Stereo Vision-based Object Detection Algorithm for USV using Faster R-CNN

By

Heesu Kim

Submitted for the Degree of Master of Philosophy

Department of Naval Architecture, Ocean and Marine Engineering

University of Strathclyde

October 2017

Declaration of Authenticity and Author's Rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in or derived from, his thesis.

Signed:

Date:

Acknowledgements

I would first like to thank my supervisor Dr. Evangelos Boulougouris of the Department of Naval Architecture, Ocean and Marine Engineering at the University of Strathclyde. He consistently allowed this thesis to be my own work but provided me with the research environment I needed. I also wish to express my thanks to my Korean supervisor Prof. Sang-Hyun Kim of the Department of Naval Architecture and Ocean Engineering at Inha University. He supported me to have the opportunity to study at the University of Strathclyde.

I would also like to thank the technicians who helped me with the towing tank experiments. Without their enthusiastic assistance, the verification experiment would not have been successful. My sincere thanks also go to Mr. Byongug Jeong, who has kindly and meticulously reviewed my thesis. He gave me his abundant knowledge in the writing of the thesis.

Finally, I must express my profound gratitude to my parents and to my friends for their heartfelt interest and continuous encouragement throughout my year of research. This achievement would not have been possible without them. Thank you.

Heesu Kim

Abstract

The missions at sea require automation due to human accessibility and labour constraints. Accordingly, the requirement for a USV is highlighted for surveillance, environment investigation, and so on. The fully automated USV requires the reliable detection system in accordance with the prerequisite to safe collision avoidance. For this, USVs are equipped with a number of equipment, but these units are expensive and demand extra loading capacity. Therefore, it is necessary to simplify such equipment, and at the same time, essential data for safe collision avoidance should be acquired without loss.

The equipment simplification potentially can be achieved by using a vision sensor. The vision sensor that has been used in the conventional USV only tracks the marine object. For safe collision avoidance, the type of object detected and the distance to the object are also required. This additional information requires direct observation from human or other equipment support. If the vision sensor can be used to estimate the distance and the object type, the equipment for USV can be simplified.

The purpose of this research is the development of vision-based object detection algorithm that recognises a marine object and estimates the position and distance to the object for USV. Faster R-CNN, a state-of-the-art image processing technique that imitates human visual perception, is used to recognise and localise object on a captured frame from a vision sensor. In order to obtain the distance to the recognised object, stereo vision based depth estimation technique is used. Therefore, a stereo camera was used in this research. By combining these two techniques, real-time marine object detection algorithm was implemented and the performance of this algorithm is verified by model ship detection test in towing tank. The test results showed that this algorithm is potentially applicable to real USV.

Key words: USV, vision-based object detection, Faster R-CNN, depth estimation

Contents

List of Tables

List of Figures

Nomenclature

AIS	Automatic Identification System
ANN	Artificial Neural Network
ASC	Autonomous Surface Craft
ASV	Autonomous Surface Vessel
AUV	Autonomous Underwater Vehicle
CNN	Convolutional Neural Network
COLREGs	International Regulations for Preventing Collisions at Sea 1972
DCPA	Distance to Closet Point of Approach
ECDIS	Electronic Chart Display and Information System
Fast R-CNN	Fast Region with Convolutional Neural Network
Faster R-CNN	Faster Region with Convolutional Neural Network
GMDSS	Global Maritime Distress and Safety System
GPS	Global Positioning System
GPU	Graphics Processing Unit
INS	Inertial Navigation System
IoU	Intersection of Union
ISLVRC	ImageNet Large Scale Visual Recognition Challenge
LiDAR	Light Detection And Ranging
LORAN	Long Range Navigation
LRF	Laser Rangefinder
mAP	Mean of Average Precision
MoZAK	Moment of Zoomed-Algorithm Kurtosis
MSE	Mean Squared Error
PASCAL	Pattern Analysis, Statistical Modelling and computational Learning
Radar	Radio Detection And Ranging
RANSAC	Random Sample Consensus
R-CNN	Region with Convolutional Neural Network
ReLU	Rectified Linear Units
RoI	Region of Interest
ROV	Remotely Operated Underwater Vehicle
RPN	Region Proposal Network

SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Feature Transform
SPPnets	Spatial Pyramid Pooling networks
SVM	Support Vector Machine
TCPA	Time to Closest Point of Approach
USV	Unmanned Surface Vehicle
VOC	Visual Object Classes
VTS	Vessel Traffic Service

1 Introduction

Autonomous systems are becoming an essential part of our life, reducing human labour and human error. The automatic system has found its way into the various control systems such as processes in factories, switching on telephone networks, heat treating, etc. Over time, the automatic system technology has been advanced and the concept of the fully automatic system, called automation, has been arising. This system is usually accomplished in combination with complex systems, such as modern factories, airplanes, and ships.

Accordingly, there is rapid growth in unmanned vehicle development such as unmanned ground and aerial vehicle for supporting transportation, surveillance environment investigation and so on. In the marine industry, there has been an effort on development of USV. It operates on the sea surface without crew and is becoming popular due to its reduced cost compared to for example research and oceanographic ships, and being more efficient than weather buoys. They are commonly designed to accomplish their mission from the commands transmitted remotely without humans' instant control or programmed to perform regularised actions repeatedly. This helps to avoid marine accidents mostly caused by human error (Campbell et al., 2012).

As vessels are automated, the significance of obtaining and processing the data surrounding the operating vehicles for safe navigation has increased. The collision avoidance through proper path planning ensures also prevention from a crash accident. Accordingly, it requires decent sensor system that detects accurately and processes the obtained data to applicable information that can be used for pertinent action.

In order to collect such data, the majority of USV is equipped with various sensors such as sonar sensor, AIS, LiDAR, Radar and vision sensor for detecting obstacles or other vessels. However, most of this equipment has disadvantages as they are expensive or difficult to install on a small ship due to their massive weight. This necessitates the simplification of the equipment and the reduction of their number. In this regard, the use of a vision sensor is powerful for USV where near obstacles are closely related to collision risk, in place of other expensive and heavy detection equipment. Furthermore, it can enhance detection by supporting existing detection system in large vessels.

Due to use of the vision sensor, it is required to process an image to recognise objects. In order for a USV to recognise an object without human intervention, it is important to possess object recognition ability comparable to that of a human being. For this purpose, this research uses the CNN which specialises in image processing more than other machine learning techniques. The CNN is a state-of-art technique of computer algorithms, mimicking animal's visual perception and learning abilities. Intelligent animals and humans obtain the ability of object recognition by learning the images and their corresponding names by experience over a long period of time. As the CNN works similarly, it requires a large number of images, many computational iterations, high computational power and time. Recently, due to the remarkable developments in data science, it is not difficult to collect a large number of datasets. Moreover, improvement of computer capacity reduced computation time significantly.

However, the brevity of research on the unmanned ship, there were no efforts or studies on the application of this method to the marine industry. Most vision-based detection systems in this area are set through the intuitive visual features observed by users (Gladstone et al., 2016, Shin et al., 2017, Sinisterra et al., 2014, Wang et al., 2011a, Wang et al., 2011b, Wang and Wei, 2013, Woo and Kim, 2016b, Woo and Kim, 2016a). Although it can be called automation, there is still a human error because it is eventually set by a human. The CNN can mitigate this problem by extracting the features on its own reducing human intervention. In this context, this research was motivated to apply the vision sensor, one of the economical and lightweight equipment for automatic navigation.

The aim of this research is to implement an algorithm to recognise other objects or ships using a stereo camera for autonomous navigation of USV. Faster R-CNN developed for real-time classification and localisation based on CNN is used, and depth estimation method is used to estimate the distance to detected objects. As a preliminary process, the CNN and RPN in the Faster R-CNN are fine-tuned. When the algorithm starts to run, a left frame passes through the whole network of the Faster R-CNN, and it classifies and localises the observed objects. After this, from the left and right vision, the 3D point cloud is created all over the pixels. By matching the local information and the 3D point cloud obtained from the Faster R-CNN and depth estimation, it estimates the distance to the objects. This process is repeated in real-time.

2 Aim and Objectives

The aim of this research is to contribute to enhancing vision-based detection system for USV. To achieve this, the following objectives are set:

1) To overview trend of an unmanned ship, existing object detection system and distance estimation techniques, and their application to review their limitations and to identify where to be improved.

2) To develop an object detection algorithm for automatic navigation for USVs.

3) To demonstrate the effectiveness of the proposed algorithm.

4) To provide recommendations for future research.

3 Literature Review

3.1 Remarks

In this section, it outlines three topics: unmanned ship, object detection system and distance estimation technique. The section of the unmanned ship indicates the general trend of unmanned ship briefly. The section of object detection system and distance estimation technique discusses their current technologies and applications in unmanned ships including other fields.

3.2 Unmanned Ship

There are many types of platforms for an unmanned ship such as USV, ASV, ASC, autonomous ship, etc. Although they are called in a different name, their ultimate purposes are similar as searching mines, surveying the ship’s bottom with ROV and detect suspicious divers; investigating the sea bottom; tracking target boat and so on (Bertram, 2008), without human control.

There have been many projects of unmanned ships for their development and application. The projects include, for example, the DELIM ASC for investigation of hydrothermal extent and the patterns of community diversity (Pascoal et al., 2000); sea surface autonomous catamaran, named SEAMO, for collection of data and samples on sea-air boundary (Caccia et al., 2005); ROAZ II ASV for search and rescue mission (Martins et al., 2007b) and for docking system interlocking with AUV (Martins et al., 2007a); MESSIN for measuring tasks in shallow water (Majohr et al., 2000); the autonomous catamaran, named Charlie for investigation of sea surface (Bibuli et al., 2008); the Springer for environmental and hydrographic surveys (Naeem et al., 2006); the non-crew commercial vessels, carried out by Rolls-Royce (Levander, 2017) etc.

In most of such projects, unmanned ships receive only mission commands without manual control of its instant movement (Naeem and Irwin, 2010). In order to satisfy it, the unmanned ships are required to be fully automated. However, they always face complex marine environment such as unexpected obstacles, harsh weather condition, and sensor signal disturbance during navigation. It attracts a lot of attention on how to process the measured or observed data reliably in order to manage the unmanned ship.

3.3 Object Detection

Unmanned ship needs data of surrounding environment, especially neighbouring obstacles, as the first step of collision avoidance. In order to collect such data, unmanned ships are equipped with various types of sensors such as LiDAR, Radar, LRF, sonar, etc. These sensors collect the dataand then the data is transformed into information that is useful to determine the unmanned ships' behaviour by certain data process technique. This serial process is called object detection. In order to obtain high-quality information from those processes, it is worth to discuss which sensors should be installed, which technique should be applied, which type of data is required, etc. The trend of them is reviewed in this chapter.

3.3.1 Non Vision-based Detection System

Presently, advanced navigational aids support navigation of ships by dedicating vessel positioning, wireless communication and the exploration of the environment (Olsson and Jansson, 2006). These aid systems assist the navigation as it localises neighbouring ships by plotting corresponding location on a map with GPS, LORAN, ECDIS etc. It includes telecommunication systems that transmit navigational mutual information by bidirectional transmission technique such as GMDSS and AIS. They communicate with the VTS or other ships directly (Ruiz and Granja, 2009).

However, it is not preferred to equip communication system to small vessels, which are commonly designed for unmanned ships due to loading capacity and cost efficiency. Therefore, most of the unmanned ships are equipped a with self-detection system such as Radar, LiDAR, a sonar sensor, etc.

For example, the Navico BR24 FMCW radar system was applied for target tracking algorithm while scanning 360 degrees (Schuster et al., 2014). A sonar sensor was applied to obstacle detection system for USV, detecting reefs or shallow banks in water (Heidarsson and Sukhatme, 2011a). A 2-D laser sensor was equipped with ASC and its detection system was tested in harbour (Bandyophadyay et al., 2010).

However, without vision sensor, there is a limit to detection of short-distance objects, which is largely associated with collision risk than long-distance objects. For example, typical radar measuring range is 0.3-5km (Ruiz and Granja, 2009), and the Furuno marine radar sensor detects from 22 meters (Onunka and Bright, 2010).

In order to overcome it, some researchers developed fusion sensory detection system such as combined system of stereo vision sensor, radar and AIS (Larson et al., 2007); Omnidirectional camera composed by INS and six cameras (Wolf et al., 2010); GPS and LiDAR (Leedekerken et al., 2010); Radar and vision sensor (Hermann et al., 2015); Radar, camera and GPS (Almeida et al., 2009); Sonar sensor and overhead image (Heidarsson and Sukhatme, 2011b);

However, these fusion sensor systems typically require high cost. If the vision sensor does not only track targets but also provide additional data of surrounding environment, the fusion sensor systems are able to be simplified alongside robust collision avoidance.

3.3.2 Vision-based Detection System

As the first step of the object detection system, it is required to determine a region where there is a possibility of an object to be present. For this, most of the detection systems borrow edge extraction method, which figures out object outline or boundary line between different coloured regions. Especially, the Canny edge detector (Canny, 1986) is used widely with several additional processes.

In the field of unmanned aerial vehicles, a method of extracting the horizon between the sky and the ground is often used in image processing to estimate their attitude. (Bao et al., 2005, Todorovic, 2002, Todorovic and Nechyba, 2004, Todorovic et al., 2003, McGee et al., 2005, Ettinger et al., 2003, Cornall and Egan, 2005, Cornall and Egan, 2004).

Similarly, in the unmanned ship sector, there has been a study of estimating the distance from an observed object by extracting a sky-sea line on a single image. Woo et al (2016) developed obstacle detection system for USV using the method of horizontal line extraction and feature extraction. RANSAC algorithm and SIFT are applied respectively. Furthermore, collision risk was estimated from motion information and DCPA and TCPA by using fuzzy estimator (Woo and Kim, 2016b). Wang et al (2011b) developed real-time obstacle detection system based on horizontal line extraction using RANSAC as well. Saliency detection and Harris corner extraction were used for feature extraction and tracking of objects (Wang et al., 2011b). Sergiy Fefilatyev (2008) developed automated ship detection system using horizontal line extraction to define the region to be processed in an image, edge extraction to determine the region to estimate objects presence, and connected components algorithm to label objects (Fefilatyev, 2008).

As other detection techniques, segmentation on the image has been studied with applicability in a diverse field (Khan and Shah, 2001, Nguyen and Wu, 2013). Pedro Santana et al (2012) developed a water detection image processing model for aquatic robots that stays on the water. In this research, water region is segmented by measuring optical flow's entropy across the frames of video (Santana et al., 2012). Daniel Socek et al (2005) developed a hybrid colour-based foreground object detection system for automated marine surveillance using colour segmentation technique with Bayesian decision framework (Socek et al., 2005).

The methods of those researches elicit the location of detected objects. However, the horizontal line cannot be extracted accurately in foggy weather, inland waterway, and raging waters due to an unclear sea-sky joined line. It leads to an error in estimating the object region on the image Furthermore, only using edge extraction method cannot draw out the type of obstacle which has the potentiality of useful information for path planning.

3.3.3 Object Classification and Localisation

In order to surmount the weakness of the conventional vision-based detection, we introduced Faster R-CNN (Ren et al., 2015), which is a type of machine learning classifying and localising objects in an image.

The Faster R-CNN is largely divided into CNN (Krizhevsky et al., 2012) and RPN parts. The CNN and the RPN are responsible for the classification of images and localisation of objects, respectively. Previously developed CNN only performed classification. In order to add the function of localisation, R-CNN (Girshick et al., 2014) was developed. However, it took a long time in computing during training and detecting, thus Fast R-CNN (Girshick, 2015) was developed to reduce the computation time. Nonetheless, it still took a long time to compute so that it was impossible to apply it to the real-time detection mission. To improve this, Faster R-CNN was developed and it dramatically decreased the computing time, making it applicable to real-time detection.

The Faster R-CNN has a great potential and is still being developed for performance enhancement in various fields such as face recognition (Sun et al., 2017, Ranjan et al., 2017, Zhu et al., 2017, Qin et al., 2016, Jiang and Learned-Miller, 2017), visual relations (Zhang et al., 2017), action detection (Peng and Schmid, 2016), person search (Xiao et al., 2017), text detection (Zhong et al., 2016), tumour detection (Akselrod-Ballin et al., 2016), traffic sign detection (Zhu et al., 2016), 3D pose estimation (Poirson et al., 2016), etc.

In order to improve the performance of classification and localisation, research on the development of network architecture is also actively carried out (Howard et al., 2017, Shrivastava and Gupta, 2016, Lin et al., 2016, Zhang et al., 2016, Kong et al., 2016, Redmon and Farhadi, 2016, Huang et al., 2016, Kang et al., 2017, Tolias et al., 2015).

With the classification and localisation ability of the Faster R-CNN, it is possible to derive the type and location of objects that are intimately related to the risk of collision.

3.4 Estimation of Distance to Objects

The distance to an object is the most relevant variable to the collision risk. The Faster R-CNN provides the location on the image but does not calculate the distance to objects. Therefore, in order to find the distance to objects in an image, depth estimation technique is used in this research.

3.4.1 Monocular Vision-Based Depth Estimation

The research of depth estimation on a single image has been studied and applied in various fields. The robot developed by Lee et al (2016) can calculate the shortest distance between the robot and an obstacle with a monocular camera (Lee et al., 2016). It extracts the closest horizontal border line of an object by using probability density function that separates the regions according to its colour and texture. Ranftl et al (2016) introduced a method of dense depth estimation from a monocular vision of a dynamic scene. They proposed a novel motion segmentation algorithm and reconstructed moving objects in company with encompassing environment (Ranftl et al., 2016). Z. Said et al (2012) indicated the reliability of monocular vision-based depth estimation method performed by Wahab et al (2011) and Jan and lqbal (2009) (Wahab et al., 2011, Jan and Iqbal, 2009). It estimates the distance to a round-shaped object with primary knowledge of object size and the geometrical relation between camera and object (Said et al., 2012). Haris et al (2011) introduced a distance estimation technique, called MoZAK, for robotic arm movement (Haris et al., 2011). It guides that movement from statistical analysis with the degree of complexity of image edges, which is extracted by the Canny filter (Canny, 1986). Saxena et al (2006) proposed a depth estimation method from a single monocular image using supervised learning approach. They adopted three types of visual information for this approach, namely texture variations, texture gradients, and haze. They also applied two types of features, namely absolute depth features and relative features. They separated the single image into small patches and estimate a depth value for each corresponding path (Saxena et al., 2006). Liu et al (2016) performed depth estimation from single monocular image combining CNN and Conditional Random Filed. They carried out network learning process introducing unary and pairwise potential functions (Liu et al., 2016).

However, the depth estimation on single image requires a lot of hand-crafted assumption that causes detection errors in a different environment.

3.4.2 Stereo Vision-Based Depth Estimation

One way to overcome the disadvantages of single vision-based depth estimation is to use two fields of vision imitating the animal’s eyes. Since stereo vision-based depth estimation is highly accurate, it is often used to calculate the distance from its location on an unmanned ship to obstacles. Sinisterra et al (2014) proposed a target tracking system with stereo vision using Extended Kalman Filter. It facilitates targeting object and estimating depth by comparing across frames (Sinisterra et al., 2014). Terry Huntsberger at el (2011) analysed Hammerhead vision system, which detects a geometric threat, for a stereo vision-based autonomous navigation system in maritime environments, especially for high-speed USV (Huntsberger et al., 2011). Wang et al (2011a, 2012) developed real-time obstacle detection system with a stereo vision based on Saliency detection and Harris corner extraction (Wang et al., 2011a, Wang et al., 2012).

In this research, one of the conventional depth estimation method, Semi-Global Block Matching (Hirschmüller, 2007), is used for the disparity.

3.5 Summary

Most of the USV are equipped with self-detection system such as Radar, LiDAR, a sonar sensor, etc. Those sensors are useful for recognising other ships or obstacles over long distances, but the factor that closely related to USV crashes are near obstacles. To remedy this, some USV equips fusion sensor system, but it also has a limit due to load capacity and it requires a high cost. It causes to streamline the equipment as the use of a vision sensor that detects relatively short distances. In order to acquire high-quality information equivalent to other sensors, Faster R-CNN that classify and localise detected objects on frame image is introduced. The distance to the detected object is also important information. For this, depth estimation technique using stereo camera is introduced.

4 Methodology

4.1 Remarks

This section describes the methodology of this research. Basically, Faster R-CNN and depth estimation techniques are used, and the detection algorithm is implemented by integrating the two techniques. This section describes only the flow of the overall method and the modified part from existing algorithms borrowed. A description of the Faster R-CNN was given in the Appendix.

4.2 Algorithm Architecture

The project is composed of two stages as shown in figure 4.1. The first stage is localisation and classification performed by Faster R-CNN. In this stage, the process is carried out with only left frame acquired from left view. It provides the information of object type and the location on the frame. The second stage is depth estimation. It utilises both side frames and figures out the depth, which represents the distance to an every pixel point. This process furnishes the information of distance to object mobilising object local data obtained from the Faster R-CNN.

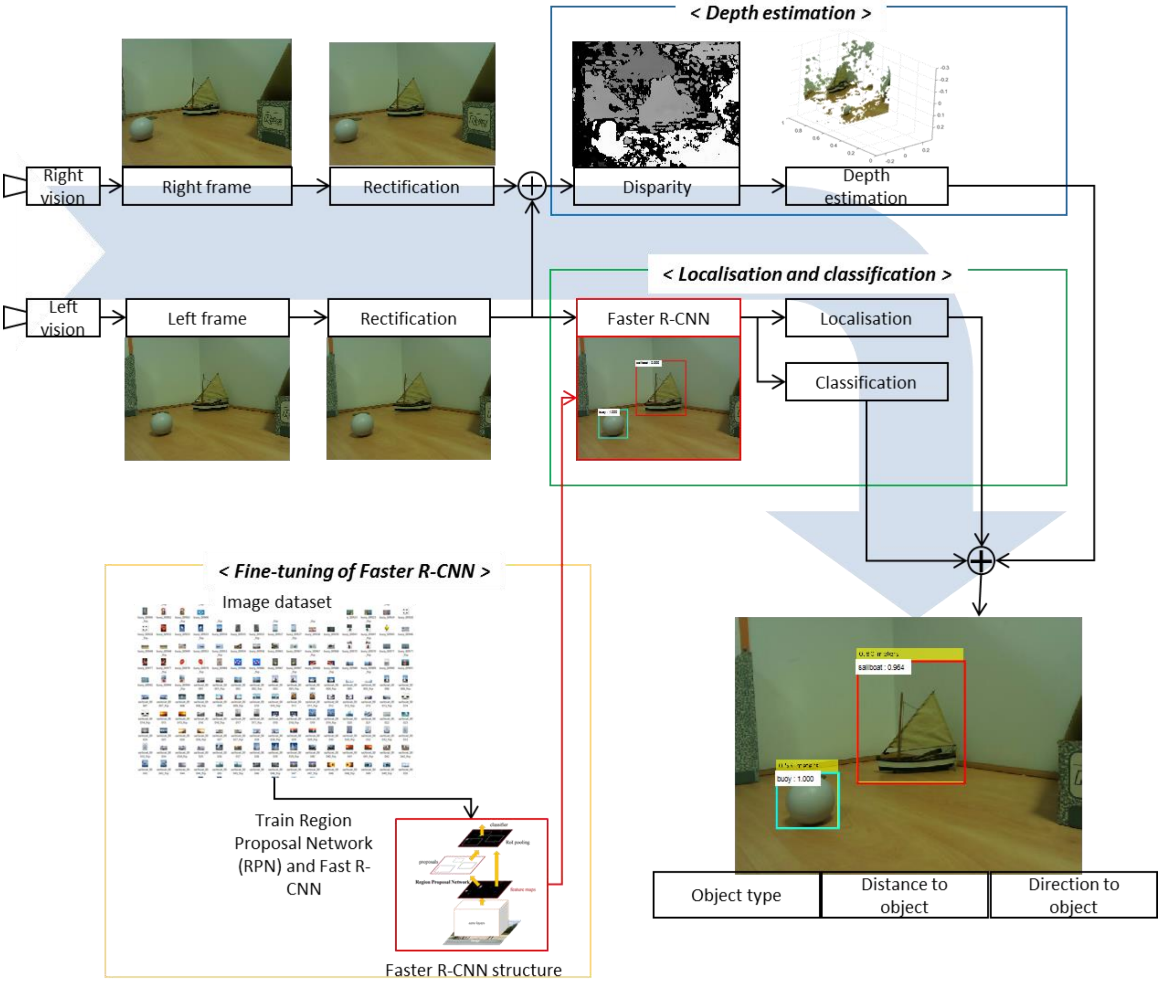


Figure 4.1 Project Architecture

4.3 Faster R-CNN

Although there is a default configuration in Faster R-CNN that gives the best performance in VOC2017 (Everingham et al., 2007), some configurations are modified to be suitable to recognise the ship as it has not been utilised in the marine industry.

4.3.1 CNN model

There are many CNN models that have been released such as ALexNet (Krizhevsky et al., 2012), ZF Net (Zeiler and Fergus, 2013), VGG Net (Simonyan and Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), Microsoft ResNet (He et al., 2016), etc. As such these networks are becoming deeper, they showed higher accuracy in classification. However, although they are improved, they also require higher GPU memory capacity as it processes more massive data. It restricted the options for using the best network among them. Due to this reason, we selected ZF Net that does not cause out of the memory of GPU that used in this research.

ZF Net is the network that won the ILSVRC 2013[1] . This model reached an 11.2% error rate and was fine-tuned more than the AlexNet architecture, which won the ILSVRC 2012. It is alike to AlexNet, but with a few slight alterations, it has improved performance. ZF net uses 7 × 7 filters instead of 11 × 11 filters used in AlexNet, and the stride is also reduced. This allows the first convolutional layer to maintain a lot of initial pixel information. ReLU is used for the activation function, the cross-entropy loss is used for error function, and batch stochastic gradient descent is used for training (Deshpande, 2016a). Its architecture is shown in figure 4.2.

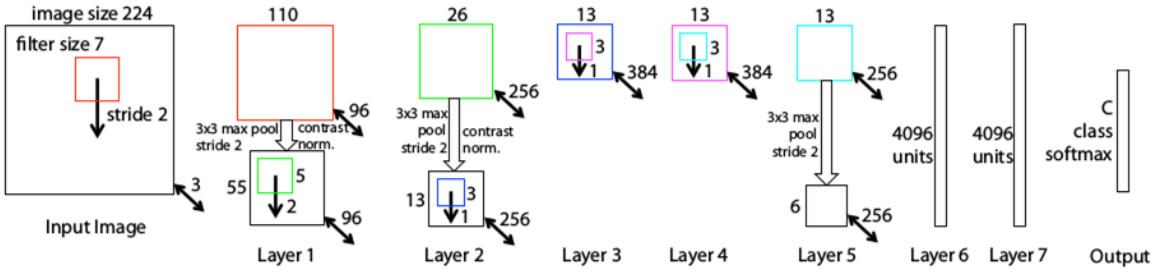


Figure 4.2 ZF net architecture (Deshpande, 2016a).

[1] http://www.image-net.org/challenges/LSVRC/2013/results.php#cls

4.3.2 Anchor

In the Faster R-CNN process, the input image is scaled such that their shorter side becomes 600 pixels while the long side does not exceed 1000 pixels before it is fed into a network. Therefore, the 640 × 480 pixels image captured by the stereo camera is scaled to 800 × 600 pixel. The anchors propose regions on this scaled image with its size of 128^2 pixels, 256^2 pixels, 512^2 pixels and its ratios of 1:2, 1:1 and 2:1 as shown in figure 4.3 and 4.4 (Ren et al., 2015).

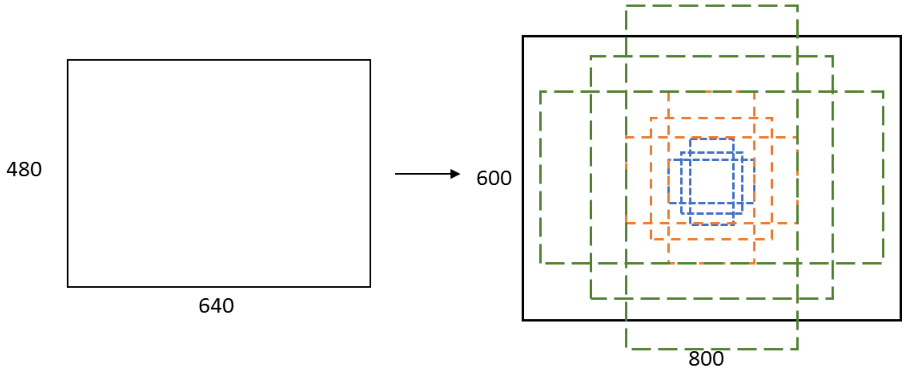


Figure 4.3 Scaled image and applied anchors.

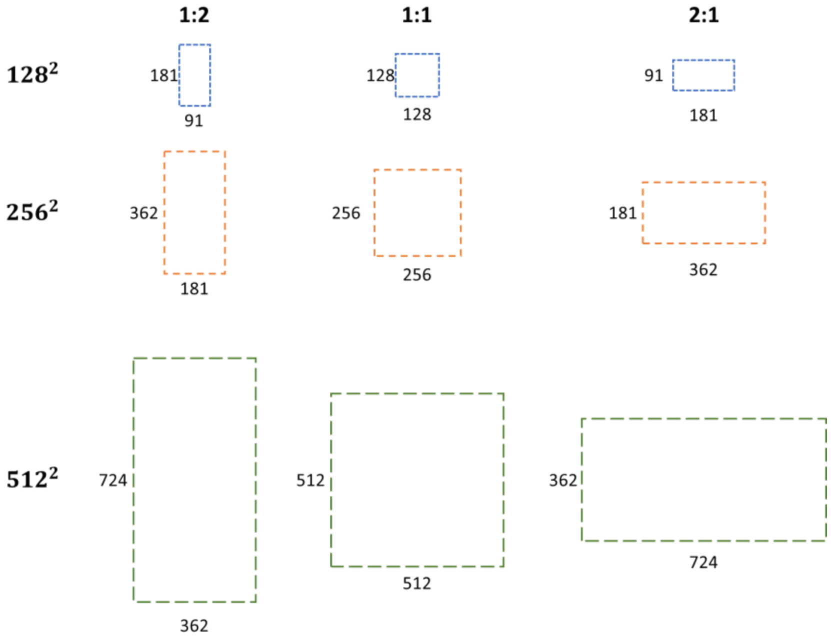


Figure 4.4 Default anchors.

In this process, there is a critical drawback to detect a small object. For example, if it detects a side of a small ship that is 3 m long and 50 m away, the ship occupies around 30×6 pixels on the captured image, and it is scaled to 37×7 pixels. At the moment the smallest anchor slides over the object region, as shown in figure 4.5, the IoU is only 0.016, which is much smaller than default IoU threshold 0.7 to be considered as positive. With this default anchor, the ground-truth box smaller than 90^2 pixels cannot be labelled as positive.

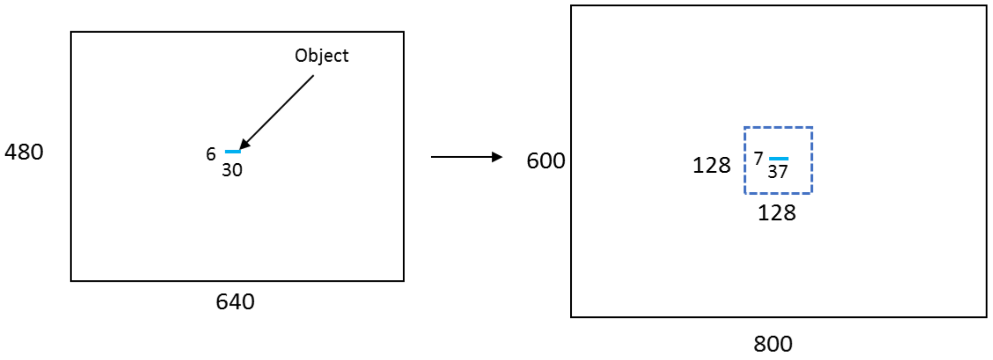


Figure 4.5 The overlap between 128^2 anchor and small object.

Therefore, the anchor size and ratio are recommended to be set to at least 16^2 pixels and 5:1, as shown in figure 4.6, respectively, to maximise the IoU.

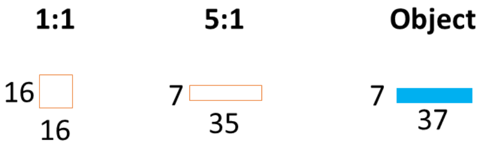


Figure 4.6 Size comparison between 16^2anchors with the ratio of 5:1.

Accordingly, we modified the anchor configuration from the default of it, to fit to detect a ship-shaped object in distance.

A 2 m-long small ship occupies 350×70 pixels at a distance of 4 m, and 30×6 pixels at a distance of 50 m, on the captured image from the stereo camera. These sizes are scaled to 438×88 pixels and 37×7 pixels, respectively. Correspondingly, the optimal range of anchor size is from 16^2 pixel to 196^2 pixel with the ratio of 5:1. Because changing the size and ratio of anchor from its default reduces its mAP (Ren et al., 2015), we followed the anchor size and IoU threshold from Faster R-CNN for small logo detection (Eggert et al., 2017), to minimise the loss of mAP, and also modified the setting to drop small boxes as changing minimum box size from 16^2 to 2^2 to enable to detect small area. The anchor configuration is shown in table 4.1.

Table 4.1 Default and modified anchor configurations.

	Default	Modified
Anchor size (pixels)	128^2, 256^2, 512^2	8^2, 16^2, 32^2, 44^2, 64^2, 90^2, 128^2, 256^2
Anchor ratio	2:1, 1:1, 1:2	4:1, 5:1, 6:1
IoU threshold	0.7	0.5
Minimum box size (pixels)	16^2	2^2

4.3.3 Dataset

The powerful advantage of CNN is that it can classify objects by generalising same labelled objects into one category, although they have various appearances. It can be proved clearly if the experiment is carried out on real sea observing various real ships. However, in this research, the actual sea area test was replaced with an experiment that detects the model ship in the towing tank because there are many practical limitations such as preparing and measuring real distance. Therefore, the dataset consists of only one class of model ship.

A notable point in this section is the size of an object in an image used for training. As the modified anchor sizes are smaller than the default anchor sizes, proposals that are assigned as positive during training are required to be considered carefully. For example, assume that there is a 600×400 pixels object in an 800×600 pixels image and the anchor size is 16^2 pixels. When the anchor slides over the ground-truth box, it labels everywhere as positive and catches all the feature of the object minutely, rather than its overall outline as shown in figure 4.7. However, when observing a distant object, the overall outline is a criterion that recognises objects more than detailed features due to the fixed resolution of the camera. It causes difficulty in recognising distant object.

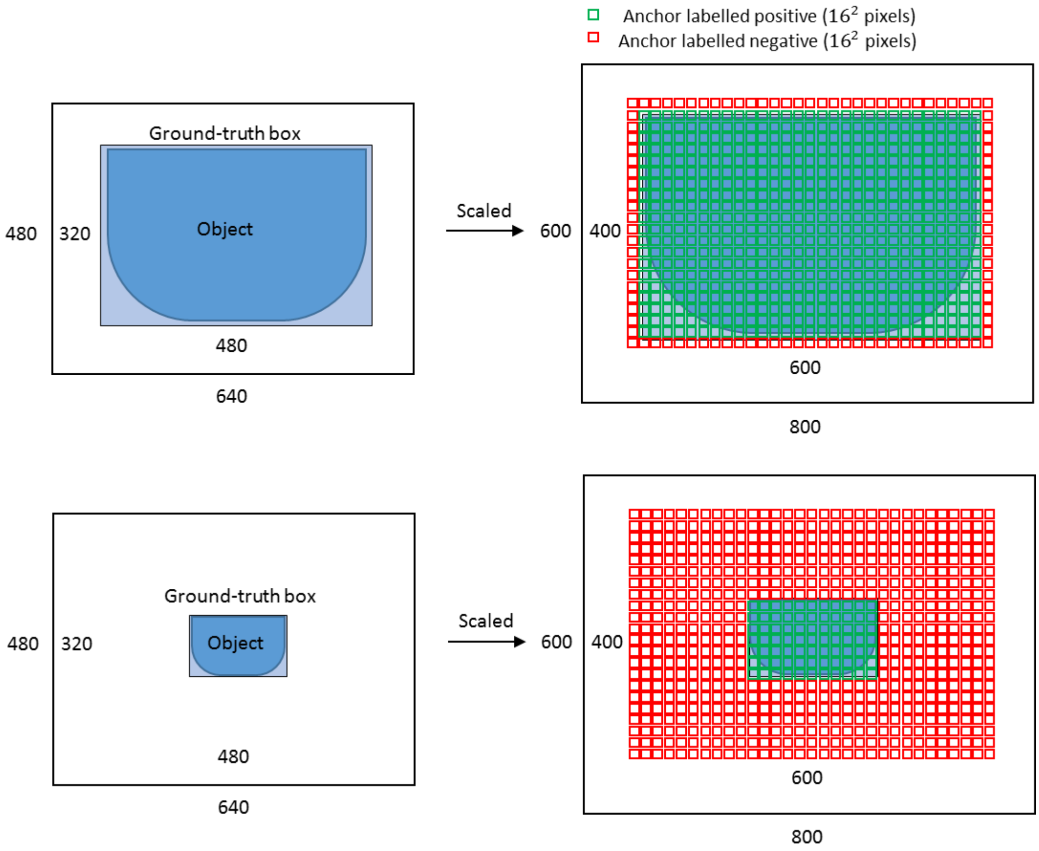


Figure 4.7 Anchors labelled positive and negative on a large object and small object.

Therefore, in order to detect small objects, anchors must capture the outline of the object as a feature. This means that the scaled object size of the dataset image should be similar to the scaled size of the object to be detected.

As the model ship is observed between the distances from 4m to 50m during the experiment, the ground-truth box of model ship occupies pixels from 37×7 pixels to 438×88 pixels, in 800×600 pixels scaled image. Accordingly, ground-truth box size in image datasets to be prepared are recommended to occupy pixels from 37×7 pixels to 438×88 pixels, where the area ratios of scaled images to the ground-truth boxes are $1{:}\,(5.40 \times 10^{-4})$ and $1{:}\,(8.03 \times 10^{-2})$, respectively. For example, if there is a 1500×1000 pixels image in a dataset, the ground-truth box area is required to occupy the pixels from $(1500 \times 1000) \times (5.40 \times 10^{-4})$ to $(1500 \times 1000) \times (8.03 \times 10^{-2})$, e.g., from 810 pixels to 120450 pixels as shown in figure 4.8. Additionally, as the aspect ratio of the ground-truth box in dataset image closes to the anchor ratio, there is a high probability that anchors sliding over the object labelled as positive.

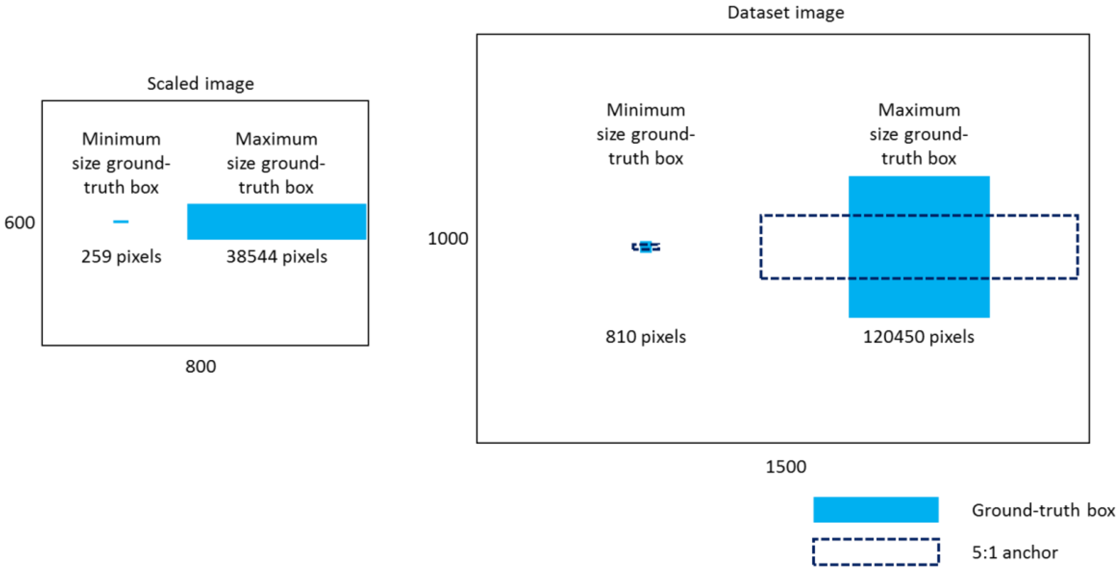

Scaled image
Minimum size ground-truth box
Maximum size ground-truth box
600
259 pixels
38544 pixels
800
Dataset image
Minimum size ground-truth box
Maximum size ground-truth box
1000
810 pixels
120450 pixels
1500
Ground-truth box
5:1 anchor

Figure 4.8 Example of recommended ground-truth box in dataset image.

The image dataset of the model ship to be used for the training was prepared by taking a picture of it. If the size of the object on the taken images is large, a margin is added to the edge of the images to satisfy above conditions. The number of dataset images is around 1000, referring to the PASCAL VOC 2007 dataset (Everingham et al., 2007).

4.4 Depth Estimation

The distance to object is important information for collision risk assessment. This section describes the process to calculate the distance to object. Unlike the case where only the left frame is used in Faster R-CNN, both frames are used in depth estimation and it calculates the distance to object based on the location of the object obtained from the Faster R-CNN.

The workflow of depth estimation is shown in figure 4.9 (Bradski and Kaehler, 2008, Dalal and Triggs, 2005). Left frame and Right frame are acquired from a stereo camera in real-time. Both frames are rectified and transform into the grey scale from RGB, and by using those, the disparity is calculated (Hirschmüller, 2007, Hirschmuller, 2005). This disparity is used to calculate the distance to each pixel point on the captured scene along with the stereo camera calibration parameters.

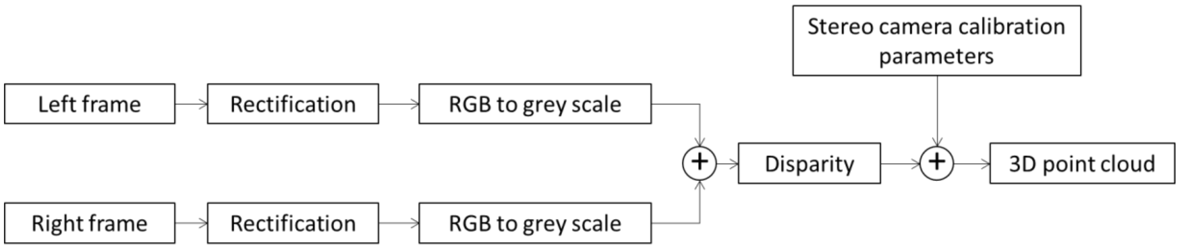
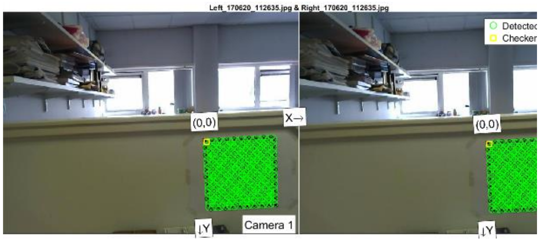


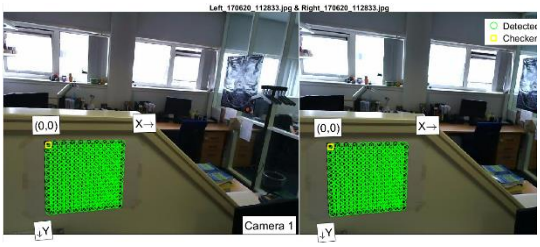
Figure 4.9 Workflow of depth estimation.

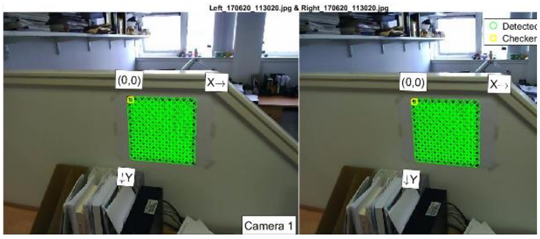
4.4.1 Stereo Camera Calibration

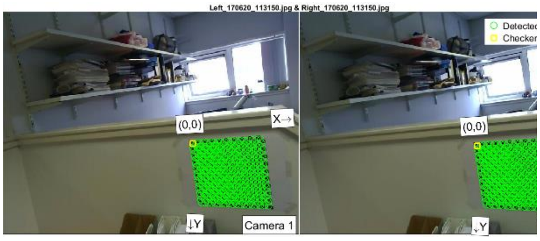
The stereo camera was calibrated with 20×20 checkerboard image with $14.4mm$ size of checkerboard square. 20 images were selected for the calibration not to exceed 0.15 of mean error in pixels. This images and mean error in pixels are shown in figure 4.10 and 4.11. The extrinsic parameters during the calibration are visualised in figure 4.12. The extrinsic parameters represent the coordinate system transformations from 3D world coordinates to 3D camera coordinates, which defines the camera's centre position and the camera's heading in world coordinates.

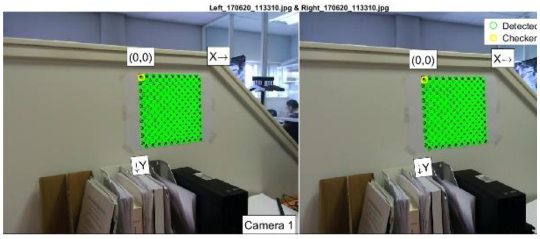

Left_170620_112635.jpg & Right_170620_112635.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker

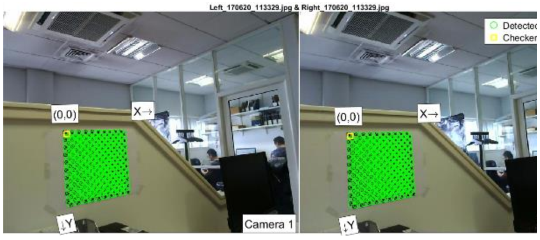
Left_170620_112739.jpg & Right_170620_112739.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker

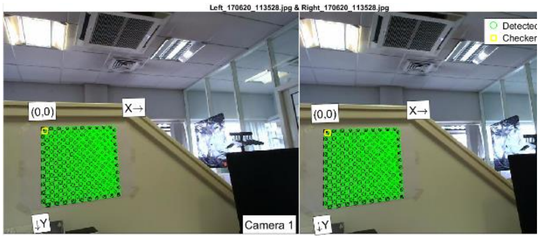

Left_170620_112833.jpg & Right_170620_112833.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker


Left_170620_113020.jpg & Right_170620_113020.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker

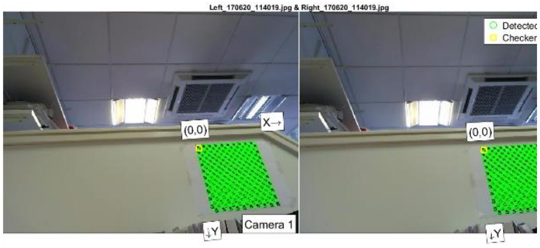

Left_170620_113150.jpg & Right_170620_113150.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker

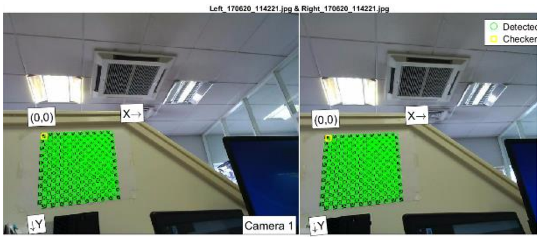

Left_170620_113310.jpg & Right_170620_113310.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker

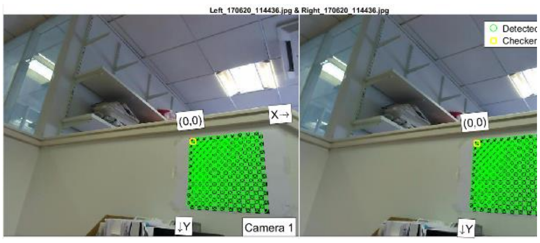

Left_170620_113329.jpg & Right_170620_113329.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker

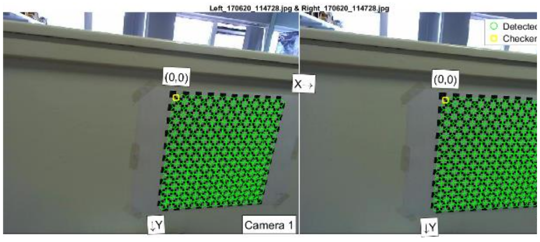

Left_170620_113528.jpg & Right_170620_113528.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker

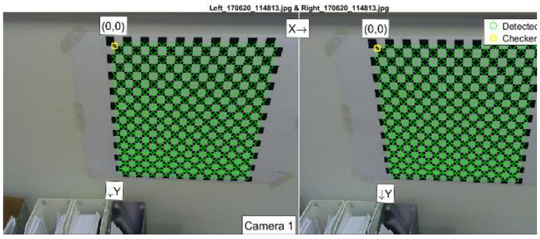
Left_170620_113935.jpg & Right_170620_113935.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker


Left_170620_114019.jpg & Right_170620_114019.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker


Left_170620_114221.jpg & Right_170620_114221.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
X→
↓Y
○ Detected
□ Checker


Left_170620_114436.jpg & Right_170620_114436.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker


Left_170620_114728.jpg & Right_170620_114728.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker


Left_170620_114813.jpg & Right_170620_114813.jpg
(0,0)
X→
↓Y
Camera 1
(0,0)
↓Y
○ Detected
□ Checker

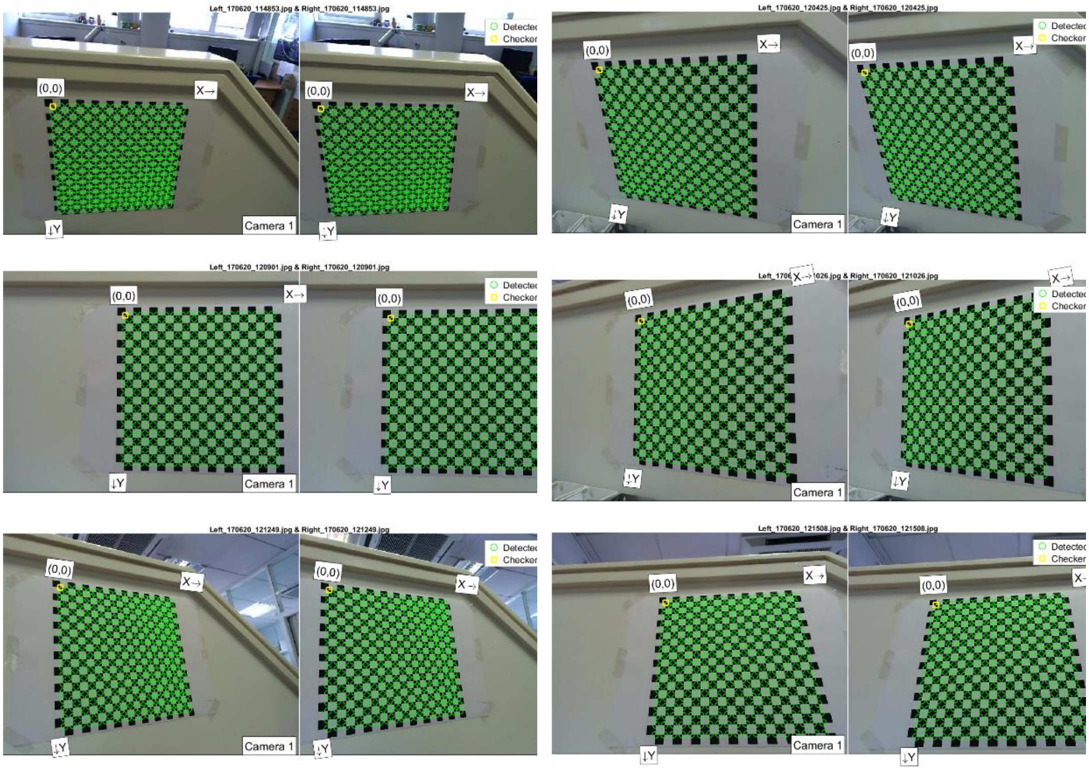

Left_170620_114853.jpg & Right_170620_114853.jpg
Left_170620_120425.jpg & Right_170620_120425.jpg
Left_170620_120901.jpg & Right_170620_120901.jpg
Left_170620_121026.jpg & Right_170620_121026.jpg
Left_170620_121249.jpg & Right_170620_121249.jpg
Left_170620_121508.jpg & Right_170620_121508.jpg
(0,0)
X→
↓Y
Camera 1
○ Detected
□ Checker

Figure 4.10 Pictures of checkerboard for stereo camera calibration.

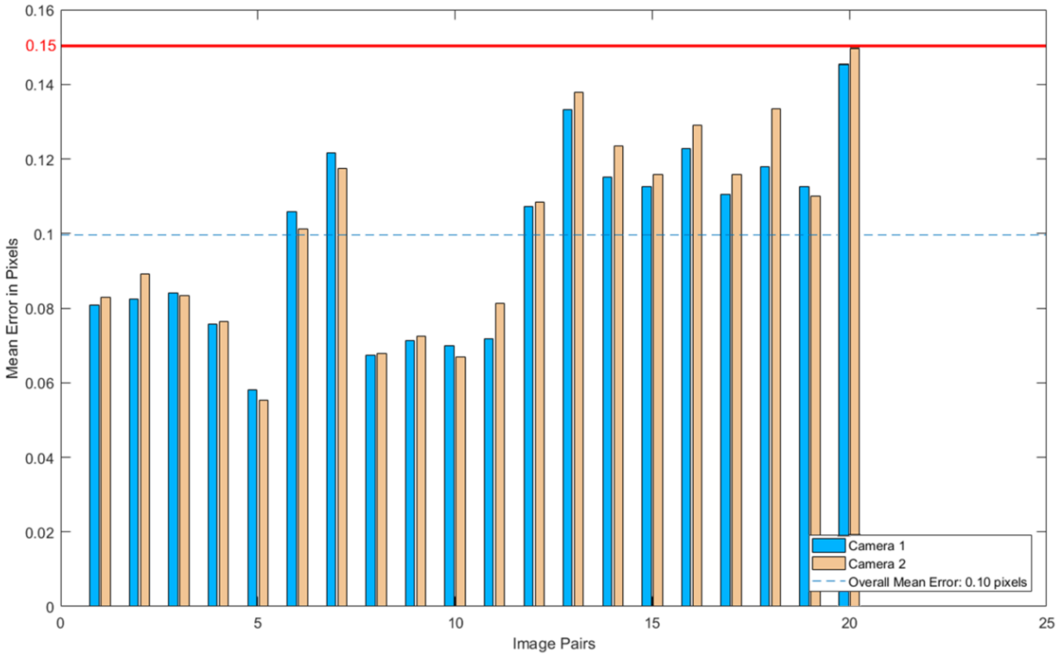


Figure 4.11 Mean error in pixels during calibration.

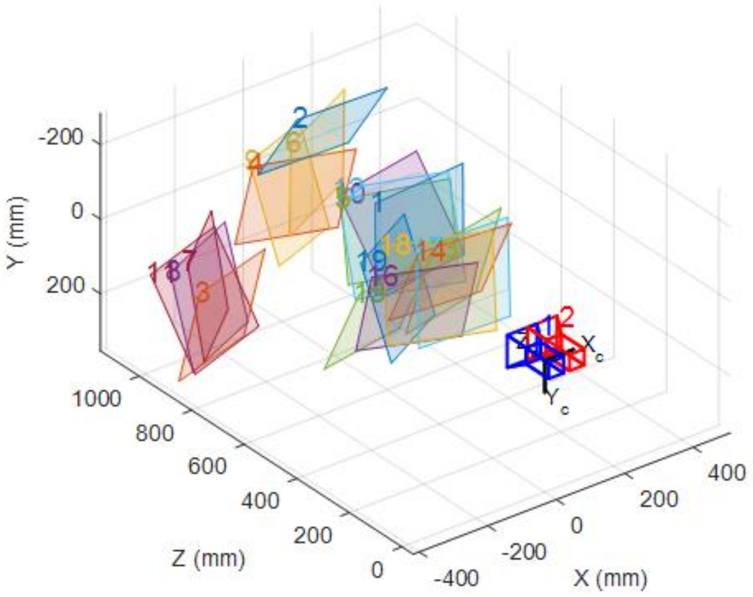


Figure 4.12 Extrinsic parameters visualisation.

4.4.2 Distance to Object

The Faster R-CNN represents the position of the object as a final output giving bounding boxes accompanying the values of left bottom point and right top point. From these values, we extracted the centre point of the bounding box as following equation 4.1.

$$(x_c, y_c) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right) \tag{4.1}$$

where (x_c, y_c) is the centre point of bounding box, (x_1, y_1) and (x_2, y_2) are the left bottom point and right top point of bounding box respectively. By matching this value to 3D point cloud map, the hypothetical distance to object, z_o, is calculated. This process is shown in figure 4.13.

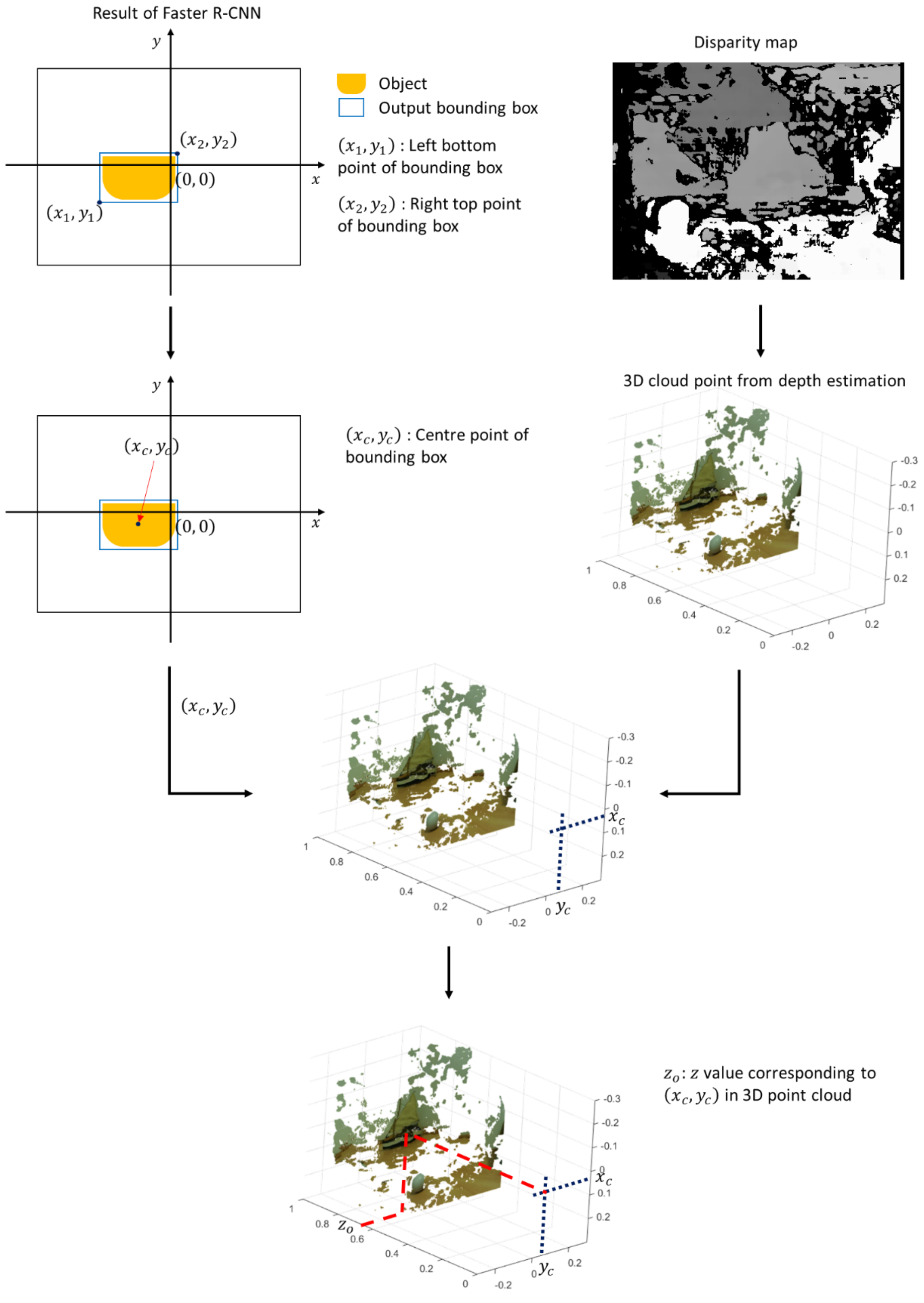

Result of Faster R-CNN
y
x
(x_2, y_2)
$(0, 0)$
(x_1, y_1)
🟨 Object
□ Output bounding box
(x_1, y_1) : Left bottom point of bounding box
(x_2, y_2) : Right top point of bounding box
Disparity map
3D cloud point from depth estimation
y
x
(x_c, y_c)
$(0, 0)$
(x_c, y_c) : Centre point of bounding box
(x_c, y_c)
x_c
y_c
z_o
z_o: z value corresponding to (x_c, y_c) in 3D point cloud
-0.3
-0.2
-0.1
0
0.1
0.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2

Figure 4.13 Workflow of distance estimation.

5 Detection Test

5.1 Remarks

Tests were carried out to investigate the effectiveness of the proposed detection algorithm. To evaluate the performance of the network according to the training condition, we divided the dataset and proposal configuration into four cases and trained the networks separately. The detection tests were carried out in towing tank observing a model ship floating on water through stereo camera hung on the carriage. By moving the carriage back and forth, the relative position of the model ship was changed and then the object recognition and the distance estimation results were evaluated.

5.2 Stereo Camera

The specification of the stereo camera used in this research is described in figure 5.1 and table 5.1. In this research, resolution and frame were set to 640 × 480 MJPEG and 30fps respectively due to memory limitation during computation.

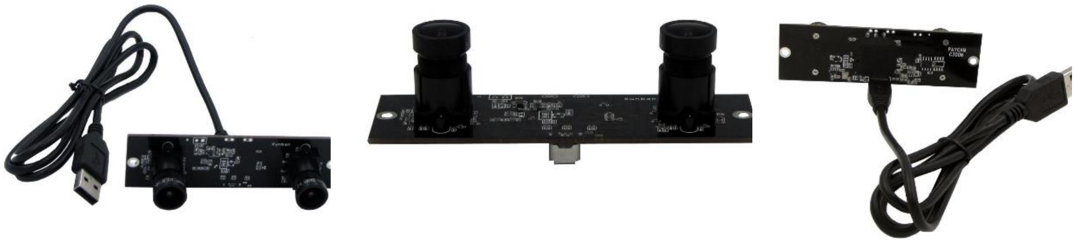


Figure 5.1 Stereo camera.

Table 5.1 Specification of the stereo camera.

Model name	KYT-U100-960R1ND
Sensor	Aptina AR0130
Focus	Manual
Synchronization	Yes
Resolution & frame	640 × 480 MJPEG 30fps, YUY2 15fps 1280 × 960 MJPEG 30fps, YUY2 5fps
Compression format	MJPEG \ YUY2
Interface	USB2.0
Lens Parameter	Non Distortion Lens, FOV 96°(D), 80°(H), 60°(V)
Voltage	DC5V
UVC (USB Video Class)	Support
OTG	Support
Auto exposure AEC	Support
Auto white balance AEB	Support
Adjustable parameters	Brightness/Contrast/Colour saturation/Definition/Gamma/WB
Dimension	74mm x 27mm
Operating Temperature	-20°C to 70°C
Support OS	Windows, Linux, MAC, Android

5.3 Computer Capacity

The computer environment in which the computation is performed is indicated in table 5.2.

Table 5.2 Computer environment.

CPU			Mainboard		
Specification	Cores	Threads	Model	Chipset	Southbridge
Intel Core i7 6700 CPU @ 3.40GHz	4	8	W650DC	Intel Skylake	Intel H170

Memory		Graphics Card			MATLAB
Type	Size	GPU	Memory type	Memory size	Version
DDR 4	8 GBytes	NVIDIA GeForce GTX 960M	GDDR5	4096 MB	R2016b

5.4 Detection Test Environment

The detection test was carried out by observing the model ship in a towing tank. The geometry is shown in figure 5.2. Since the LRF outputs the voltage according to the distance to the board, we measured the voltage and actual distance at three points and calibrated it.

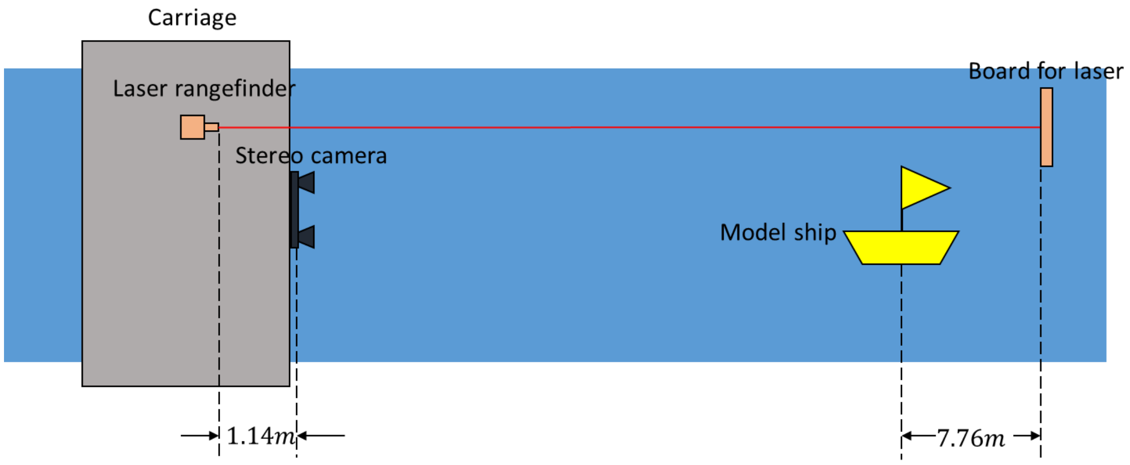


Figure 5.2 Towing tank geometry during detection test.

As shown in table 5.3 and figure 5.3, the voltage of the LRF and the distance to the board was calibrated.

Table 5.3 The corresponding values between LRF voltage output and real distance to the board.

LRF voltage output (V)	Real distance between LRF and board (m)
7.1902	53.8
5.4024	40.45
3.1187	23.23

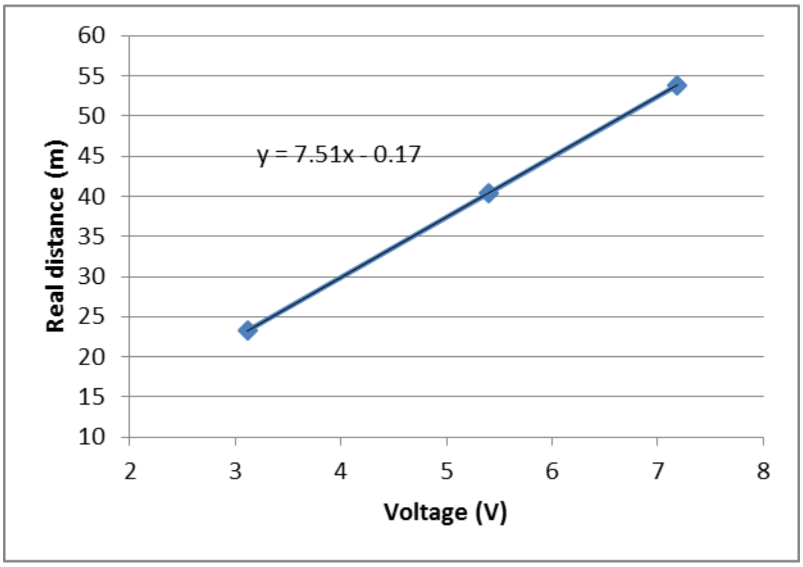


Figure 5.3 Regression line and equation between LRF voltage and real distance to board.

The regression equation is as following equation 5.1,

$$y = 7.51x - 0.17 \tag{5.1}$$

where x is voltage output from the LRF and y is a real distance between LRF and board. The distance between the stereo camera and the model ship was calculated by taking the geometry of the towing tank into consideration as following equation 5.2.

$$y^{*} = 7.51x + 8.73 \tag{5.2}$$

where y^* is the distance between stereo camera and model ship, and x is the voltage output from the LRF.

The length of the model ship used in the experiment is shown in figure 5.4 and was around 2 m, and only the side was observed during the detection test.

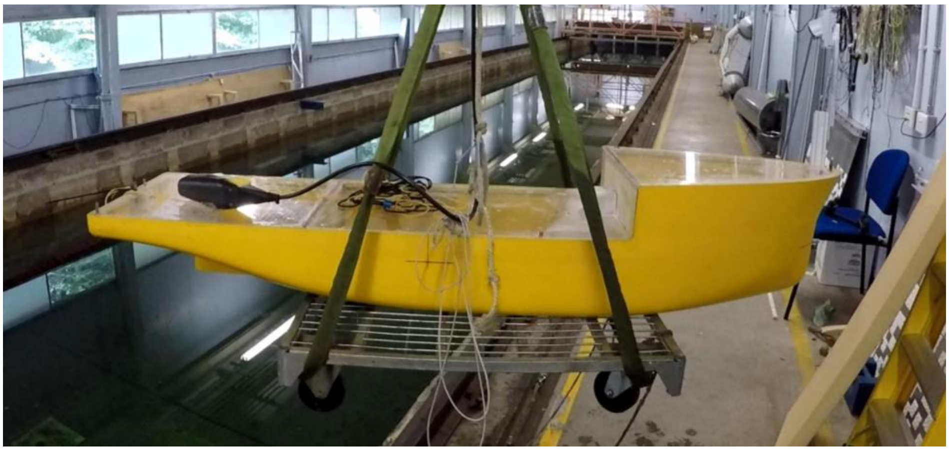
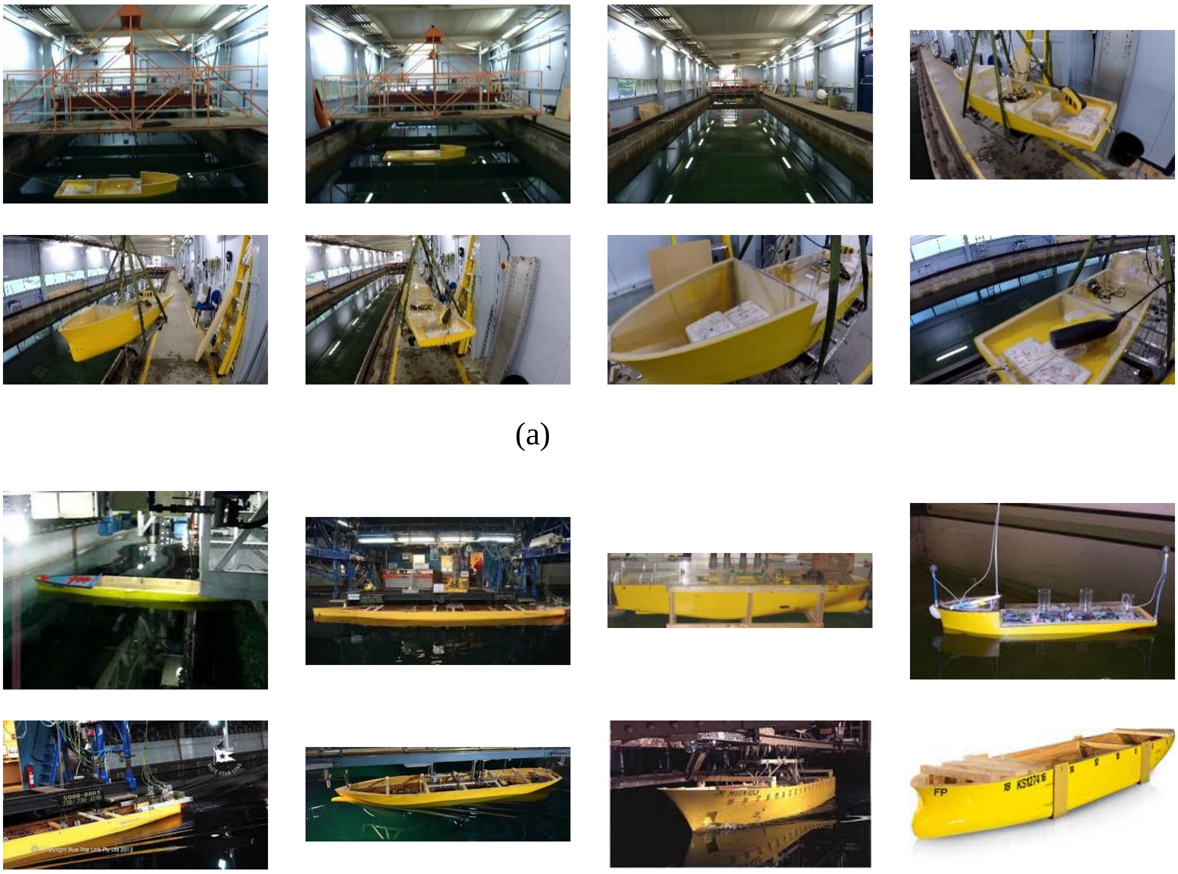


Figure 5.4 The model ship used in detection test.

5.5 Dataset

In order to observe the performance of the algorithm proposed in this research, the detection tests were carried out by changing dataset image, which has a large effect on the CNN performance. As the dataset, images of the model ship used in this detection experiment and of the other model ships have prepared as shown in figure 5.5.



(a)

(b)

Figure 5.5 Image samples in the dataset. (a) Image of model ship used in the detection system. (b) Image of the model ship not used in detection test.

5.6 Network Configuration

The detection test was carried out in the following cases to observe the performance of the algorithm according to the dataset type and proposal configuration as described in table 5.4.

Table 5.4 Dataset type and proposal configuration of networks for detection test.

	Dataset type		
Network case	Same model ship	Different model ship	Total amount
Case 1	930	104	1034
Case 2	930	104	1034
Case 3	0	303	303
Case 4	241	0	241

	Proposal configuration		
	Anchor size	Anchor ratio	IoU threshold
Case 1	$128^2, 256^2, 512^2$	1:2, 1:1, 2:1	0.7
Case 2	$4^2, 8^2, 16^2, 32^2, 44^2,$ $64^2, 90^2, 128^2, 256^2$	1:4, 1:5, 1:6	0.5
Case 3	$4^2, 8^2, 16^2, 32^2, 44^2,$ $64^2, 90^2, 128^2, 256^2$	1:4, 1:5, 1:6	0.5
Case 4	$4^2, 8^2, 16^2, 32^2, 44^2,$ $64^2, 90^2, 128^2, 256^2$	1:4, 1:5, 1:6	0.5

The network for case 1 is set to default proposal configuration and trained with the same model ship dataset image that will be observed in the test. This is for taking a see how powerful the existing CNN is with default configuration and for comparison with other modified networks. In case 2, the dataset is same to case 1 but the proposal configuration is changed. This configuration is modified for the purpose of small object detection. In case 3, the proposal configuration is same to case 2 but the dataset is composed of other model ship that is different from what will be observed. This is to see how much the network recognises when it is trained with a limited dataset. Case 4 is to see the effect of the amount of dataset. It has a relatively small amount of dataset. In order to calculate mAP, the dataset consists of the 70% of train images and the 30% test images.

5.7 Test Result

5.7.1 Network training result

The results of training each network are shown in table 5.5. The mAP is a factor that evaluates the quality of dataset and is an index of how much the test set relates to the train set. The higher the mAP is, the higher the associativity is between images of the datasets. Since the mAP of the ZF net trained with the PASCAL VOC 2007 dataset is 59.9% (Ren et al., 2015), the dataset used in this research is judged to be collected appropriately.

Table 5.5 Training and detection results of networks for each case.

Network case	Train-Time (hour)	mAP (%)
Case 1	18.57	66.04
Case 2	19.75	72.58
Case 3	18.27	79.87
Case 4	18.83	63.27

5.7.2 Detection Test Result

In the first detection test, the detection algorithm ran while the carriage with the stereo camera approaches to the model ship. It was carried out for each network in four cases. The initial distance between the stereo camera and the model ship is $47.86\ m$, and the speed of the carriage is $0.1\ m/s$. It starts moving after $10\ sec$ from the start of the camera recording. The results of detection test are shown in figure 5.6-9. In all cases, the mean computing time per frame was $0.33\ sec$ so that it is considered that there is no problem in real-time detection.

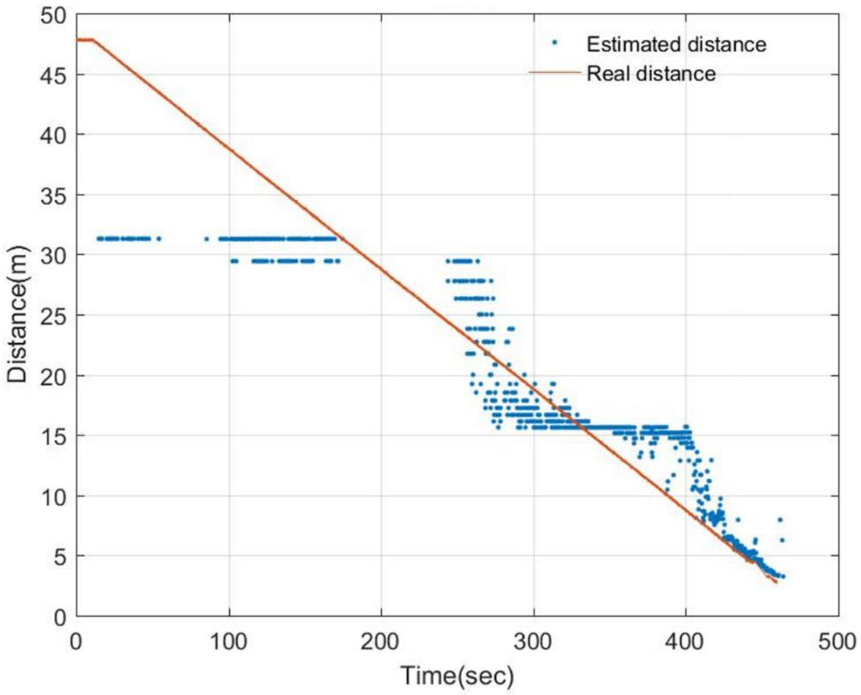


Figure 5.6 The result of detection and distance estimation result of network case 1.

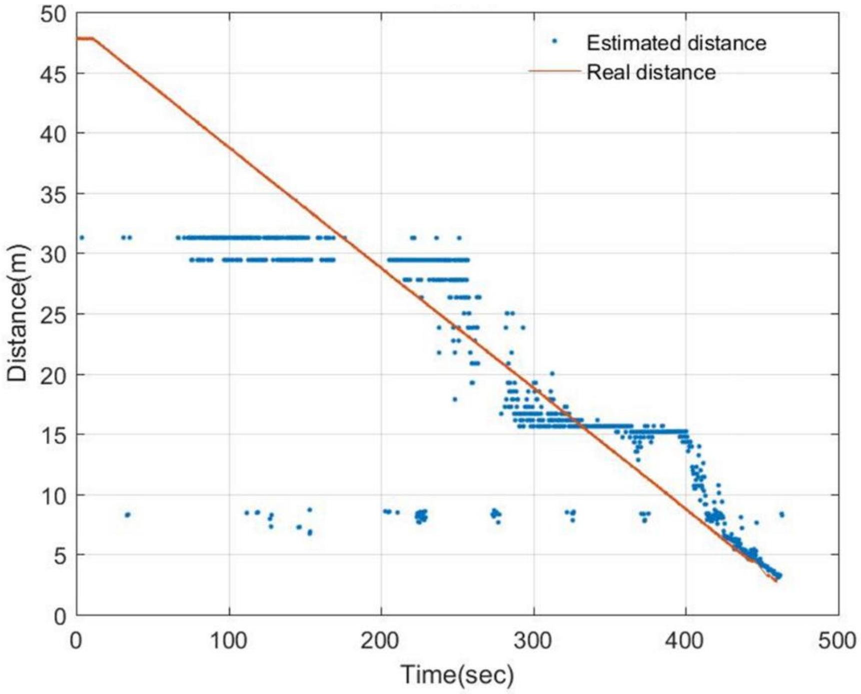


Figure 5.7 The result of detection and distance estimation result of network case 2.

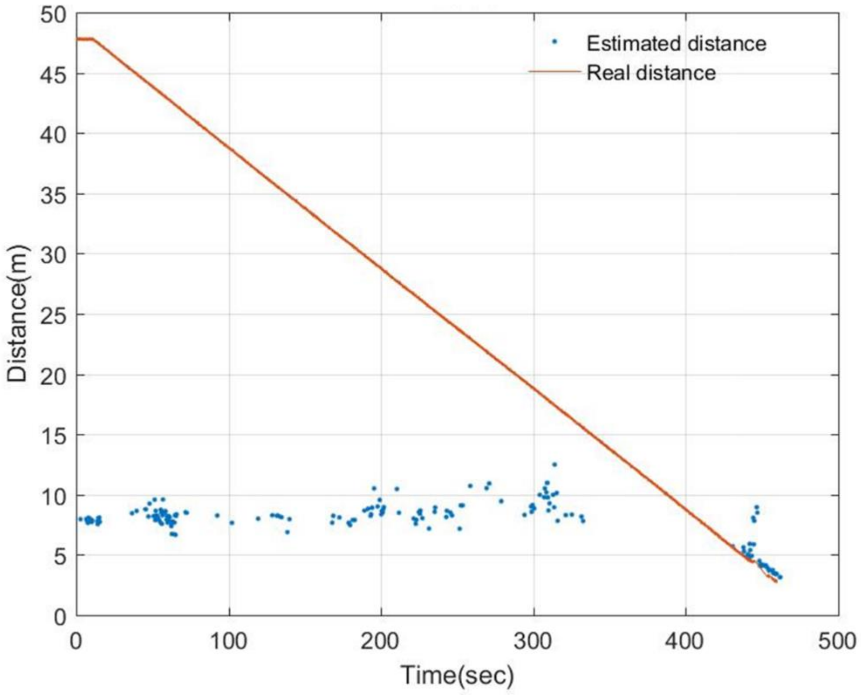


Figure 5.8 The result of detection and distance estimation result of network case 3.

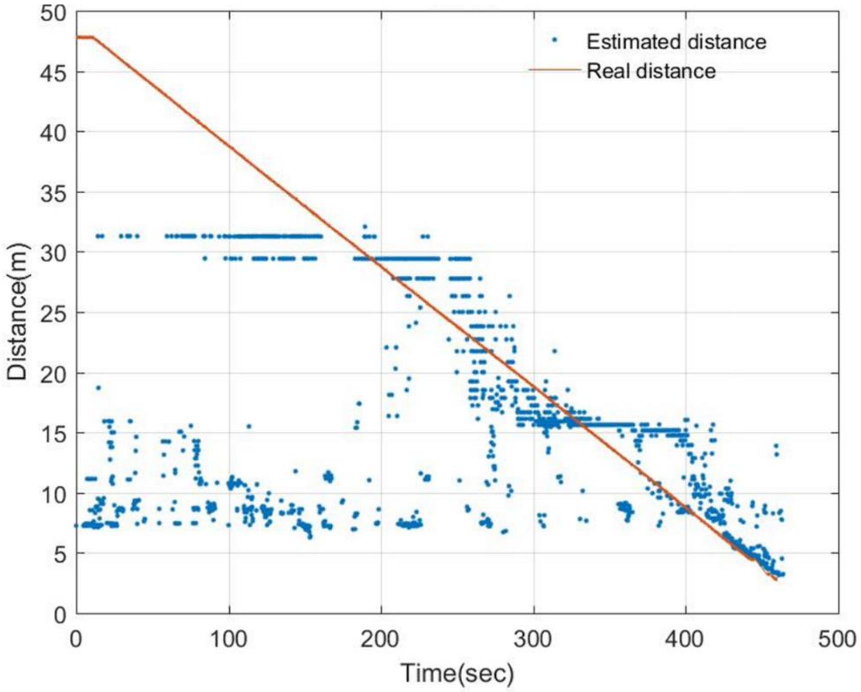
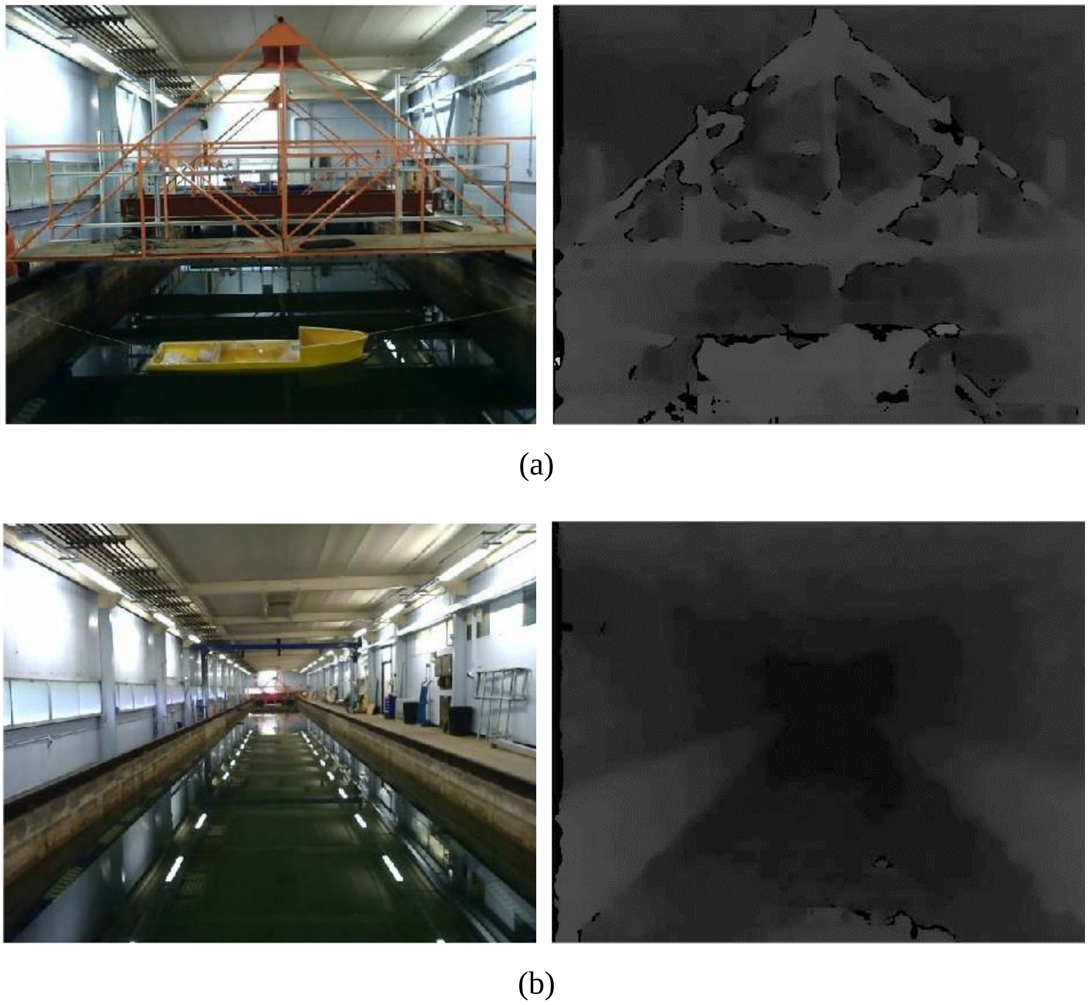


Figure 5.9 The result of detection and distance estimation result of network case 4.

However, it was not able to estimate the distance more than $31.3\ m$. The reason is due to the depth estimation technique, which is built on disparity images. Since the disparity images are drawn based on the texture of the image, wrong disparities can be included due to low texture, low pixel, etc. (Hirschmüller, 2007). As the distance increases then the pixel containing visual information reduces, inaccurate disparities are generated and the accuracy of the distance estimation decreases. The example of disparity images at the distance of $47\ m$ and $3\ m$ is shown in figure 5.10. The plateau between 300 and $400\ sec$ in the cases 1, 2, and 4 (figure 5.6, 5.7, and 5.9) is also explained for the same reason.



(a)

(b)

Figure 5.10 Disparity image. (a) Original frame and disparity image at the distance of $3\ m$. (b) Original frame and disparity image at the distance of $47\ m$.

Except for the detection farther than $31.3\ m$, cases 1, 2, and 4 generally estimated distances close to the actual distance. However, in case 3, only the distance within $5\ m$ was estimated appropriately, and in case 4, excessive wrong detection occurred.

The percentages of the well-detected frame, calculated as in equation 5.3, and mean distance errors excluding the distance of more than $31.3\ m$ are shown in table 5.6. Case 2, the network trained with the same model ship image dataset with the proposed proposal configuration, showed the highest detection performance. On the other hand, a network trained with a small amount of different model ship image dataset scarcely detected the model ship. The mean distance error was the smallest in the default network, case 1.

$$\frac{The\ number\ of\ well-detected\ frames}{Total\ number\ of\ frames} \times 100\ (\%) \qquad (5.3)$$

Table 5.6 Detection test results according to network case[2].

Network case	Total #	Well-detected #	Wrong-detected #	No-detected #	Percentage of well-detected frame (%)	Mean of distance error (m)
Case 1	1392	832	0	560	59.77	**2.36**
Case 2	1350	927	51	372	**68.67**	2.90
Case 3	1376	38	135	1203	2.76	11.34
Case 4	1318	812	438	68	61.61	3.67

1) Result comparison between case 1 and case 2; focusing on proposal configuration

In cases 1 and 2, the dataset image is the same as the model ship used in detection test, and the number of those was large enough. The difference between the two cases was the proposal configuration, where case 2 has more anchor sizes than case 1 and the anchor ratio is closer to the size of the model ship used in this test. The IoU threshold of case 2 was also set to a smaller than case 1.

The percentage of well-detected frame in case 1 and case 2 was 59.77% and 65.85%, respectively. This shows that the modified proposal configuration improved the recognition success rate by 6.08% from the default. The greatest improvement was to detect at a distance more than $25\ m$ as shown in figure 5.11-13. For example, At the

[2] In this paper, '#' refers the number of frames.

time of 231 sec, network case 1 was not able to recognise the model ship, whereas network case 2 recognised it.

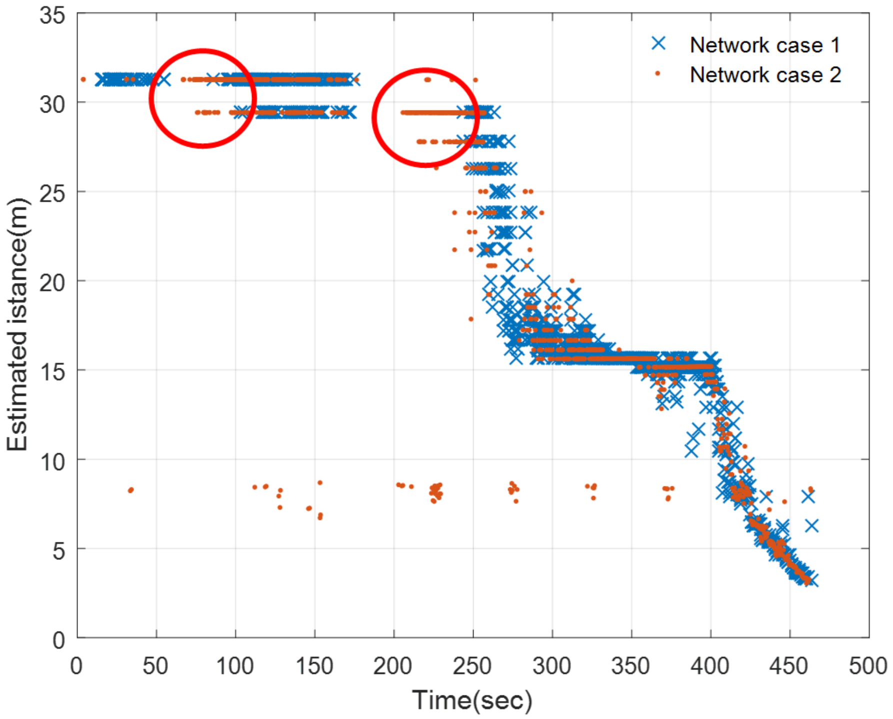


Figure 5.11 Comparison of well-detected distance range between case 1 and case 2. It shows the improvement of detection at the distance more than 25 m as marked in circles.

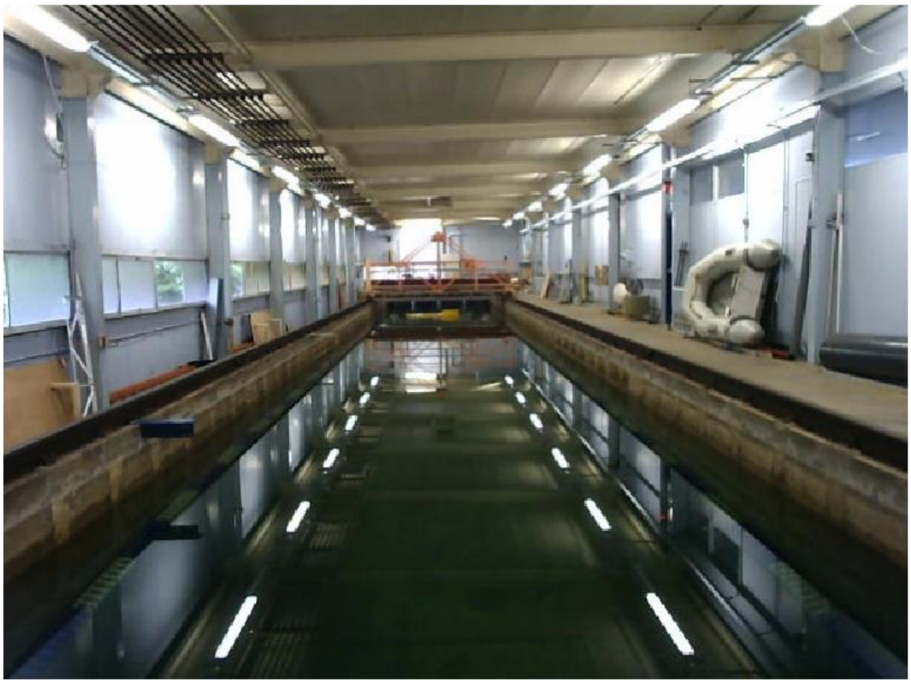


Figure 5.12 Visualisation of distance estimation and object recognition at the time of $231\ sec$ in case 1. Nothing detected.

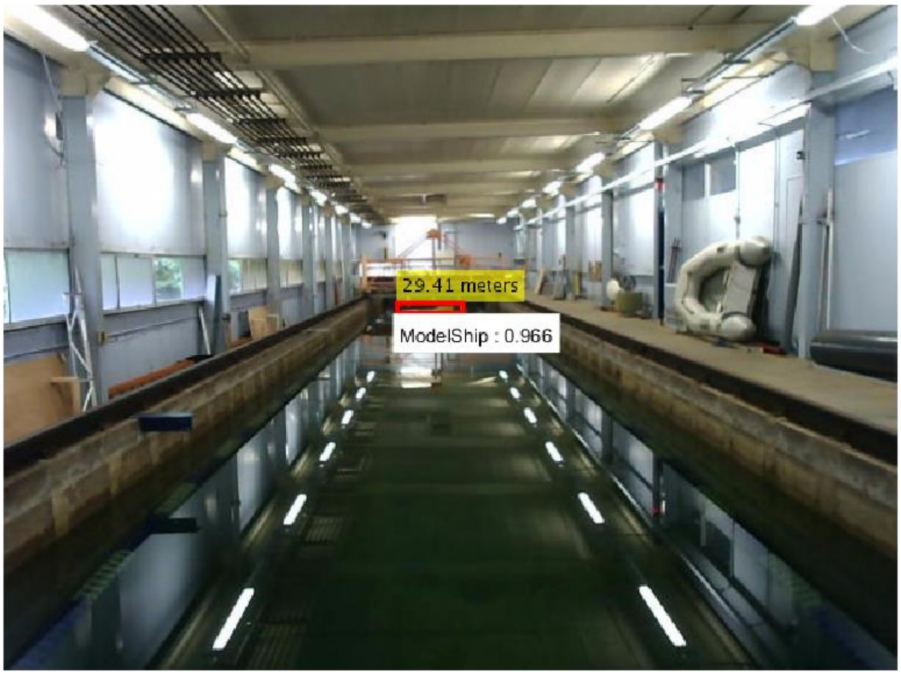


Figure 5.13 Visualisation of distance estimation and object recognition at the time of $231\ sec$ in case 1. The numbers above the bounding box indicate the estimated distance, and the text below indicates the classification result and corresponding matching probability.

However, there was the wrong-detected frame in case 2 as shown in figure 5.14-16. At the time of 277 sec, network case 1 recognised model ship properly, but network case 2 showed wrong recognition result. The percentage of the wrong-detected frame in case 2 was 3.78% higher than case 1. This is why the mean distance error of case 2 is higher than case 1. Nevertheless, as the no-object-detected frame reduced from 40.23% to 27.56%, so that overall model ship recognition success rate increased.

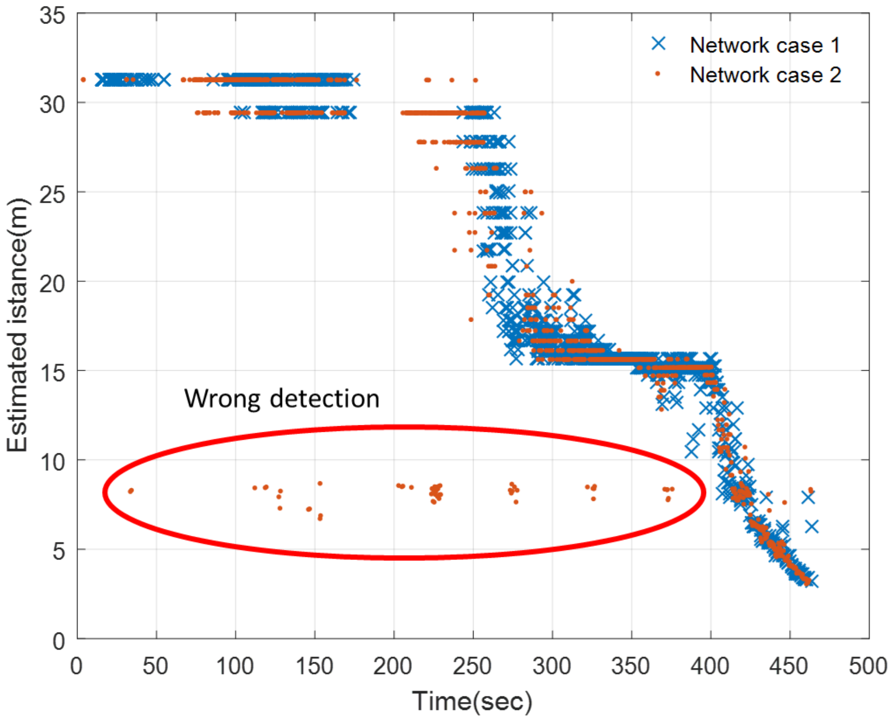


Figure 5.14 Wrong detection in case 2 compared to case 1. The parts where wrong recognition are marked as circles

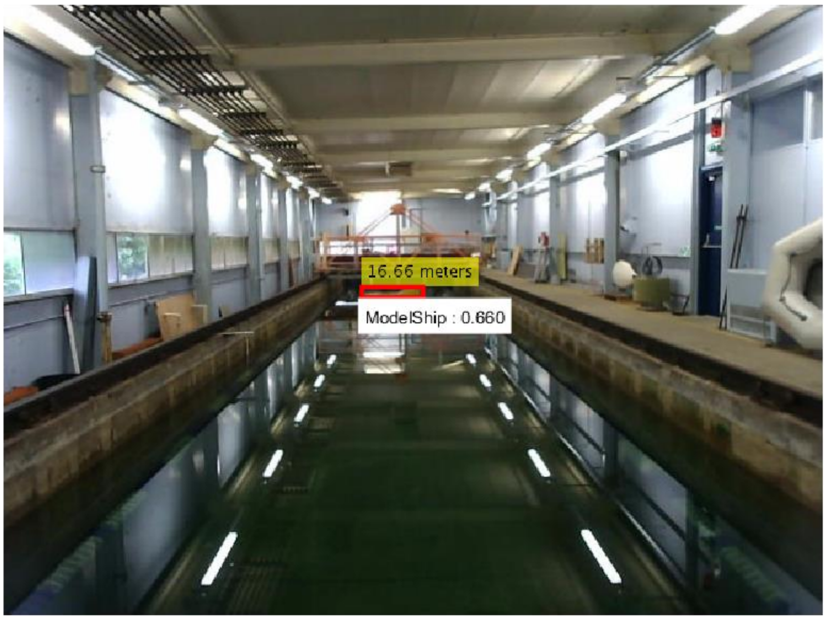


Figure 5.15 Visualisation of distance estimation and object recognition at the time of 277 sec in case 1. The numbers above the bounding box indicate the estimated distance, and the text below indicates the classification result and corresponding matching probability.

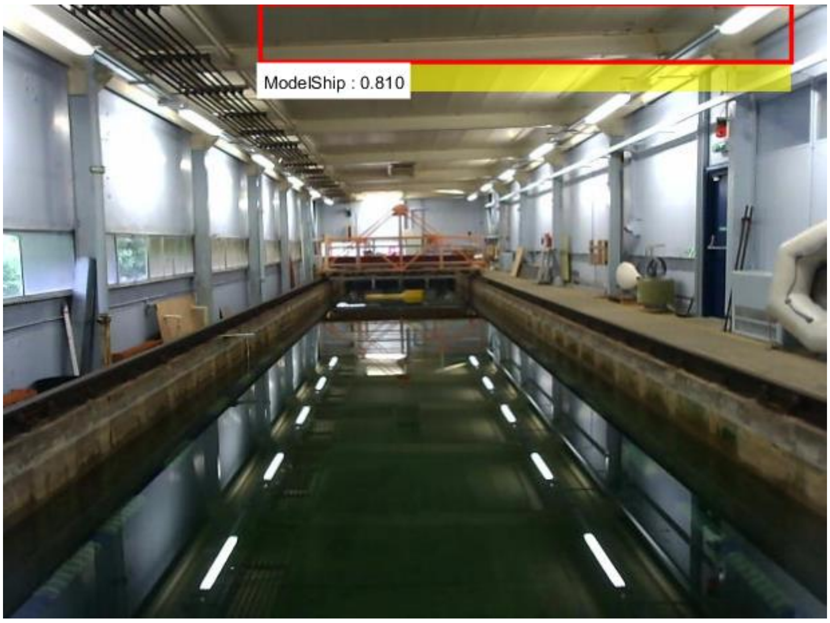


Figure 5.16 Visualisation of distance estimation and object recognition at the time of 277 sec in case 2. The numbers above the bounding box indicate the estimated distance, and the text below indicates the classification result and corresponding matching probability.

2) Result comparison between case 2 and case 3; focusing on dataset quality

Cases 2 and 3 have the same proposal configuration but the dataset image was different. The dataset images in case 2 have mainly consisted of the images of the same model ship that has been used in the detection test. On the other hand, the dataset image in case 3 is composed of images of completely different model ships that have not been used in the detection test. The purpose of arranging the dataset was to test the CNN's strength that it can recognise certain objects even if it has not been trained with the same image. However, there was a limit to collect enough amount of images of different model ships, so that only 303 images were contained in the dataset in case 3.

As shown in figure 5.8, network case 3 did not detect the model ship at the distance more than 6 m, and it misrecognised or did not recognised at all in 97.24% of the frames as described in table 7. This implies that the quality of the dataset has the greatest effect on the performance of the detection algorithm, especially CNN, and its effect is extremely critical.

3) Result comparison between case 2 and case 4; focusing on dataset quantity

Case 2 and case 4 have same proposal configuration and they both were trained with images of the model ship that has been used in the detection test. The difference between them is a number of dataset images, 1034 images for case 2 and 241 images for case 4. The purpose of this comparison is to see the influence of the dataset quantity.

As indicated in table 5.6, network case 4 made a result that a percentage of the well-detected frame is 61.61%, which is 7.6% lower than case 2. From this, it is considered that the amount of dataset affects the CNN. The larger the amount of dataset, it is expected that the better performance of the detection algorithm.

5.8 Summary

Overall, the proposed algorithm was impossible to estimate over a certain distance due to disparity-based calculation in terms of distance estimation. On the object detection side, detection noise was occurred due to false recognition, but changing the proposal configuration showed a slight improvement in performance compared to the default. Due to the characteristics of CNN, the performance of the proposed algorithm was more dominantly influenced by the quality of dataset than the proposal configuration and quantity of dataset.

6 Discussion

The aim of this research was to develop a vision-based detection algorithm for USV. For this, the Faster R-CNN is used to recognise and localise objects on frames, and the depth estimation with a stereo camera is used to estimate the distance to detected objects. In order to evaluate the proposed algorithm and to examine the factors that affect the performance of the algorithm, several case studies were carried out with model ship detection test.

First of all, the average computation time per frame was 0.33 sec, revealing that it is practical for real-time detection. When CNN is trained with high quality and quantity of dataset, it detected the model ship with a probability of almost 70% and the average distance error was within 3 m. Unlike conventional vision-based detection system, the proposed algorithm clarifies the type of object through classification so that it derives additional factors that contribute to the collision risk. It thus seems that it is possible to support the automation of the USV with low cost by simplifying existing expensive equipment.

However, the proposed detection algorithm required a high quality and a large amount of dataset for high performance. In particular, the quality of the dataset has had the greatest impact on the performance. This is due to the nature of artificial intelligence that draws erroneous results when it learns with incorrect information. This is why it needs a large amount of dataset to cover this enough. Another limitation observed in this test is that it was impossible to estimate the distance over $30\ m$. Since the depth estimation computes the disparity based on the texture difference between the left and right frames of the stereo camera, if the texture or resolution is low, the distance estimation is limited. This limitation in this test was because the frame was taken at a resolution of 640×480.

The most important point for the real application of this algorithm is a large amount of high-quality image dataset of marine obstacles. Due to advances in data science, it is expected that organisations providing image databases increases then it will be able to collect these vast amounts of datasets effortlessly in the future. We plan to train the CNN by collecting image datasets of various objects that may exist in actual sea, not model ship, and to make a more powerful detection algorithm by using a high-resolution stereo camera. In addition, since the ultimate goal of the USV detection system is collision avoidance, we plan to devise a method to calculate the collision risk using the information of the type of object, direction to object, and distance to object, which are derived from the proposed algorithm.

7 Conclusion

As automation in various fields receives the spotlight, it is identified that USV in the marine field has also been attracting much attention and has been actively researched. For its complete automation, a secure autopilot is needed, so we have looked at how the USV acquire information about its surrounding that may have the risk of collision. It was necessary to simplify the equipment in terms of economic and light weight of the USV, so we have studied how to handle the information with the equipment. We have confirmed that this can be achieved by using the Faster R-CNN and depth estimation with a stereo camera. This has proven the feasibility of developing an algorithm that simultaneously calculates the type, position, and distance of detected objects differentiated from existing vision-based detection system.

We implemented an object detection algorithm combining Faster R-CNN and depth estimation with a stereo camera. The dataset for Faster R-CNN has been collected as the images of model ship used in detection test and other model ships. The Faster R-CNN has trained with that dataset for its fine-tuning. In this process, we have felt the need for object recognition that occupies a small area on the frame, so that we accordingly modified the existing default configuration of Faster R-CNN and resized the images in the dataset.

In order to examine the efficiency of the proposed algorithm and its influencing factors, Faster R-CNN has been trained in four cases by varying the quality, quantity of dataset, and proposal configuration. Test results have shown that the quality of the dataset has the greatest effect on the performance of the algorithm. In this research, the Faster R-CNN has shown almost 70% recognition ability if such dataset condition is satisfied.

The distance estimation using depth estimation in this test cannot estimate the distance over $30\ m$. This happens because the depth estimation technique computes the disparity based on the texture of the frame. As the distance to the model ship increases, the number of pixels containing the visual information of the area decreases. On the other hand, when estimating the distance within $30\ m$, the average distance error has been only within $3\ m$. Therefore, if the resolution of the camera is high, then the depth estimation technique seems to be well worth applying.

Finally, the average computation time per frame has been $0.33\ sec$ when computing with the above two techniques combined. Therefore, it has been confirmed that there is a possibility of real application if high quality and quantity dataset can be collected and a high-resolution stereo camera is used.

8 Future work

8.1 Improvement of recognition and distance estimation

The proposed algorithm is a combination of Faster R-CNN and depth estimation. Notably, Faster R-CNN is a state-of-the-art image processing technique that imitates animal neuron structures and is constantly being studied to improve its performance and speed. However, it is not simple for users to intuitively optimise its structure and other configurations due to the hidden layer of neural network, which is called the black box, and the processing of a large amount of data with enormously deep layers. Research on the structure development of CNN is carried out by experimenting with various changes of network structure and examining the results. Furthermore, such research requires a lot of high-performance GPU due to the characteristic of CNN that process large amounts of data. Therefore, there is a limitation in developing the structure of CNN or optimising other configurations personally.

However, the CNN structures released so far has no problems in practical use if the quality and quantity of the dataset are high enough. Some institutions have provided image databases for CNN, and the amount of these is steadily growing. Using these databases, CNN can be fine-tuned to recognise more accurately and more diversely. Therefore, we will focus on collecting these datasets rather than optimising configurations that have a relatively small impact on CNN.

In this detection test, it was not able to estimate the distance more than $30\,m.$ As described in chapter 7, this is mainly due to the resolution of the stereo camera. Therefore, we will test with a higher resolution stereo camera in the future. Moreover, it will help improve the classification of CNN by contributing to catch features by including more pixels in the same region.

8.2 Path planning for obstacle avoidance

The main reason for acquiring information from the proposed detection system is to support USV to avoid a collision. The information about the type of detected object and the distance to object obtained by the proposed algorithm can be used in various ways to create a safe path of an unmanned ship as an indicator to calculate the collision risk.

My future research is path planning for USV with the method of machine learning based on the index of collision risk, extending proposed detection system. Since the proposed algorithm obtains the information by the vision sensor, there is a lack of information diversity than a system equipped with a relatively large number of sensors. Therefore, in order to constitute a reliable collision avoidance system, the obtained information is required to be processed advisedly. One idea to solve this problem is to use Q-learning (Watkins and Dayan, 1992), which is a model-free reinforcement learning technique. This is a method to find an optimal solution based on fitness. We plan to implement a path planning algorithm by creating a fitness function considering the type, position, distance information of the object, obtained by the proposed algorithm, and COLREGs.

Reference

AKSELROD-BALLIN, A., KARLINSKY, L., ALPERT, S., HASOUL, S., BEN-ARI, R. & BARKAN, E. A region based convolutional network for tumor detection and classification in breast mammography. International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, 2016. Springer, 197-205.

ALMEIDA, C., FRANCO, T., FERREIRA, H., MARTINS, A., SANTOS, R., ALMEIDA, J. M., CARVALHO, J. & SILVA, E. Radar based collision detection developments on USV ROAZ II. Oceans 2009-Europe, 2009. IEEE, 1-6.

BANDYOPHADYAY, T., SARCIONE, L. & HOVER, F. S. 2010. A Simple Reactive Obstacle Avoidance Algorithm and Its Application in Singapore Harbor. *In:* HOWARD, A., IAGNEMMA, K. & KELLY, A. (eds.) *Field and Service Robotics: Results of the 7th International Conference.* Berlin, Heidelberg: Springer Berlin Heidelberg.

BAO, G.-Q., XIONG, S.-S. & ZHOU, Z.-Y. 2005. Vision-based horizon extraction for micro air vehicle flight control. *IEEE Transactions on Instrumentation and Measurement,* 54**,** 1067-1072.

BERTRAM, V. 2008. Unmanned surface vehicles-a survey. *Skibsteknisk Selskab, Copenhagen, Denmark,* 1**,** 1-14.

BIBULI, M., BRUZZONE, G., CACCIA, M., INDIVERI, G. & ZIZZARI, A. A. Line following guidance control: Application to the Charlie unmanned surface vehicle. Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 2008. IEEE, 3641-3646.

BRADSKI, G. & KAEHLER, A. 2008. *Learning OpenCV: Computer vision with the OpenCV library*, " O'Reilly Media, Inc.".

CACCIA, M., BONO, R., BRUZZONE, G., SPIRANDELLI, E., VERUGGIO, G., STORTINI, A. & CAPODAGLIO, G. 2005. Sampling sea surfaces with SESAMO: an autonomous craft for the study of sea-air interactions. *IEEE robotics & automation magazine,* 12**,** 95-105.

CANNY, J. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence***,** 679-698.

CORNALL, T. & EGAN, G. Calculating attitude from horizon vision. Eleventh Australian International Aerospace Congress, Melbourne, 2005.

CORNALL, T. D. & EGAN, G. Measuring horizon angle from video on a small unmanned air vehicle. 2nd international conference on autonomous robots and agents, 2004. 7.

DALAL, N. & TRIGGS, B. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005. IEEE, 886-893.

DESHPANDE, A. 2016a. The 9 Deep Learning papers You Need To Know About. Available from: https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html [Accessed 7 July 2017].

DESHPANDE, A. 2016b. A Beginner's Guide To Understanding Convolutional Neural Networks. Available from: https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/ [Accessed 7 July 2017].

EGGERT, C., ZECHA, D., BREHM, S. & LIENHART, R. Improving Small Object Proposals for Company Logo Detection. Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, 2017. ACM, 167-174.

ETTINGER, S. M., NECHYBA, M. C., IFJU, P. G. & WASZAK, M. 2003. Vision-guided flight stability and control for micro air vehicles. *Advanced Robotics,* 17**,** 617-640.

EVERINGHAM, M., GOOL, L. V., WILLIAMS, C., WINN, J. & ZISSERMAN, A. 2007. *The PASCAL Visual Object Classes Challenge 2007* [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/ [Accessed 02 September 2017].

FEFILATYEV, S. 2008. Detection of marine vehicles in images and video of open sea.

GIRSHICK, R. Fast r-cnn. Proceedings of the IEEE international conference on computer vision, 2015. 1440-1448.

GIRSHICK, R., DONAHUE, J., DARRELL, T. & MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2014. 580-587.

GLADSTONE, R., MOSHE, Y., BAREL, A. & SHENHAV, E. Distance estimation for marine vehicles using a monocular video camera. Signal Processing Conference (EUSIPCO), 2016 24th European, 2016. IEEE, 2405-2409.

GLOSSER.CA 2013a. Artificial neural network with layer coloring. *In:* NETWORK.SVG, C. N. (ed.).

GLOSSER.CA 2013b. Dependency graph for an artificial neural network. *In:* (GRAPH).SVG, A. D. (ed.).

GRAHAM, B. 2014. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*.

HARIS, S. M., ZAKARIA, M. K. & NUAWI, M. Z. Depth estimation from monocular vision using image edge complexity. Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on, 2011. IEEE, 868-873.

HE, K., ZHANG, X., REN, S. & SUN, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. European Conference on Computer Vision, 2014. Springer, 346-361.

HE, K., ZHANG, X., REN, S. & SUN, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 770-778.

HEIDARSSON, H. K. & SUKHATME, G. S. Obstacle detection and avoidance for an Autonomous Surface Vehicle using a profiling sonar. 2011 IEEE International Conference on Robotics and Automation, 9-13 May 2011 2011a. 731-736.

HEIDARSSON, H. K. & SUKHATME, G. S. Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011b. IEEE, 3160-3165.

HERMANN, D., GALEAZZI, R., ANDERSEN, J. C. & BLANKE, M. 2015. Smart sensor based obstacle detection for high-speed unmanned surface vehicle. *IFAC-PapersOnLine,* 48**,** 190-197.

HIRSCHMULLER, H. Accurate and efficient stereo processing by semi-global matching and mutual information. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005. IEEE, 807-814.

HIRSCHMÜLLER, H. Stereo processing by semi-global matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. Citeseer.

HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W., WEYAND, T., ANDREETTO, M. & ADAM, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

HUANG, J., RATHOD, V., SUN, C., ZHU, M., KORATTIKARA, A., FATHI, A., FISCHER, I., WOJNA, Z., SONG, Y. & GUADARRAMA, S. 2016. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*.

HUBEL, D. H. & WIESEL, T. N. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology,* 160**,** 106-154.2.

HUNTSBERGER, T., AGHAZARIAN, H., HOWARD, A. & TROTZ, D. C. 2011. Stereo vision–based navigation for autonomous surface vessels. *Journal of Field Robotics,* 28**,** 3-18.

JAN, I. U. & IQBAL, N. A new technique for geometry based visual depth estimation for uncalibrated camera. Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on, 2009. IEEE, 286-291.

JIANG, H. & LEARNED-MILLER, E. Face detection with the faster R-CNN. Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on, 2017. IEEE, 650-657.

KANG, K., LI, H., YAN, J., ZENG, X., YANG, B., XIAO, T., ZHANG, C., WANG, Z., WANG, R. & WANG, X. 2017. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*.

KARPATHY, A. 2015. *CS231n Convolutional Neural Networks for Visual Recognition* [Online]. Available: https://cs231n.github.io/convolutional-networks/ [Accessed 27 September 2017].

KHAN, S. & SHAH, M. Object based segmentation of video using color, motion and spatial information. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001. IEEE, II-II.

KONG, T., YAO, A., CHEN, Y. & SUN, F. Hypernet: Towards accurate region proposal generation and joint object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 845-853.

KRIZHEVSKY, A., SUTSKEVER, I. & HINTON, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012. 1097-1105.

LARSON, J., BRUCH, M., HALTERMAN, R., ROGERS, J. & WEBSTER, R. 2007. Advances in autonomous obstacle avoidance for unmanned surface vehicles. SPACE AND NAVAL WARFARE SYSTEMS CENTER SAN DIEGO CA.

LAZEBNIK, S., SCHMID, C. & PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Computer vision and pattern recognition, 2006 IEEE computer society conference on, 2006. IEEE, 2169-2178.

LEE, T.-J., YI, D.-H. & CHO, D.-I. 2016. A Monocular Vision Sensor-Based Obstacle Detection Algorithm for Autonomous Robots. *Sensors,* 16**,** 311.

LEEDEKERKEN, J. C., FALLON, M. F. & LEONARD, J. J. 2010. Mapping complex marine environments with autonomous surface craft.

LEVANDER, O. 2017. Autonomous ships on the high seas. *IEEE Spectrum,* 54**,** 26-31.

LIN, T.-Y., DOLLÁR, P., GIRSHICK, R., HE, K., HARIHARAN, B. & BELONGIE, S. 2016. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*.

LIU, F., SHEN, C., LIN, G. & REID, I. 2016. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence,* 38**,** 2024-2039.

MAJOHR, J., BUCH, T. & KORTE, C. 2000. Navigation and Automatic Control of the Measuring Dolphin (Messin™). *IFAC Proceedings Volumes,* 33**,** 399-404.

MARTINS, A., ALMEIDA, J. M., FERREIRA, H., SILVA, H., DIAS, N., DIAS, A., ALMEIDA, C. & SILVA, E. Autonomous surface vehicle docking manoeuvre with visual information. Robotics and Automation, 2007 IEEE International Conference on, 2007a. IEEE, 4994-4999.

MARTINS, A., FERREIRA, H., ALMEIDA, C., SILVA, H., ALMEIDA, J. M. & SILVA, E. Roaz and roaz ii autonomous surface vehicle design and implementation. International Lifesaving Congress 2007, 2007b.

MCGEE, T. G., SENGUPTA, R. & HEDRICK, K. Obstacle detection for small autonomous aircraft using sky segmentation. Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, 2005. IEEE, 4679-4684.

MIRJALILI, V. 2017. *How des the region proposal network (RPN) in Faster R-CNN work?* [Online]. Available: https://www.quora.com/How-does-the-region-proposal-network-RPN-in-Faster-R-CNN-work [Accessed 27 September 2017].

NAEEM, W. & IRWIN, G. W. 2010. An Automatic Collision Avoidance Strategy for Unmanned Surface Vehicles. *In:* LI, K., LI, X., MA, S. & IRWIN, G. W. (eds.) *Life System Modeling and Intelligent Computing: International Conference on Life System Modeling and Simulation, LSMS 2010, and International Conference on Intelligent Computing for Sustainable Energy and Environment, ICSEE 2010, Wuxi, China, September 17-20, 2010. Proceedings, Part II.* Berlin, Heidelberg: Springer Berlin Heidelberg.

NAEEM, W., SUTTON, R. & CHUDLEY, J. Modelling and control of an unmanned surface vehicle for environmental monitoring. UKACC International Control Conference, 2006. 1-6.

NGUYEN, T. M. & WU, Q. J. 2013. A nonsymmetric mixture model for unsupervised image segmentation. *IEEE transactions on cybernetics,* 43**,** 751-765.

OLSSON, E. & JANSSON, A. 2006. Work on the bridge – studies of officers on high-speed ferries. *Behaviour & Information Technology,* 25**,** 37-64.

ONUNKA, C. & BRIGHT, G. Autonomous marine craft navigation: On the study of radar obstacle detection. 2010 11th International Conference on Control Automation Robotics & Vision, 7-10 Dec. 2010 2010. 567-572.

PASCOAL, A., OLIVEIRA, P., SILVESTRE, C., SEBASTIÃO, L., RUFINO, M., BARROSO, V., GOMES, J., AYELA, G., COINCE, P. & CARDEW, M. Robotic ocean vehicles for marine science applications: the european asimov project. Oceans 2000 MTS/IEEE Conference and Exhibition, 2000. IEEE, 409-415.

PENG, X. & SCHMID, C. Multi-region two-stream R-CNN for action detection. European Conference on Computer Vision, 2016. Springer, 744-759.

POIRSON, P., AMMIRATO, P., FU, C.-Y., LIU, W., KOSECKA, J. & BERG, A. C. Fast single shot detection and pose estimation. 3D Vision (3DV), 2016 Fourth International Conference on, 2016. IEEE, 676-684.

QIN, H., YAN, J., LI, X. & HU, X. Joint training of cascaded CNN for face detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 3456-3465.

RANFTL, R., VINEET, V., CHEN, Q. & KOLTUN, V. Dense monocular depth estimation in complex dynamic scenes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 4058-4066.

RANJAN, R., SANKARANARAYANAN, S., CASTILLO, C. D. & CHELLAPPA, R. An all-in-one convolutional neural network for face analysis. Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on, 2017. IEEE, 17-24.

REDMON, J. & FARHADI, A. 2016. YOLO9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*.

REN, S., HE, K., GIRSHICK, R. & SUN, J. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015. 91-99.

RUIZ, A. R. J. & GRANJA, F. S. 2009. A Short-Range Ship Navigation System Based on Ladar Imaging and Target Tracking for Improved Safety and Efficiency. *IEEE Transactions on Intelligent Transportation Systems,* 10**,** 186-197.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A. & BERNSTEIN, M. 2014. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*.

SAID, Z., SUNDARAJ, K. & WAHAB, M. 2012. Depth estimation for a mobile platform using monocular vision. *Procedia Engineering,* 41**,** 945-950.

SÁNCHEZ, J. & PERRONNIN, F. High-dimensional signature compression for large-scale image classification. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011. IEEE, 1665-1672.

SANTANA, P., MENDONÇA, R. & BARATA, J. Water detection with segmentation guided dynamic texture recognition. Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, 2012. IEEE, 1836-1841.

SAXENA, A., CHUNG, S. H. & NG, A. Y. Learning depth from single monocular images. Advances in neural information processing systems, 2006. 1161-1168.

SCHUSTER, M., BLAICH, M. & REUTER, J. 2014. Collision Avoidance for Vessels using a Low-Cost Radar Sensor. *IFAC Proceedings Volumes,* 47**,** 9673-9678.

SHIN, B.-S., MOU, X., MOU, W. & WANG, H. 2017. Vision-based navigation of an unmanned surface vehicle with object detection and tracking abilities. *Machine Vision and Applications***,** 1-18.

SHRIVASTAVA, A. & GUPTA, A. Contextual priming and feedback for faster r-cnn. European Conference on Computer Vision, 2016. Springer, 330-348.

SIMONYAN, K. & ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

SINISTERRA, A. J., DHANAK, M. R. & VON ELLENRIEDER, K. Stereo vision-based target tracking system for an USV. Oceans-St. John's, 2014, 2014. IEEE, 1-7.

SOCEK, D., CULIBRK, D., MARQUES, O., KALVA, H. & FURHT, B. 2005. A hybrid color-based foreground object detection method for automated marine surveillance. *Lecture notes in computer science,* 3708**,** 340.

SPRINGENBERG, J. T., DOSOVITSKIY, A., BROX, T. & RIEDMILLER, M. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

STOWELL, D. 2015. Rectifier and softplus functions. *In:* FUNCTIONS.SVG, R. A. S. (ed.).

SUN, X., WU, P. & HOI, S. C. 2017. Face detection using deep learning: An improved faster rcnn approach. *arXiv preprint arXiv:1701.08289*.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V. & RABINOVICH, A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015. 1-9.

SZYMKOWIAK, T. 2017. Compilation of free resourcese to get started with Artificial Intelligence quickly. *Medium* [Online]. Available from: https://medium.com/@thoszymkowiak/compilation-of-free-resources-to-get-started-with-artificial-intelligence-24a48bdb4357 [Accessed 20 Feburary.

TODOROVIC, S. 2002. *Statistical modeling and segmentation of sky/ground images.* University of Florida.

TODOROVIC, S. & NECHYBA, M. C. 2004. A vision system for intelligent mission profiles of micro air vehicles. *IEEE Transactions on Vehicular Technology,* 53**,** 1713-1725.

TODOROVIC, S., NECHYBA, M. C. & IFJU, P. G. Sky/ground modeling for autonomous MAV flight. Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, 2003. IEEE, 1422-1427.

TOLIAS, G., SICRE, R. & JÉGOU, H. 2015. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879*.

UIJLINGS, J. R., VAN DE SANDE, K. E., GEVERS, T. & SMEULDERS, A. W. 2013. Selective search for object recognition. *International journal of computer vision,* 104**,** 154-171.

WAHAB, M., SIVADEV, N. & SUNDARAJ, K. Development of monocular vision system for depth estimation in mobile robot—Robot soccer. Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2011 IEEE Conference on, 2011. IEEE, 36-41.

WANG, H. & WEI, Z. Stereovision based obstacle detection system for unmanned surface vehicle. Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on, 2013. IEEE, 917-921.

WANG, H., WEI, Z., OW, C. S., HO, K. T., FENG, B. & HUANG, J. Improvement in real-time obstacle detection system for USV. Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on, 2012. IEEE, 1317-1322.

WANG, H., WEI, Z., WANG, S., OW, C. S., HO, K. T. & FENG, B. A vision-based obstacle detection system for unmanned surface vehicle. Robotics, Automation and Mechatronics (RAM), 2011 IEEE Conference on, 2011a. IEEE, 364-369.

WANG, H., WEI, Z., WANG, S., OW, C. S., HO, K. T., FENG, B. & LUBING, Z. Real-time obstacle detection for unmanned surface vehicle. Defense Science Research Conference and Expo (DSR), 2011, 2011b. IEEE, 1-4.

WANG, X., YANG, M., ZHU, S. & LIN, Y. Regionlets for generic object detection. Proceedings of the IEEE International Conference on Computer Vision, 2013. 17-24.

WATKINS, C. J. & DAYAN, P. 1992. Q-learning. *Machine learning,* 8**,** 279-292.

WOLF, M. T., ASSAD, C., KUWATA, Y., HOWARD, A., AGHAZARIAN, H., ZHU, D., LU, T., TREBI-OLLENNU, A. & HUNTSBERGER, T. 2010. 360-degree visual detection and target tracking on an autonomous surface vehicle. *Journal of Field Robotics,* 27**,** 819-833.

WOO, J. & KIM, N. 2016a. Vision-based target motion analysis and collision avoidance of unmanned surface vehicles. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment,* 230**,** 566-578.

WOO, J. & KIM, N. Vision based obstacle detection and collision risk estimation of an unmanned surface vehicle. Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on, 2016b. IEEE, 461-465.

XIAO, T., LI, S., WANG, B., LIN, L. & WANG, X. Joint detection and identification feature learning for person search. Proc. CVPR, 2017.

ZEILER, M. D. & FERGUS, R. 2013. Visualizing and understanding convolutional networks. *arXiv preprint arXiv:1311.2901*.

ZHANG, H., KYAW, Z., CHANG, S.-F. & CHUA, T.-S. 2017. Visual translation embedding network for visual relation detection. *arXiv preprint arXiv:1702.08319*.

ZHANG, Y., WEI, X.-S., WU, J., CAI, J., LU, J., NGUYEN, V.-A. & DO, M. N. 2016. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Transactions on Image Processing,* 25**,** 1713-1725.

ZHONG, Z., JIN, L., ZHANG, S. & FENG, Z. 2016. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv preprint arXiv:1605.07314*.

ZHU, C., ZHENG, Y., LUU, K. & SAVVIDES, M. 2017. CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection. *Deep Learning for Biometrics.* Springer.

ZHU, Z., LIANG, D., ZHANG, S., HUANG, X., LI, B. & HU, S. Traffic-sign detection and classification in the wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2110-2118.

Appendix

A Faster R-CNN

In recent years, due to the improvement of data science and computer performance, deep learning that requires a large number of datasets is attracting attention. Deep learning is a field of machine learning, and it enables to perform a task by learning a large number of datasets. The architecture of this has been applied to various fields such as computer vision, speech recognition, natural language processing, and machine translation, by replacing what human experts are performing. Among them, CNN is a specialised architecture for image processing, which classifies objects on the screen. However, in order to utilise it practically, it was necessary to not only classify but also localise the object on the screen. This led to the emergence of R-CNN, which was able to identify object location as well. However, this had also a limitation in applying to real-time object detection due to excessive computing time. As a result, Faster R-CNN has emerged as an algorithm for real-time object detection that dramatically shortens computing time. This chapter describes the ANN, CNN, R-CNN, Fast R-CNN and RPN that make up the Faster R-CNN.

A.1 ANN

ANN is a computing system that mimics the neuron structure of the animal brains. Unlike other systems that are artificially programmed to perform a task, ANNs are programmed themselves by learning various examples of the task. This has the advantage of being able to program much easier than conventional programming if it is difficult to create rules by hand.

An ANN consists of a collection of units called artificial neurons, which are interconnected. They transmit signals to other neurons, and then the neurons that receive the signal transmit the processed signal to another neuron. Neurons have their own unique state, usually represented by real numbers, and they have their own weight formed by the learning process. These weights serve to reinforce or weaken the signal. Neurons are normally arranged in layers, and depending on the type of layer, the methods of transforming the input signal are different. The signals are passed in succession from the first layer to the last layer in one direction as shown in figure A.1.

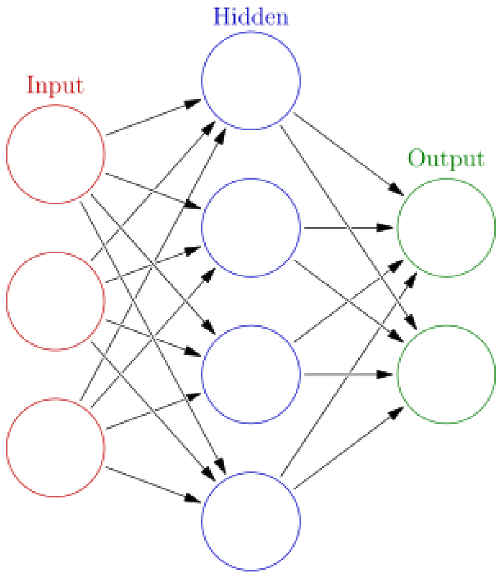


Figure 8.2.1 ANN structure (Glosser.ca, 2013a).

The important elements of ANN are neurons, weights, propagation functions and learning rules. A neuron represented by j that receives input $p_j(t)$ from upper neuron contains several components. This neuron has activation $a_j(t)$ according to the discrete time parameter. It also has a fixed threshold value θ_j if it is not changed by the learning function. An activation function f computes a new activation for time $t + 1$ from $a_j(t)$, θ_j and input $p_j(t)$ with the relation as following equation A.1.

$$a_j(t+1) = f\left(a_j(t), p_j(t), \theta_j\right) \qquad \text{(A.1)}$$

An output function f_{out} computes the output from the activation as follows:

$$o_j(t) = f_{out}\left(a_j(t)\right) \qquad \text{(A.2)}$$

At neuron connection, the output of the upper neuron i is passed to the input of the lower neuron j, and each of connection is allocated a weight ω_{ij}. The propagation function computes the input $p_j(t)$ to be passed to the neuron j, from the output $o_j(t)$ of the upper neuron, and it is normally expressed as follows:

$$p_j(t) = \sum_i o_i(t)\omega_{ij} \qquad \text{(A.3)}$$

The learning rule is a rule for producing the output of a network in preferred form, usually by modifying weights and thresholds.

Neural network models can be represented by a simple mathematical model defined as $f: X \rightarrow Y$. Mathematically, the network function $f(x)$ is defined from other functions $g_i(x)$, and these $g_i(x)$ functions are also defined from other functions, as visualised in figure A.2. This is expressed in the form of a nonlinear weighted sum as $f(x) = K(\sum_i \omega_i g_i(x))$, where K is a predefined function called activation function such as hyperbolic tangent or sigmoid function. This activation function serves to smooth the change in the input value.

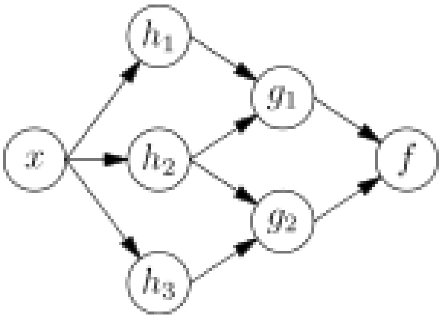


Figure 8.2.2 ANN dependency (Glosser.ca, 2013b).

In such a neural network, given a class of function F, learning refers to the use of observations to find $f^* \in F$ that solves given task in terms of optimisation. This involves defining the cost function $C: F \rightarrow \mathbb{R}$ for the optimal solution f^*, and the cost function depends on the task. Another learning feature of the neural network is backpropagation. Backpropagation is a technique to calculate the gradient of the loss function using the weights in ANN. Backpropagation weight updates are performed through a stochastic gradient descent as follows:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \frac{\partial C}{\partial \omega_{ij}} + \xi(t) \qquad \text{(A.3)}$$

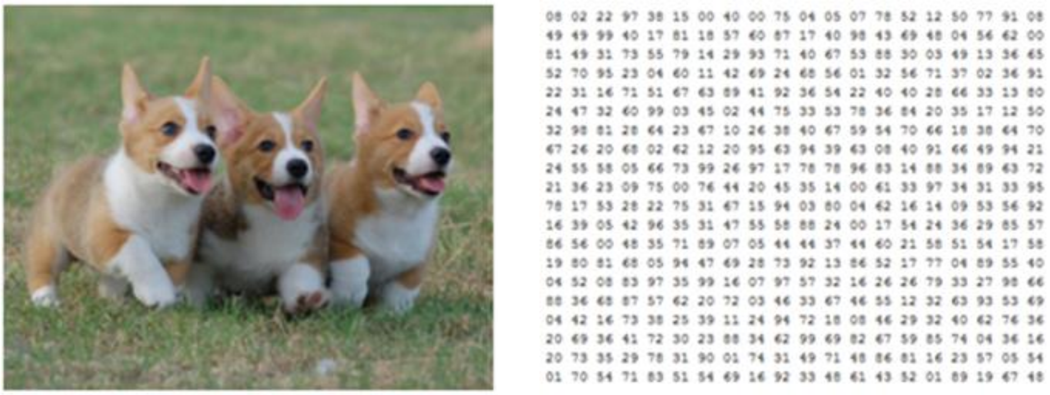
where η is the learning rate, C is the cost function and $\xi(t)$ is a stochastic term. The cost function depends on the learning type and the activation function.

A.2 CNN

CNN is a specialised network for image classification based on the basic concepts of ANN and derives a class of image as output from an input image. The pioneering model of CNN was published in 2012 by Alex Krzheevsky, Ilya Sutskever, and Geoffrey Hinton, as the name of AlexNet (Krizhevsky et al., 2012). They designed a "large, deep convolutional neural network" and then won the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) (Russakovsky et al., 2014) by using that. ILSVRC is an annual worldwide competition of computer vision, to see which computer vision model is the best for classification, localisation, detection, etc. Through the AlexNet, a top 5 test error rate for classification was reduced down to 16.4%, which was astonishingly less than the best (Sánchez and Perronnin, 2011) in the previous competition, 2011 ILSVRC, which was 25.8%. (Russakovsky et al., 2014)

CNN captured a biological idea from visual cortex that receives visual information in the brain. In the experiment conducted by Hubel and Wiesel in 1962 (Hubel and Wiesel, 1962), some individual neuronal cells only reacted to the appearance of edges in the scene, which means they respond particular visual component.

For humans, when they classify an object, the input is the scene that they look at, and the output is their judgment what the object is. Humans classify objects with identifiable features such as two legs of a human, four wheels of a car, etc. They instantly label objects from the input based on the generalised visual pattern they learned, despite different image environment. This ability of object recognition is typically formed by learning from what they look at around them as growing up naturally. On the other hands, the computer accepts an image as an array of pixel intensity. The array is represented as $width \times height \times depth$, and each number is represented from 0 to 255 which illustrates the pixel intensity at each point, as shown in figure A.3. Width and height depend on the pixel the image has, and depth is generally 3, each of them demonstrates the intensity of red, green and blue respectively. In a computer, the concept of classification is outputting probability of the class of the image via the series of convolutional layers, where low-level features such as edges and curves are extracted first and then more abstract features are derived.



```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

(a) (b)

Figure 8.2.3 Visual perception (Deshpande, 2016b). (a) What human see. (b) What computers see.

A.2.1 CNN Architecture

The convolutional neural network is characterised by spatial processing of data, unlike regular neural network. A regular neural network receives a single vector input, processes it, and sends it to the next layer, a series of hidden layers. All neurons in the hidden layer are fully connected to all neurons in the previous layer, and each neuron in one layer is absolutely independent of each other as shown in figure A.4.

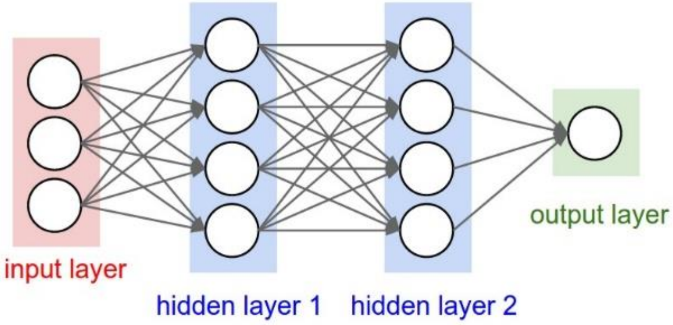


Figure 8.2.4 Typical neural network (Szymkowiak, 2017).

On the other hand, in a convolutional neural network, data is arranged in 3 dimensions with width, height, and depth, where depth represents the third dimension of the volume as shown in figure A.5. For example, an image with 32×32 pixel with three colours is represented by a volume of $32 \times 32 \times 3$. The neurons in the layer are not fully connected but are connected only to a small area of the previous layer. Besides, by continuing to reduce width and height during the forward pass, the final result is output as a single vector of $1 \times 1 \times n$ representing the class score, where the n is a final depth that is different for network structure.

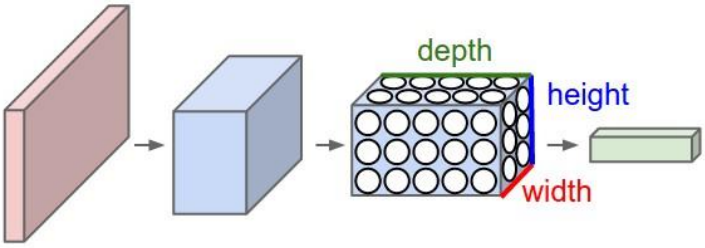
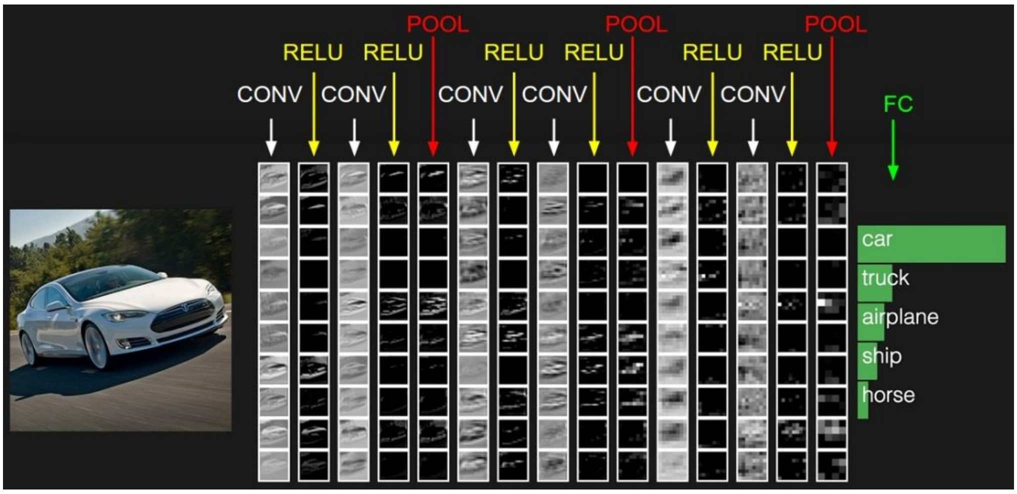
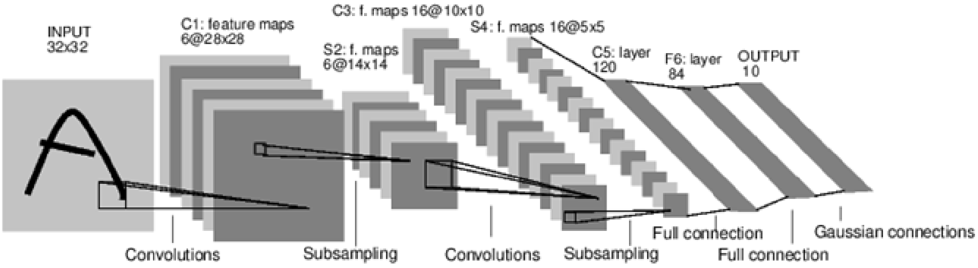


Figure 8.2.5 Regular 3-layer neural network (Karpathy, 2015).

The structure of CNN is formed by stacks of different layers that convert the input volume to the output volume visualised in figure A.6. These CNN layers generally include a convolutional layer, a polling layer, a ReLU layer, a fully connected layer, and a loss layer.



(a)



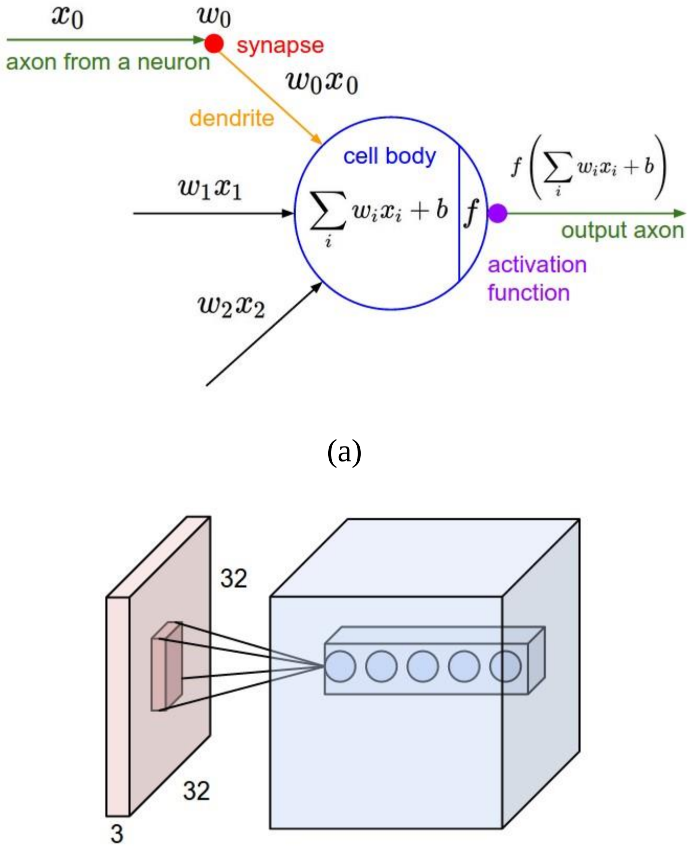
(b)

Figure 8.2.6 Convolutional neural network architecture. (a) Visualisation of activation and volume's slices (Karpathy, 2015). (b) Visualisation of activation and volume's slices of LeNet (Deshpande, 2016b).

A.2.2 Convolutional Layer

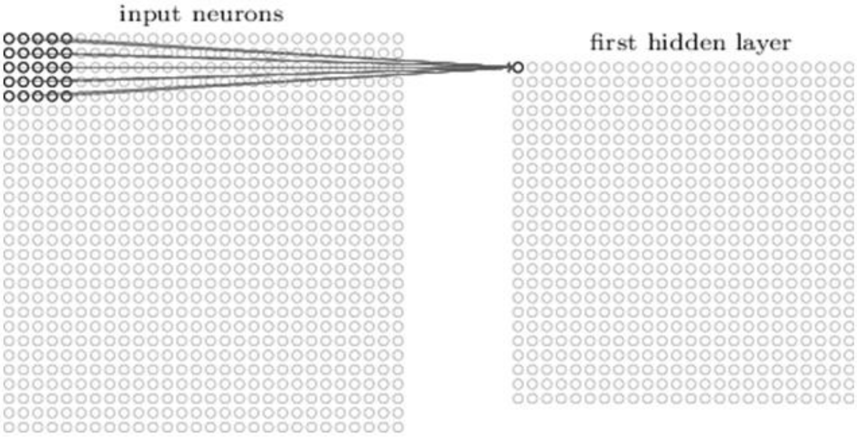
The convolutional layer is a fundamental building block of CNN and the first layer of CNN is always the convolutional layer. This layer has a parameter called filter, which is learnable and has a small receptive field. This filter has its own unique weight and generates a 2-dimensional activation map of the filter by calculating the dot product between filter and input by convolving the entire volume of the input volume during the forward pass. During convolving, a filter with a larger weight creates an activation map with a larger value of the receptive field. As a result, these filters only activate for a particular feature in the space of the input volume. This convolving process is repeated for the number of filters, and each activation map generated in this process is stacked in the direction of the depth dimension. These activation maps, stacked by all filters, form the output volume of the convolutional layer.

The computing process in this convolutional layer is distinguished from other common neuron connection structures, that is, local connectivity. When processing high-dimensional input volume such as an image, connecting all the upper and lower neurons is inefficient because the spatial structure of the data is not taken into consideration. The convolutional layer has overcome the inefficiency of existing regular neuron connection by connecting each neuron only to the local region of the input volume as shown in figure A.7. The spatial extent of this connectivity is expressed as the receptive field of the neuron.



(a)

(b)



(c)

Figure 8.2.7 Local connectivity. (a) Existing ANN that does not consider local connectivity (Karpathy, 2015). (b) CNN considered local connectivity (Karpathy, 2015). (c) CNN considered local connectivity (Deshpande, 2016b).

Another feature of the convolutional layer is how to arrange the data spatially. The convolutional layer controls the size of output volume with three parameters: depth, stride, and zero-padding.

The depth is the size of the output volume in the third dimension and corresponds to the number of user-defined filters. It controls the number of neurons in a layer that is connected to the same region of the input volume. The stride defines how many pixels the filter moves at a time during convolving and adjusts the spatial dimension of the output volume in the width and height directions. The zero-padding adds a value of 0 outside the boundary of the input volume, which also controls the width and height of the output volume. The spatial size of the output volume is calculated as a function of the input volume size W, the kernel field size K, stride S, and zero padding size P. The formula for calculating the number of neurons for a given volume is

$$(W - K + 2P)/S + 1 \qquad \text{(A.4)}$$

If the number calculated by this formula is not an integer, the stride is erroneously set and the neuron cannot be tiled to fit the input volume.

Another feature of the convolutional layer is that it shares parameters. This reduces the number of free parameters that are generated excessively. The parameter sharing is based on a reasonable assumption that if a particular patch feature is useful for computing at a specific spatial location, then this patch feature is also useful in other locations. That is, each filter has the same weight and bias. Therefore, since all neurons in a filter share the same parameters, each filter computes by convolving its weight to the input volume during the forward pass. As a result of this convolution, activation maps are generated, and these maps, which are created with different filters, are stacked all together to depth dimension to create output volume.

A.2.3 Pooling Layer

The basic concept of the pooling is that the precise location of the features is not important, and only the approximate relative positions need to be matched. The polling layer reduces the spatial size of the input volume and reduces the parameters and computation of the network. Most commonly, max pooling is used, which samples the maximum value at each divided part of the single depth slice of the input volume as shown in figure A.8.

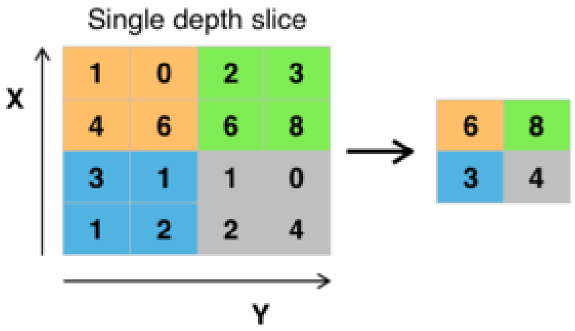


Figure 8.2.8 Example of max pool with a 2 × 2 filter and a stride of 2 (Deshpande, 2016b).

Pooling is applied independently to all depth slices in the input volume. The most common form is a polling layer with a filter size of 2 × 2 and stride of 2, which abandon 75% of the activation. Due to the elimination of data that may be used, there is a tendency to use smaller filters (Graham, 2014) or not to use the pooling layer itself (Springenberg et al., 2014). However, pooling is an important component in the Fast R-CNN architecture, which is part of the main algorithm of this study.

A.2.4 ReLU Layer

ReLU stands for Rectified Linear Units, where the non-saturating function $f(x) = max(0, x)$, as shown in figure A.9, is applied. It enhances the nonlinearity of the decision function and the whole network without involving the receptive field of the convolutional layer. Other activation functions include hyperbolic tangent and sigmoid function, but ReLU is preferred because of the low penalty of training speed and accuracy.

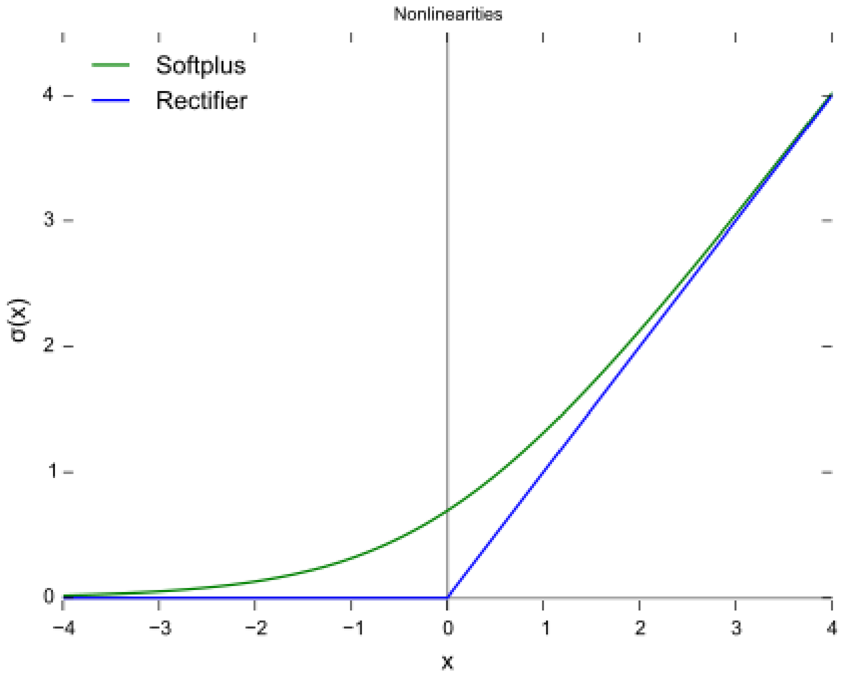


Figure 8.2.9 Rectifier and softplus functions (Stowell, 2015).

A.2.5 Fully Connected Layer

Once the input volume passes through several convolutional and max pooling layers, the final high-level features are created. These higher-level features are fed into a fully connected layer where inferences are made. All neurons in this layer are connected with all activations of the previous layer, and these activations are calculated on the same principle as normal ANN.

A.2.6 Loss Layer

The loss layer is the final layer of the CNN architecture, which stipulates how to penalise the deviation between the predicted and actual labels. The loss function applied to this layer is appropriately selected according to the task. These loss functions include the sigmoid cross-entropy loss used to predict independent probabilities in $[0, 1]$, the Euclidean loss used to regress to the real-value label $(-\infty, \infty)$, and softmax loss used to predict a single class among classes. In this paper, Fast R-CNN uses softmax loss and its function is given as follows:

$$\sigma(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k-1}^{K} e^{z_k}} for j = 1, 2, \ldots, K. \tag{A.4}$$

where $\boldsymbol{z}$ is a K-dimensional vector of arbitrary real values, and $\sigma(\boldsymbol{z})$ is a K-dimensional vector of real values.

A.2.7 Training of CNN

A training process follows the idea of how humans learn to classify objects. For humans, when they were born, they cannot distinguish objects at all due to the inability of feature cognition and previous visual knowledge. As they grow up, by learning the corresponding labelled image, they obtain the ability of classification by observing the features of objects such as colour, edge, the shape of a curve, etc. Likewise, before the training of network, it doesn't output the correct probability of corresponding class due to randomly initialised filters, which support feature cognition. Through a training process, the filters are optimised to catch appropriate features using given input images and corresponding label. This process for CNN is called backpropagation. Backpropagation consists of four parts, the forward pass, the loss function, the backward pass and the weight update. During the forward pass, the array of numbers of input image passes through the entire network. It doesn't output acceptable probability of class at first due to randomised filter.

A.2.8 Loss Function in Training

Generally, a loss function is defined as following MSE.

$$E_{total} = \sum \frac{1}{2}(target - output)^2 \tag{A.5}$$

In order to achieve the correct classification result with an equal label corresponding to given image, the weights are required to be modified when it outputs unexpected classification result. The loss value from the loss function is utilised to modify the weights.

A.2.9 Backward Pass in Training

During a backward pass, the contribution from weight to loss is calculated as dL/dW, where L is loss and W is weight. It determines how much the weight will be updated. Afterwards, weights are updated based on the contribution of each weight, dL/dW, as the following equation.

$$w = w_i - \eta \frac{dL}{dW} \qquad \text{(A.5)}$$

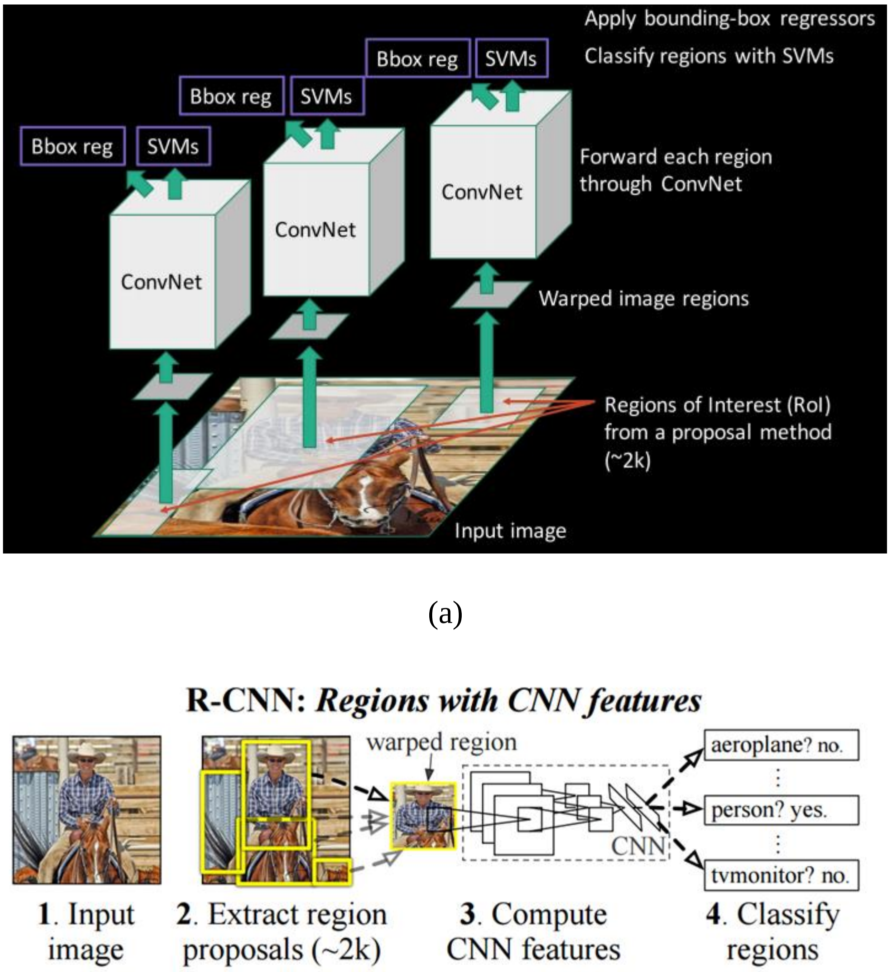
where w is weight, w_i is initial weight, and η is learning rate. The learning rate, η, is a manual parameter that determines the convergence speed of weight. If it is high, the weights are changed rapidly, but the final optimised weight is not accurate. These processes constitute one iteration of training, and it is repeated by the number that programmer sets.

From this basic CNN structure, several groups developed advanced structures such as AlexNet (Krizhevsky et al., 2012), ZF net (Zeiler and Fergus, 2013), VGG Net (Simonyan and Zisserman, 2014), GoogleNet (Szegedy et al., 2015) and ResNet (He et al., 2016), in order to improve classification performance. The performance of these structures was evaluated by Simonyan et al.(Simonyan and Zisserman, 2014)

A.3 R-CNN

R-CNN (Girshick et al., 2014) was first introduced in 2013, by Ross Girshick, Jeff, Trevor Darrell and Jitendra Malik. It does not only classify but also localise the objects in an image with a bounding box. The R-CNN consists of three modules: independent region proposals, large convolutional neural network, and a set of class-specific linear SVMs. In the region proposals module, it extracts candidate sets that the detector can detect. In the convolutional neural network module, a feature vector with a fixed-length is extracted through the forward pass of the input volume. In the linear SVMs module, it finally classifies the image. In region proposal process, Selective Search (Wang et al., 2013, Uijlings et al., 2013) is used to extract the region that contains the object with high probability. It generates 2000 independent region proposals from an image and extracts the feature vector from proposals using CNN. In classification process, it classifies each region proposed from Selective Search with linear SVM. The feature vector is applied bounding box regressor to reap precise coordinates as well.

Non-maximum suppression is applied to restrain the worthless bounding boxes that overlap the region of IoU in order to display one bounding box for one object. The R-CNN workflow is shown in figure A.10.


Apply bounding-box regressors
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image
R-CNN: Regions with CNN features
warped region
aeroplane? no.
person? yes.
tvmonitor? no.
CNN
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

(a)

(b)

Figure 8.2.10 R-CNN workflow (Deshpande, 2016a).

A.4 Fast R-CNN

The R-CNN (Girshick et al., 2014) has succeeded in locating objects on the screen, but it had a major drawback (Girshick, 2015). First, training is performed in a multi-stage pipeline. The R-CNN first fine-tunes the convolutional network for objects proposals. The SVMs then is fitted to the features of this convolutional network. These SVMs function as object detectors, and the final step is to learn bounding-box regressor. Second, training takes a long time and requires a lot of storage space. For each object proposal in each image, the feature is extracted and recorded on a disk. For deep networks such as VGG16, training for 5000 images takes about 2.5 days on a GPU and requires hundreds of gigabytes of storage. Third, object detection is slow. Detection with VGG16 takes 47 seconds per image based on GPU.

In order to alleviate the speed problem, SPPnets (He et al., 2014) was proposed that share a portion of the computation. The SPPnet does not extract the convolutional feature map for the object proposals, but preferentially extracts the convolutional feature map for the entire input image, and then classifies the object proposals from this feature map. Higher-level features are extracted from this feature map into a fixed feature map with the size of 6 × 6 through max pooling. These pooled feature maps are constructed as in spatial pyramid pooling (Lazebnik et al., 2006). The SPPnet reduced test time of R-CNN by 10 to 100 times and training time by 3 times, but training was still multi-stage pipeline and learning of preceding layer of spatial pyramid pooling was impossible. Due to the fixation of this first convolutional layer, recognition accuracy has been limited, so Ross Girshick has developed Fast R-CNN that improves both accuracy and speed to solve these problems of R-CNN and SPPnet.

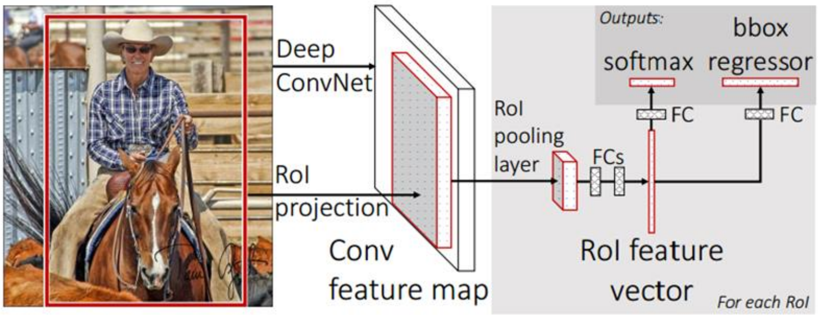


Figure 8.2.11 Fast R-CNN workflow (Girshick, 2015).

The architecture of Fast R-CNN is shown in figure A.11. In training of the Fast R-CNN, like the SPPnet, it first extracts the convolutional feature map for the entire image. Afterwards, In RoI pooling layer, it extracts the fixed-length feature vector from RoI projected on the convolutional feature map. The extracted vectors then fed into a series of fully connected layers, and enter to softmax and bounding-box regressor, respectively. The softmax estimates the probability of each object class including the background class, and the bounding-box regressor extracts four real numbers that optimise the bounding-box for the class.

A.4.1 RoI Pooling Layer

In the RoI pooling layer, it transforms the projected RoI into a fixed-size small feature map of size ($H \times W$) by max pooling, where the H and W are the height and width of pooled feature map respectively. It produces the output by dividing this RoI window with sub-window of size ($h/H \times w/W$) and calculating each corresponding sub-windows, where the h and w are the height and width of projected RoI (He et al., 2014).

A.4.2 Fine-tuning

The shortcoming of SPPnet and R-CNN is that back-propagation is extremely inefficient because each RoI comes from a different image, which means it processes vast amounts of data during training. In the Fast R-CNN, it increases efficiency by sharing features, where SGD mini batches are sampled hierarchically. It samples N images and then samples R/n RoIs from each image, from which RoIs of the same image share memory in forwarding and backwarding passes. In addition, instead of training softmax classifier, SVMs, and regressors in three divided stages, Faster R-CNN performs simplified training with only one stage to optimise softmax classifier and bounding-box regressors mutually (Deshpande, 2016a).

A.5 Faster R-CNN

The experiment result of Fast R-CNN disregarded the time spent on region proposal, so it was still unsatisfactory to apply to real-time object detection. The main factor of this problem was region proposal that is overburdened in computation. Shaoqing Ren, Kaiming He, Ross Girshik and Jian Sun solved it by inserting an RPN instead of Selective Search after the last convolutional layer (Ren et al., 2015) as shown in figure A.12. The Faster R-CNN reduced running time by sharing the convolutional computations in contrast with Fast R-CNN that conduct region proposal separately (Deshpande, 2016a).

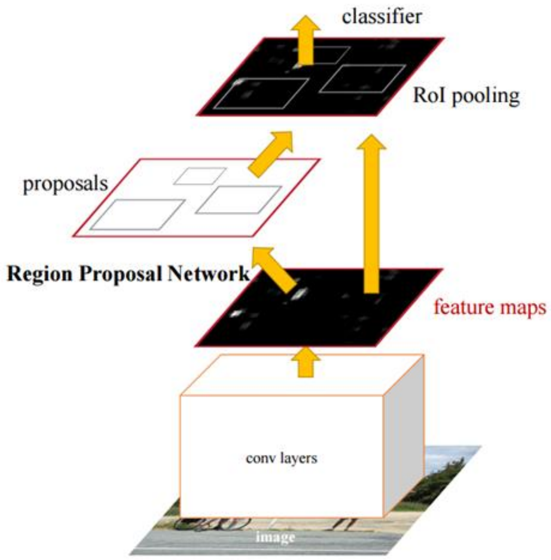


Figure 8.2.12 Faster R-CNN workflow (Deshpande, 2016a).

A.5.1 RPN

In the first step of the RPN, the images are fed into a network and a set of convolutional feature maps output. Thereafter, a sliding window is prepared to spatially explore these feature maps. Each sliding window has 9 anchors that have 3 different aspect ratio and scales as shown in figure A.13. Then a sliding window slides over these feature maps.

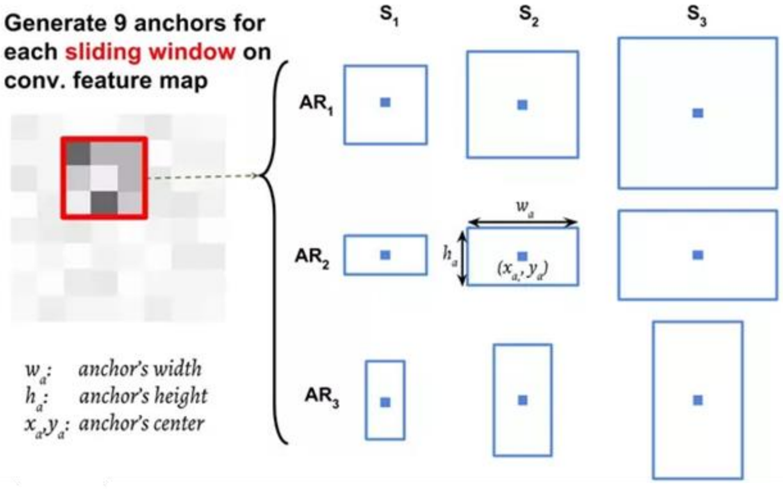


Figure 8.2.13 Generated anchors for one sliding window (Mirjalili, 2017).

During this, the sliding widow computes the value, p^*, which refers how much these anchors overlap with the ground-truth bounding box as follow.

$$p^* = \begin{cases} 1 & if \quad IoU > 0.7 \\ -1 & if \quad IoU < 0.3 \\ 0 & \quad otherwise \end{cases} \tag{A.6}$$

IoU is the overlapped region between anchor and ground-truth box and defined as follows.

$$IoU = \frac{(Anchor) \cap (Ground\ truth\ box)}{(Anchor) \cup (Ground\ truth\ box} \tag{A.7}$$

After this, spatial features extracted by sliding window from feature maps go through the network that performs classification and regression. The regressor outputs a predicted bounding box, and the classifier computes the probability for each box if it includes the object or not as shown in figure A.14 (Mirjalili, 2017).

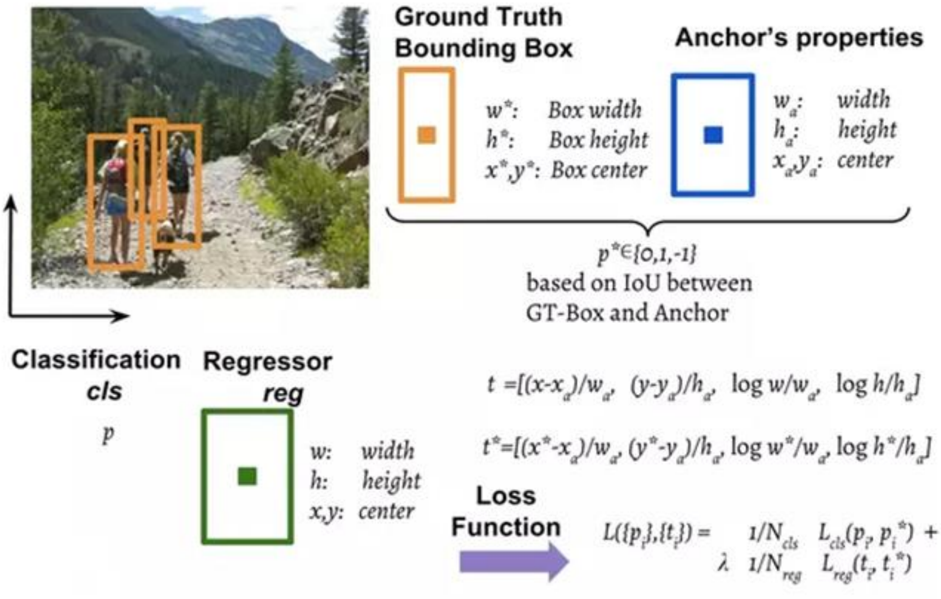

Ground Truth Bounding Box
w*: Box width
h*: Box height
x*,y*: Box center
Anchor's properties
w_a: width
h_a: height
x_a,y_a: center
p*∈{0,1,-1}
based on IoU between GT-Box and Anchor
Classification
cls
p
Regressor
reg
w: width
h: height
x,y: center
t =[(x-x_a)/w_a, (y-y_a)/h_a, log w/w_a, log h/h_a]
t*=[(x*-x_a)/w_a, (y*-y_a)/h_a, log w*/w_a, log h*/h_a]
Loss Function
L({p_i},{t_i}) = 1/N_cls L_cls(p_i, p_i*) + λ 1/N_reg L_reg(t_i, t_i*)

Figure 8.2.14 RPN workflow (Mirjalili, 2017).