecloth	0.86
Tennis ball container	0.74	Toilet brush	0.38
Windex	0.9	Robots everywhere	0.98
Balloons	0.9	Hanes socks	0.7
Irish spring soap	0.6	Mouse traps	0.96
Scotch sponges	0.8	Pie plates	0.98

Figure 27 Recognition accuracy of CNN + fixed sliding window

CNN + dynamic sliding window

CNN + dynamic sliding window is a recognition method in which the sliding window sizes are related to the items sizes. It is proved to be suitable for the items with all sizes after the experiments with the same test images. Table 7 and Figure 28 show the results for CNN + dynamic sliding window. The total average recognition accuracy is 0.8795.

Table 7 Recognition accuracy of CNN + dynamic sliding window

Item name	Identify accuracy	Item name	Identify accuracy
Measuring spoons	0.82	Crayons	0.86
Table cloth	0.96	Tissue box	0.88
Epsom salts	0.96	Laugh out loud jokes	0.94
Burts bees baby wipes	0.96	Ticonderoga pencils	0.9

Plastic wine glass	0.84	Composition book	0.98
Speed stick	0.94	Mesh cup	0.86
Expo eraser	0.58	Band aid tape	0.78
Hinged ruled index cards	0.94	Hand weight	0.9
Glue sticks	0.86	Avery binder	0.82
Poland spring water	0.86	Bath sponge	1
Ice cube tray	0.98	Marbles	0.94
Flashlight	0.74	Robots DVD	0.96
Black fashion gloves	0.88	Fiskars scissors	0.76
Colgate toothbrush 4pk	0.92	Reynolds wrap	0.86
Duct tape	0.88	White facecloth	0.96
Tennis ball container	0.84	Toilet brush	0.56
Windex	0.94	Robots everywhere	0.98
Balloons	0.96	Hanes socks	0.8
Irish spring soap	0.74	Mouse traps	0.98
Scotch sponges	0.92	Pie plates	0.94

Figure 28 Recognition accuracy of CNN + dynamic sliding window

CNN + dynamic sliding window + GMS

Table 8 and Figure 29 show the result of the CNN + dynamic sliding window + GMS method which finds the item location by applying the CNN to find an approximate location and using GMS to give out a more accurate location. This hybrid method improves the average accuracy from 0.8795 to 0.938.

Table 8 Recognition accuracy of CNN + dynamic sliding window + GMS

Item name	Identify accuracy	Item name	Identify accuracy
Measuring spoons	0.94	Crayons	0.96
Table cloth	0.98	Tissue box	0.94
Epsom salts	0.98	Laugh out loud jokes	0.98
Burts bees baby wipes	0.98	Ticonderoga pencils	0.9
Plastic wine glass	0.9	Composition book	1
Speed stick	1	Mesh cup	0.96
Expo eraser	0.76	Band aid tape	0.9
Hinged ruled index cards	0.98	Hand weight	0.98
Glue sticks	0.9	Avery binder	0.9
Poland spring water	0.94	Bath sponge	1
Ice cube tray	0.98	Marbles	0.94
Flashlight	0.92	Robots DVD	0.96
Black fashion gloves	0.92	Fiskars scissors	0.88
Colgate toothbrush 4pk	0.96	Reynolds wrap	0.92
Duct tape	0.92	White facecloth	0.98
Tennis ball container	0.88	Toilet brush	0.74
Windex	0.96	Robots everywhere	1
Balloons	0.96	Hanes socks	0.86
Irish spring soap	0.9	Mouse traps	1
Scotch sponges	0.96	Pie plates	1

Figure 29 Recognition accuracy of CNN + dynamic sliding window + GMS

CNN + dynamic sliding window + image differencing

Table 9 and Figure 30 show the result for the CNN + dynamic sliding window + image differencing method. The average successful rate is 0.9525 which is the highest one among all the recognition methodologies.

Table 9 Recognition accuracy of CNN + dynamic sliding window + image differencing

Item name	Identify accuracy	Item name	Identify accuracy
Measuring spoons	0.92	Crayons	0.96
Table cloth	0.98	Tissue box	0.98
Epsom salts	0.98	Laugh out loud	0.98

		jokes	
Burts bees baby wipes	0.98	Ticonderoga pencils	0.9
Plastic wine glass	0.9	Composition book	1
Speed stick	1	Mesh cup	0.96
Expo eraser	0.82	Band aid tape	0.9
Hinged ruled index cards	0.98	Hand weight	0.98
Glue sticks	0.9	Avery binder	0.94
Poland spring water	0.92	Bath sponge	1
Ice cube tray	1	Marbles	1
Flashlight	0.94	Robots DVD	0.96
Black fashion gloves	0.92	Fiskars scissors	0.94
Colgate toothbrush 4pk	0.96	Reynolds wrap	0.98
Duct tape	0.96	White facecloth	0.98
Tennis ball container	0.92	Toilet brush	0.82
Windex	0.96	Robots everywhere	1
Balloons	1	Hanes socks	0.86
Irish spring soap	0.94	Mouse traps	1
Scotch sponges	0.98	Pie plates	1

Figure 30 Recognition accuracy of CNN + dynamic sliding window + image differencing

5.4 Discussion

In this section, the improvement process and the comparison results of all the methods will be discussed. The improvement process will show the developing idea of the 5 methods progressively. The comparison result gives the analysis about the advantages and disadvantages of the 5 recognition methodologies.

Improvement process

The five methodologies are developed in a logical sequence to improve the accuracy of the recognition performance.

A traditional feature matching method is first applied rather than a deep learning method because of its convenience to set up reference database and limited preparation time without a training process. GMS is chosen to be used with sliding window to realize the recognition in cluttered workspace. However, the recognition accuracy of some specific items are pretty low such as measuring spoons, plastic wine glass, ice cube tray, flashlight, mesh cup, bath sponge, marbles, face cloth .etc. After analyzing the result, the most possible reason for the result is that the featureless items are difficult to be identified by feature matching. The GMS matching results of the tissue box shown in Figure 31 and Figure 32 explain the reason for the low accuracy. The up-side face of the tissue box leads to a better matching result than the downside face because the upside face has more obvious and distinctive features. Since the GMS is a traditional vision method to identify the target by detecting and matching the features of the object, the item will not be easy to be detected by GMS if the items is uniform or transparent.

Figure 31 GMS + fixed sliding window result for the item with feature

Figure 32 GMS + fixed sliding window result for the item without feature

In order to realize the generalization of the recognition method, CNN is applied to replace the traditional feature matching method and sliding window is kept to overcome the cluttered environment. As shown in the experiment results, CNN + fixed sliding window is used and tested which leads to a better recognition performance for all kinds of items.

After analyzing the results for the CNN + fixed sliding window, some small-size items cannot be recognized because the fixed window size seems not suitable for the small sizes. CNN + dynamic sliding window is then applied to make a changeable window according to the sizes of the items. By doing this, a proper window can contain proper information for easier recognition. The accuracy of small-size items increases to

an average level. Until now, the objective of recognition is basically achieved with an acceptable accuracy.

However, a problem of poor suction point for CNN cannot be avoided because the location of the target item is decided by the center of the sliding window with the highest score. Figure 33 shows an example problem of the location of the toilet brush. The left image is the current result in which the location of the toilet brush is regarded as the center of the sliding window. But a more accurate result is the right image in which the suction point is properly located on the toilet brush itself. Items especially the ones in stick shapes like toilet brush really tend to be located incorrectly when the only method of sliding window is applied because we regard the center of the window as an approximation location of the target item.

Figure 33 Problem of the wrong suction point location

After comparing the advantages and disadvantages of the previous methods, we propose a hybrid method to combine CNN + dynamic sliding window with other

traditional methods like GMS and image differencing to find a more accurate suction point.

The operation process of the two hybrid methodologies is applying CNN + dynamic sliding window, finding the window with the highest score, operating to find the center of the item part in the window and giving the suction point. GMS and image differencing are all applied to the window. GMS works by extracting the features and finding the match points. Image differencing works by removing the background and remaining the target item region. Both of the two methods increase the recognition accuracy and CNN + dynamic sliding window + image differencing tends to find a more accurate suction point for item picking.

Results comparison

In this section, the recognition results for the 5 methodologies are compared to discuss and analyze the advantages and disadvantages. First GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window are compared to research the differences between traditional methods and deep learning methods. Then based on CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing are compared to analyze the performance of finding accurate suction points.

Table 10 and Figure 34 show the results which compare the accuracy of GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window. The total average recognition accuracies of GMS + fixed sliding window,

CNN + fixed sliding window and CNN + dynamic sliding window are 0.4925, 0.7955 and 0.8795.

Table 10 Result comparison of GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window

Item name	GMS + fixed sliding window	CNN + fixed sliding window	CNN + dynamic sliding window
Measuring spoons	0.06	0.64	0.82
Table cloth	0.38	0.96	0.96
Epsom salts	0.84	0.96	0.96
Burts bees baby wipes	0.44	0.96	0.96
Plastic wine glass	0.12	0.68	0.84
Speed stick	0.64	0.7	0.94
Expo eraser	0.52	0.66	0.58
Hinged ruled index cards	0.52	0.86	0.94
Glue sticks	0.48	0.86	0.86
Poland spring water	0.42	0.78	0.86
Ice cube tray	0.04	0.96	0.98
Flashlight	0.02	0.28	0.74
Black fashion gloves	0.36	0.92	0.88
Colgate toothbrush 4pk	0.82	0.9	0.92
Duct tape	0.14	0.5	0.88
Tennis ball container	0.72	0.74	0.84
Windex	0.82	0.9	0.94
Balloons	0.32	0.9	0.96
Irish spring soap	0.74	0.6	0.74
Scotch sponges	0.32	0.8	0.92
Crayons	0.76	0.6	0.86
Tissue box	0.36	0.88	0.88
Laugh out loud jokes	0.7	0.94	0.94
Ticonderoga pencils	0.94	0.84	0.9
Composition book	0.92	0.96	0.98
Mesh cup	0.06	0.82	0.86
Band aid tape	0.66	0.8	0.78
Hand weight	0.32	0.62	0.9
Avery binder	0.24	0.96	0.82
Bath sponge	0.16	1	1
Marbles	0.28	0.8	0.94

Robots DVD	0.88	0.94	0.96
Fiskars scissors	0.34	0.5	0.76
Reynolds wrap	0.74	0.74	0.86
White facecloth	0.16	0.86	0.96
Toilet brush	0.36	0.38	0.56
Robots everywhere	0.98	0.98	0.98
Hanes socks	0.76	0.7	0.8
Mouse traps	0.66	0.96	0.98
Pie plates	0.7	0.98	0.94
Average accuracy	0.4925	0.7955	0.8795

Figure 34 Result comparison of GMS + fixed sliding window, CNN + fixed sliding window and CNN + dynamic sliding window

The accuracy of CNN + fixed sliding window is higher than the one of GMS + fixed sliding window especially for the featureless items or items with uncertain poses which cannot be identified by the GMS method such as measuring spoons, plastic wine glass, ice cube tray, black fashion gloves, tissue box, avery binder, bath sponge, marbles and white facecloth. After applying the dynamic sliding window for all the items, the accuracy of the CNN method increase especially for the small items like flashlight and scissors.

To make a summary for the existing results, the advantages and disadvantages of GMS and CNN methods are described in Figure 35.

Figure 35 Advantages and disadvantages of GMS and CNN methods

However after analyzing the results, the accuracies of expo eraser, flashlight, irish spring soap, fiskars scissors and toilet brush are considered lower than the average accuracy which need to be increased. The hybrid methods are tested to find a more

accurate suction point to improve the recognition accuracy. Table 11 and Figure 36 show the result which compares the accuracy of CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing. The total average recognition accuracies of CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing are 0.8795, 0.938 and 0.9525.

Table 11 Result comparison of CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing

Item name	CNN + dynamic sliding window	CNN + dynamic sliding window + GMS	CNN + dynamic sliding window + image differencing
Measuring spoons	0.82	0.94	0.92
Table cloth	0.96	0.98	0.98
Epsom salts	0.96	0.98	0.98
Burts bees baby wipes	0.96	0.98	0.98
Plastic wine glass	0.84	0.9	0.9
Speed stick	0.94	1	1
Expo eraser	0.58	0.76	0.82
Hinged ruled index cards	0.94	0.98	0.98
Glue sticks	0.86	0.9	0.9
Poland spring water	0.86	0.94	0.92
Ice cube tray	0.98	0.98	1
Flashlight	0.74	0.92	0.94
Black fashion gloves	0.88	0.92	0.92
Colgate toothbrush 4pk	0.92	0.96	0.96
Duct tape	0.88	0.92	0.96
Tennis ball container	0.84	0.88	0.92
Windex	0.94	0.96	0.96
Balloons	0.96	0.96	1
Irish spring soap	0.74	0.9	0.94
Scotch sponges	0.92	0.96	0.98
Crayons	0.86	0.96	0.96
Tissue box	0.88	0.94	0.98
Laugh out loud jokes	0.94	0.98	0.98

Ticonderoga pencils	0.9	0.9	0.9
Composition book	0.98	1	1
Mesh cup	0.86	0.96	0.96
Band aid tape	0.78	0.9	0.9
Hand weight	0.9	0.98	0.98
Avery binder	0.82	0.9	0.94
Bath sponge	1	1	1
Marbles	0.94	0.94	1
Robots DVD	0.96	0.96	0.96
Fiskars scissors	0.76	0.88	0.94
Reynolds wrap	0.86	0.92	0.98
White facecloth	0.96	0.98	0.98
Toilet brush	0.56	0.74	0.82
Robots everywhere	0.98	1	1
Hanes socks	0.8	0.86	0.86
Mouse traps	0.98	1	1
Pie plates	0.94	1	1
Average accuracy	0.8795	0.938	0.9525

Figure 36 Result comparison of CNN + dynamic sliding window, CNN + dynamic sliding window + GMS and CNN + dynamic sliding window + image differencing

After comparing the results of the 3 methods, the accuracy of recognition increases especially the one of some problem items like measuring spoons, expo eraser, flashlight, irish spring soap, crayons, band aid tape, fiskars scissors and toilet brush. These results indicate that the problem of poor suction can be fixed by applying the GMS method or image differencing method to find the item in the extracted window. This combination method can really improve the recognition accuracy and the suction point location.

With careful observation and analysis, GMS cannot work sometimes because the background is not totally black and it may be regarded as the feature. What's more, GMS uses ORB as a feature detector for key points especially the corners. Therefore, if the target part in the extracted window only contains the edge information, then the method will not work very well. Compared to this, CNN + dynamic sliding window seems have a better characteristic and performance. A comparison summary is shown in Figure 37 to analyze the two hybrid methods.

Figure 37 Comparison of the two hybrid recognition methods

CHAPTER6 Conclusions and Perspectives

6.1 Conclusions

The system aims to achieve the order picking in unstructured environment in e-commerce logistics. Due to the significance of the perception method in the system integration, the objective for this thesis is to implement the object recognition and localization of the items in Amazon Robotic Challenge.

Considering the complexity of all the items and system environment, two recognition methods are used and tested to achieve the target. One method is a traditional method called GMS (Grid-based Motion Statistics) and another one is a deep learning method called CNN (convolutional neural network using GoogLeNet Inception v3). While applying the two methods, sliding window is used instead of segmentation because of the cluttered environment. By doing this, more complete information is kept and makes it easier to realize the recognition. The size of the sliding window is tuned and decided by the experiment performance which quarter-image size window for GMS and a (80×80) - pixels window for CNN.

Between these two methods, GMS is found to be suitable for the items with obvious features since it proceeds the recognition by detect and match the features. In order to find a method to identify the items without features and generalize the method, CNN is used. The biggest problem of CNN is the training time, so Google inception v3 model is chosen and transfer learning is used to remove the final full-connected layer and retrain the model with the own database for the e-commerce system. This method

minimum the training and preparation time for the creation of the database and make it become feasible. After testing the CNN method, the average accuracy is 0.7955 which becomes a lot higher than the one of GMS of 0.4925 which shows the capability of CNN for the featureless items.

After making sure that the CNN method's advantages and analyzing the results, it's found that the small items are less possible to be identified because of the large difference between the sizes of the sliding window and item. In order to improve this, dynamic sliding window is used based on the item size and increase the recognition accuracy from 0.7955 to 0.8795. It solves part of the problem and increases the accuracy of small items. After the analysis, the remained problem becomes the poor suction point of CNN for irregular or small item because the center of the sliding window may not locate on the target items accurately.

In order to improve the method and get more accurate suction points, GMS and image differencing are combined with CNN + dynamic sliding window. The basic process is to get the sliding window with the highest score which means the window has a high possibility of containing part of the target item and use the image to proceed GMS matching or comparison with the background. By doing this, a suction point on the item part in the extracted window will be given out which will increase the recognition accuracy because of the erase of the target point for the item. CNN + dynamic sliding window + GMS method increases the accuracy to 0.938, and CNN + dynamic sliding window + image differencing increases the accuracy to 0.9525.

The summary of all the 5 methods are listed in Table 12 to show the improvement of the accuracy of the recognition.

Table 12 Result comparison of all the recognition methods

Item name	GMS + fixed sliding window	CNN + fixed sliding window	CNN + dynamic sliding window	CNN + dynamic sliding window + GMS	CNN + dynamic sliding window + image differencing
Measuring spoons	0.06	0.64	0.82	0.94	0.92
Table cloth	0.38	0.96	0.96	0.98	0.98
Epsom salts	0.84	0.96	0.96	0.98	0.98
Burts bees baby wipes	0.44	0.96	0.96	0.98	0.98
Plastic wine glass	0.12	0.68	0.84	0.9	0.9
Speed stick	0.64	0.7	0.94	1	1
Expo eraser	0.52	0.66	0.58	0.76	0.82
Hinged ruled index cards	0.52	0.86	0.94	0.98	0.98
Glue sticks	0.48	0.86	0.86	0.9	0.9
Poland spring water	0.42	0.78	0.86	0.94	0.92
Ice cube tray	0.04	0.96	0.98	0.98	1
Flashlight	0.02	0.28	0.74	0.92	0.94
Black fashion gloves	0.36	0.92	0.88	0.92	0.92
Colgate toothbrush 4pk	0.82	0.9	0.92	0.96	0.96
Duct tape	0.14	0.5	0.88	0.92	0.96
Tennis ball container	0.72	0.74	0.84	0.88	0.92
Windex	0.82	0.9	0.94	0.96	0.96
Balloons	0.32	0.9	0.96	0.96	1
Irish spring soap	0.74	0.6	0.74	0.9	0.94
Scotch sponges	0.32	0.8	0.92	0.96	0.98
Crayons	0.76	0.6	0.86	0.96	0.96
Tissue box	0.36	0.88	0.88	0.94	0.98
Laugh out loud jokes	0.7	0.94	0.94	0.98	0.98
Ticonderoga	0.94	0.84	0.9	0.9	0.9

pencils					
Composition book	0.92	0.96	0.98	1	1
Mesh cup	0.06	0.82	0.86	0.96	0.96
Band aid tape	0.66	0.8	0.78	0.9	0.9
Hand weight	0.32	0.62	0.9	0.98	0.98
Avery binder	0.24	0.96	0.82	0.9	0.94
Bath sponge	0.16	1	1	1	1
Marbles	0.28	0.8	0.94	0.94	1
Robots DVD	0.88	0.94	0.96	0.96	0.96
Fiskars scissors	0.34	0.5	0.76	0.88	0.94
Reynolds wrap	0.74	0.74	0.86	0.92	0.98
White facecloth	0.16	0.86	0.96	0.98	0.98
Toilet brush	0.36	0.38	0.56	0.74	0.82
Robots everywhere	0.98	0.98	0.98	1	1
Hanes socks	0.76	0.7	0.8	0.86	0.86
Mouse traps	0.66	0.96	0.98	1	1
Pie plates	0.7	0.98	0.94	1	1
Average accuracy	0.4925	0.7955	0.8795	0.938	0.9525

Another issue for the recognition process is the training process and training time. The scanning system which is used to create the database and the whole training process are described in the previous part of the article. As for some small patch of the items, it takes only 8 minutes for 5 items, 15 minutes for 10 items and 20 minutes for 20 items. From these data, it really becomes a feasible way to deal with the goods and sets up the database and model for e-commerce system rapidly. Through this method, it's possible to realize the picking and sorting in unstructured environment with high efficiency and accuracy.

As a conclusion, CNN + dynamic sliding window + image differencing is good perception method for object recognition in unstructured environment. This method is improved to realize the recognition for the items with different features, sizes and

shapes. The general process of the method is to use CNN + dynamic sliding window to get an approximate location and find an accurate suction point by image differencing. This method can reach the recognition accuracy up to 95%.

6.2 Perspectives

With the development of the e-commerce system, many relative problems are needed to be solved like automated order picking in unstructured environment. Based on the items from Amazon and their different characteristics, a recognition and classification method is proposed by implementing CNN with GMS and image differencing. The thesis tells about the whole developing progress to achieve a high recognition accuracy of the object recognition.

After a summary of the whole method, some problems still exist for further improvement. One main objective is the speed for the algorithm. While applying the method, it usually takes 6 – 15 seconds to finish finding the item location since every sliding window needs to be passed into the inception model for classification. Considering the whole system, the speed is acceptable because the robot needs time to react and pick the item. But it's still necessary to improve the speed of the vision method to realize a wider application and higher efficiency.

REFERENCE

1. Meng, Y., *Relationship of e-commerce, logistics and information technology*. 2011 International Conference on E-Business and E-Government (ICEE), 2011: p. 1-4.
2. Yang, Y. and W. Hao-yu, *Mechanism of Logistics information in reverse tracking system under E-commerce*. Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics, 2011: p. 177-181.
3. Andrea, R.D., *Guest Editorial: A Revolution in the Warehouse: A Retrospective on Kiva Systems and the Grand Challenges Ahead*. IEEE Transactions on Automation Science and Engineering, 2012. **9**(4): p. 638-639.
4. Pochyly, A., et al., *3D vision systems for industrial bin-picking applications*. Proceedings of 15th International Conference MECHATRONIKA, 2012: p. 1-6.
5. M. Hassaballah, H.A.A., A.A. Abdelmgeid, *Image Feature Detectors and Descriptors*. Studies in Computational Intelligence, ed. M.H. Ali Ismail Awad. Vol. 630. 2016, Switzerland: Springer International. 11-45.
6. Wikipedia. *Computer vision*. System methods 10 August 2017 [cited 2018 February 2]; Available from: https://en.wikipedia.org/wiki/Computer_vision.
7. Lisin, D.A., et al., *Combining Local and Global Image Features for Object Class Recognition*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, 2005: p. 47-47.
8. WilliamThomas, H.M. and S.C.P. Kumar, *A review of segmentation and edge detection methods for real time image processing used to detect brain tumour*. 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2015: p. 1-4.
9. Jain, R., R. Kasturi, and B.G. Schunck, *Machine Vision*. 1995: McGraw-Hill.
10. Canny, J., *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **PAMI-8**(6): p. 679-698.
11. Ram, P. and S. Padmavathi, *Analysis of Harris corner detection for color images*. 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), 2016: p. 405-410.
12. Jianbo, S. and C. Tomasi, *Good features to track*. 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994: p. 593-600.
13. S. M. Smith, J.M.B., *SUSAN – a new approach to low level image processing*. International Journal of Computer Vision, 1997. **23**(1): p. 45-79.
14. Trajković, M. and M. Hedley, *Fast corner detection*. Image and Vision Computing, 1998. **16**(2): p. 75-87.
15. Rosten, E. and T. Drummond, *Machine Learning for High-Speed Corner Detection*. Proceedings of 9th European Conference on Computer Vision, Graz, Austria, Part I, 2006: p. 430-443.
16. Lindeberg, T., *Image Matching Using Generalized Scale-Space Interest Points*. Scale Space and Variational Methods in Computer Vision: 4th International Conference, SSVM 2013, Schloss Seggau, Leibnitz, Austria, June 2-6, 2013.

- Proceedings, ed. A. Kuijper, et al. 2013, Berlin, Heidelberg: Springer Berlin Heidelberg. 355-367.
17. Lindeberg, T., *Scale Selection Properties of Generalized Scale-Space Interest Point Detectors*. Journal of Mathematical Imaging and Vision, 2013. **46**(2): p. 177-210.
 18. Hough, P.V.C. *Method and Means for Recognizing Complex Patterns*. 1962 1962-12-18 [cited 2018 February 2]; Available from: <http://www.osti.gov/scitech/servlets/purl/4746348>.
 19. Hough, P.V.C. and B.W. Powell, *A method for faster analysis of bubble chamber photographs*. Il Nuovo Cimento (1955-1965), 1960. **18**(6): p. 1184-1191.
 20. Mikolajczyk, K. and C. Schmid, *Scale & Affine Invariant Interest Point Detectors*. Int. J. Comput. Vision, 2004. **60**(1): p. 63-86.
 21. Mikolajczyk, K. and C. Schmid, *An Affine Invariant Interest Point Detector*. Proceedings of the 7th European Conference on Computer Vision-Part I, 2002: p. 128-142.
 22. contributors, W. *Visual descriptor*. 2017 4 July 2017 [cited 2018 February 2]; Available from: https://en.wikipedia.org/w/index.php?title=Visual_descriptor&oldid=788982618.
 23. B. S. Manjunath, J.R.O., V. V. Vinod, and A. Yamada, *Color and Texture descriptors*. IEEE Trans. Circuits and Systems for Video Technology, 2001. **11**(6): p. 703-716.
 24. Hassaballah, M., A.A. Abdelmgeid, and H.A. Alshazly, *Image Features Detection, Description and Matching*. Image Feature Detectors and Descriptors : Foundations and Applications, ed. A.I. Awad and M. Hassaballah. 2016: Springer International Publishing. 11-45.
 25. Lowe, D.G., *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 2004. **60**(2): p. 91-110.
 26. Bay, H., T. Tuytelaars, and L. Van Gool, *SURF: Speeded Up Robust Features*. Proceedings of 9th European Conference on Computer Vision, Part I, 2006: p. 404-417.
 27. Mikolajczyk, K. and C. Schmid, *A performance evaluation of local descriptors*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005. **27**(10): p. 1615-1630.
 28. Ojala, T., M. Pietikainen, and D. Harwood, *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions*. Proceedings of 12th International Conference on Pattern Recognition, 1994: p. 582-585 vol.1.
 29. Wang, X., T.X. Han, and S. Yan, *An HOG-LBP human detector with partial occlusion handling*. 12th International Conference on Computer Vision (IEEE), 2009: p. 32-39.
 30. Calonder, M., et al., *BRIEF: binary robust independent elementary features*. Proceedings of the 11th European conference on Computer vision: Part IV, 2010: p. 778-792.

31. Calonder, M., et al., *BRIEF: Computing a Local Binary Descriptor Very Fast*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012. **34**(7): p. 1281-1298.
32. Rublee, E., et al., *ORB: An efficient alternative to SIFT or SURF*. 2011 International Conference on Computer Vision, 2011: p. 2564-2571.
33. Muja, M. and D.G. Lowe, *Scalable Nearest Neighbor Algorithms for High Dimensional Data*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014. **36**(11): p. 2227-2240.
34. Bian, J., et al., *GMS: Grid-based Motion Statistics for Fast, Ultra-robust Feature Correspondence*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: p. 2828-2837.
35. Lecun, Y., et al., *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998: p. 2278-2324.
36. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet classification with deep convolutional neural networks*. Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012: p. 1097-1105.
37. Zeiler, M.D. and R. Fergus, *Visualizing and Understanding Convolutional Networks*. European Conference on Computer Vision (ECCV), 2014: p. 818-833.
38. Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*. International Journal of Computer Science & Communication (CoRR), 2014: p. 1409-1556.
39. Szegedy, C., et al., *Going deeper with convolutions*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: p. 1-9.
40. Ioffe, S. and C. Szegedy, *Batch normalization: accelerating deep network training by reducing internal covariate shift*. Proceedings of the 32nd International Conference on Machine Learning, 2015: p. 448-456.
41. Szegedy, C., et al., *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. Advancement of Artificial Intelligence (AAAI), 2017: p. 4278-4284.
42. Szegedy, C., et al., *Rethinking the Inception Architecture for Computer Vision*. Computer Vision and Pattern Recognition (CVPR), 2016: p. 2818-2826.
43. He, K., et al., *Deep Residual Learning for Image Recognition*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: p. 770-778.
44. He, K., et al., *Mask r-cnn*. IEEE International Conference on Computer Vision (ICCV), 2017: p. 2980-2988.
45. LLC, A.R. *Amazon Robotics Challenge*. 2015 [cited 2018 February 2]; Available from: <https://www.amazonrobotics.com/#/roboticschallenge/results>.
46. Wurman, P.R. and J.M. Romano, *The Amazon Picking Challenge 2015 [Competitions]*. IEEE Robotics & Automation Magazine, 2015. **22**(3): p. 10-12.
47. Correll, N., et al., *Analysis and Observations From the First Amazon Picking Challenge*. IEEE Transactions on Automation Science and Engineering, 2017. **99**: p. 1-17.

48. Hernandez, C., et al., *Team Delft's Robot Winner of the Amazon Picking Challenge 2016*. International Journal of Computer Science & Communication (CoRR), 2016. **abs/1610.05514**.
49. stanford, c. *Convolutional Neural Networks for Visual Recognition*. [cited 2018 February 2]; Available from: <http://cs231n.github.io/convolutional-networks/>.
50. Donahue, J., et al., *DeCAF: a deep convolutional activation feature for generic visual recognition*. Proceedings of the 31st International Conference on Machine Learning, 2014: p. I-647-I-655.