Multiple and Deep Learning Networks for Dense Stereo Correspondence in Computer Vision

A Thesis Submitted to the University of Kent at Canterbury
for the Degree of Doctor of Philosophy
in the Subject of Electronic Engineering

by

Miao Wang

Jun 2018

Acknowledgements

First and foremost, I wish to take the opportunity to thank people have helped me during the challenging study so as to accomplish a significant achievement in my life.

I would particularly like to express my heartfelt and deepest gratitude to my supervisor Dr Konstantinos Sirlantzis. He taught me plenty of knowledge and the spirit of research and helped me to overcome various problems. Following his supervision, support, advice, guidance and helps, my study can keep proceeding smoothly, and considerable amounts of works have been carried out and rich results and achievements have been harvested.

A very great thank goes to my parents that they have supported me along with the journey of the degree. They are always there whenever I am in need of assistance and encourage me when difficulties arise in my entire life.

Finally, many thanks to all my colleges and friends who gave me suggestions and helps, also shared the experience of researchers.

Abstract

The inspiration for stereo vision in computer vision systems is derived from the human binocular visual system. In this, two views are captured by the left and right eyes and are merged into one three-dimensional (3D) scene representation in the brain. One of the important elements of ‘stereopsis’ involves the stereo correspondence estimation. Stereo correspondence refers to generating the location disparity for the same object in two images to retrieve distance information. A disparity map contains all 3D depth cues of a scene, and it is estimated by using stereo matching algorithms. Therefore, the quality of correspondence matching is an essential component, which affects the accuracy of 3D scene reconstruction.

Dense stereo correspondence is one category of matching methods, which, normally, operates at pixel level in order to reconstruct 3D views of the real world. As this is not a trivial task, among others, neural networks have been introduced and studied by researchers as a powerful nonlinear method. In this thesis, we introduce systems to perform dense stereo correspondence for disparity generation based on simple neural networks (NNs), multiple neural networks, and convolutional neural networks (CNNs). These calculate matching degrees between paired pixels in order to identify the best matched pair at maximum disparity range of stereo images. The contributions of this thesis refer to feature analysis for network training and disparity computation, network design involving structure construction and model optimisation, speed improvement for the disparity map computation, post processing for raw disparity maps, and comparisons: a) between the three networks and state-of-the-art-methods, and, b) among the three different types of networks, on the basis of quality of the generated disparity map.

Experimental investigations for feature selection and network model optimisation are discussed to define specific network architectures and model parameters. Moreover, the performance of the three networks introduced are compared with state-of-the-art approaches. Our results show that these three networks are all capable of matching corresponding pixels between left and right (stereo) images. The multiple neural networks-based system outperforms the other two in general. However, the system, which uses convolutional neural networks produces very similar performance in all cases. Finally, although both multiple and convolutional neural networks systems were shown to have the capacity for high performance in dense stereo correspondence estimation, the convolutional NNs shows better scalability.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Applications endowed with intelligent autonomous functions which are related to visual analysis nowadays are being widely used in areas such as vehicle automation, visual object recognition and robotics path planning. Products integrating this type of intelligent system are becoming to play assistant roles in the society of human being more and more in order to improve the qualities of human lives.

In order to reach a higher intelligent level, the essential theory for this category of technology is found on emulating the behaviour of human vision so that to make machines to acquisition the ability to perform automatic vision interaction from surrounding environment involved with visualisation information gathering and understanding. Computer vision represents such interdisciplinary scientific field of intelligent computer or machine systems mincing human vision system.

1.1 Research background

The concept of Computer vision can trace its history back to the early period of the 1970s, which was used to provide human imitation robots with intelligent competency by the pioneer of artificial intelligence, at that point, a computer was connected with a camera for attempting to accomplish the goal [1]. From then on, computer vision as a visual understanding theory has been continued investigating and developing by researchers. At the present time, computer vision especially the strategy of stereo vision is being used to extensively range of applications domain. Figure 1.1.1 shows an application instance of an automatic driving system utilizing the technologies of stereo vision [2]. Examples of popular fields with computer vision technique are listed as below [1]:

- Biometrics verification and recognition for identity authentication.
- Self-driving vehicle: environment sensing, automatic pilot and obstacles avoidance.
- Object recognition: to recognise objects from given images of video sequences.
- Traffic detection: monitoring of traffic flow and pedestrians and road conditions.
- Automatic navigation of robotics: path planning, automatic localisation and environmental map analysis.

- Medical symptoms analysis: diagnosis of a state of an illness through the medical visual instrument.
- 3D model reconstruction: to reconstruct objects or scenes into three-dimensional form.
- Motion inspection and tracking: to detect movements of subjects to structure motion flow models.

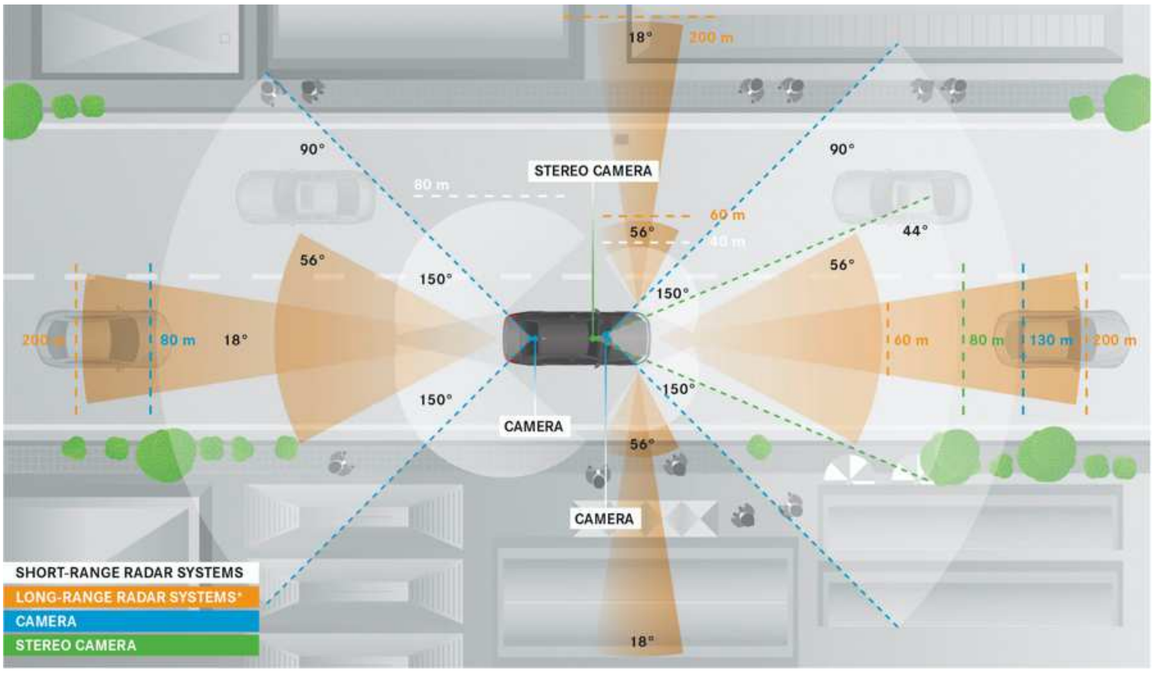


Figure 1.1.1 Drive system with stereo vision

The definition of making a computer or machine to have sight is not only just the meaning of image capture, but also includes the capability of brain function on processing graphical information. Visual information of the real world is captured and passed through on to the retina by Human eyes, following by transmitting signals of perception data to be processed by the brain to finally make decision of which action to execute.

According to such theory, moreover, as the purpose of computer vision is to interpret human vision for computer or machine by implementing mathematical algorithms to reconstruct the world, the processing procedure can be divided into three main steps: image extraction, data processing and output decision. Data extracted from images usually captured by media like cameras, and are applied with algorithms of image processing and analysis to produce output for the system to determine which decision would be the appropriate choice.

In terms of the characteristic of computer vision, depth estimation referring to retrieve three-dimensional depth of objects and scenes has been considered as an important component of intelligent applications that require visual perceptions in the form of machine visualization. For this reason, how to retrieve depth information significantly affects the performance of such applications. The fundamental core theory for depth extraction is accordingly in respect of how to determine the depth in a precise way. Theoretical processes of computer vision on depth recovery combine a series of techniques that obtaining, retrieving and analysing localization information in a three-dimensional spatial architecture. Depending on the inherent typicality of visual perception, the most common theory for estimating depth can be categorised into following two methodologies: depth estimation from a single image or stereo vision. According to the observations of state-of-the-art, stereo vision possesses the closest nature to the trait of human binocular vision and is the focus of most researches.

To recover depth from a single image is also called monocular depth estimation. This single vision approach examines values of 3D depth from a monocular image by analysing monocular cues. Human has the talent of recreating 3D scenes in the brain with a single view, which is to be achieved by estimating clues from single vision [3].

For example, objects with larger size would be considered as closer than smaller objects that both have similar known size. When an object is closer, the appearance on this object is sharp and clear, on the other hand, the texture shows undetailed surface for a farther object. Lighting produces different shadow and bight areas on objects. Those cues together contribute to in aid of perceiving depth in monocular vision.

Stereo vision based on video extracted images generates 3D views simulating the process of human vision with two eyes. After the images are captured, the 3D coordinates are computed based on the intersection of lines from the coordinate frame of each image. Then the global 3D coordinate is derived in accordance with the mathematical principle of binocular vision which is known as triangulation computation. The final global 3D coordinate is the 3D position of the object. In respect that the computation is in need of disparities between corresponding points in stereo images, one of the core processes of building the stereo vision from images is disparity estimation. The computation of disparity is to obtain the difference value of X coordinates of corresponding points. An example of robotics navigation adopting stereo vision is given in Figure 1.1.2 [4].

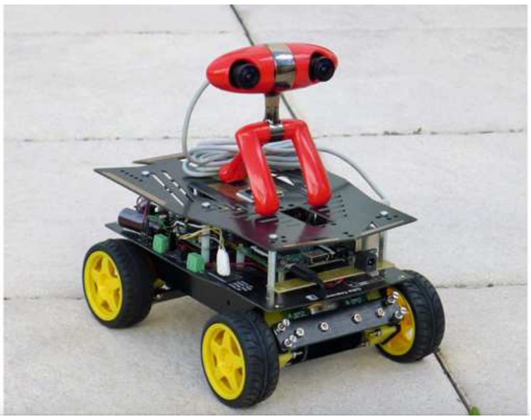


Figure 1.1.2 A robot setting up with stereo vision for navigation

Stereo matching is a field of approaches to match points have correspondence. This is implemented by searching corresponding pixels on the same row between rectified left and right images. Based on the observation of stereo correspondence methods, normally, strategies of stereo vision can be divided into two categories: dense and sparse [5]. Dense correspondence focus on global estimation (every pixel computation) in contrast to sparse method involves feature points based analysis.

This thesis focuses on investigating and developing techniques to improve the estimation of the 3D depth of stereo vision and in particular by investigating algorithms on dense disparity estimation by using artificial intelligence techniques such as Artificial Neural Networks (ANNs) and deep learning. Deeper descriptions of state-of-the-art will be presented in Chapter 2.

1.2 Challenge, motivation and objective

Along with the development of computer vision, the implementation of simulating vision system results in a variety of challenges which affect the quality and reliability of applications. The fundamental traits of its nature with a specific type of data and applying field cause issues to be faced.

The trends of researches in the community have met problems to be addressed in terms of how to make systems able to perform the level of the simulation as high as possible in an efficient way. General prominent problems issuing in wretched performance, in the current scenario, that have the need to be settled for application domain utilising computer vision technologies are shown below:

- Quality of Data:

  One in the first place is data collection. The input image data for the computer vision system plays an essential role as the starting point of the whole system. Noises could make an influence on and reduce the quality of data such as alt and pepper noise, image distortion and periodic noise. The performance of entire system thereby will be seriously dragged to low phase

- Real-time implementation:

  Real-time execution of computer vision has been a hard topic to be tackled all the time, as one of the characteristics of vision processing is to process huge amount of data as input, in this matter, to develop algorithms which can deal with vast quantities of information as speedy as possible to satisfy the demand of intelligent function nowadays is being attempted to solve by scientist.

- Hardware limitation:

  More advanced in technology, more hardware resources are required for the improved vision system design.

  Currently, there is particular necessary of improvement on the processing ability of hardware such like the capability of memory, central processing unit (CPU), and especially the graphics processing unit (GPU) that specifically is useful for vision processing, and as well as the long-lasting power support. The inventions of hardware resources especially for adapting computer vision system are one problem imperatively to be resolved.

- Autonomous algorithm improvement:

  By adopting the vision system, a human can determine their geographical and physical position effortlessly by themselves to accomplish diverse interactions with the real world. Being successful at simulating this talent so as to endue computers and mechanical applications with closest intelligent competence to human has been being desired by human society since the concept of AI appeared, which makes this target extraordinarily challengeable.

- Learning efficiency:

  As one of the domains constituting the interdisciplinary architecture of computer vision, Machine learning provides the intelligent capability of understanding concepts of images videos. In the interest of superior intelligent level training stage in most cases is given copious data, in this circumstances, efficiency on time-consuming and abstraction of integrant knowledge during the learning procedure leads to difficulties on keeping up the accelerated advancement of technology.

- Complexity understanding:

  For the purpose of to be suitable for wide range of applications, artificial vision systems are requested to equip with functions of understanding complex environment in consideration of the complexity of the actual world. For instance, there are various objects and subjects could be included in an image like animals, human, plants, cars, etc. These scenarios increase the complicacy of vision sensing. How to precisely recognise and distinguish every identity and relationship with each other and even further context of what are their intentions from motions comes to be a quite a challenge to attain.

The outline illuminated above are the general ones for computer vision. Regarding conceptions of stereo vision that imitating human stereopsis, the following is the digest of typical challenges and problems on this scope to be considered:

- Rectification quality:

  The matching of corresponding points in a stereo-pair image is carried out by contrasting similarities of potential points on the same row coordinates of left and right images. A stereo-pair is thus rectified to be projected onto a conjunct plane with the aim of decreasing matching procedure to the one-dimensional issue. On account of this, the quality of rectification directly affects the accuracy of matching outcome.

- Correspondence matching performance:

  To match the corresponding points is the centre phase of stereo vision, in view of functional duty supplied that the projection intersection of matched points represents the true point in a scene. This is a difficult topic, especially with dense computation. A large number of researches has been carried out on developing algorithms to improve the performance of stereo matching. Nevertheless, there is still a distance to be shortened to actually recurrence the theme of human perception.

- Disparity map computation speed

  The computation of dispraise is a time expensive issue, especially when implementing with dense algorithms. This work is done by searching corresponding pixels for every reference pixel in the reference image within a disparity range, which occupying a load of hardware resources, in consequence longer processing time. The expectation of The development of optimization algorithms has become a growing area of interest.

- Disparity range accuracy:

  As mentioned above, the principle of searching correspondence indicates that the matching procedure between a reference pixel and candidate pixel performed within a range that the potential matched pixel should not be beyond. The inaccurate defined maximum range of disparity could result in an incorrect pixel matched to reference pixel.

- Occlusion issue:

  Due to the peculiarity of visual perception, occlusions always appear in stereo images, which missing matched region in the other image. By consequence of this phenomena, the matching process may cause error detection that regions in another image could be erroneously matched to occlusion region

At the very least, computer vision has been developed steadily and appeared to have a rapid growth of advancement during the recent period. More and more issues are being attempted to solve by researches gradually.

The motivation of investigating in this subject above all is that the stereo vision has been considered as a technology in computer vision which can generate the depth in the way of simulating human vision instead of using extra facilities, and the accuracy of depth retrieve of the stereo vision is one of the core factors to affect the accuracy of the 3D reconstruction and depth localisation. In respect that stereo vision can extract the 3D data, more information can be extracted from digital images as compared to the traditional two-dimensional information.

Moreover, the usage of stereo vision is to gather the depth information to perform the 3D reconstruction and depth localisation, therefore, such accuracy of the 3D reconstruction and depth localisation are then dependent on the depth information in the stereo representation. Furthermore, the depth information is one of the significant foundations for establishing 3D scene and localisation for application such as robotics navigation. Thus, it can be seen that, in order to have a better result of 3D re-establishment and depth localisation, the improvement of the performance of the depth information generated in the stereo vision has a significant effect on improving the localisation and environment mapping processes, wherefore disparity estimation comes to a considerable status.

A large amount of methodologies has been recommended for disparity estimation, meanwhile, each approach is being constantly reformed in a way of fusion or parallel with each other. Algorithms that exploiting techniques of Neural Network (NN) and deep learning (DP) in the family of Artificial Neural networks (ANN) appears to be started to investigate to the accompaniment of evolving. ANN simulates neural systems of the brain to create artificial intelligent models for computer evolvement. On the basis of the inspiration coming from human neural architecture, NN and DP have been proved that can be adequate to tackle complex problems in need of human thinking pattern.

NN is capable to train and learn a circumstance with non-linear and intricate connections, also after training NN can infer the nature of unknown scenario from unobserved information, in addition, a variety of variables can be adapted as input by this special neural structure. DP as a form found on NN has an advanced ability that creating features in layers itself and the capability of adapting a vast number of input data without decreasing proceeding speed. Moreover, the concept of DP can generate plenty of flexible models with curtailing the cost of feature engineering simultaneously to suitable for handling the diversity of occasions among the real living lives. Both of these two systems are been adopting by the field of computer vision. These motivations have made this study in the interest of investigating the implementation of disparity computation on the basis of stereo correspondence approaches with NN and DP.

Different algorithms can produce different outcomes. The objective of this study is to investigate the effectiveness of dense stereo matching algorithms found on different designs of NNs and DP systems that find corresponding points between stereo images to form dense disparity maps which represent overall depth related representations for captured scenes. In order to achieve the goal, there are three major architectures used to carry out the evaluation: simple neural network, multiple structured neural networks and convolutional neural network. The research is mainly carried out in aspects as shown below:

- Feature engineering
- Design of network structures
- Model optimisation for stereo estimation networks
- Disparity map optimisation
- And so on.

To summarise, this research aims to discover and present the effect of stereo vision exploiting different types of ANNs cooperating with the dense matching method in the hope of revealing the potential benefits of proposed methodologies. Detail methodologies, experiments and evaluations will be expatiated from Chapter 3 to Chapter 6.

1.3 Contribution to knowledge

Systems that produce dense disparity map have been developed by adopting the theory of simple, multiple and convolutional neural networks to derive stereo correspondences for disparity computation. Some issues are addressed by this thesis on the basis of examining through these systems. Summarised contributions of this thesis are outlined below:

- *Feature Extraction*:
  Methodologies in relation to data selection for extracting features from left and right images with their reference pixels as the centre of feature windows form matched and unmatched pairs for network training, where the extraction are based on some designed constraints.

- *Feature Types Impact*:

  Estimation of performances uses three types of features in respect to the degree of numerical information contained in feature vectors, as these conditions affect the level of accuracy on account of the aspect that computer understands images in the form of numerical values.

- *Network Structure Design*:

  An assembled multiple neural networks consisting of three backpropagation sub-networks, and a convolutional neural network have been devised. Each network structure is designed in a basic form initially and improved to reformative architecture. The experimental results uncover the path of creating the advanced structure of networks.

- *Network Layers and Parameter and Model Optimizations*:

  Typical parameters for training involve learning rates and training functions for simple neural network, hidden layer construction for multiple neural networks, fully connected layer design and parameters for convolutional layer and training algorithms for convolutional neural network are examined with different settings to show the effect on accuracies. Detailed experiments and evaluations are discussed to present the connections among those layers and parameters for providing the idea of adjusting cooperation between them in order to maximize performance.

- *Speed Improvement for Disparity Map Computation*:

  The formation of a dense disparity map normally consumes long execution time, that makes difficulty on experiments or applying with applications. On account of this matter, to a degree, optimisation of processing time is created with two phases: features extraction from stereo images, and disparity computation in accordance with matching degrees.

- *Refinement of Raw Disparity Map*:

  A refinement procedure is often applied to the raw disparity map to enhance the quality in respect to cleanness and resolution. A series of refinement methodologies always been implemented at this stage. This study implements refinement methods that can provide functional post-processing in a compact way for the purpose of reducing the complicacy causing resources consuming.

- *Comparisons*:
  The performances of the three networks are compared in two aspects. The first comparison is between three networks and state-of-the-art approaches, and the second comparison is carried out among three networks. The results of comparisons can give clues of systems are good at which aspects, moreover, the effectiveness of each network is as well as observed and evaluated, which is conducive to understand the recondite principles behind the surface that how algorithms using such networks to make a service to the field of stereo correspondence estimation.

The work of this study makes contributions to stereo vision, also with the hope of making assistant for other further researches.

1.4 Structure of thesis

This Chapter as the start point of the thesis presents the background of research field as well as involving the issues in such region, which gives us the motivation to study proposed topic, accordingly, address the goal expected to achieve. Contributions from our research are also depicted in this Chapter.

The second Chapter systemically introduces the family of computer vision from the theory of visual perception system to practical algorithms for implementing applications, especially stereo vision reconstructing 3D view in the way of estimating stereo correspondence.

Chapter three presents the pipeline of feature extraction on the basis of epipolar geometry, and methodologies of feature selection and different data type, furthermore, relative experiments are evaluations are explicated. Moreover, the dataset used in our research are illustrated.

In the fourth Chapter, neural networks and state-of-the-art of stereo correspondence algorithms with neural network and two types of neural networks built in our research will be presented for stereo corresponding task. The first one is a simple neural network and the second one is a multiple neural networks found on the first one. Interrelated methodologies and experiments are explicated and analysed.

Chapter Five presents deep learning and state-of-the-art stereo corresponding algorithms that adopting deep learning techniques. A devised convolutional neural network for the purpose of finding pixel corresponding is given in the Chapter, moreover, methodologies will be depicted and experiments with analysis will be discussed.

The evaluation with disparity maps for all three network structures will be presented in Chapter Six. The methodologies of disparity map computation and algorithms for computational speed optimization are explicated. After retrieving initial disparity maps, post-processing is introduced to implement refinement. The systems with three types of networks are evaluated with different image datasets and are compared with state-of-the-art, and with each other.

The last Chapter makes conclusions based on all of the investigation and contributions explored in this thesis, and the possible future work will be listed for further study so as to further advance development on the interrelated field.

1.5 Chapter summary

On the whole, the theoretical concepts that have been summarised in this Chapter depicted research background regarding computer vision, moreover, practical applications were introduced. furthermore, the importance of stereo correspondence for stereo 3D vision reconstruction and relevant challenges have been addressed in this Chapter. The general ideas of system constructions with technologies of ANN (simple neural network, multiple neural networks and convolutional neural network built in our research) have been presented. Relevant contributions derived from motivations and objectives were as well as listed in this Chapter. Finally, the organization of the thesis was outlined by breaking down into each individual Chapter introduction.

Chapter 2: Computer vision and 3D stereo view reconstruction

According to the objective of this project, the literature survey has focused on several different, but closely related areas of previous work. This includes the visual perception system in computer vision and specifically stereo vision as a popular thread for disparity estimation and comparisons with other

2.1 Overview of computer vision

Computer vision can be referred to make improvements of intelligence of computer based on the simulation of activities of human vision. By applying computer vision, the ability and efficiency of a computer interacting with human and environment can be significantly improved.

To reach to the level today, computer vision has been experiencing a long evolutionary process. As introduced by Szeliski [1], in the earlier age of 1970s, the original techniques of computer vision started from processing images extracted through digital means which focused on understanding the scene through 2D information processing with the hope of computer recognizing the world easier like mentioned in Chapter 1, such as making labels for edges in a form of 2D lines to extrapolate 3D figure, furthermore, later in the 1970s, 3D construction and stereo matching begun to be studied.

In the wake of advancement, during 1980s, researches were concentrated on developing mathematical algorithms to resolve the quantitative problems [1]. In this period, a method called image pyramids which could implement down-sampling to break an image into a series of sub-images to obtain the required information. The stereo technique was utilised to deal with shape related targets, for instance, to extract a shape from shading, focus and texture. Moreover, Markov Random Field was introduced and tried to tackle regularization issues as an alternative optimization method.

From the 1990s to 2000s, some of the fields developed previously became more popular than others on the area of recognition, e.g. projective reform, multiple views 3D analysis, advanced segmentation of images, and one of the significant milestones is learning theories got arisen primarily on the range of facial identification [1]. The learning algorithms increased the capability of image understanding from vast inform in an effective route. Till the century nowadays, by combining different algorithms have been discovered, computer vision has become a theoretical system that is used on various of areas, such as robotics navigation, clinical image diagnosis, and so on, which are supplying the demand of computational machine mainly on providing intellectual power at the present time [6], as also mentions in Chapter 1.

The goal of computer vision is to re-establish the world in a reverse means which is to derive attributes of nature and living creature, such like a form of an outline, colour, texture, lighting and depth from data captured through physical equipment [1], [6]. In order to achieve this simulation, scenario information like animals and houses is generally given to computer as input, under these circumstances, how to distinguish which one belongs to which category is simple and natural ability for human to work out, while comparatively speaking, this is a formidable job to complete for computer which including quantities of mathematical algorithms targeting on every area of visual data analysis.

Human is naturally born with powerful brain architecture that is able to recognize everything observed through eyes and reflected on the retina in this world. In computer vision, the eyes are represented by tools like cameras, and brain functions consist of arithmetical algorithms and various of electrical hardware, which requests a great variety of elaborate, intricate and advanced scientific approaches. As pointed out, a human can achieve these actives without an effort, nevertheless which is a hard topic to accomplish with a computer, in the respect that the theory of such area is to transform the data obtained from video or image to mathematical functions to generate computational determination [1], [7].

The whole process of computer vision can amply and principally be divided into three aspects: feature extraction, reprocessing features and rebuilding vision [6], [8]. The main work of image extraction is to gather images from cameras following by feature extraction from images according to the requirements of applications such as gradient, texture and so on. During reprocessing phase, the extracted features are computed using different methodologies for instance image processing and machine learning to output decisions. In reference to which type of visual perception to be reconstructed in the form of simulating human abilities, the most common visual re-establishment is 3D reconstruction.

At the recent period, 3D scene extraction is quite a popular research area in computer vision and has been widely used for a variety of fields, e.g. biometrics application, building models, robotics, etc. For example, one of the famous robotics application that utilizing 3D depth technologies is the Mars rover created by NASA in Figure 2.1.1 ([9], image is used following NASA copyright guidelines: https://www.nasa.gov/multimedia/guidelines/index.html). Face recognition as one of the common biometrics fields has started to adopt 3D information to improve performance. After all, the concept of computer 3D sense is the approach that can represent the intelligent ability of human visual activities as much as possible.

Figure 2.1.1 Mars rover

On account of the reconstructed 3D scene will represent a 3D environment, which contains 3D information for computer understanding its surrounding better. The more accurate 3D scene reconstruction is, the more efficient interacting capability of the computer will be. Therefore, how to improve the accuracy of 3D scene reconstruction is a significant matter. The core component, which affects accuracy in establishing the 3D scene, is the 3D depth estimation that obtaining 3D localization information from the real world. 3D depth can be recovered from images (single or multiple images), or from the flow of a video, or from a 3D sensor.

2.2 Computer depth perception

A distance of an object in the real world is represented as the depth from its surroundings, furthermore, the localization information can be determined by predicting depth value [10]. The field of depth estimation is in relation to a collection of approaches that recover depth data of every projection spot of the digital visual prospect through mathematical theories so as to endue machines with the autonomous ability of this area.

2.2.1 Overview of depth estimation

Figure 2.2.1.1 shows the theory of how scenarios are mapped into an image. As shown in this figure, every point of the scene projects its reflection onto a common plane by shooting a series of rays passing through an aperture, in other terms, this is like a “mirror” standing in the front of the scene, thereupon all reflections shape an image representing the entire projection of visual scene [10].

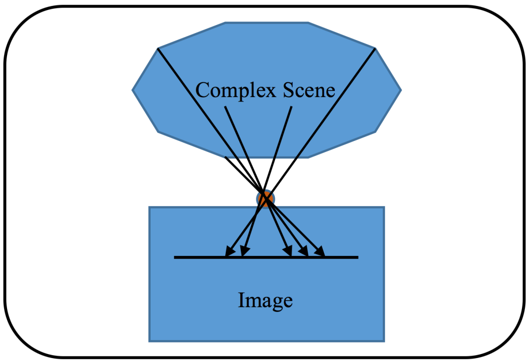


Figure 2.2.1.1 Scene projection

By the reason of this principle, some issues are caused during this projection procedure. the projection leads to the loss of 3D information that is the depth information. A formed image normally does not contain the data of the third dimension indicating the relative depth of spatial position in the real scene. In another word, the trait of this process is to acquire a 2D image representing a vision that has the benefit of decreasing implementing duration but the deficit of 3D depth information. Under these circumstances, an approach which can extrapolate depth data is in the need of implementation. 3D depth estimation hereby playing an essential role is exploited to handle the challenge.

Generally speaking, depth estimation takes into effect from two aspects: definite region, and occlusion region [10]. Furthermore, a projection mapping to form an image regularly happens at a certain point from an angle. In the real world, arrangements of objects in a scene place intricately with each other in most cases, as well as including complicated interactions. Some objects may be blocked by other objects, that is to say, parts of objects locate at the back of other objects since vision performs from a particular point of view. In the consequence of sight angle, some regions are certainly projected into images, in contrast to there are some fragments of objects are missing in the formation of optic perception, this is so called as occlusion area. The research of this these concentrates on estimating definite area.

2.2.2 Active and passive schemes

Although many techniques are investigated around the form of analyzing images to make computer predict depth itself automatically, other methods that exploit extra instruments to detect depth as well as reveal the productive capability at the meantime Among a quantity of approaches has been discovered, active and passive as two classes can be the representative of common schemes for depth estimation technologies nowadays [10].

The scheme of active algorithms aims to derive depth value through a way of directly gathering clues from objects through means of functions supplied by devices. The feedback information generated by those media imply and evolve as qualitative trails for producing a depth map. As illustrated by Bhatti [10], two types of such class are regarded as popular theoretical bases: illumination and ultrasound.

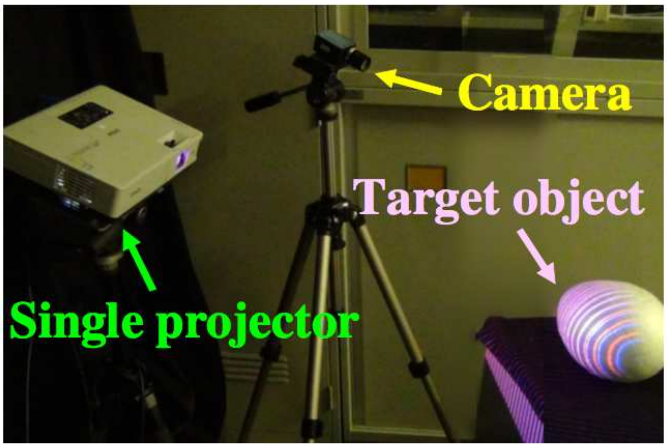


Figure 2.2.2.1 Layout of structure light approach

Such kind of active methods applies lighting irradiation and ultrasonography on objects in a fashion of scan to attain depth cues. As an example, the paper [11] describes a typical method that is to make use of structured light generated by projectors to produce a pattern for labelling each pixel following by matching computation with those decoded cues. Figure 2.2.2.1 displays one of standard setup for a structur light system [12]. The research of [12] uses the flow of light to tackle challenges of exterior enhancement. From the figure we can see, such structure contains one or more projector casting light on objects to create lighting stripes to real distance hints for camera capture. Another common method in the illumination field is called Time-of-flight (ToF) approach that counting the duration of light reach on objects [13], [14]. Techniques based on ultrasound adopting techniques of ToF act on and are widely implemented in aid of medical inspections [10], [15].

Passive based algorithms seek the path of evaluating depth on the basis of ideas that putting much more effort on computational terminal instead of early data acquiring stage. The purpose of this strategy is to devise algorithms that can formulate human behaviour for creating human imitation based intelligent machine. In general terms, there are three categorise as passive approaches to accomplish this mission, that is monocular vision, stereo vision and over two visions.

The basic theory is to implement a series of mathematical algorithms on images captured by devices as to say the most common one camera, in this matter, these technologies great benefits in aid of improving the capability of computational intelligence [10]. As likewise introduced in Chapter 1, the human is not only good at vision reconstruction from stereoviews, but is also adept in estimating depth from a single view.

According to this inspiration and thinking of the intention of developing and advancing intelligent applications, scientific models are being devised based upon these strategies: monocular computer vision that retrieving spatial sight from one captured view, and thus the visual perception re-establishment from stereo image pair representing the principle of stereo vision, moreover a scene restoration through more than two views. Further discussion of these algorithms will be expounded next in detail.

No matter active or passive scheme, their own characteristics make these approaches can exert their functions effecting on different ranges of interests. Some area may require active contributions, and some quest may apply with passive functions. Even more, some applications not only utilise an active approach, but also take advantages of a passive method, in other terms, the fusion of active and passive, which sometimes can produce a superior class outcome. For example, researches [13], [16], [17] suggests approaches that can improve depth detection of scene and surface content by combing active technique (structured light) and passive (stereo measurement) method.

2.3 Monocular vision with depth estimation

In computer vision, depth perception can be achieved by studying not only multiple images but also a single image. To obtain depth information from one image is a methodology inspired in terms of monocular vision which representing a field of visual ability in reference to perceiving spatial environment with one eye/view. This scheme is always adopted as a low-cost solution for applications that do not need in detail estimation, which drew our attention on depth estimation at the very first beginning.

Monocular computer vision as a challenging approach has started to be employed with various regions. For examples, embedded system cooperating with single visual perception for mobile appliance of tracking face [18], applying monocular with vision identity detection of robot [19], monocular vision based independent and self organized actions of robots on navigating and localizing [20], and application of autonomous obstruction avoidance found on sensing monocular depth clues [21].

As mentioned before, monocular perception unlike stereo vision does not generate straightforward depth information, which results in even more difficulty in obtaining spatial data. Roughly, there are three sort of areas introduced by Bhatti [10]: structure analysis, points movements and defocus measurement. The first two are the approaches only can produce relative determinations.

The way of examining structure merely estimates relative distances between objects with presumed structure. For the second method, image patches are labelled with points and those points are tracked following time elapse to observe changes between different time domain. Defocus approach take the measure of the degree of defocus on each pixel of images which can create definite space mensuration. In contrast with structure and points strategies, this approach can provide more precise measurement.

Among various technologies, one of common means is to follow the theory of imitating perception system of human on the basis of estimating the signal from a single view which is known as monocular cues[3]. Monocular cues that are the most widely used for such depth estimation are listed as following [3], [22], where Figure 2.3.1 gives the schematic examples of monocular cues of relative size, texture gradient and overlap [23]:

- Size and shape: size and shape between objects (relative aspect), the physical size of objects, the familiarity of objects size. Bigger size may indicate a closer object among objects which are the similar known size. Closer objects present detailed shape in contrast to the outline of shape fading from far away.

- Texture: gradient of texture, variations of texture. The texture of closer objects is normally clearer and more detailed on the texture content, and more visible in a sight than farther objects.

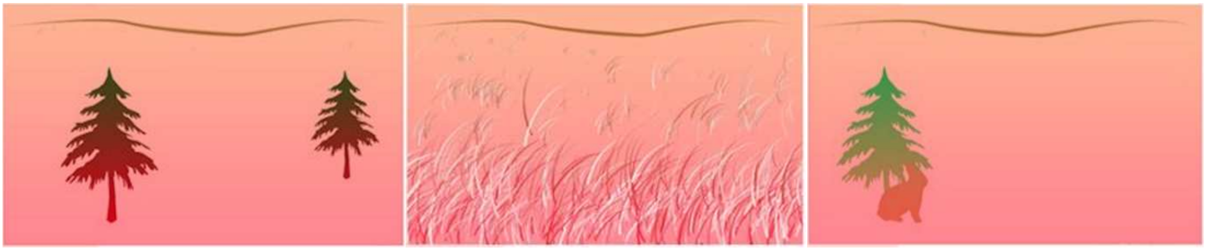
- Light and shade: colour or haze of objects caused by angels of lighting. Brighter and legible areas of a scene may be closer than areas with more shades and hardly visual observation.

- Focus: regarding the use of the lens, nearer objects require more accommodation than objects in a further distance.

- Overlap: due to the complexity characteristics of the real world, contents of a sight is interlaced a blocked object can be normally determined as in farther distance than the one blocking it.

- Motion parallax: in relation to a reference subject, its surrounding objects pass through faster than distant objects, in other terms, the speed of movements is faster for closer objects in visual perception.

- And so on.



(a) Relative size (b) Texture gradient (c) Overlap

Figure 2.3.1 Examples of monocular cues

Research [3], [24] and [25] uses Markov Random Field to analyze local and global monocular cues (such as texture gradients, haze, and relative occlusions) for retrieving depth values by presuming variables containing location and orientation information on patches of the plane. The paper [26] proposes an approach that estimates depth from a single image using cues like shape, colour, and texture features through the method of breaking down matter into segmentation analysis level, moreover with short processing time.

During the recent period, the technique of machine learning has increased its roles in the field for depth estimation. The study of [27], [28] uses Hidden Markov Model (a probability model) for 3D depth estimation from one 2D image to reconstruct surface. The theory of research [27], [28] is to reconstruct the 3D model from a single 2D image with Subband Pseudo 2D Hidden Markov (SPHMM), which is trained in advance. The result of this study shows that applying machine learning with monocular depth reconstruction can produce effective performance [27], [28].

In recent years, deep learning as an advanced and novel approach along with the development of computational learning techniques is extensively adopted for monocular depth perception [29]. By making the use of deep learning, the fusion of global and local information on images can reach to an effective class. An approach of combining two deep scaled networks is proposed to tackle this task that first to handle global data with one network producing depth values on this stage and second to refine the output depth from the first network with local data using another deep network [30].

The most adopted model of deep learning for depth retrieve from a single view is convolutional neural network, which has dominated in the field of computer vision recently. Most researches create their algorithms based upon the idea of deep learning especially CNN. Paper [31] introduces a strategy which expanded the idea from [30] that connecting three CNN networks to form a learning model: depth prediction and refinement are processed by first and second networks, and the third network increases the resolution for output map.

It is a hard work to recover 3D depth from one 2D image, since the cues from a single image only show local features, moreover, the 3D information loss during the projection from the real world. Most studies of 3D depth estimation have centred on stereopsis field. Next section will interpret stereo vision.

2.4 Stereo vision with depth perception

Although monocular vision has the benefit that with low-cost implementation consuming, the accuracy level still cannot reach to a very high performance due to its perception characteristics leading to ambiguities and uncertainty. Multiple views based depth perception can, in contrast, provide higher performance which is mainly divided into two categories: two vision based and more than two vision based.

The first one is known as stereo vision which is also the most widely exploited one. The second category is three or more visions based visual reconstruction, which utilizes over two cameras in terms of this scheme can re-establish different perspective angles of an object simultaneously. Comparing this two methodologies, a system with over two visions requires more resources supplies like the setup of cameras in surrounding and only suitable for specific circumstances, while stereo vision can adapt to practically every application with simple setup requirements. On account of these matters, stereo vision so then is the focus of our research.

2.4.1 General introduction of applications

Vision reconstruction has been always a challenge mission to achieve by reason that computer perceives this real world in the form of numerous quantity of numerical data. To seek an approach that can accomplish this task efficiently is in a qualitative manner significant. The most extensively investigated and employed algorithms in computer vision is stereo vision as the principle of stereo perception is good at the ability to predict depth information.

There are a number of applications have started to integrate with stereo vision system such as autonomous driving vehicle that is capable of recognizing obstruction [32], navigation and localization while mobile robotic shifting around in the real world with complicated circumstances [33], [34], and also medical field like [35] that using stereo systems to reconstruct the surface of retina for surgery preplan purpose. Stereo perception contributes essentially on promoting the advancement of robotics on account of making robots possess vision ability like a human. Lots of projects have been attempting to explore more and more progressive algorithms on this infusive area.

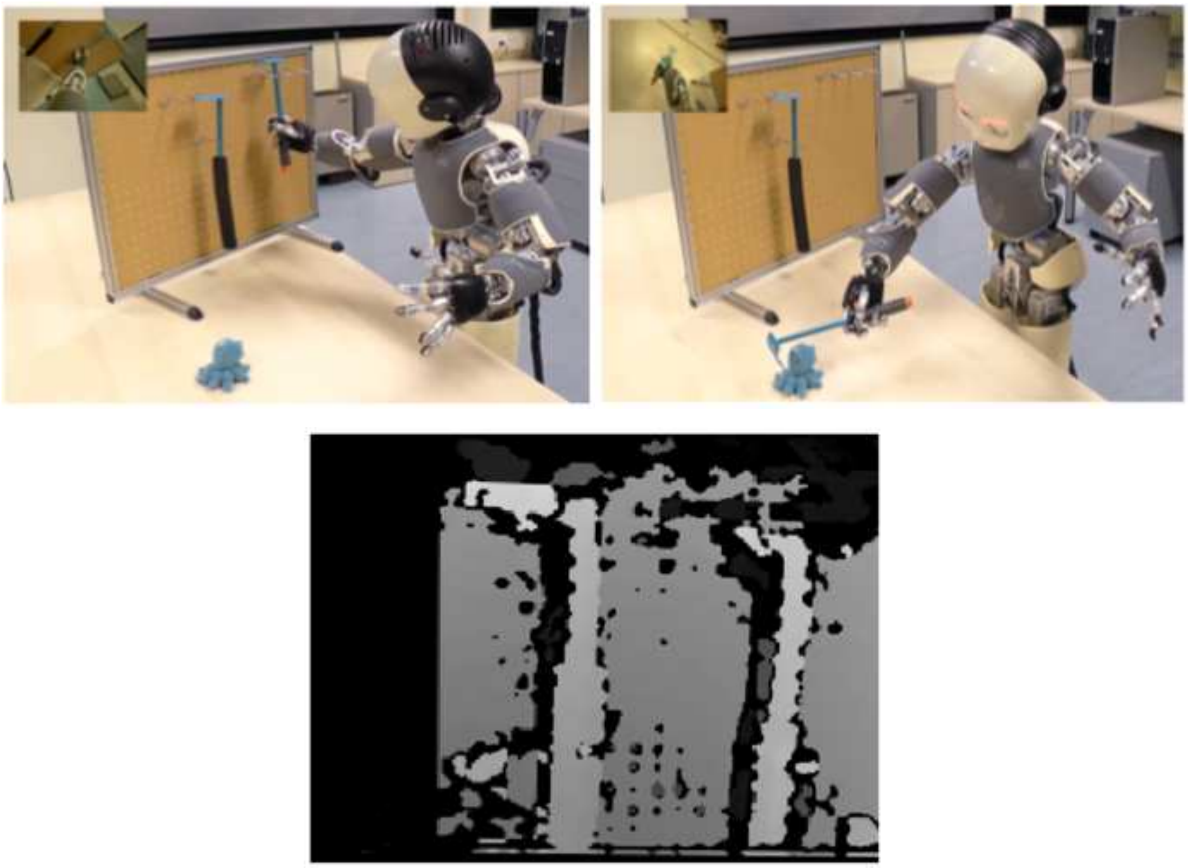


Figure 2.4.1.1 Robot hand tracking with stereo disparity

Paper [36] describes a project investigating on that sophisticated humanoid robot studies and trains the ability on its own that uses stereo depth based perception to cooperate with hand function in the interests of performing actions like picking and grabbing objects as presented in Figure 2.4.1.1. Research [37] introduces a similar project but with rather a simple setup of the experimental environment. An approach [38] talks about one interesting research area for the intelligent autonomous robot, which is to recognize stairs based on stereo vision for the purpose that a robot can move without restrictions caused by stairway.

2.4.2 Principle of binocular vision and computer stereo vision

The inspiration of computer stereo vision is derived from the human perception system so as to achieve an intelligent level of human imitation as higher as possible. A human can view the real world in a three-dimensional structure effortless owing to the functions of brain processing coordinates of two views on the left and right side that are captured by the two eyes, which is known as binocular vision [39].

2.4.2.1 Theoretical presentation: computer simulating biological system

The theory of binocular vision is to re-establish a scene at the visual cortex through neural systems by processing vision signals obtained by two eyes in the way of recombining two stereo signal streams [39], [40], [41]. Figure 2.4.2.1 shows the schematic description of how binocular perception works [41].

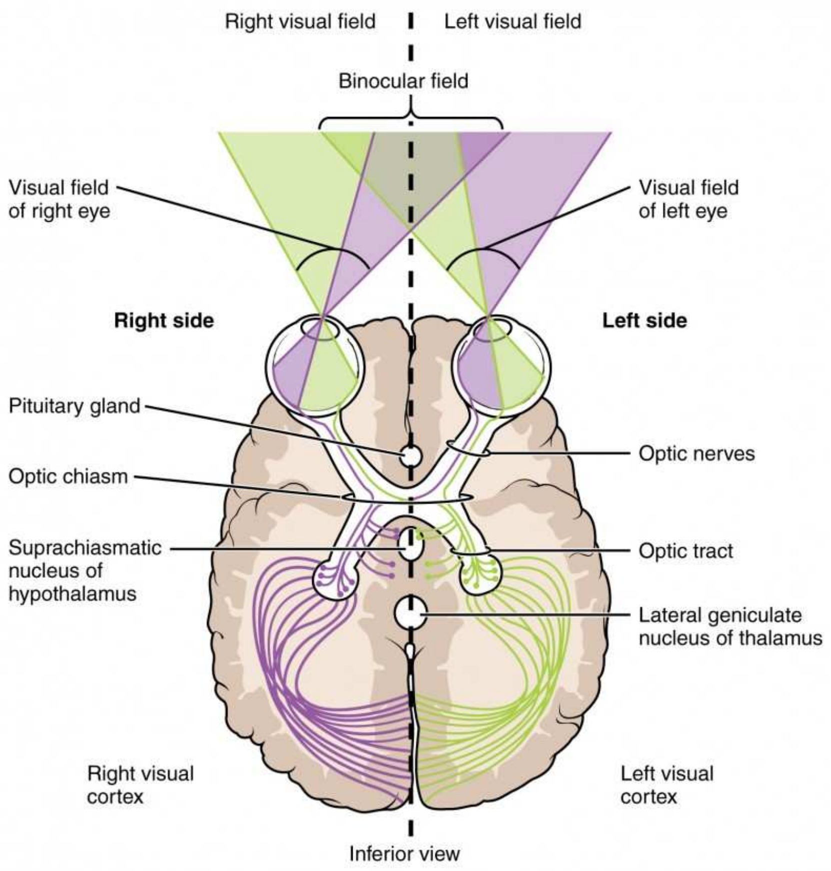


Figure 2.4.2.1 Binocular vision principle

First, at the vision capture terminal, a sight projection passes through the left and right eye lenses and projects on the surface of the retina which is composed of millions of neurons transforming lighting projection into nerve signals. And then the two signal flows are transported by optic nerve and cross at optical chiasm, at this point, half of two flows head to invert left-right direction to converge with main left and right flows. At the last stage, the left and right optical flows arrive at the left and right brain hemisphere, and the sight is then reconstructed into one 3D presentation at visual cortex on the basis of the retinal disparity between left and right views.

In accordance with the idea of simulating human binocular vision, the architecture of stereo computer vision at the very beginning level, that is to say, to acquire the input information normally employs two cameras which is the representative of two eyes, where features form signal flow. The system then makes the use of various mathematical algorithms to cooperate with each other to imitate and accomplish the functions of the neural system and visual cortex of the brain, which is a quite challenging as the difficulty of being brought into effect.

A typical systematic illustration for the formation of an image is presented by Figure 2.4.2.2. Just the same as two eyes align in a certain distance with each other, the basic arrangement of stereo cameras is normally set up with a spatial length between them, which refers to baseline.

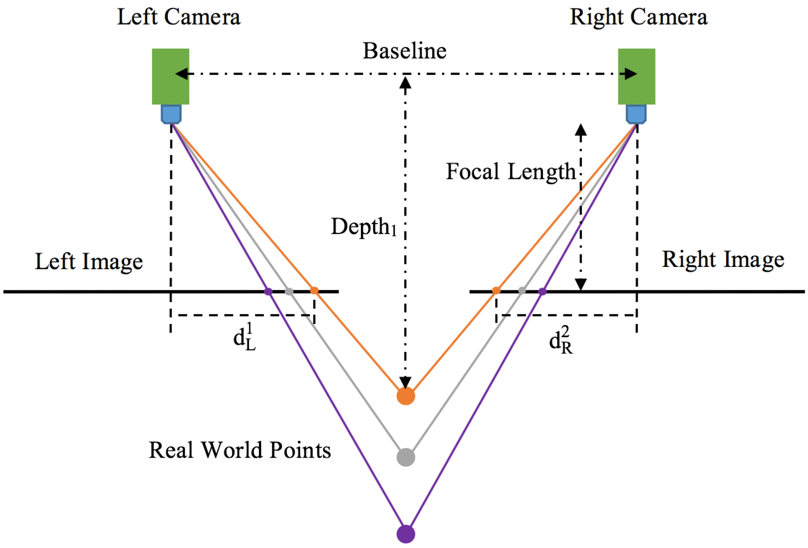


Figure 2.4.2.2 Theory of image formation in computer stereo vision

The point in the real world is found as the intersect of projections from the perspectives of left and right cameras as the schematic displayed in Figure 2.4.2.2. The depth of the point that is from the intersection to the baseline between cameras is accordingly computed in relation to the differences (disparities) produced by the shifted distance among cameras on image plane within a focal length, which is known as triangulation equation [42].

2.4.2.2 The process of stereo vision reconstruction

The entire process for reconstructing 3D values can be divided into four main stages from stereo images capture to 3D depth generation [43], [44], [45], [46] as the illustrated pipeline given by Figure 2.4.2.2.1 [43]. Camera Calibration, Stereo Image Rectification, Stereo Correspondence and Triangulation Computation.

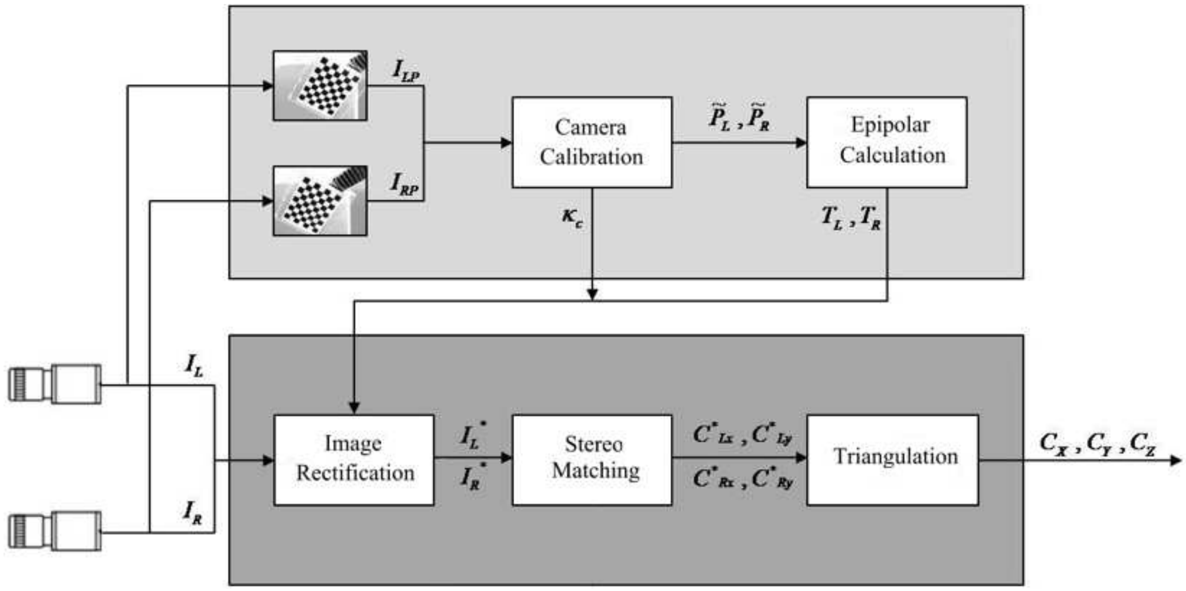


Figure 2.4.2.2.1 Process for stereo 3D reconstruction

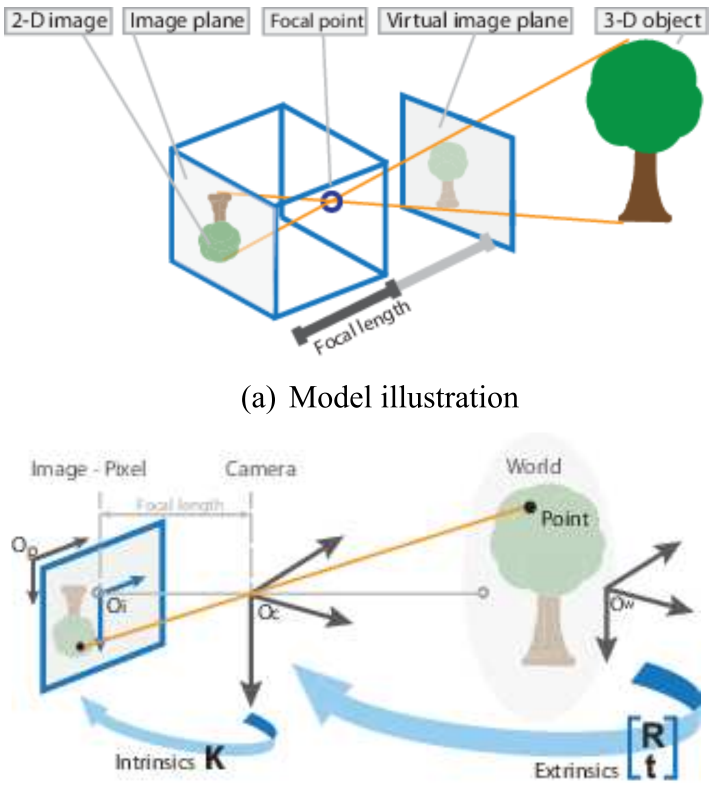
On the stage of camera calibration, the parameters of cameras are calculated for epipolar transformation computation with the purpose of rectifying images. The stereo images are then rectified with transformation matrices to transform two stereo images into a same horizontal plane so as to simplify the signal processing procedure (details can be found in Section 3.1).

Since the images are aligned into the same horizontal line, the stereo corresponding process can be implemented in one horizontal dimension. Corresponding of points between left and right images are estimated on stereo matching step so as to produce disparity values that are the representative of depth information, in addition, the performance of this matching conduct principally has an influence on the quality of produced depth value.

The final stage of reconstructing visual perception is to determine the third-dimensional value – distance from epipolar geometry. By being aware of the depth value, the spatial position of a point combining with horizontal and vertical values can be identified in relation to the real world. The distance value of a point can be calculated with disparities after discovering its projections (a matching pair) in left and right images in accordance with the triangulation principle (see Chapter 6, Section 6.1).

Camera Calibration:

The area of camera calibration is in respect of estimating the connection between the structures of camera and real-world spaces such as finding out the precise coordinates of pixels in relation to the real point location through the way of parameter matrix transformation, and as well as the correction of distortion, in addition that most calibration algorithms are found on the principle of homographies [7], [8], [47], [48], [49], [50].



(a) Model illustration

(b) Spatial relationship

Figure 2.4.2.2.2 Pinhole camera model

The parameters of a camera that is in need of calibration for representing camera stat are intrinsic and extrinsic parameters [7], [8], [47], [48], [49], [50]. Intrinsic (K: projective conversion between the camera and pixel/image coordinates - 3D to 2D) contains three coefficients which are listed below:

- Focal length: the distance from the focal pinhole to the image plane.
- Principal point: the pixel position of the image centre.
- Skew coefficient: angel between skew and perpendicular axes.

The correction of distortion is handled by the model of radial distortion which happens more around the border of the lens in the form of bent light rays and tangential distortions caused by not parallel between lens and image plane. And the components of the extrinsic parameter (rigid conversion between world and camera coordinates - 3D to 3D) are rotation (R) and translation (t).

These parameters are computed with the algorithms of calibrations that are based upon the geometry of the camera model, which the most basic and broadly adopted model is the pinhole model [51] as given in Figure 2.4.2.2.2 [47]. The first image of Figure 2.4.2.2.2 presents the principle of pinhole model and the second image shows the relation between world, camera and image plane in respect to intrinsic and extrinsic parameters.

In the pinhole camera model, the projections of objects are mapped on to an image plane through a focal point in the form of pinhole. This process transforms the real scene from 3D to a reversed 2D image in a certain length between the plane of pinhole and image mapping. The transformation is accomplished by a matrix (camera matrix - M) with a 3×4 structure that consists of intrinsic - K and extrinsic - (R|t) coefficients as denoted by Equation 2.1 [7], [8], [47], [48], [49], [50].

$$M = (R|t)K \tag{2.1}$$

There are two major toolboxes provides very useful tools to implement calibration. One is from leaning OpenCV library [7] and the other one is created based on Matlab functions [48]. These toolboxes have plenty of functions for carrying out calibration algorithms and detailed descriptions to explain how to operate with these built-in functions, which is very helpful for performing camera calibration. Our research focuses on exploring the field of stereo correspondences for disparities estimation which is a core aspect for depth estimation.

2.5 Stereo correspondence

Scene reconstruction based on stereo images has been a popular topic in the field of computer perspective as the most adapted technologies for perceiving 3D information from the real spatial environment. The society is the of opinion that one of the cores in re-establishment engineering of stereo visual perception is meant to be stereo disparity estimation that is accomplished by a known methodology which is called stereo correspondence for the purpose of binocular depth recovery. Such corresponding methodologies consists of stereo matching and disparity computation algorithms, which found on the theory of epipolar geometry [52]. The principle is to search matching point between left and right images captured by a pair of stereo cameras. These points that are determined as corresponding to each other are the projections of a point in the real world.

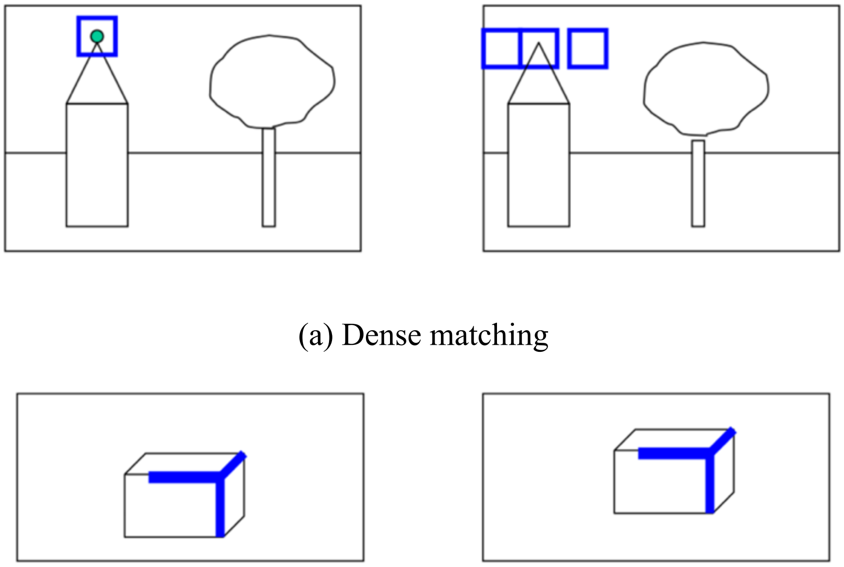
Correspondence issue can mainly refer to a term of matching points. Stereo matching procedure as the primary step plays a significant role in depth estimation. The task devotes to discover the projections of real points in stereo images which normally appears as two points have correspondence on their attributes.

2.5.1 Stereo matching theory

A great range of interests has focused on exploring the field of matching stereo correspondences. As the process of recognizing correspondences happens between two images, in most cases, one of them has to be chosen as a reference image. The correspondences are accordingly estimated by scanning points (in a paired image) that appear to have the same identity in regard to a reference point (in reference image) [52].

There are two types of stereo matching schemes in respect to pixel or feature based, that is to say, dense and sparse, and the illustration giving a schematic explanation is shown in Figure 2.5.1.1 [53].

Figure 2.5.1.1. (a) demonstrates an instance of dense matching theory that making use of pixel identification in regard to its paired possible pixels. Stereo dense correspondence normally performs matching methods on pixel level which can retrieve details of a view as much as possible. Figure 2.5.1.1. (b) illustrates the points of interests that sparse approaches concentrate on which term of correspondence attempted to tackle. Sparse correspondence matches feature in the form of various zones that have extrusive appearances e.g. segments, SIFT (scale-invariant feature transform) [54] and speeded up robust features (SURF) [55], which can play a discriminative role between different relationships in a scene [5], [53].



(a) Dense matching

(b) Sparse matching

Figure 2.5.1.1 Correspondence scheme

The scheme of dense matching can produce disparities for the entire image by searching correlations between left and right images in the way of pixel-by-pixel examining, as this trait, dense matching is more required by task regarding entire plane analysis. Sparse methods deal with a certain number of segments with given specific definitions in advance, such as edges, corners and so on, which adapt to partial area enhancement. The dense approach is good at the whole surface estimation in contrast to the sparse approach perform more on local features [56], [57].

On account of the distinct outputs, dense based approaches demand heavy processing resources to support analysis of a huge quantity of data stream. In order to achieve speedy performance, the progress of dense matching issues in serious computational cost, while in contrast, sparse correspondence is less affected by this restriction due to only cope with a small amount of feature points, however, the most range of interests currently requests as much detail as possible on the global surface level like applications of robotics [5], [58]. For this reason, dense correspondence is concentrated on by most reaches at the present time.

According to these comparisons, stereo correspondence based on the dense strategy attracts our interests and therefore is one centre study of our research by considering the advantages of dense output that producing global disparities for every pixel in a scene.

As indicated by [57], a classic stereo matching method implements in four stages as listed below: (1) Computation of matching cost, (2) Aggregation of cost, (3) Optimization/ computation of disparities (4) Refinement of disparity. The practical stages may vary in relation to the diversity of concrete applications. On the basis of these fundamental steps, many researchers have carried out studies on investigating algorithms for stereo matching. Mainly, there are two groups of stereo algorithms: Local algorithms and global algorithms [57].

Generally speaking, in respect of the four stages, local algorithms normally complete with first three stages step-by-step, however, global algorithms sometimes unite stage (1) and (2) following by stage (3). Moreover, the methodologies of disparity optimization regarding local and global methods distinguish from each other. Local methods normally adopt a scheme called Winner-Take-All (WTA) [57], [45], [59] to estimate disparities. The principle of WTA is to take pixel with the minimum value of aggregated matching cost as correspondence in relation to its reference pixel. Global algorithms determine the correspondence in the way of estimating a disparity value that can minimize a function of global energy which is known as energy function minimization approach [57], [45].

Both local and global methods have their own special advantages and disadvantages [60]. Local algorithms generally utilise the block matching method that cost is calculated on the basis of the windowed pixel block, which makes efficient process on high textured areas with low computational cost but lack accuracy on the occluded and less textured areas. The theory of global algorithms is to compute the minimum cost of energy functions following with the result of higher accuracy subject to ambiguities, however in exchange for processing speed consuming.

Considering global algorithms mostly require specific environment such as high-performance hardware to make effective, our research explores algorithms on the basis of the principle of a local strategy for the system design as our goal is to create low-cost stereo architecture.

2.5.2 Common stereo correspondence algorithms

Along with the development of stereo vision, the diversity of stereo correspondence algorithms have been created, among a variety of approaches four groups can be used for presenting current study categorizes which are local, global, semi-global algorithms and cooperation of local and global [61].

The first stage of the stereo correspondence approaches involves matching cost computation. The most common ways of local algorithms formulate matching cost with squared differences and absolute difference (simple algorithms - sum of squared differences (SSD) and sum of absolute differences (SAD) [45]), and normalized cross correlation (NCC) [62].

NCC that finding the disparity with the best correlations can increase accuracy in contrast to SSD and SAD which are sensitive to intensity changes, nevertheless, it is inclined to mismatch depth discontinues that lead to fuzziness on such regions in consequence [63], [64]. Paper [65] proposes a method for the purpose of improving the performance produced by NCC called summed NCC in two steps: (1) to calculate normalized cross-correlation, (2) to sum values of normalized cross-correlation in order to deal with issue that SSD and SAD as the most traditional but are lack ability of handling changes of intensity computation function of matching cost.

A non parametric transforms called Census Transformation that estimate the order relationship between pixel intensities is proposed as an possible matching cost function to tackle disparity estimation of outliers with the advantage of being independent from intensity data [64], [66], which is then modified with different scan patterns such as an improved census transformation integrates with pattern of star scan introduced by [67] for the purpose of dealing noise sensitive issue.

Cost aggregation approaches aggregate computed matching cost in the way of computing sum or average of chosen areas for the final decision with WTA approach [57]. Widely used approaches involve shape based adaptive support window, segmentation support and adaptive support weight that are the widely adopted aggregation methods [68], [69].

Normally there are two forms for shape support window: rectangular and constrained which are sensitive to depth discontinues. Paper [69] introduces a systematic stereo matching algorithms for dense disparity estimation adopting Census Transformation aggregated with the method on the basis of cross-based window that is proposed by [59] for computing shape adaptive full support region with varying scale polygon. Segment based support approach is suggested as an optional method to overcome the issue caused by depth discontinues between arbitrary shapes [70], [71], which forms adaptive support windows with segmentations with random shapes and size. These methods follow the idea of finding an optimal window in contrast to adaptive support weight method utilises the way of regulating pixel weights within a predefined specific support window, moreover, performs more computational cheap and higher accuracy than the approach of adaptive support window [72].

Research [73] proposed an adoptive support weighted window method implements aggregation procedure with support weights formulated by the proximity of geometry and the similarity of colour. An improved approach introduced by paper [61] that two initial disparity maps for left and right images are generated respectively steps: cost calculation using Census Transform function, cost aggregation exploiting successive weighted summation function based on the similarity ratio of intensity to obtain horizontal and vertical support and the pixels that have the minimum aggregation costs selected as the matching pixels. Bilateral filter [74], guided filter [75] and furthermore recursive edge-aware filter are as well as commonly used to compute adaptive weights with the benefits of edge aware capability [76].

Local algorithms can produce a high performance on high textured areas but occurring disparity noises on depth discontinues, low and repetitive textured areas, and occlusions as local methods determine optimal disparities depending on support windows which cannot conclude enough global information. Global algorithms can overcome these issues in the way of global energy estimation.

In the field of global algorithms, a global energy function consists of data and smoothness energies. Data energy estimates the compatibility between disparity function and image pair, and smoothness energy encodes the smoothness of disparity solutions in respect to piecewise smooth [57]. The term of data energy involves the integration of cost e.g.: square and absolute difference, difference, mutual information [77] and census transformation. For smoothness estimation, Markov Random Field (MRF) method with the advantage of discontinue preserving has been commonly used for smoothness energy encoding [57], [78], [79].

After formulating global energy function, energy minimization is estimated by various optimization approaches. One of the conventional ways is to utilize MRF [57], moreover, belief propagation (BP) can be used to solve the Markov network [80], for example [81]. Dynamic programming [57] and graph cuts [82] play as well as popular roles in optimization algorithms.

Dynamic programming computes the global minimization in the way of finding a path of minimum matching cost in a cost volume integrated with two scan lines that are corresponding to each other [57], [83]. Graph cuts approach performs minimization by mapping energy function to a specific graph and minimises the energy from the minimum cut [84]. Graph cuts produce precise disparity map in contrast to dynamic programming that causing streaking issues.

Apart from local and global approaches, there are also approaches perform in the form of semi-global which adopts a way of the cooperation of local and global algorithms. A semi-global approach is proposed by the study [85] that using mutual information as matching cost and aggregating cost with a global energy function to compute pixel level correspondence with WTA. Generally speaking, a typical algorithm of local and global cooperation performs matching with blocks of the 2D window first and then global computation with a volume in the form of the 3D box [86].

Besides traditional approaches, machine learning in the form of an effective approach presenting the ability of learning complex as well as attracts attention and has been fused with local and global theories to make effective in improving the performance of methodologies for stereo depth estimation at the recent period [87]. For examples, [88] presents a work that using learning conditional field to formulate the relationship between smoothness and the changes of color for energy function, and [89] learns non parametric cost function with structured support vector machine, and a method of cost aggregation that exploiting hidden markov tree is proposed by [87].

Among a variety of matching learning technologies, neural networks and deep learning such like convolutional neural network as advanced learning techniques on the basis of imitating human nervous system attracts researchers interest and is also exploited to solve stereo correspondence problem such like proposed by [90], [91]so as to improve the performance by offering artificial intelligent capability. The details referring to this field will be explicated in Chapter 4 and Chapter 5.

2.6 Chapter summary

Computer vision as a technique that can endue machine with intelligent visual perception capability has participated in a wide range of artificial intelligent applications along with the historical line up to the present time.

The theory of computer vision has been presented by this Chapter and technologies in respect to an important field of visual system which is depth estimation for 3D reconstruction were depicted. Generally speaking, the types of vision systems include single vision and multiple vision. This Chapter has included literature reviews of monocular vision, and in particular, the stereo vision for depth estimation. The principle of stereo visual perception and widely used algorithms referring to stereo correspondence for stereo views reconstruction were systematically explicated in this Chapter.

Chapter 3: Feature construction and datasets

Feature extraction is the first step of stereo correspondence which normally refers to obtain information from rectified pairs of stereo images as mentioned in Section 2.4.2. This Chapter will introduce a basic pipeline of feature extraction from stereo pairs on the basis of the theory of image rectification without camera calibration. And then following this pipeline, the estimation of the effectiveness of feature selection schemes will be presented. Moreover, different types of input features will be evaluated for model determination. In respect to the experiments carried out, this Chapter will also explicate the adopted datasets that participating in our study.

3.1 Image rectification with epipolar geometry

For the purpose of reducing the complexity of correspondence searching, stereo images are normally preprocessed with rectification algorithms to convert the 3D computational problem into the 2D matter [92]. This rectification procedure transforms images on the basic principle of epioplar geometry as the schematic in Figure 3.1.1 [93].

Hartley and Zisserman indicated that: "The epipolar geometry is the intrinsic projective geometry between two views" [8]. There are three main concepts in epipolar geometry: epipolar plane, epiploar line and epipoles.

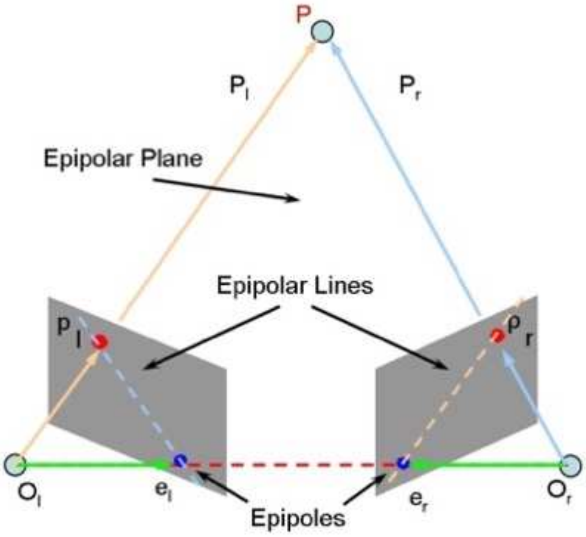
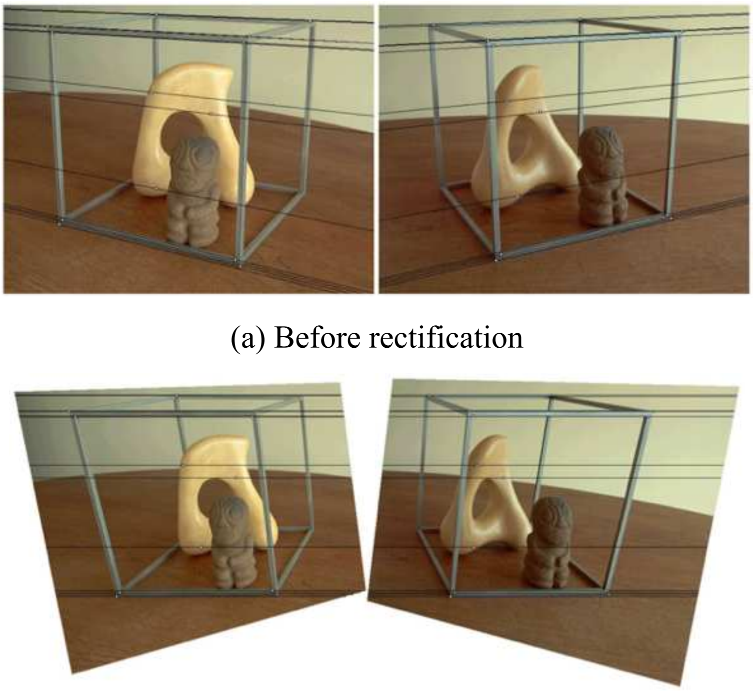


Figure 3.1.1 Epipolar geometry

The epipolar plane is a geometry presentation formed by lines concatenating three points occurring while intersecting at image planes [7], [8], [50], [93], [94]. As shown in Figure 3.1.1, O_l and O_r are the origins of two cameras, and between them is a line known as the baseline. The projection rays from O_l and O_r to intersection point P interest and create two points (P_l and P_r) on two image planes which are the projection points of P on images. Three lines (O_lP, O_rP and O_lO_r) together construct the plane of epipolar geometry. The crossing points that are produced by a line of O_lO_r passing through two image planes, in another term, the intersections of epliploar plane and image planes are epipoles (e_l and e_r). The lines from P_l and P_r to e_l and e_r are so denoted as epipolar lines. P_l and P_r are corresponding points with each other in stereo image pair.



(a) Before rectification

(b) After rectification

Figure 3.1.2 Image rectification

Stereo image rectification aims to transform two epipolar lines of two corresponding points into one epipolar line which is parallel to the axis of horizontal so that these two matching points position on the identical coordinate of row. In consequence, all the epipolar lines on rectified images should be parallel to each other as shown in Figure 3.1.2 [94]. Figure 3.1.2 (b) shows that the epipolar lines on rectified images identically locate on a parallel axis to horizontal in contrast to Figure 3.1.2 (a) that before rectification. Rectification process is achieved by a transformation rule of Formula 3.1 [94], that is to say, an epipolar constraint.

As the Equation 3.1 indicates, the corresponding point pair can be retrieved by transformation F which is commonly called fundamental matrix, accordingly, the epiplolar lines between corresponding points in left and right images are linked by the Fundamental matrix. Common algorithms estimate coefficients of the fundamental matrix by given a certain number of matching pairs, in which the typical algorithm is known as eight-point algorithm. Furthermore, another term of fundamental matrix is an essential matrix in the case of when the transformation is formulated with calibrated parameters [7], [8], [50], [93], [94].

$$P^{r^T} F P^{l} = 0 \tag{3.1}$$

The transformation matrix can be defined with or without parameters from camera calibration. In this case, rectification procedure can be applied on calibrated or un-calibrated images.

3.2 Feature extraction from un-calibrated images

A standard process of feature extraction from un-calibrated images presented by the paper [90] has made a guide on how to extract features from un-calibrated images for our study. Moreover, a toolbox provided by Mathworks that can implement rectification with un-calibrated image, which includes the most popular estimation algorithms and very handful functions for experiments on such topic [47].

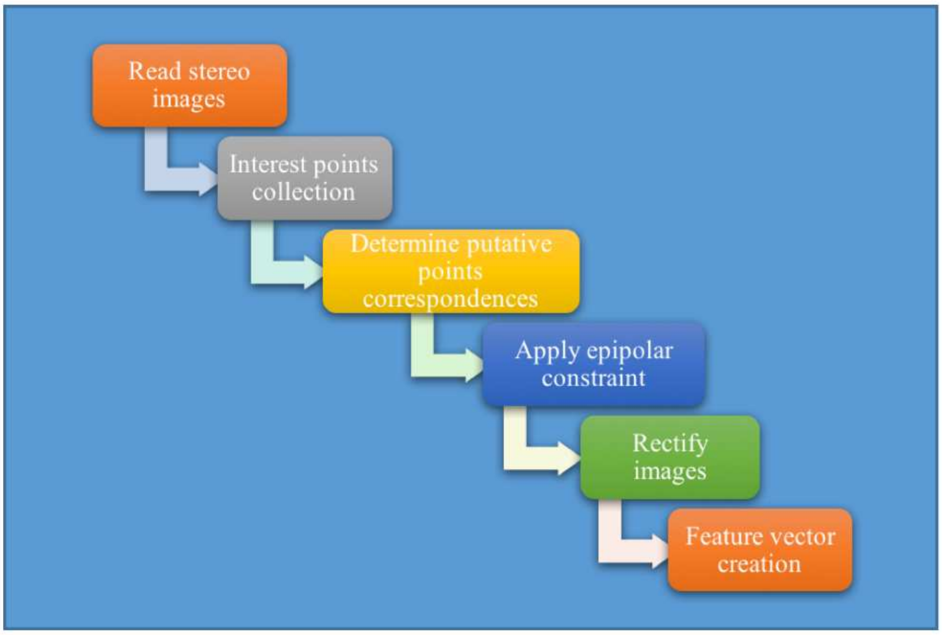

Read stereo images
Interest points collection
Determine putative points correspondences
Apply epipolar constraint
Rectify images
Feature vector creation

Figure 3.2.1 A pipeline of feature extraction form un-calibrated images

A pipeline proposed by such scheme and toolbox of feature extraction in our research is illustrated in Figure 3.2.1. The whole process can be mainly grouped into six steps: stereo images reading, interest points obtaining, putative correspondence determination, epiploar constraint implementation, stereo images rectification, and feature vector construction. In our experiments, the implementations from step one to step five were accomplished by the toolbox [47]. The detailed procedures are explained below:

1) Read stereo images:

When the system is given a pair of stereo images, the first step is to load the stereo images in the way of converting the colour images into grey scale images (from RGB images to one channelled intensity images) so as to prepare for obtaining interesting points that possess characteristic attributes.

2) Interest points collection:

On account of the transformation matrix for rectifying two stereo images needs to be determined on the basis of correspondences between points, the interest points have to be gained. On the second step, interest points between stereo images are extracted with SURF features in both left and right images. Those points can then be used to find possible points correspondences in a pair of stereo images.

3) Determine putative points correspondences:

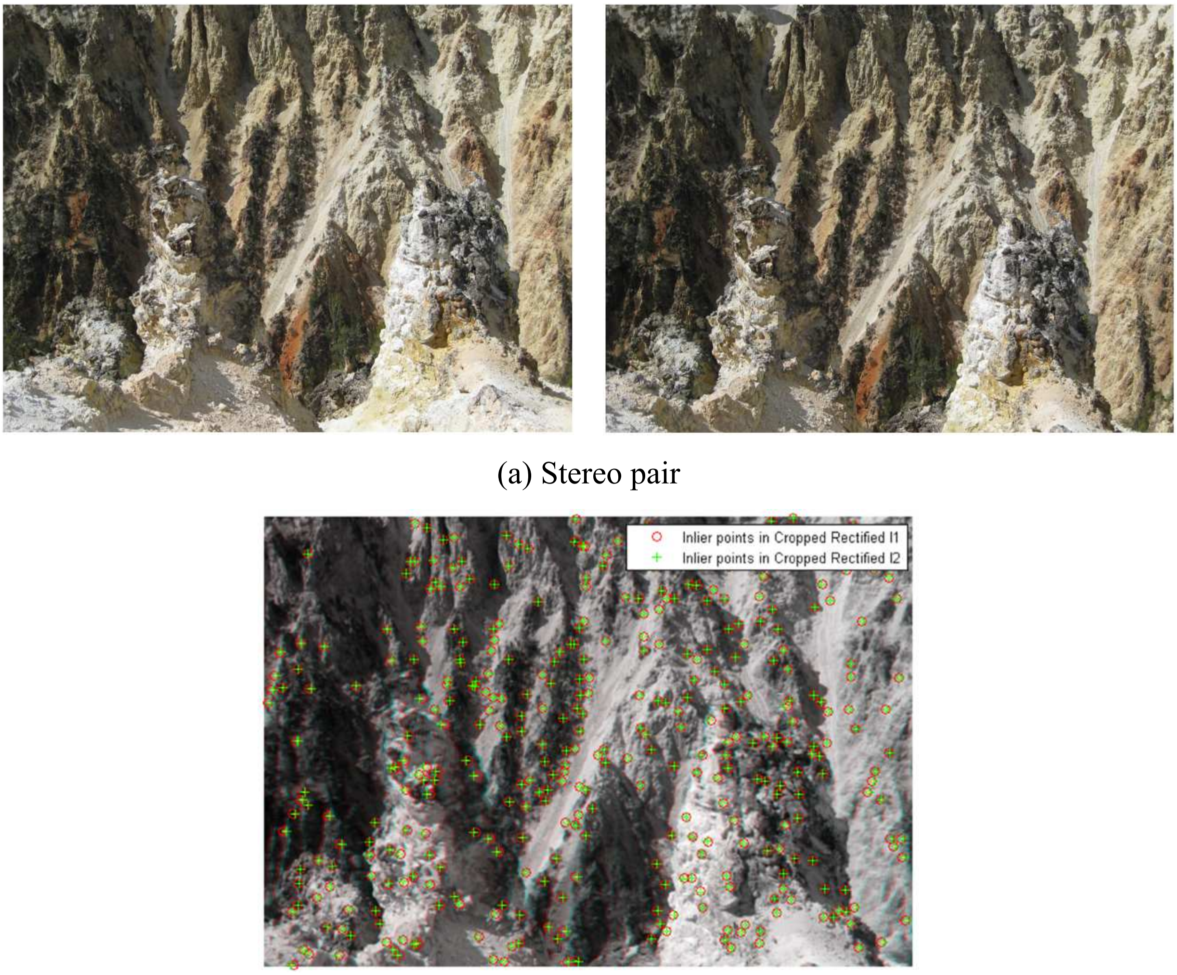
Once the interest points are found, SURF features are formed into a vector for putative matching computation. Accordingly, the putative correspondences of these points are determined with SAD. At this point, each point has its corresponding point locating in the other image. All the locations of matched pairs of points in left and right images are then recovered for further estimation.

4) Apply epipolar constraint:

According to the theory of epipolar geometry, matched points must fit with epipolar constraint. Although the most results of matching satisfy the condition, there are some that do not meet the constraint. This step implements epipolar constraint with matched pairs to refine the output from step three in the way of estimating whether two matched points can lie on the same epipolar line by adopting fundamental matrix computation. These output pairs of matched points then can be stored for training dataset preparation.

5) Rectify images:

The stereo images can now be rectified with the fundamental matrix generated from the previous step. The toolbox computes two projective transformations from the fundamental matrix, and then rectifies the stereo images into a common image plane, moreover, normally the stereo images are cropped with the common after images are rectified. The results are shown in Figure 3.2.2, which the data images are from the toolbox [47]. Furthermore, the transformed location of matched points can be retrieved through projective transformations.


○ Inlier points in Cropped Rectified I1
+ Inlier points in Cropped Rectified I2

(a) Stereo pair

(b) Rectification results

Figure 3.2.2 Example of un-calibrated rectification

6) Feature vector creation:

At this moment, all corresponding points locate in the same horizontal line in left and right images. The feature vectors can be constructed in the formation of 2D feature extraction from rectified image pairs. First is to select reference points in one of the stereo images. Secondly, in respect to the reference point, a set of points that locate at the same horizontal line (the same number of row) in the other image are obtained for matching estimation. All location of points is converted into indices of pixels by using the projective transformations and then each pixel is formed with a specific designed type of features. The formative features from a pixel constitute an input feature vector for this pixel in the form of combination. This general pipeline contributes handful procedures for creating feature vector and has especially helped our research with estimating the selection of training feature. Next section will present selection analysis on the set of training feature.

3.3 Schemes of data selection for training

The purpose of our research is to estimate the effectiveness of stereo correspondences estimation with neural networks and deep learning. On account of this objective, the designs of our systems have found on the principle of machine learning. The primary stage of such a learning system involves the training data preparation. The more effective and efficient training data, the better performance occurs.

In consideration of such aspect, how to sufficient select training data plays an import role. This section will evaluate the approaches of feature set selection on construction training dataset. Regarding the characteristics of stereo matching, both matched and unmatched pairs are used for training to improve knowledge level of learning ability. The two schemes estimated in our project are hereby denoted as below:

- Random based selection scheme (RS) based on [90].
- Stereo and rectification constraints based selection scheme (SRC) proposed in our study.

The paper [90] proposed a selection scheme that training data of matched and unmatched pairs are randomly selected after initial matching points. Nevertheless, this scheme does not include the consideration of stereo and rectification traits that correspondence between two pixels in stereo images generally appears within a certain range of disparities on the same number of row, therefore, we have designed another scheme to achieve data selection on the basis of such stereo and rectification constraints. With our scheme, the pairs must meet these constraints so that they can be eligible to be used for training.

3.3.1 Two selection schemes for matched training data

In respect to the learning capability of correspondence, matching relation is considered at the first place, therefore, matched pairs are involved in training. The selections are performed on sets of initial corresponding points generated from the pipeline of feature vector extraction introduced previously. The qualifications for training are then determined with the following strategies.

The basic strategy for both RS and SRC:

The two matched pixels of a matched pair in left image and right image must not be on the boundary of their images in order to assure the matched pixels can have required feature window, as features are extracted within a certain size of window in our designed systems.

Strategies for RS:

The set of initial matched pixels used by RS scheme is retrieved from Step Two in the pipeline presented in Section 3.2. According to [90], RS adopts the way of random collection without any extra constraints. Consequently, matched pairs for training are arbitrarily selected from this initial set.

Strategies for SRC:

1) The row indices of the two matched pixels of a matched pair in left image and right image must be equal. For this reason, the training pairs regarding matching traits are selected from the initial set of matched pairs obtained from Step Three in Section 3.2.

2) The two matched pixels of a matched pair in left image and right image are not black pixel generated by rectification filling the extra empty boundary and their corresponding unmatched pixels are also not black pixel generated by rectification, which is also the reason of directly selecting matched pairs from rectified images applied with procedure of common area cropping.

3) For every pixel in matched pairs, at least there is one unmatched pixel in the maximum disparity range except for its corresponding matched pixel in the other image and is not beyond and not on the boundary of the other image to make sure each pixel of this matched pair is able to have an unmatched pixel with demand feature window in the other image.

3.3.2 Two selection schemes for unmatched training data

With provision for the diversified and flexible capability of system recognising the relationship between pixels, training should not only involve matched relation but also an unmatched relation. In order to increase the flexible ability, unmatched pairs are as well as picked up for training the system to acquire the ability to distinguish the traits of dis-correspondence.

The basic strategy for both RS and SRC:

This is the same as matched pairs selection. Unmatched pixels in stereo images must not be on the boundary of respective images so as to guarantee unmatched pixels can produce a demanded feature window.

Strategies for RS:

As described in [90], unmatched pairs for training are as well as picked up randomly. Accordingly, RS employs the way of arbitrarily selecting an unmatched pixel for each pixel in a matched pair respectively, which the matched pair is generated from the selection of matched data set for training. Thereupon, every pixel from a matched pair has its unmatched pixel to for an unmatched pair.

Strategies for SRC:

This strategy selects unmatched pixels in the maximum disparity range based on corresponding pixels in selected matched pairs (produced by SRC scheme) to constitute unmatched pairs. Supposing a matched pair contains (Pixel1, Pixel2), which Pixel1 is from in Image Left and Pixel2 is from in Image Right. The index of Pixel1 is (r1, c1), and the index of Pixel2 is (r2, c2), where {r1, r2} indicates the number of row and {c1, c2} indicates the number of column. According to the rectification theory, r2 is equal to r1. The maximum disparity (MaxDisp) points out the maximum distance of a potential matched pixel in the other image regarding its reference pixel. The unmatched pixel index in the maximum range is (r_m, c_mith), which c_mith ∈ c_m. The method of selecting unmatched pixels to form unmatched pairs are showing below:

a. Using Pixel1 as reference pixel:

   The indices of unmatched pixels in the maximum range are:

$$\text{In Right Image} \begin{cases} \text{r\_m} = \text{r1} \\ \text{c\_m} = \{\text{c1, c1-MaxDisp}\} \end{cases}$$

   If (r_m, c_mith) in Image Right is not beyond and not on the boundary of Image Right, which c_mith ≠ c2, then a matched pair consists of:

((r1, c1) in Image Left, (r_m, c_mith) in Image Right)

b. Using Pixel2 as reference pixel:

   The indices of unmatched pixels in the maximum range are:

$$\text{In Left Image} \begin{cases} \text{r\_m} = \text{r2} \\ \text{c\_m} = \{\text{c2, c2+MaxDisp}\} \end{cases}$$

   If (r_m, c_mith) in Image Left is not beyond and not on the boundary of Image Left, which c_mith ~= c1, then a matched pair consists of:

((r2, c2) in Image Right, (r_m, c_mith) in Image Left)

According to the methodologies of RS and SRC, RS has fewer constraints comparing to SRC, however, however, SRC focuses on revealing the specific character of stereo vision itself. The evaluation between RS and SRC performances will be presented in the next section.

3.3.3 Experimental evaluation of two selection schemes

The RS and SRC schemes have their own characteristics. The dataset built with RS have the traits of randomness, in contrast to the dataset constructed on the basis of SRC. The SRC dataset involves the special characteristics of stereo vision in relation to epipolar geometry. In order to have a sense of the effectiveness of these two schemes, this section investigates the performance of RS and SRC schemes.

Experimental process:

We adopted a way of evaluating the training and test performances using two schemes respectively to find out the effectiveness. The output performances can show the distinguish outcome. With the help of these results, the evaluations of the two schemes can be carried out. The experimental process was mainly divided into three stages: construction of feature vector set, training and test sets division, and neural network training, which are introduced as below:

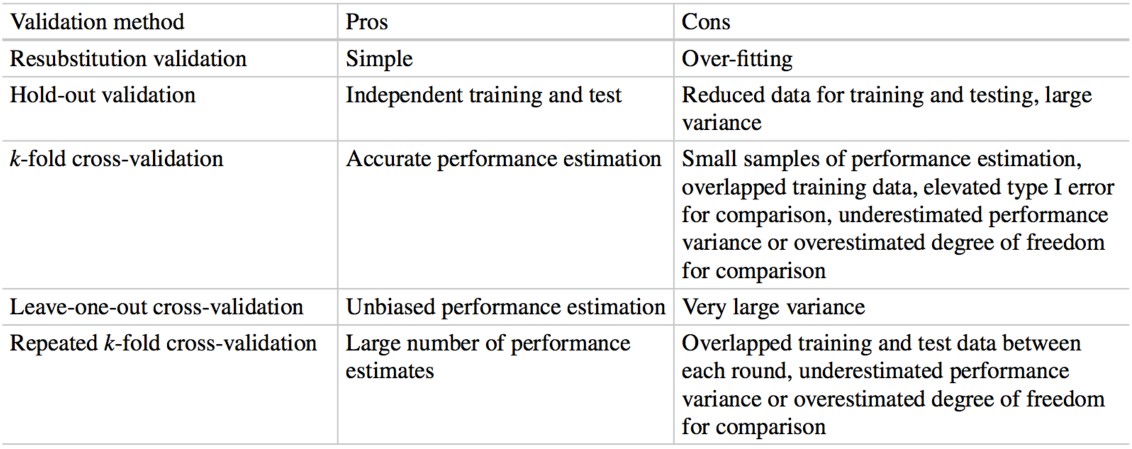
- The first stage built two feature vector sets for training following the pipeline with RS and SRC schemes as described in previous sections. The final output of feature dataset creation produced two sets that have the same size of training samples on the selection of both matched and unmatched pairs.

- Normally the learning procedure of machine learning requires training and test sets to estimate the performance. The produced feature datasets from stage one were then divided into training and test sets. The procedure of splitting datasets to form training and test sets were implemented according to the principle of cross-validation which denoted percentages for training and test sets.

- The structure of neural network adopted for training was the one proposed by [90]. The network consisted of one input layer, one hidden layer and one output layer as shown in Chapter 4 Section 4.3 The split datasets were trained and test for k times regarding cross-validation with this simple network to produce average performances for both RS and SRC based learning.

Performance evaluation with cross-validation:

Cross-validation provides statistical evaluation for learning techniques in the way of splitting the dataset into two parts, that is to say, one is created for training and the other one is for model validation [95].



Validation method	Pros	Cons
Resubstitution validation	Simple	Over-fitting
Hold-out validation	Independent training and test	Reduced data for training and testing, large variance
k-fold cross-validation	Accurate performance estimation	Small samples of performance estimation, overlapped training data, elevated type I error for comparison, underestimated performance variance or overestimated degree of freedom for comparison
Leave-one-out cross-validation	Unbiased performance estimation	Very large variance
Repeated *k*-fold cross-validation	Large number of performance estimates	Overlapped training and test data between each round, underestimated performance variance or overestimated degree of freedom for comparison

Table 3.3.3.1 Pros and cons comparisons between cross-validation algorithms

As the taxonomy given by [95], there are six sorts of methodologies in the family of cross-validation: resubstitution validation, hold-out validation, k-fold cross-validation, leave-one-out cross-validation and repeated k-fold cross-validation, and the comparisons between their pros and cons are given in Table 3.3.3.1. Resubstitution validation uses all the data for systems to learn and test with the same data. Hold-out validation holds out one part of the dataset as a test set which does not participate in training.

K-fold cross-validation splits dataset into k sets that have the same size and performs k times of training, which uses k-1 sets for training and one set for the test each time, furthermore, if k equals to the total number of samples in the dataset, k-fold cross-validation becomes Leave-one-out cross-validation. Repeated k-fold cross-validation repeats k-fold cross-validation more than one time and shuffles the dataset at the beginning of each turn of k-fold cross-validation.

In consideration of the efficiency and comparisons in Table 3.3.3.1, we adopted k-fold cross-validation to perform generalised performance estimation. Our experiments set the number of k equal to ten for k-fold cross-validation as ten is the most common value for k in machine learning estimation [95]. In this case, each dataset based on RS and SRC was split into ten equal set and trained with the neural network ten times where each iteration used nine sets for training and one set for the test. The final evaluation was computed as the average of performances of ten turns as shown in Equation 3.2.

$$\text{K-FCV} = \frac{1}{k}\sum_{i=1}^{k} P_i \qquad (3.2)$$

Where k = 10, and P is the performance of every iteration, and i is the number of iterations. The illustration of 10-fold cross-validation is given in Figure 3.3.3.1.

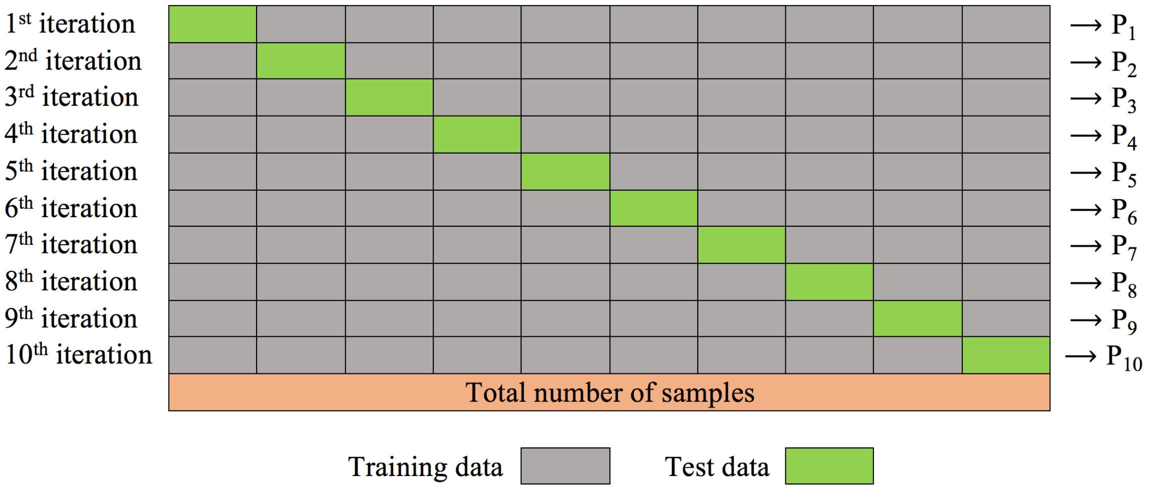


Figure 3.3.3.1 10-fold cross-validation

Each performance was computed with the most common algorithm of error estimation that is known as Mean Squared Error (MSE). Equation 3.3 denotes the computational theory of MSE.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(t_i\text{-}o_i)^2 \qquad (3.3)$$

Where n is the number of instances, i indicates which instance, t is the target and o is the output from the network. By combining with the 10-fold cross-validation, the computation can be defined as Equation 3.4.

$$\text{MSE-10-FCV} = \frac{1}{10}\sum_{i=1}^{10}\mathrm{MSE}_i \qquad (3.4)$$

The evaluation results of 10-fold cross-validation for RS and SRC based training are shown in Figure 3.3.3.2. From the results, we can have a view of the impact of RS and SRC on system learning. The stereo images pair used for this experiments was from [47].

In Figure 3.3.3.2, the shorter the bar is, the smaller the error is. As the results shown in Figure 3.3.3.2, performance of SRC produced shorter bar than RS, which means when choosing SRC scheme that involving specific selection regarding to the traits of stereo geometry to pick up training instances for dataset construction the system can be able to acquire higher possibility to generate better performance in contrast to exploit RS scheme that performing generalised selection.

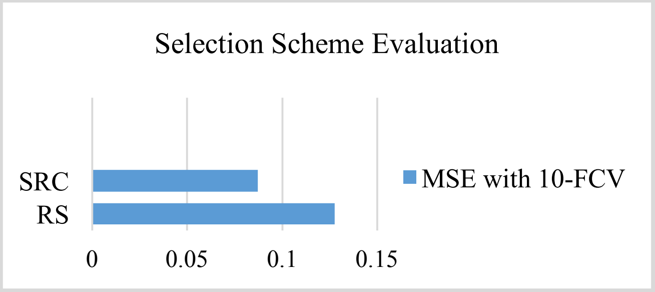


Figure 3.3.3.2 Selection scheme evaluation

Furthermore, this also can imply that stereo correspondence learning for neural network requires training data that can represent the characteristics of stereo vision rather than arbitrary representation. In other words, according to the results, SRC has appeared to be more suitable for the task of stereo matching. For this reason, all other experiments of this project adopted SRC strategy to carry out implementations.

3.4 Feature modality analysis

Features represent the traits of pixels so as to help system to study in the way of target analysis. Different applications may demand diversified feature engineering. The inspiration of features design for our research was derived from the paper [90].

3.4.1 Basic Feature design

While it comes to the stage of extracting features from captured images, the primary matter that is normally considered at the first point involves the decision referring to which type of images to be utilised in terms of shades: colour and grey scale [10]. A colour image is made up of three channels (red, green and blue), that pursuantly including more abundant data in comparison of a grey scale image represented by one channel. Both colour and grey scale images are possessed of their own advantages and disadvantages.

With a plenty of information provided by three channels of colour images, performance can easily reach to a higher point than single channelled grey scale image. Nevertheless, the operation time is raised due to a bigger amount of data generated from RGB images, on the opposite side, grey scale images show its preponderance on this aspect that less information conduces elapsed duration.

A grey scale image can be converted from RGB channels. Different applications require different input, in other words, the chosen of RGB and grey scale depends on practical matter. Our research utilises grey scale images by considering the processing time and the computational engineering cost of RGB images. Once images are converted into grey scale, the attributes representing characteristics information can be extracted in a way of simplified implementation.

In machine learning, the input feature vectors are made up of series of attributes for the computer to learn the properties through specific cues. There are three differential features suggested by [90], which are the most commonly used attributes for grey scale image: Intensity Differences, Magnitude Differences and Orientation Differences, where the terms of intensity, magnitude and orientation are defined as follows. Supposing the intensity (Int) of a pixel is f (x, y), then its gradient can be denoted as Equation 3.5, accordingly, the magnitude (Mag) of the pixel is formulated as Equation 3.6, and the orientation (Ori) is defined as Equation 3.7 [90]:

$$\mathrm{Int} = \left[G_x, G_y\right] = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] \tag{3.5}$$

$$\mathrm{Mag} = \left[G_x^2 + G_y^2\right]^{\frac{1}{2}} \tag{3.6}$$

$$\mathrm{Ori} = \tan^{-1}\left(\frac{G_x}{G_y}\right) \tag{3.7}$$

The differences of intensity, magnitude and orientation are computed as the subtraction between left and right features of pixels, supposing a pair of pixels from left and right images for correspondence estimation is (P_l, P_r), then the three features of this stereo pair are defined as below:

$$\text{Intensity Differences} = \text{Int}^{P_l} - \text{Int}^{P_r} \qquad (3.8)$$

$$\text{Magnitude Differences} = \text{Mag}^{P_l} - \text{Mag}^{P_r} \qquad (3.9)$$

$$\text{Orientation Differences} = \text{Ori}^{P_l} - \text{Ori}^{P_r} \qquad (3.10)$$

Pixels in each pair are extracted with these three features in a certain window. However, only the attributes from single pixel cannot provide enough clues in respect to the relations with surrounding context for the system to learn, for this reason, feature extraction of a pixel is obtained in a certain window, where the pixel is in the centre of this window [90]. The centre pixel and all other pixels belonging to the same window are all computed with three features so as to include more information. The differential features are then calculated with two feature windows of stereo pixels.

A schematic example of feature formation on the basis of the window approach is illustrated in Figure 3.4.1.1. In Figure 3.4.1.1, the orange squares combining with one green square represents a selected feature window of a pixel in an image, where the green square in the centre of the window is the representative of this pixel, all the orange squares are the neighbour pixels.

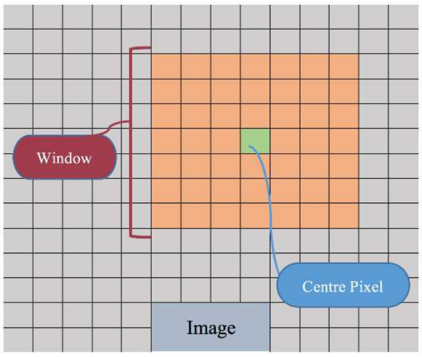


Figure 3.4.1.1 Window based feature

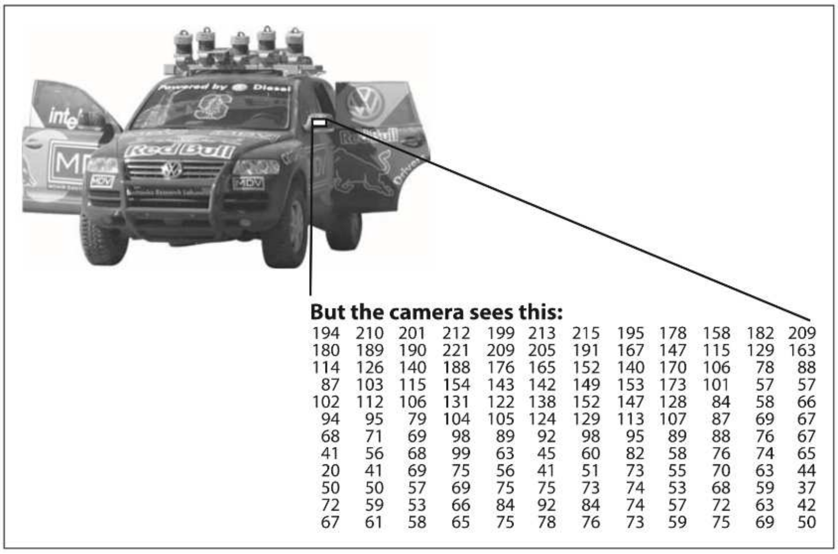
[90] suggests the size of feature window to be 7×7. Each 7×7 differential feature window is generated by left 7×7 window minus right 7×7 window, where two windows are centred with

P_l and P_r. The three differential Feature windows of a stereo pair are then formatted into a vector as the input of network, in the case here, the size of each differential vector is 49, moreover, the three differential feature vectors are combined into one final vector that has 147 attributes.

In our research these three features were computed as 7×7 for SNN and Multiple NNs and final vector utilized the approach introduced in [90], however considering the different input structure from SNN and Multiple NNs, CNN adopted different multiple window sizes and different input structure in the form of channels (details will be presented in Chapter 5).

3.4.2 Numerical data types of differential features and experimental evaluation

The computer sees the image in the form of numerical values. Figure 3.4.2.1 shows an example that how computer understanding a digital image given by Bradski and Kaehle [7], as this illustration, what human sense in this figure, a car is classified, whereas computer only can recognize a series of digital numbers.



But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	74	65
20	41	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Figure 3.4.2.1 Computer understanding a scene

In other words, the way of computer to comprehend images is to analyze binary data which images are turned into. Every patch of those numbers containing not only the actual information but also a lot of clattered data that disturbing the correct decision to be made by a computer. Even more, the state of affairs is to convert 2D input images back to the 3D concept as output, which makes the mission of computer vision rather difficult to be accomplished.

On account of this reason, the type of numerical values for the learning of intelligent system plays special roles in learning procedures, furthermore, the term of numerical types can be referred to the range of values such like from negative number to positive number in the form of integer that is more widely employed rather than double type. Double-number requires more computational resource than integer variables. Our research has utilised integer type for feature extraction.

Regarding the design of features for our research as presented in the previous section, the differential features were in the form of the results produced by substitution, in consideration of this property, there would be three types of numerical data that could be involved into consideration, which are 8-bit unsigned integer, 16-bit signed integer, and absolute integer as specifications below:

- The type of 8-bit unsigned integer (uint8) excludes negative numbers in the way of converting negative values to zeroes, and the rage is from 0 to 255. If the subtraction values were negative, then the numerical values of these features would be equal to zeroes.

- The rage of 16-bit signed integer (int16) begins with -32768 and ends with 32767, which keeps the negative values. Therefore, the negative values of the subtraction output can remain in the feature vectors.

- The approach using absolute integer extracts absolute values from subtraction results. By adopting this method, the negative values were kept in the form of positive values. The absolute difference is also one of the most common matching cost for solving the task of stereo correspondence.

Evaluations were carried out in order to observe the impact of these three numeric types on learning capability. Figure 3.4.2.2 shows the 10-fold cross-validation for this evaluation. The network structure utilised in training was the same as the one adopted by the evaluation in the previous section, which was introduced by [90]. Moreover, the evaluation was implemented with 10-fold cross-validation computed with MSE.

In Figure 3.4.2.2, lower bar means lower error generated, in other words, the higher bar represents lower performance. From the results we can see, int16 type produced the worst performance in three types, and the results of training using uint8 type closed to int16 method but appeared slightly better performance, among these results, absolute approach generated the best performance.

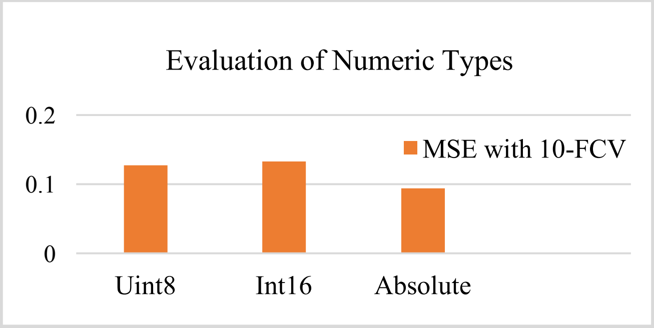


Figure 3.4.2.2 Feature numeric types evaluation

These three features contain different specific numerical information which could result in different performances. Numeric type with 8-bit unsigned integer contains the least amount of numerical values in these three types. The type of 16-bit signed integer has a more amount of data with a negative sign. The absolute type retains negative numbers by removing negative sign so as to maintain the same quantity of values as the method of 16-bit.

The results imply that system learning requires a certain amount of values that contain more numerical data rather than many zeroes, moreover, the negative values do not make an improvement on the performance, however, decrease the level of accuracy. In accordance with this outcome, our research has considered the scheme based on absolute principle to be preferred data type which are shown as follows:

$$\text{Absolute Intensity Differences} = |\text{Int}^{P_l} - \text{Int}^{P_r}| \tag{3.11}$$

$$\text{Absolute Magnitude Differences} = |\text{Mag}^{P_l} - \text{Mag}^{P_r}| \tag{3.12}$$

$$\text{Absolute Orientation Differences} = |\text{Ori}^{P_l} - \text{Ori}^{P_r}| \tag{3.13}$$

3.5 Datasets

Apart from the stereo images Mountain used in Section 3.2, the Middlebury benchmark [96] provides a database contains several datasets for implementation of stereo matching experiments, which are extensively popular datasets used in the community of stereo matching research, moreover is the main dataset used in our research. Most researches have been carried out with these datasets. The datasets have been used for our research are 2003 Datasets [11], 2005 Datasets and 2006 Datasets [63], [88].

The 2003 Datasets include two pairs of stereo images which also have been used in our study: Cones and Teddy. There are three sizes for each pair as listed below. In consideration of time consuming, our research used quarter size of this datasets as large sizes demand high computational resources to support. The two sets of stereo images and their left disparity maps are shown in Figure 3.5.1.

- Full size: 1800 × 1500
- Half size: 900 × 750
- Quarter size: 450 × 375

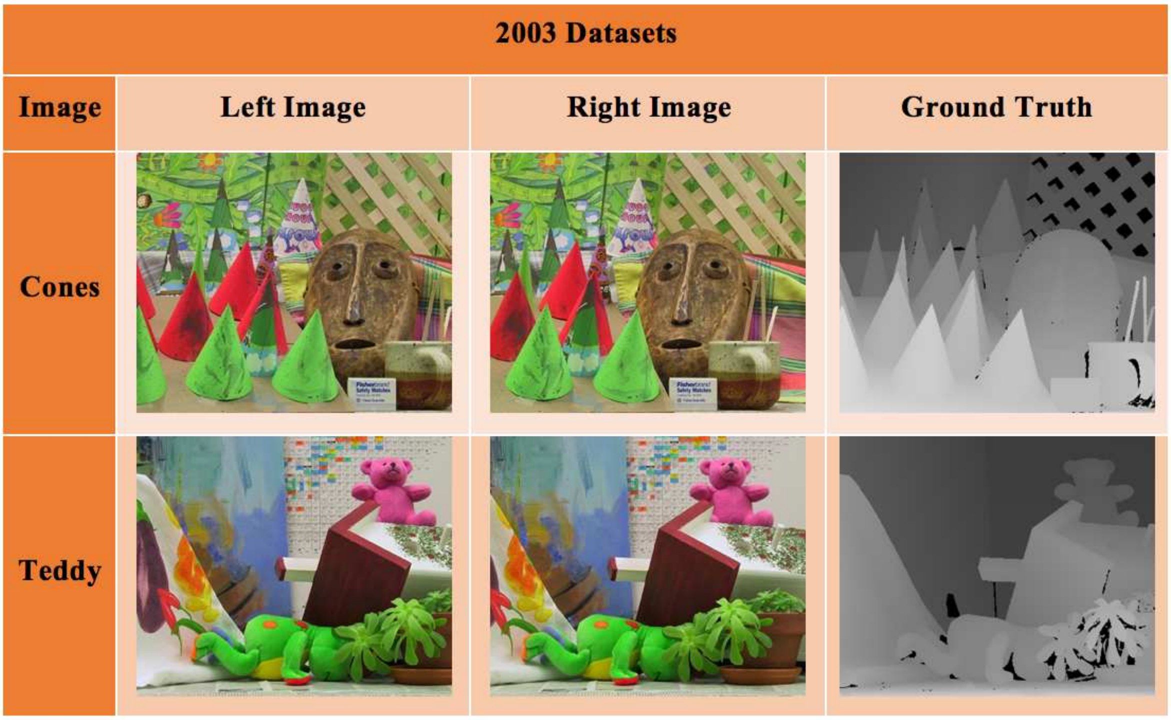
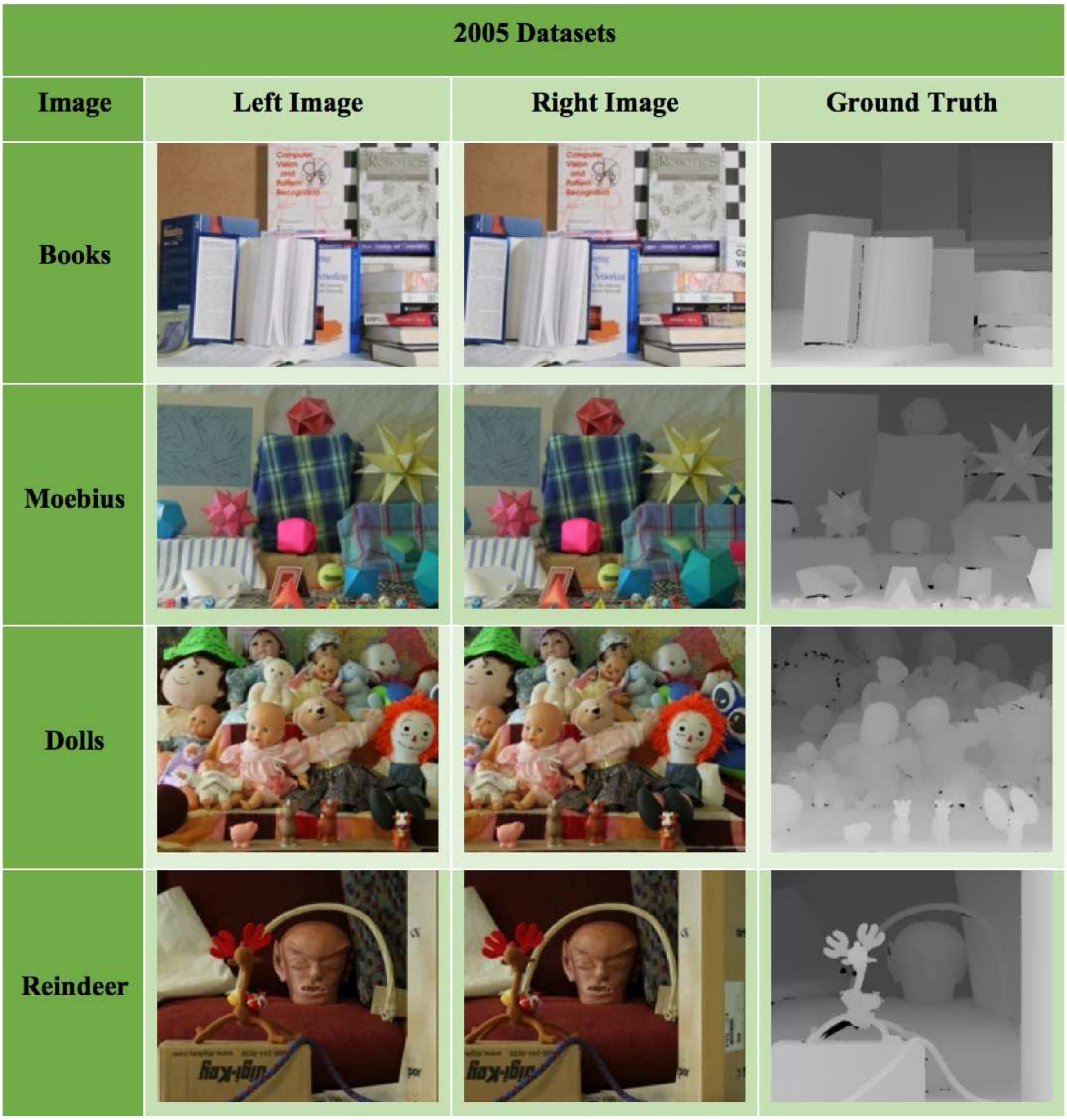


Figure 3.5.1 Stereo pairs from 2003 Datasets

2005 datasets and 2006 datasets were created for increasing the stereo pairs for algorithms test by Middlebury benchmark. Six pairs of stereo images in 2005 Datasets were utilised in our research: Books, Moebius, Dolls and Reindeer. Our research used five pairs from 2006 Datasets: Aloe, Baby3, Bowling2.

Figure 3.5.2 and Figure 3.5.3 illustrates pairs and their corresponding left disparity maps that have participated in our study, which are from 2005 Datasets and 2006 Datasets.



2005 Datasets

Image	Left Image	Right Image	Ground Truth
Books			
Moebius			
Dolls			
Reindeer			

Figure 3.5.2 Stereo pairs from 2005 Datasets

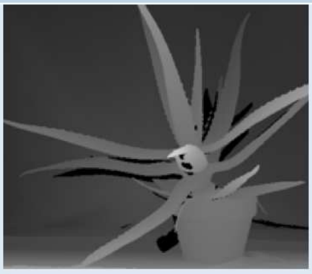
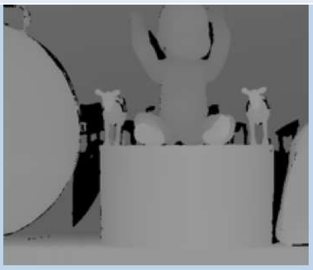
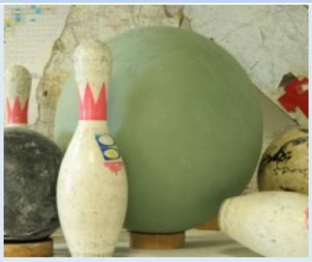
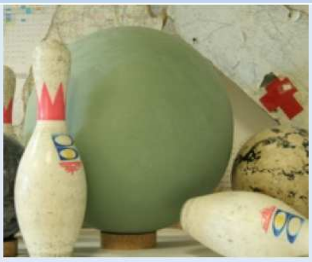
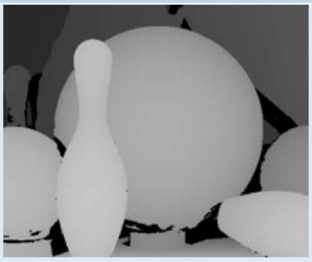
2006 Datasets			
Image	**Left Image**	**Right Image**	**Ground Truth**
Aloe			
Baby3			
Bowling2			

Figure 3.5.3 Stereo pairs from 2006 Datasets

The image sizes for 2005 Datasets and 2006 Datasets are as follows, moreover the third size of these two datasets was chosen by our experiments for the purpose of low computational cost:

- Full size: $2005\mathrm{D}\ (1330 \ldots 1390) \times 1110$, and $2006\mathrm{D}\ (1240 \ldots 1396) \times 1110$
- Half size: $2005\mathrm{D}\ (665 \ldots 695) \times 555$, and $2006\mathrm{D}\ (620 \ldots 698) \times 555$
- Third size: $2005\mathrm{D}\ (443 \ldots 463) \times 370$, and $2006\mathrm{D}\ (413 \ldots 465) \times 370$

The maximum disparity defines the possible distance for the corresponding pixel referring to its reference pixel. The Middlebury datasets have different maximum disparity ranges, the specifications given for each dataset are: 2003 Datasets are 64 pixels, 2005 Datasets and 2006 Datasets are 80 pixels. These disparity ranges have been used for feature extraction and disparity computation in our research. Due to the sizes of datasets utilised were not full size, the actual disparity values should be encoded with a scale factor, where 4 for 2003, 3 for 2005 Datasets and 2006 Datasets in accordance with the selected image sizes.

The Middlebury datasets were used throughout our experiments. Firstly, provided datasets were implemented with feature analysis for investigating better suitable feature engineering. Secondly, training and test datasets were extracted from those datasets for the experiments of improvement on the level of accuracies regarding learning capability that can be affected by system parameters. Furthermore, a series of methodologies for network architectures and disparity computations with different system designs were evaluated with Middlebury datasets so as to observe the effectiveness of our proposed systems.

3.6 Chapter summary

A systematic pipeline of feature extraction has been explained. This pipeline can extract feature vectors as the input of intelligent systems in the way of implementing rectification for un-calibrated stereo images. Moreover, in consideration of the effectiveness of input datasets, two feature selection schemes (RS and SRC) have been evaluated based upon the pipeline in order to improve the performance of learning ability. Estimation of numeric types (uint8, int16 and absolute types) derived from basic feature design have as well as been carried out to find out which numerical data is more fit to stereo correspondence problem. After the observations from experiments, SRC and absolute type have been decided as the main methods for feature engineering with the datasets from Middlebury benchmark that have been the main experimental datasets in our study.

Chapter 4: Neural networks for stereo correspondence

As mentioned in Section 2.5, main types of stereo matching algorithms are on the basis of local and global schemes. Along with the improvement, various approaches of stereo corresponding estimation have been investigated and carried out for tackling a variety of issues. Up to the present time, many novel algorithms have been investigated, techniques of neural networks as well as have started to perform an effective character in the realm of stereo correspondence.

This Chapter introduces the novel approach in relation to adopt neural networks to perform the procedure of stereo matching. Especially, the innovative one created and designed by the project is formed with multiple neural networks that has been created based on the simple neural network, which the detailed design from network structure, training function, parameter optimization to performance optimization of network model and evaluation of produced performance are particularly presented step by step.

4.1 Neural network overview

Artificial neural network (ANN) has proved that its learning skills can make huge benefits on pattern recognition and classification tasks in a way of self-organization, moreover, trained ANN becomes an expert in the area of given examples and can predict meanings for unseen data [97]. Such strategies are in the light of the interest in the path of simulating human nervous system specifically on learning principles. The functional activations provided by the brain are generated from the power of enormous nervous networks. The brain processes information that is represented by electric signals through billions of neurons in the way of emitting signals between neurons as the schematic shown in Figure 4.1.1 [98].

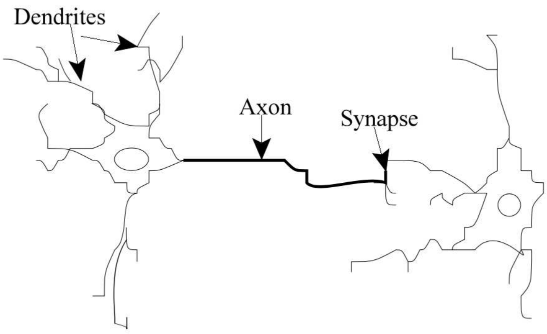


Figure 4.1.1 Biological neurons

The dendrites, cell body and axon as elementary functions constitute a biological neuron. Impulses in the form of signals reaches dendrites of a neuron through synapse that is in between dendrites of the neuron and terminals of another neuron strands, if there are strong signals received the cell body is activated to process these signals then following by transmission along the axon further to strands and their sub-strands to the next neuron that again performs the same procedure [98], [99].

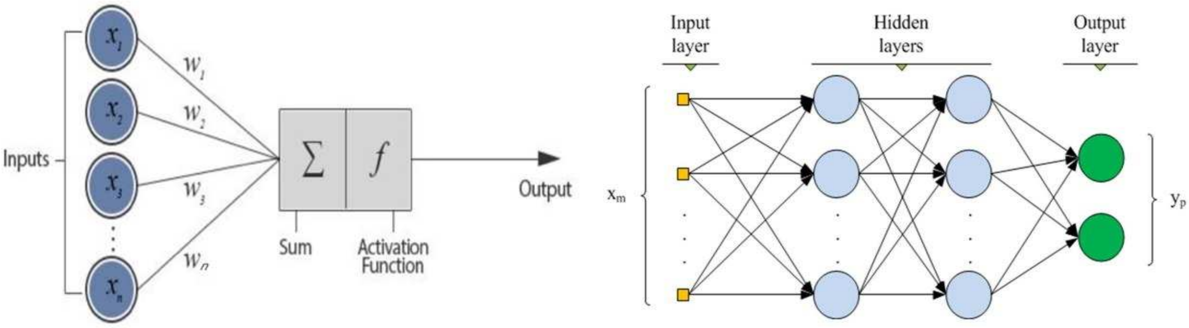


Figure 4.1.2 Computational model of a neuron (left) and ANN example (right)

The technique of ANN simulates a biological neuron as a node formed with a computational model in a network [99]. Figure 4.1.2 presents the schematics of the artificial neuron and an ANN network models [100]. An artificial neuron computes the output in two steps: first step is to implement the summation of weighted input data and second step computes output with an activation function, moreover, an ANN consists of these artificial neurons that are divided into different layers connected by weighted directions, which consists of input and output layers and hidden layers in between them that do not directly communicate with external circumstances [98], [99], [100], [101]. The term of weight represents the strength of the connection between neurons [102]. Basically, there are four common activation functions: threshold, piecewise linear, sigmoid and Gaussian as shown in Figure 4.1.3 [99].

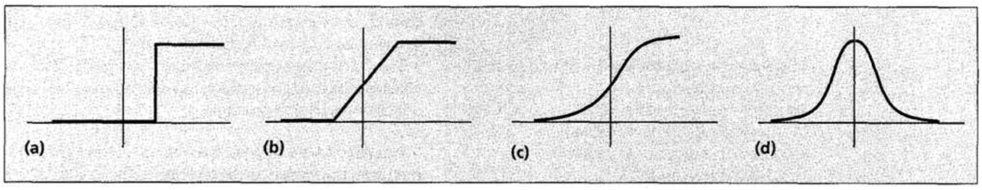


Figure 4.1.3 Common activation functions: (a) threshold, (b) piecewise linear, (c) sigmoid and (d) Gaussian

According to the pattern of connections, there are two main classes of ANN: feed-forward network that connects layers in one direction without loops and feedback/recurrent that occurs loops in connections as a result of data feedback [99], [100], a taxonomy of architectures based on these two categories is given in Figure 4.1.4 [99]. Static feed-forward network with low computational requirement produces one set of output instead of a sequence of output generated by dynamic feedback network, which accordingly performs with low computational cost in contrast to feedback structure [99].

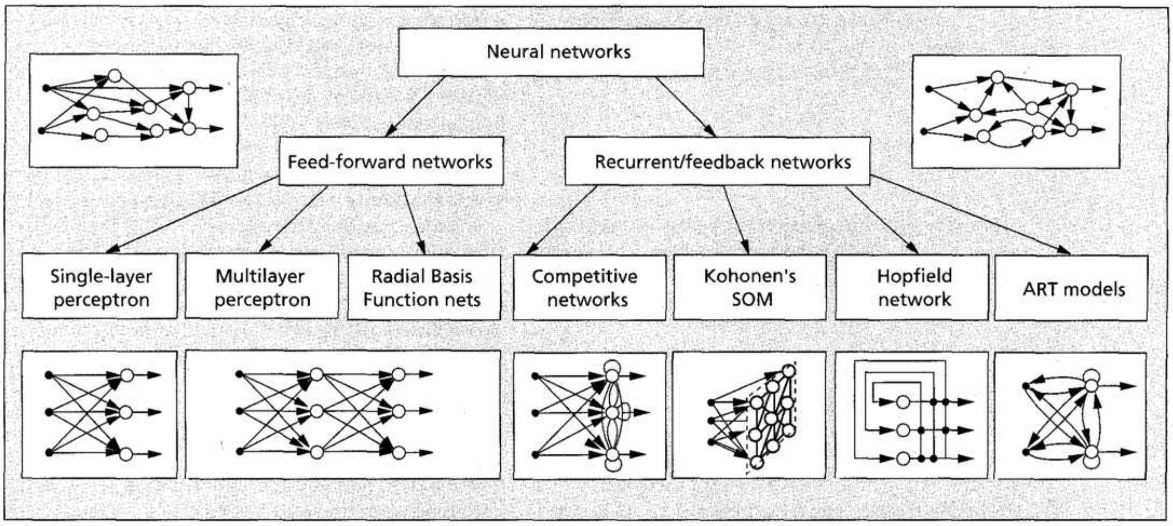


Figure 4.1.4 Feed-forward and recurrent/feedback architectures taxonomy

Learning schemes of ANN referring to the adjustment of weights group into three fundamental categories: supervised, unsupervised and hybrid [99], [101]. According to the literature, supervised learning gives each sample with correct answers, and modifies weights on the basis of errors in respect to the answers so as to generate outputs approaching to correct answers. Unsupervised learning categorises samples without correct answers but based on the correlations of samples such as underlying data structure and comparability between. Hybrid learning associates supervised learning with unsupervised learning in the way of dividing weights for two learning schemes estimating respectively.

The update of weights for input samples can be carried out in two forms: incremental training updates weights sample by sample and batch training updates weights with the entire set of samples [101], [103]. Incremental training has the advantage of estimation with less storage demand and less possibility of falling into a local minimum, however, may start with a bad sample resulting in a wrong searching path in contrast to batch training that having better estimation measurement owing to plenteous representatives [101]. Both incremental and batch strategies can be implemented in static and dynamic networks, while incremental training commonly participates in dynamic networks [103].

4.2 State-of-the-art of stereo corresponding algorithms using neural networks

As presented in the previous section, neural networks can provide many functions in the realm of intelligent learning. On one hand, the imitation of nervous system is able to provide vastly parallel distribution architecture that can significantly improve computational speed which makes favor for tasks with huge amount of dataset, on the other hand, the learning power of neural network can provide generalization analysis to the input information that has not been seen in advance, moreover, the ability of processing inherent contextual knowledge can sufficiently take advantage of local information, in addition, neural network system can be stable at a certain degree owing to fault tolerance capability [99]. On account of dominate aspects of ANN, disparity estimation with neural networks as a popular approach has a growing domain of interest and been studied by many types of research.

At the 1970s, neural networks had started its role in stereopsis. Dev [104] introduced an examination that detected depth with a neural model formulating surface segmentations integrated with random-dot stereograms, which represented an early application of stereo depth detection with neural network, after a decade, along with the development more and more stereo corresponding researches showed interests and attempted to exploit neural networks which can be found in [105].

Up to the recent period, different types of neural networks have been implemented with stereo matching task on different aspects to generate disparity map. These algorithms can be roughly grouped in unsupervised and supervised strategies. Paper [106] introduces an unsupervised strategy that implements disparity computation by using self-organizing mapped neural network. An approach that adopts the combination of Hopfield neural network for finding the most interest area and the maximum neural network for detecting its best location stereo matching is proposed by [107], in addition, this approach is extended to compare with implementation with self-correcting networks in [108].

Comparing with unsupervised algorithms, supervised algorithms can perform a matching process in accordance with the given targeting values. Article [109] proposes a methodology that can perform disparity map generation with FPGA in real by formulating differential features between paired pixels in the form of disparity space image, and computing the final disparity with a feed-forward neural network including two hidden layered. Backpropagation (BP) network as a common supervised learning approach has participated in disparity map estimation [110]. The algorithms of [110] involve the estimation of the matching level between stereo pixels as a classification task by taking advantage of BP algorithm, furthermore, this idea has made contributes to such research direction.

An algorithm presented by work [90] that extends the idea of [110] finds out the matched pixel on right image of a reference pixel on left image in a maximum disparity range by determining the matching degree of reference pixel and its potential corresponding pixels with a BP Neural Network that is the same as the part of computing matching level in [110], which involves image rectification at the beginning and refinement approach based on segmentations estimation with the BP network at the final stage. This approach is then modified by paper [111] with the same differential features and the same structure of BP network, however with different refinement algorithm implemented by a simple network consisting of input and output layers only. The start point of our study has been found on symmetrical methodologies proposed by [90].

4.3 Initial simple neural network

The first model of the neural network in our research was created based upon [90]. We adopted the network structure proposed by the study of [90], which is illustrated in Figure 4.3.1. The primary role of the proposed simple neural network plays a role in estimating the level of similarities between paired pixels so as to obtain corresponding left and right pixels as much accurate as possible. The function of this neural network involves estimating the level of correspondence between stereo pixels which is also denoted as matching degree. The higher the values of matching degree, the more corresponding two pixels are.

As shown in Figure 4.3.1, the neural network is a multilayer perceptron network constructed with three layers. There are 147 neurons on the input layer that input the feature vectors consisting of three differential features for every pixel pair contacting 147 attributes, and one hidden layer has 49 neurons. The output layer with one node outputs a vector containing computed matching degrees for all pairs in respect to each reference pixel. The final disparity is assigned with the one has the best matching degree among all the candidate pixels.

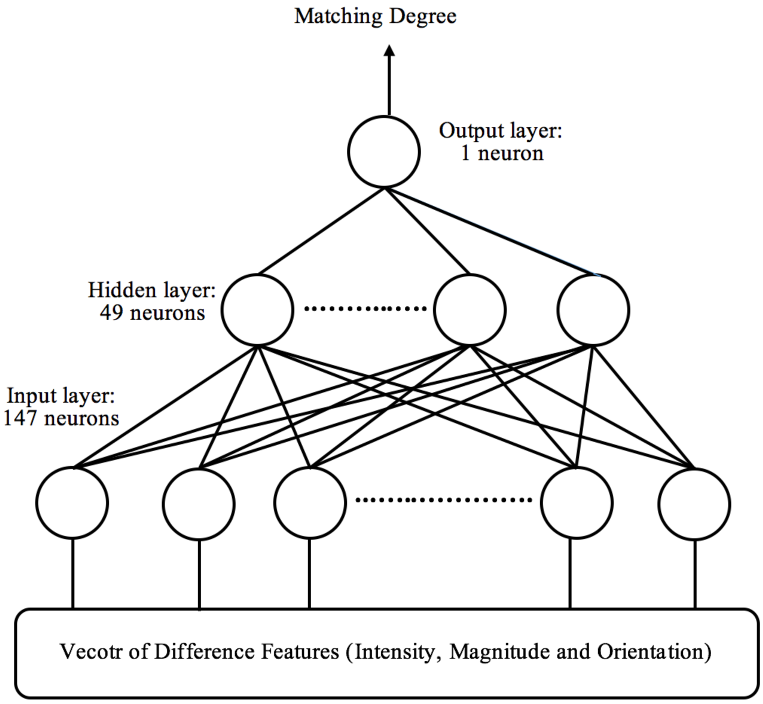


Figure 4.3.1 Structure of simple neural network (SNN)

This simple neural network employs backpropagation method to perform batch training procedure. Backpropagation algorithm is one of the widely implemented learning algorithms in ANN [112].

BP learning that is based on the principle of error correction minimizes the error function (the most common one – cost function found on squared error) in a way of performing backward error estimation and parameters update between output layer and input layer [99], [101] as shown in Figure 4.3.2 [113]. The process of a BP network learning can be divided into four stages [112]: computation of feed-forward, backpropagation between output and hidden layers, backpropagation between hidden layers and input Layers and update of all weights.

The first stage of a BP network completes a turn of feed forward computation with a set of randomly chosen weights. Once the output is generated from the feed-forward process, errors of nodes retrieved by loss function at the output layer are computed and the weights leading to them can be updated according to the estimated errors. Next, when the BP process reaches to the hidden layer, the procedure performed at the previous layer are implemented again so as to obtain corrected weights between input and hidden layers. One full backpropagation process stops until accomplishing the computation between the input layer and the first hidden layer. At this point, all the weights can be updated to a certain level of correction. The entire stages then perform again and again until deriving the desired minimum error which can have output close to the given true target as much as possible, where Equation 4.2 formulates this process to achieve parameters updating.

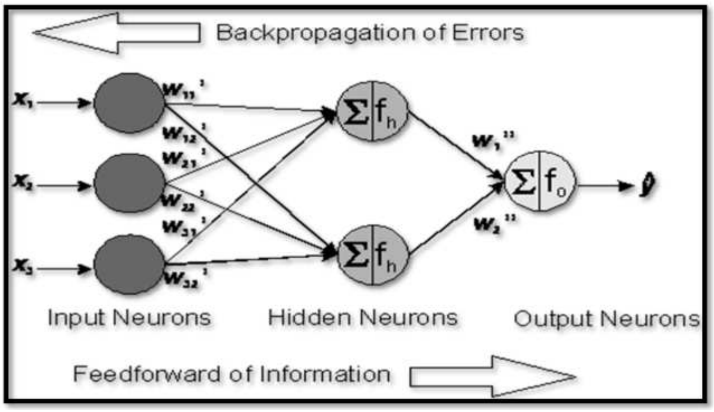


Figure 4.3.2 Backpropagation network

As explicated in Chapter 3, there are three differential features extracted for pixels: absolute intensity differences, absolute magnitude differences and absolute orientation differences. These three features are computed as combinations of pixels in selected windows. Each differential feature window is the subtraction of two windows that are centred with stereo pixels. The three differential feature windows of a pixel pair are then formatted into a vector as the input of BP neural network.

This simple neural network (hereby denoted as SNN) as the first network in our study was built with the same network architecture introduced by [90], however, with different methodologies regarding feature engineering as presented in Chapter 3. Moreover, experiments of parameter settings for SNN have been carried out to make the improvement on the level of performance contrast to the original network design.

4.4 Model selection experiments and evaluations for SNN

Artificial neural networks normally involve a diversity of parameters settings to optimize models. These settings affect the performance of designed networks. Among these parameters, learning rate and training algorithms are the two fundamental factors making effects on the degree of learning accuracy. The network specification used in experiments is given in Figure 4.4.1. Each input instance belonging to a pair of stereo pixels consisted of 147 inputs. These input values were summed with weight and bias operations at hidden layer. The term of bias measures how simply to make a neuron activate [102]. After the summation computation, the activation function transferred the values as the output of the hidden layer to the output layer. The output layer performed the same procedure as the hidden layer to produce one output denoting the matching degree of a pixel pair.

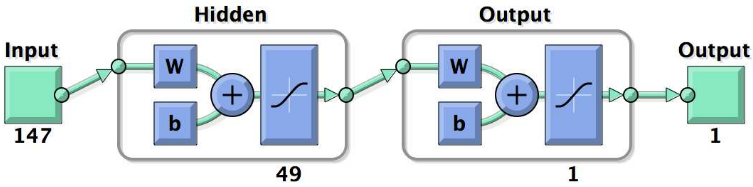


Figure 4.4.1 Experimental network structure for SNN

As shown in Figure 4.4.1, the activation function participated in the network for the experiments was sigmoid function. The sigmoid function is the most extensively used one, where the definition can be defined as Equation 4.1 [102]. The curve of the sigmoid function presents a shape of "s" as shown in Figure 4.1.3. In the mathematical sense, the sigmoid function can receive any value of real numbers in the range including both positive and negative, and converts output values into the scope zero to one [103].

$$f(x) = \frac{1}{1 + \exp^{-x}} \tag{4.1}$$

4.4.1 Model selection with learning rate

Learning rate is a parameter required by the most training algorithms in neural network as it directly affects the computation of weights. Diversified models demand specific learning rate to cooperate with themselves so as to reach to optimal performance as much as possible. Therefore, it is important to estimate which learning rate could be a suitable one for an intended network model. The possible situations caused by learning rates are given as Figure 4.4.1.1 by [114].

Small learning rate may cost a long time to reach to a significant improvement on performance where error rates drop with a large value as the representation by the blue curve in Figure 4.4.1.1, where the changes of errors appear very small between each step. The larger the value of learning rate, the faster the jump from high to low error rate would be at the beginning, however, this may cause the missing of actual minimum error point and lost in finding subsequently such as the green curve. With very large learning rate such as the yellow curve, the procedure may appear difficulties in minimizing the errors and generate extremely high errors. The red curve shows that an optimal learning rate can produce a performance with a smooth curve within an ideal range of processing epochs, moreover, the improvements of reducing errors between each epoch perform with reasonable reduction rate.

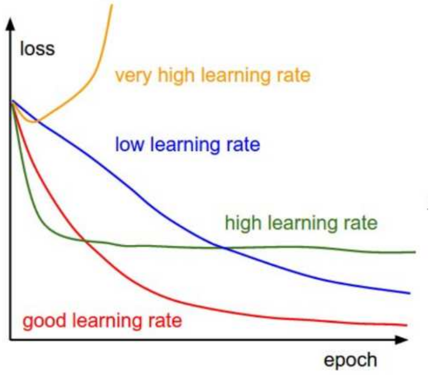


Figure 4.4.1.1 learning rate performance

On account of this reason, the learning rate in regard to the most common training function which is known as gradient descent algorithm (details are given in next section) was estimated in our research. The observation of the effect with different learning rate involves evaluates performance produced by networks with different learning rates. For this experimental evaluation, the error rates were computed with mean squared error as presented in Section 3.3.3 to reveal the training and test performances. The most common selected learning rate is 0.01. We adopted this values as one of the choices of learning rate as the basic rate, moreover, in order to observe impact generated on the basis of larger and smaller learning rate, other values were included for estimation: 0.05 and 0.005. Furthermore, we used neural network toolbox provided by MathWorks [103], [115] to implemented experiments. This toolbox is very handful for achieving neural network task.

LR	0.01	0.05	0.005
MSE with 10-FCV	0.0941	0.0941	0.0942

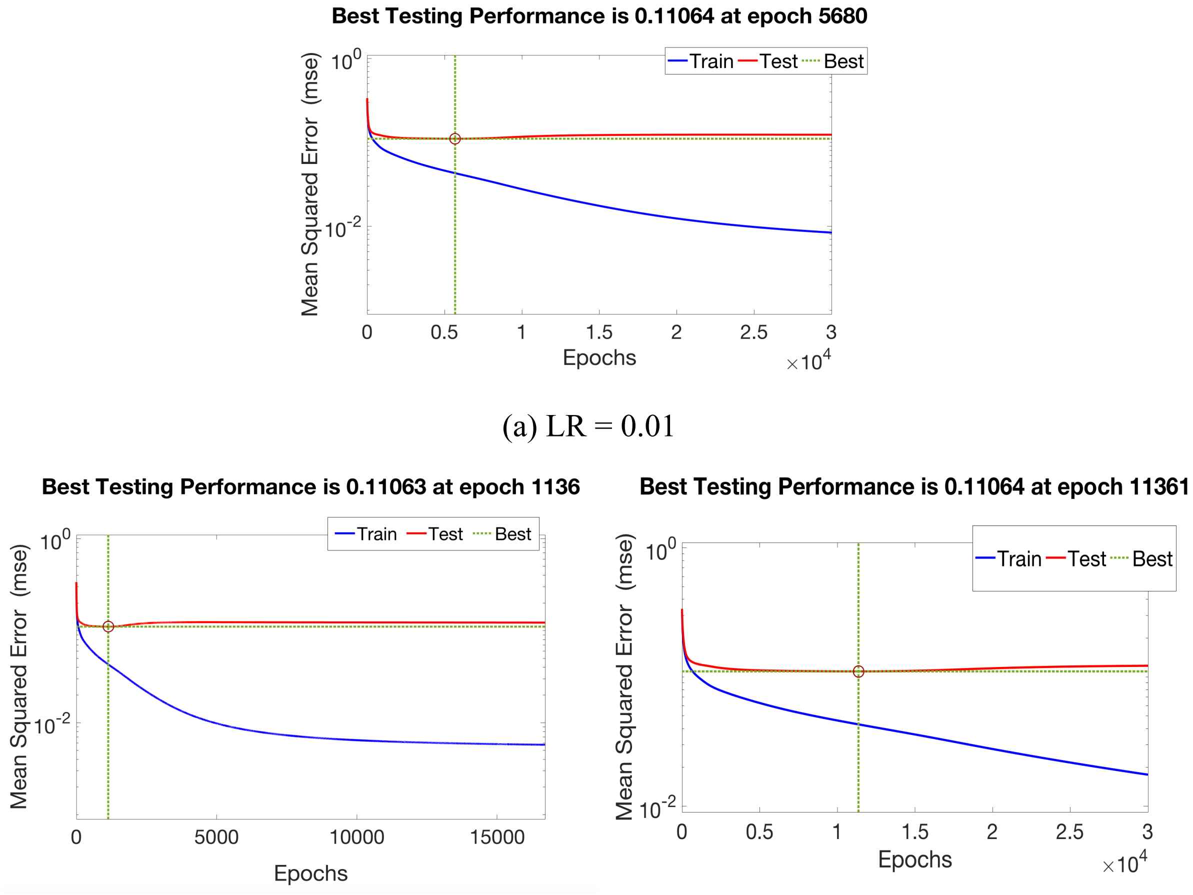
Table 4.4.1.1 Performance comparisons for learning rates for SNN

Table 4.4.1.1 lists the output performances relating to the network training with three learning rates. From the table we can see, these three learning rates produced close performance, where the average performance of LR 0.01 and LR 0.005 appeared the same results and only slightly higher than the output values generated by LR 0.005. The very similar performances generated by three learning rates imply that, without in the consideration of processing speed, the SNN integrated with gradient descent algorithm to perform matching estimation between stereo pixels can maintain stable performances regarding the changes of different learning rate.

The characteristic learning curves presenting the training effectiveness of SNN in respect to each learning rate are given in Figure 4.4.1.2, where Figure 4.4.1.2 (a) is the learning curve with learning rate 0.01, and Figure 4.4.1.2 (b) and (c) are the learning curves showing the characteristics of training with LR 0.05 and 0.005. Three learning curves all demonstrated similar smooth shape in respect to the test process. LR 0.01 and 0.005 had the similar slope for training performance while LR 0.05 made significantly decreasing training slope, which means LR 0.01 and 0.005 can balance the learning ability among training (with the seen set) and test (with the unseen set). Determining from this aspect, LR 0.01 or 0.005 could be a better selection in three learning rates.

Although these three rates can produce the very close outcomes, the actual elapsed time varies from each other. The high rate LR 0.05 reached to the low error rate in a short time, and the low rate LR 0.005 took long processing time with many epochs to minimise values of errors. The computational speed of LR 0.01 appeared to be approximately in the middle between LR 0.05 and 0.005. If making the choice of learning rate form this condition, LR 0.01 or 0.05 may come to the first place.

In the view of these outcomes, we chose LR 0.01 as the learning rate for SNN trained with gradient decent function on account of LR 0.01showed average benefits in both situations described above rather than the other two rates.



(a) LR = 0.01

(b) LR = 0.05

(c) LR = 0.005

Figure 4.4.1.2 Learning curves produced with three learning rate for SNN

4.4.2 Model optimization with training functions

In the learning procedure of neural network, the training algorithms minimize the loss function (e.g. MSE) in the way of adjusting the learnable parameters such like weight and bias so as to make a network produce errors as low as possible [102]. There were two training algorithms adopted in experiments to find out an optimal training function for SNN: gradient descent and scaled conjugate gradient.

The gradient descent backpropagation is deemed to be the basic and simplest approach among optimization algorithms. Training function integrating this algorithm modifies learnable parameters in accordance with gradient descent as defined in Equation 4.2 by computing derivatives of performance - p regarding current variable vector - x_k containing values for weight and bias [102], [103], [116], where μ is the learning rate as mentioned in the previous section and x_{k+1} represents the update of two learnable parameters.

$$x_{k+1} = x_k - \mu \frac{\partial p_k}{\partial x_k} = x_k - \mu \nabla E_k \tag{4.2}$$

The gradient descent (GD) algorithm exploits a constant value for the learning rate and negative gradient as search direction performs approximation. Moreover, in this case, all directions integrating with one learning rate, however, one learning rate cannot be the optimal solution for all conditions.

One algorithm that is known as Scaled Conjugate Gradient (SCG) proposed by [117] to solve the issue caused by GD adopts second order information to implement cost function minimization. As a result of these properties, a network utilising SCG can produce better performance within faster computational speed in contrast with using GD [103], [117], [118]. The SCG algorithm computes the learning rate (Equation 4.3) and direction (Equation 4.4) as given below (see details in [117], [118]):

$$\mu_k = \frac{-d_k \nabla E_k}{d_k (s_k + \lambda_k d_k)} \tag{4.3}$$

Where d_k is the current direction, and the denominator is scaled to positive by fudge factor λ, s represents a difference approximation of Hessian metric with a direction. The conjugate direction is formulated as Equation 4.4 by setting the initial direction as $d_k = -\nabla E_k$:

$$d_{k+1} = -\nabla E_{k+1} + \beta_k d_k \tag{4.4}$$

Where β_k is defined as Equation 4.5:

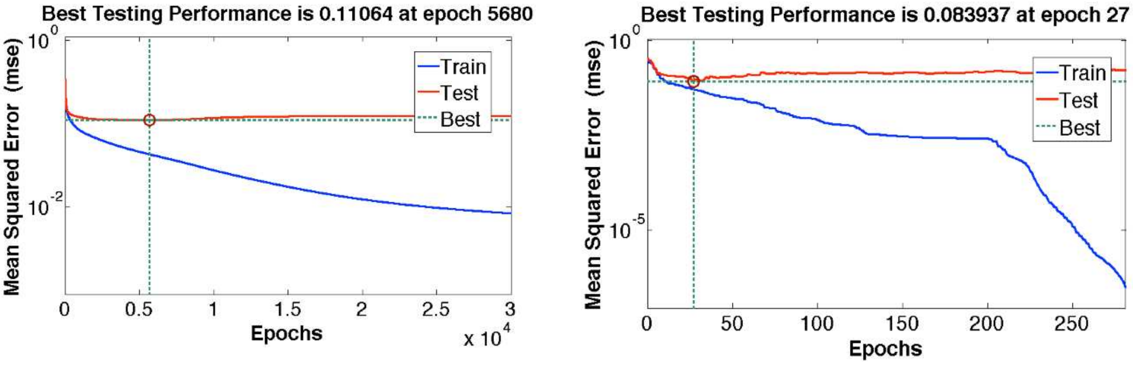
$$\beta_k = \frac{|-\nabla E_{k+1}|^2 - (-\nabla E_{k+1}) d_k}{\mu_k} \tag{4.5}$$

The experiments using GD and SCG for SNN were carried out so as to compare and evaluate the effect for model selection. The results of performance regarding each training function are given in Table 4.4.2.1.

The training and tests sets were the same datasets as the previous section, which contained matched pairs and unmated pairs. The practical experiments in respect to GD and SCG functions were also implemented with the toolbox provided by MathWorks [103], [115]. As listed in Table 4.4.2.1, SNN using SCG produced better performance than using GD according to the 10-fold cross-validation computed based on MSE, which means the optimal solution among these two raining functions for SNN should be SCG.

Training function	**GD**	**SCG**
MSE with 10-FCV	0.0941	0.0858

Table 4.4.2.1 Performance comparisons for training functions



(a) Learning curve of GD

(b) Learning of SCG

Figure 4.4.2.1 Learning curves in respect to GD and SCG

The representative learning curves produced by GD function and SCG function are illustrated in Figure 4.4.2.1 (a) and (b). The point with the best performance on the curve represents a convergence point of network learning. After the best point resenting a convergence point, the network was outfitted which led to the network learnt seen set (training set) better than the unseen set (test set). The learning curve of SCG algorithm shows in Figure 4.4.2.1 that the best performance, in other words, the convergence of the learning can be determined in a very small iteration, comparatively, training function with GD algorithm took long time to reach to an ideal performance, moreover, the best performance generated by SCG function was outperformed GD function.

In this evaluation, SCG function proved the capability of producing standout outcome relating to obtain high accuracy in a short computational time. Therefore, SCG was deemed as the optimal training function for neural networks in our research. The structure of SNN was found on a single and simple neural network. In consideration of this structure, a more complex design of networks was thereupon considered to investigate the influence of stereo matching with different network structure.

4.4.3 Impact with dataset size

In the previous sections, experiments that used datasets with small dataset produced a low performance. Normally, the overall performance can be improved by increased the number of training samples on account of the level of learning generalization, in other words, the network can learn more conditions instead of giving narrow information. For this purpose, we employed increasing sizes of dataset to observe if a larger training dataset can increase the level of performance.

This dataset was extracted with the algorithms presented in Chapter 3 Section 3.2 which relies on finding interest points. However, the number of interest point is normally limited, in order to build more datasets with sample numbers from small to large, we extracted datasets from stereo images using SRC proposed in Chapter 3 Section 3.3 according to the ground truth provided as ground truth provides the disparities that imply the position of corresponding pixels. which is good for generalization learning.

There were three datasets extracted, and their sizes were: 23400 (23400D), 46800 (46800D) and 93600 (93600D). The size of the extracted dataset in the previous section was 423, in order to implement 10-fold cross-validation, 423 instances were divided into a training set containing 378 samples and a test set containing 42 samples for each fold. The sizes of training and test sets of thee datasets obtained from the ground truth for each fold of 10-fold cross-validation are listed in Table 4.4.3.1.

Dataset size	23400	46800	93600
1-fold training set size	21060	42120	84240
1-fold test set size	2340	4680	9360

Table 4.4.3.1 Sizes of three datasets extracted according to ground truth

As the list in Table 4.4.3.1, the number of instances in these three datasets were increased the double amount of the one had a smaller number in respect to its one previous step. The experiments were implemented with the SNN using SCG function. Figure 4.4.3.1 presents the comparisons between performances regarding three datasets.

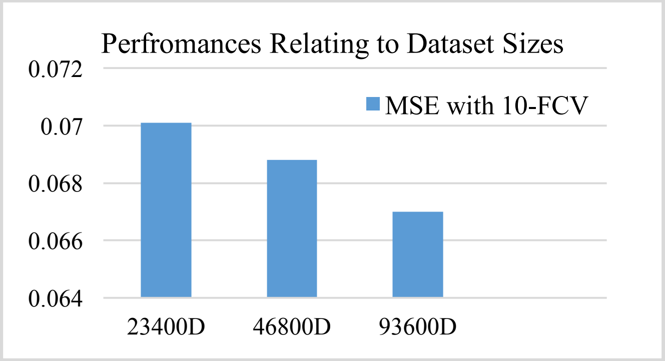
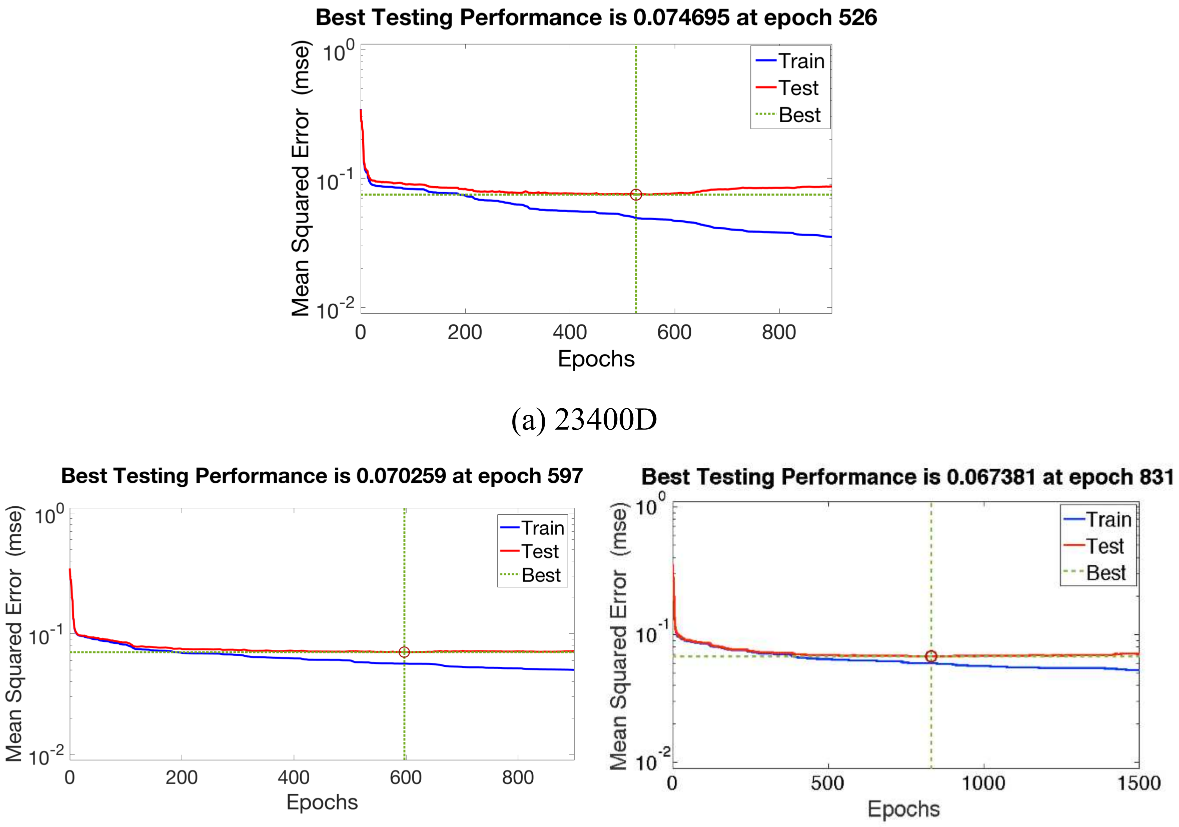


Figure 4.4.3.1 Performance of different size of datasets

In Figure 4.4.3.1, the shorter the bar, the lower the error rates generated. From this results we can see, along with the increasing number of samples, the performance increased as the decreasing bar shown, in other words, accuracy can be advanced with a large training dataset. Furthermore, as the typical learning curves presented in Figure 4.4.3.2, overfitting issue appeared to gradually reduce when the size of dataset increased, moreover, the largest dataset required the longest processing time of finding the convergence point, which is reasonable and acceptable as the computational time depends on the number of samples.



(a) 23400D

(b) 46800D

(c) 93600D

Figure 4.4.3.2 Learning curves generated by three datasets

4.5 Multiple neural networks for stereo matching

Since neural networks have been in the spotlight of the stereo matching realm, various types of neural networks have been applied to disparity map estimation. However, most of the methodologies are found on single network structure, thereupon, one question came to us, which referred to what the performance would be if estimating stereo corresponding with architectures consisting of multi-networks as the human brain not only processes with one network but also with a diversity of packs of networks. In neural network technologies, models have such characteristic involves multiple and modular neural networks.

4.5.1 Multiple and modular neural networks

Multiple neural networks (Multiple NNs) offers architecture consisting of individual networks as depicted in Figure 4.5.1.1 [119]. Each network in the system is designed and trained for their particular duties separately, and a final network produces a final decision on the basis of outputs from each individual network.

Moreover, on one hand, such architecture often plays a role in applications that adopts input data captured from different sources like shown in Figure 4.5.1.1, on the other hand, tasks require to process input information in different means as well as exploit Multiple NNs [119]. In Multiple NNs system, each output of the individual network is the input for final network comparing with a single network system that the input is the original data.

Modular neural network (MNN) aims to modularize task into different subtask, which each module of the network connects to other modules instead of neurons in the network [119] as illustrated in Figure 4.5.1.2 [120]. MNN can be mainly grouped into two categories depending on the scheme of joining modules: tightly and loosely. A tight MNN trains modules parallel by utilising a way of interacting and updating parameters of all modules at each single learning step, while error correction in loose model happens at hierarchical or sequential learning stages regarding correlations between networks [121].

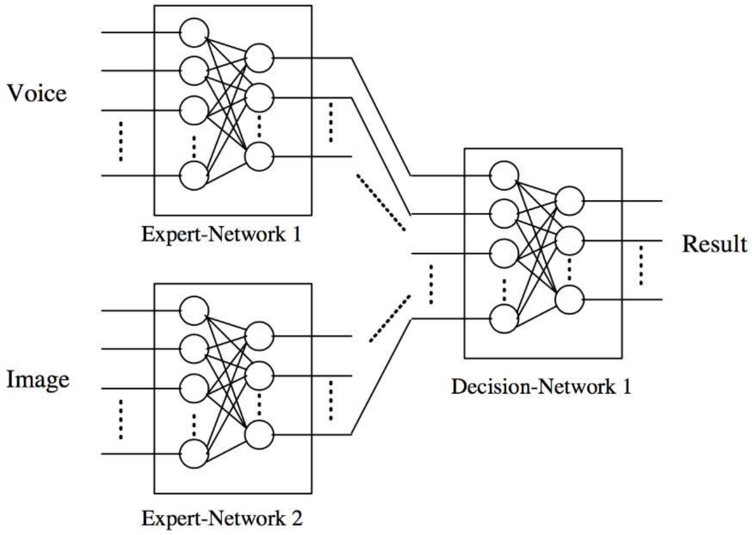


Figure 4.5.1.1 Multiple Neural Networks example

Training of MNN performs on a certain but not completely independent level on account of interactions occurring by some gating network to enable joint work between modules [119], [120] as shown in Figure 4.5.1.2. In contrast, according to the principle of Multiple NNs system, each sub neural network is independent of each other and resolves specific problem so called ‘expert’, which means, in this case, each sub-network can be in any form is trained individually and all outputs of these sub-nets are then combined for a main network to produce final output. This architecture has inspired us to apply its scheme with stereo matching task in respect of analyzing the independent correlation between two pixels from different aspects.

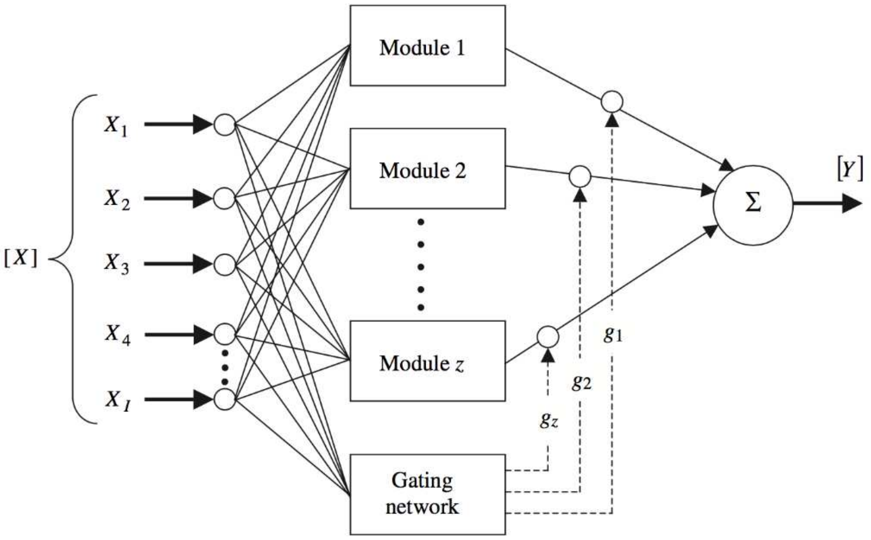


Figure 4.5.1.2 Structure of Modular Neural Network

Multiple NNs techniques are getting popular in the respect that the capability of resolving intricate jobs with results of enhanced performance in contrast to single network approaches [122]. This architecture breaks down the problem into different subproblems so as to be solved by several proper networks, therefore, Multiple NNs has been adopted as an optional strategy for creating reliable neural network system [123].

In a broad sense, the design of Multiple NNs systems can be generally divided into two classes: ensemble methods train sub-networks with the same dataset and final decision is made on the basis of integrating decisions of each sub-network to the same mission, and modular methods gives each sub-network with different tasks such as different attributes of samples and produces final decision by synthetically estimating presentations of these attributes generated by sub-networks [124]. Moreover, multiple structures could be a solution for huge datasets problems. Paper [125] proposes an approach for the purpose of improving the computational speed with a large dataset by using Multiple NNs architecture to share training responsibilities. Rather than MNN, the characteristics of Multiple NNs appear more suitable for our research purpose considering individual learning of different features.

4.5.2 The architecture of devised multiple neural networks

As described in the previous section, multiple neural networks are good at performing expert learning by dividing tasks into separate missions that are processed by the diversity of individual networks. The architecture of Multiple NNs specifically in a way of modular approach is adopted by our research in consideration of different features for correspondence estimation between left and right pixels on stereo images.

On account of three differential features (absolute intensity difference, absolute magnitude difference, and absolute orientation difference introduced in Chapter 3 Section 3.4.2) adopted for the network to learn, we designed architectures based on the principle of Multiple NNs so as to make the system learn each differential feature respectively with three sub-networks. The devised Multiple NNs were as well as used to compute matching scores for pixel pairs to measure the degree of corresponding.

The design of this Multiple NNs exploited SNN as a basic concept and extended to the more composite structure. A Multiple NNs normally contains a main network for final decision computation and sub-networks to deal with specific tasks, moreover, these networks can be any type of architecture. In our design, the main network and sub-networks were all created with the structure of the multilayer perceptron network, and hereby the devised Multiple NNs is denoted as d-Multiple NNs. Figure 4.5.2.1 illustrates the main architecture of d-Multiple NNs and the structure of a sub-net is presented in Figure 4.5.2.1.

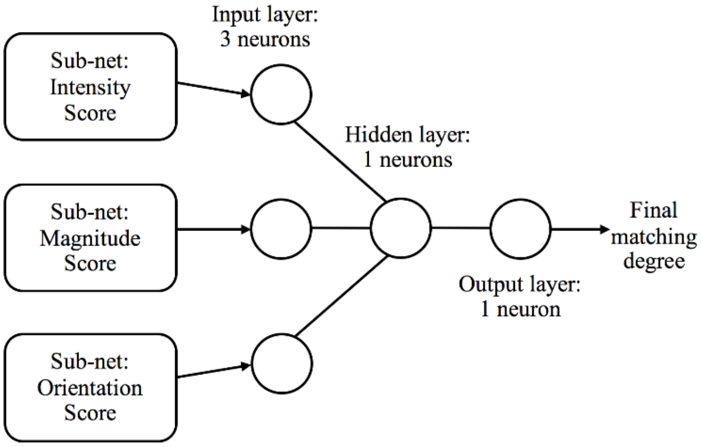


Figure 4.5.2.1 The main network of d-Multiple NNs

In our research, the main network of d-Multiple NNs consists of three layers as shown in Figure 4.5.2.1. The input layer contains three neurons that take over and transforms the input feature vector including three attributes that are the output of three sub-nets to the hidden layer. For the hidden and output layer, there is one neuron for each layer. The output of this main network represents the final matching degree for given stereo pixels. This matching degree is then used to determine if the two pixels are corresponding with each other.

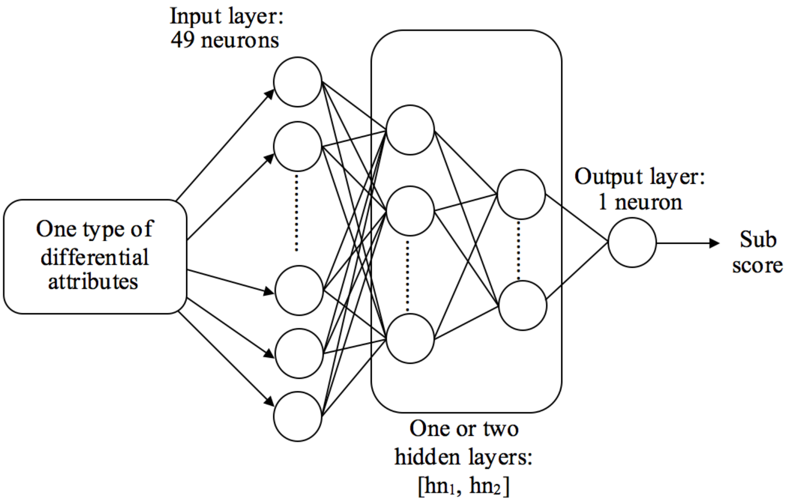


Figure 4.5.2.2 A sub-net of d-Multiple NNs

The three sub-networks use the same structure as presented in Figure 4.5.2.2, where each sub-network is created for training with one specific differential feature. The size of input feature window is also 7×7 as SNN system, therefore, the absolute difference vector of two pixels should contain 49 attributes, moreover instead of the input vector contains three differential features, the sub-network only receives one of the differential features respectively, accordingly, there are 49 neurons constitute the input layer of a sub-network.

Different plans for hidden layer architecture were built in d-Multiple NNs for investigating the effectiveness regarding hidden layers, where one was integrated with one hidden layer and another one consisted of two hidden layers, moreover, with different numbers of neurons on each hidden layer. The three sub scores (IS – intensity, MS - magnitude and OS - orientation) generated by each sub-network are combined into one vector as the input vector for the main network as below:

$$\text{Input Vector of Main Network} = [IS, MS, OS] \tag{4.6}$$

In addition, according to the evaluation carried out in Section 4.4.2, the training function for d-Multiple NNs has been considered to be SCG algorithm. Based on these methodologies, experiments for evaluating different patterns of d-Multiple were implemented in order to find out the best model design for d-Multiple NNs which will be presented in the next section.

4.6 Experiments and evaluations for d-Multiple NNs optimization

The basic pattern of d-Multiple NNs involves one simple main network and three sub-networks. The main network simply merges three scores of each differential feature to output one overall score where the practical structure as the graphical outline in Figure 4.6.1. The function of each sub-network refers to specifically learns one of the differential features. All experiments were implemented with neural network toolbox [103], [115] and the sigmoid function as activation function that is explicated in Section 4.4.

In general speaking, there are two elements involved in the design of a network, which are the type and number of layers and number of neurons at each layer. In our study, we designed d-Multiple NNs on the basis of SNN especially the layout for sub-nets. In respect that a sub-net only relates to one feature type, the number of input neurons was set to the number of attributes that one differential feature providing. Hidden layer plays a fundamental role in the architecture of neural network, which affects the learning capability of a network. Considering the importance of hidden layer, different designs of hidden layers for sub-net were experimented so as to optimize d-Multiple NNs.

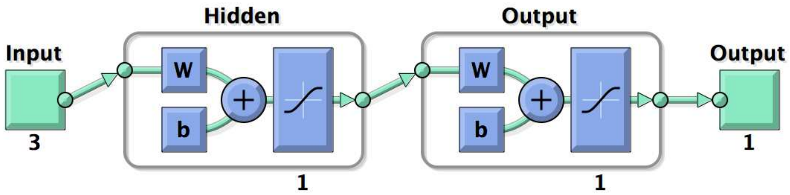


Figure 4.6.1 Layer specification of main network for d-Multiple NNs

4.6.1 Model estimation for sub-network with one hidden layer

In a type of multilayer perceptron network, when comes to the task regarding projecting layers, one primary issue is to determine the number of hidden layers. The sub-net model with one hidden layer was estimated at the initial point according to the architecture of SNN in our investigations. Figure 4.6.1.1 provides a detailed schematic represents the layer outline of a sub-net integrated with one hidden layer, where the number of neurons denoted as k.

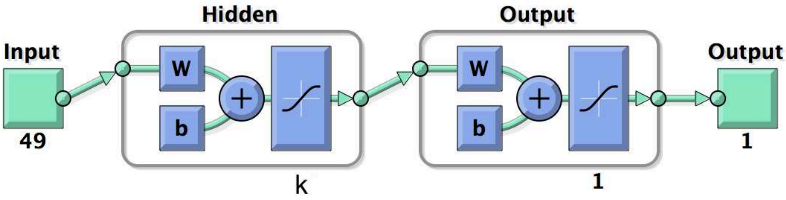


Figure 4.6.1.1 Layer specification of a sub-network with one hidden layer for d-Multiple NNs

Once the number of the hidden layers have bee confirmed, the next problem then relates to how many neurons should be at the layer. Training with too many neurons could result in each neuron is able to only understand one or a few properties, in this condition, a neuron loses the ability of generalization analysis. Inversely, with a small amount of neurons, information can be learnt by every neuron in n general phase, however, this could lead to unstable performance when given targets beyond the limitation of learnt generalization knowledge. For this reason, an appropriate number has to be found out to fit a network.

We carried out experiments in order to determine an optimal value. The performance results generated by different numbers are given in Figure 4.6.1.2. The results of 10-fold cross-validation in Figure 4.6.1.2 show that the performance improves along with the increasing number of neurons as presented by gradually shorter bars.

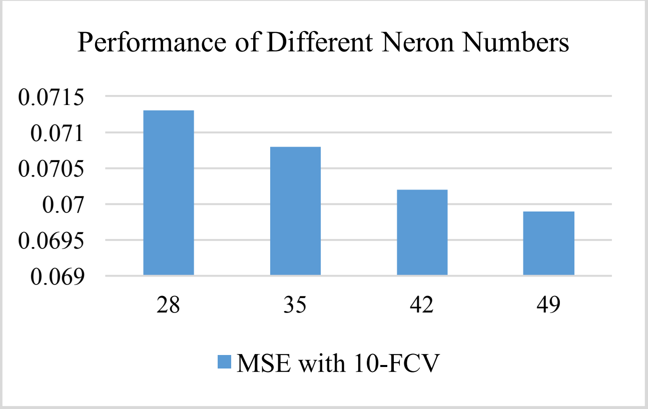
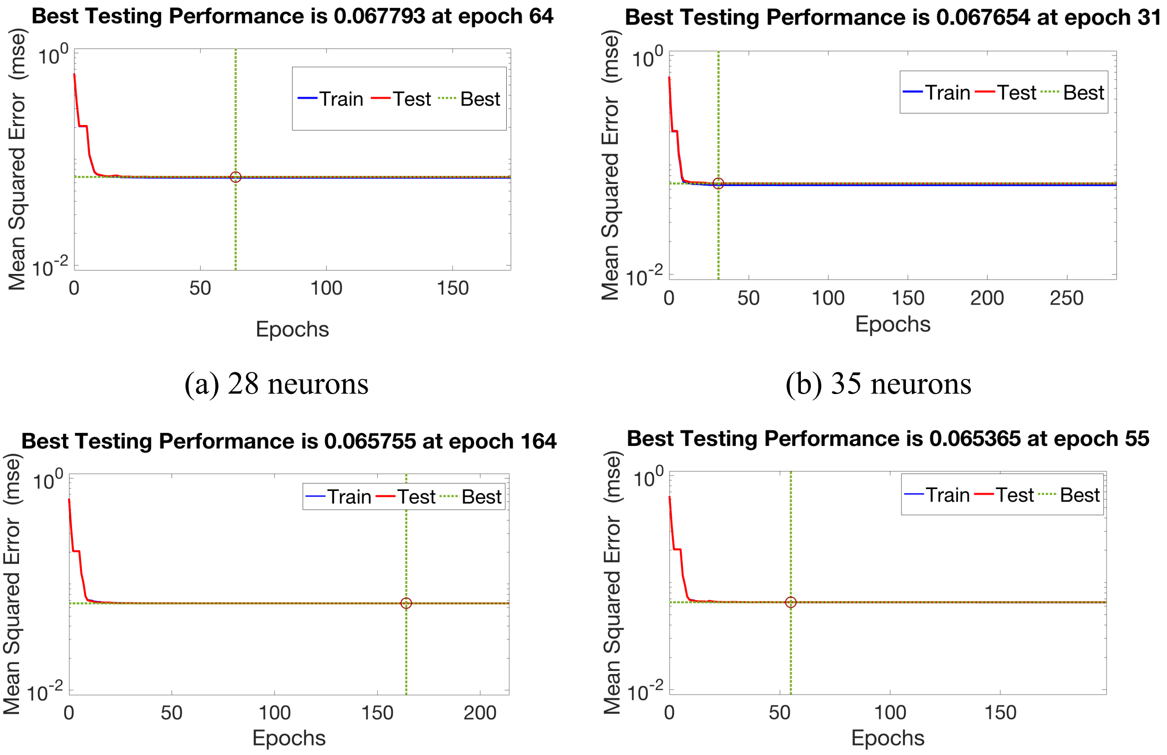


Figure 4.6.1.2 Performance with four neuron number settings at hidden layer for d-Multiple NNs



(a) 28 neurons

(b) 35 neurons

(c) 42 neurons

(d) 49 neurons

Figure 4.6.1.3 Learning curves with four neuron number settings at hidden layer for d-Multiple NNs

The experiment adopted four values as potential number setting of neurons to figure out one optimal solution or the single hidden layer. These selected values for k was found on SNN that contains 49 neurons at the hidden layer. According to this number, a factor-seven was used as the step between candidate values (28, 35, 42, 49) by taking 7×7 feature window into consideration as shown in Figure 4.6.1.2. In the view of this observation, 49 produced the best performance with relative short epochs (in Figure 4.6.1.3) and was chosen to be an optimal value as the number of neurons at this single hidden layer.

The learning procedures up to the main network using four neurons settings appeared the same characterises as illustrated in Figure 4.6.1.3, which means the learning process at the stage of the main network can be accomplished with stable competence. In four learning curves given in Figure 4.6.1.3, blue and red lines representing both training and test performances nearly overlaps each other. In other terms, the overfitting problem cannot easily arise with this design of the network. Accordingly, the main network kept this architecture for d-Multiple NNs.

4.6.2 Model estimation for sub-network with two hidden layers

On the basis of the single hidden layered sub-network, we increased one hidden layer to observe the performance changes between models integrating with single and double hidden layers. The first hidden layer adopted 49 neurons as discussed in the previous section. There were three values exploited to find a better neuron number for the second hidden layer, which are 28, 35 and 42 neurons. The comparisons between three values referring are given in Figure 4.6.2.1.

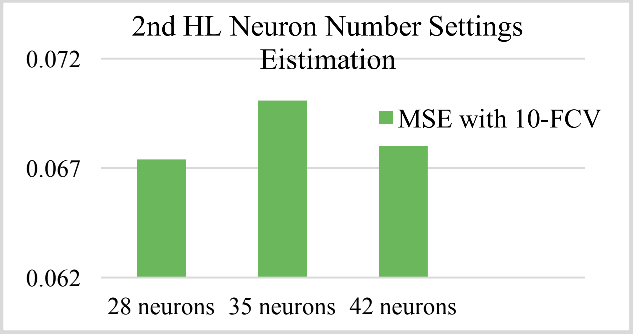
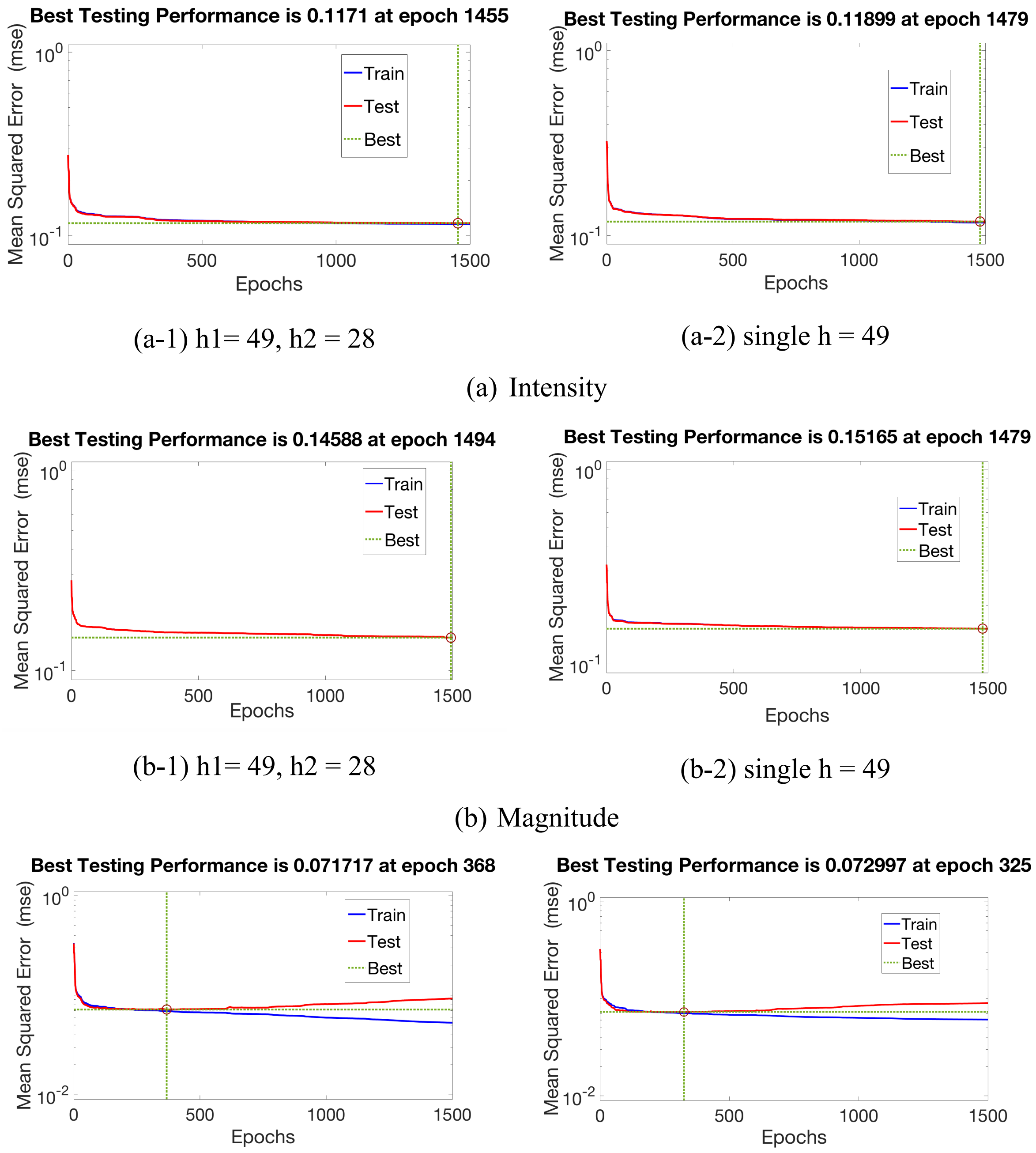


Figure 4.6.2.1 Neuron numbers estimation with two hidden layers for d-Multiple NNs

In three values, the largest number-42 neurons had the middle performance, and the worst performance was produced by training with 35 neurons at the second hidden layer, moreover, the smallest number-28 neurons output the best result, that is to say, the lowest error rate. This interesting revelation can imply that when the number of neurons at the second hidden layer is set to be approximately three steps smaller than the number at the previous hidden layer could provide better performance rather than other solutions, where the step factor is obtained according to the feature window size as explained in the previous section.

Learning characterises in respect to each differential feature (absolute intensity difference, absolute magnitude difference and absolute orientation difference) and the comparisons between single and double layer are presented in Figure 4.6.2.2, where the second hidden layer consisted of 28 neurons for this contrast.

From Figure 4.6.2.2 we can see, the training of intensity and magnitude based features require long processing time with large epochs to find the convergence, while orientation feature learning can be achieved with fast computation time.



(a-1) h1= 49, h2 = 28

(a-2) single h = 49

(a) Intensity

(b-1) h1= 49, h2 = 28

(b-2) single h = 49

(b) Magnitude

(c-1) h1= 49, h2 = 28

(c-2) single h = 49

(c) Orientation

Figure 4.6.2.2 Learning curves (three differential features) with two and single HL for d-Multiple NNs

The shapes of learning curves produced by both double and single hidden layer models appear to have the similar trend, which means the increased layer can maintain a stable training procedure, moreover, a double hidden layered model for three features all generated higher accuracy than the single-layered model. The final overall performance computed by the corresponding final decision network is shown in Figure 4.6.2.3.

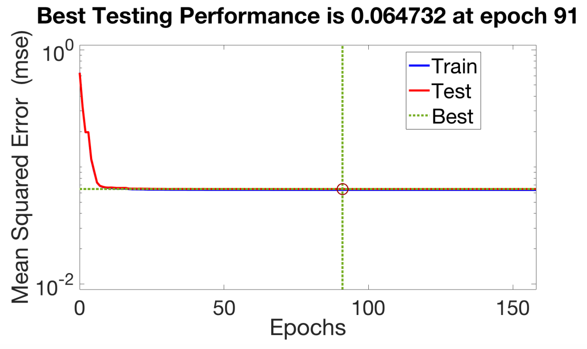


Figure 4.6.2.3 Learning curve with two hidden layers (h1 = 49, h2 = 28) for d-Multiple NNs

In Figure 4.6.2.3, the final learning curve as well as keeps the stable trend among training and test performances. The final output showed that d-Multiple NNs adopting double hidden layers (the first hidden layer-49 neurons, and the second hidden layer-28 neurons) improved the performance in contrast to single hidden layered d-Multiple NNs (single hidden layer-49 neurons) as shown in Figure 4.6.1.3 (d). Generally speaking, the model with two hidden layers can be an optimal one rather than the model with one hidden layered.

As d-Multiple NNs was created on the basis of SNN, the next section will compare the performances between SNN and d-Multiple NNs in respect to train networks to an optimal convergence point as close as possible.

4.6.3 Comparison between SNN and d-Multiple NNs

The d-Multiple NNs was created on the basis of SNN for the purpose of investigating the influence of complex network design on the stereo corresponding problem. We made the comparison in respect to the capability of learning based on the mean squared errors computed with the 10-fold cross-validation for both SNN and d-Multiple with two hidden layers (hereby is denoted as THL-Multiple NNs), and the results are given in Table 4.6.3.1.

The specifications of layers are shown below, where IL and HL denote input and hidden layers:

- SNN: IL - 147 neurons, HL - 49 neurons, OL - 1 neuron
- THL-Multiple NNs: IL - 49 neurons, HL – [49, 28] neurons, OL - 1 neuron

	1500 Epochs	6500 Epochs
SNN	0.0670	0.0670
THL-Multiple NNs	0.0693	0.0668

Table 4.6.3.1 Performances of SNN and THL-Multiple NNs

The training procedures in Table 4.6.3.1 were implemented with 93600 dataset introduced in Section 4.4.3. There were two settings for maximum training epochs adopted for the comparative evaluation. The training stops when the process reaches to the max epochs. The first comparison involves training in maximum 1500 epochs, and this number was then increased to 6500 to see if the performance could be improved.

SNN produced better performance than THL-Multiple NNs when training in 1500 epochs. However, when the number of max epochs increased, the error rate of THL-Multiple NNs dropped and was lower than performance of SNN, while SNN remained at the same accuracy level as 1500 epochs.

One one hand, this reveals that SNN can find the convergence point faster than THL-Multiple NNs. On the other hand, in consideration of THL-Multiple NNs generated very close but slightly higher performance than SNN when training with more epochs, THL-Multiple NNs showed the possible flexibility of improving performance with further training. In view of these outcomes, the system designed with complex architecture can achieve better performance than the simple neural network with a longer training period, in other terms, with the capability of further learning to produce higher accuracy.

4.7 Chapter summary

Neural networks have been exploited for stereo corresponding estimation, which inspired us to discovered the practical possibility in this field. Two systems using two types of neural networks (SNN involving one standard network and d-Multiple NNs integrating multiple networks combination) for finding the matching pixels have been described in this Chapter.

Especially, the architecture design, model selection and optimization related experiments for system optimization for the novel and innovative approach using d-Multiple NNs that were created and carried out by the project were presented in this Chapter, and related methodologies and the results analysis were also explicated. Moreover, parameter settings such like learning rate, training algorithms for d-Multiple NNs on the basis of SNN and layer designs for d-Multiple NNs were as well as in particular estimated by the project. The effectiveness of SNN and d-Multiple NNs were investigated on the basis of produced performance, and the results show that both systems are capable with the stereo corresponding task.

Chapter 5: Stereo correspondence with convolutional neural network

During recent years, researchers have a growing range of interests in integrating deep learning with the various applications so as to reach an advanced degree of marching learning referring to artificial intelligence. Deep learning presents a mighty series of learning technologies in the form of artificial neural networks, and has supplied optimal solutions in the fields of image, speech and language processing [126], moreover is the crucial study field for achieving success at the realm of computer vision [127].

In this Chapter, a novel approach is introduced for stereo corresponding estimation, which this approach matches stereo pixels by a designed convolutional neural network. The innovative design of the convolutional neural network created by this project in relation to the architecture of this convolutional neural network, training and parameter combination optimization, performance optimization of this network model from different constructions of layers, and evaluation of produced performance are presented in detail by this Chapter.

5.1 Deep learning with computer vision

For many years, deep learning has been a progressive and popular technique to resolve issues in the community of artificial intelligence, as it unlocks the capability of data distribution so that computational models can combine the diversity of processing layers to learn data from representations obtained in each level [127], [126]. Moreover, along with the development of deep learning technologies, a relatively mature community has formed to provide supporting resources such as huge amount of dataset, and pre-trained deep networks, furthermore, deep learning with the advantage of advanced learning technique can become practical methodology in accelerated computational process when integrated with the power of GUPs [128], which makes it outstanding from artificial intelligence and easily to be adopted by a variety of tasks [129]. An architecture of deep learning normally is considered as an updated neural network with complex layered components. In deep learning, the concept of deep relates to how many hidden layers there are, which normally comes with big number and even can reach to hundreds, in contrast to a common neural network that has one to three hidden layers [129].

The [130] presents a comparison between the traditional neural network and a deep learning architecture in Figure 5.1.1, which shows the distinguished part that deep learning has more hidden layers than the simple neural network. Such deep structure enables the ability of efficient and effective learning involving an intricate cognitive problem in the way of breaking down into hierarchical analysis [121], [126]. Deep learning also has the benefit of little engineering involving the establishment of data architecture so as to reduce the cost of handwork, in other terms, a deep learning network can create features and optimize required data automatically by itself [126], [130].

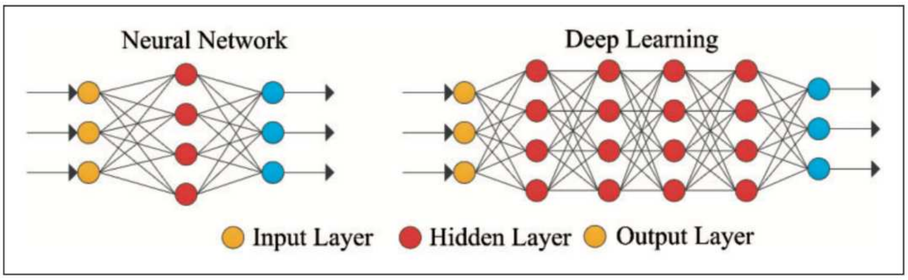


Figure 5.1.1 Construction comparison (neural network and deep learning)

A deep learning network consists of more neurons with complex connections, based on this principle, many sorts of networks have been investigated, furthermore, there are four main architectures of deep learning networks: Unsupervised Pretrained Network (e.g. Autoencoders, Deep Belief Network and Generative Adversarial Network), Convolutional Neural Network, Recurrent Neural Network (e.g. Long Short-term Memory and Gated Recurrent Unit) and Recursive Neural Network [131]. These models can achieve tasks in the realm of popular studies that have been tackled by neural networks for long period, moreover, a rough taxonomy is listed in Figure 5.1.2 [128], [131].

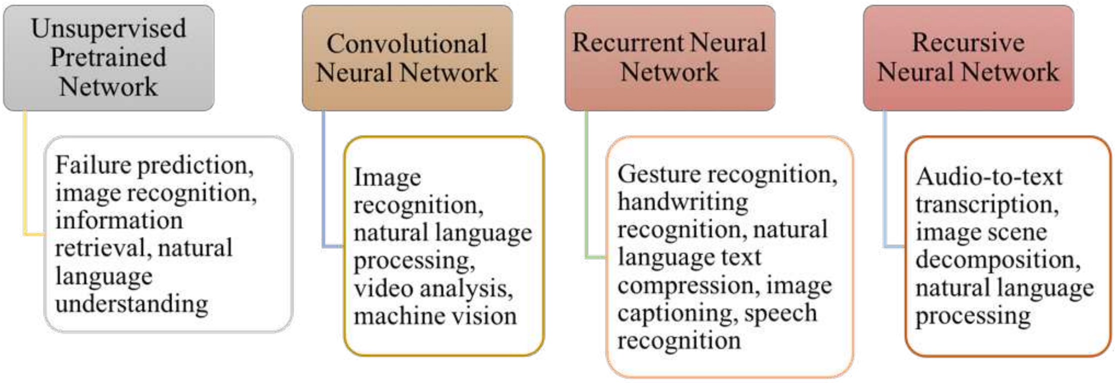

Unsupervised Pretrained Network
Failure prediction, image recognition, information retrieval, natural language understanding
Convolutional Neural Network
Image recognition, natural language processing, video analysis, machine vision
Recurrent Neural Network
Gesture recognition, handwriting recognition, natural language text compression, image captioning, speech recognition
Recursive Neural Network
Audio-to-text transcription, image scene decomposition, natural language processing

Figure 5.1.2 A taxonomy of deep learning applications

CNN has been considered as the most outstanding model in ANN [132]. This study focuses on implementing a convolutional neural network with stereo vision. Convolutional neural networks (CNN) as one most famous deep learning technique with hierarchical presentation referring to the realm of image and video processing [129] is extensively exploited in the field of computer vision [133], [134]. The architecture of CNN is inspired by the biological system of visual cortex that sensing input signals by cells breaking down information into sub-regions, according to this theory, such structure allows CNN to be able to recognise and classify the diversity of visual data such as faces, street signs and individuals [131]. A standard architecture of CNN is given by Figure 5.1.3 [129].

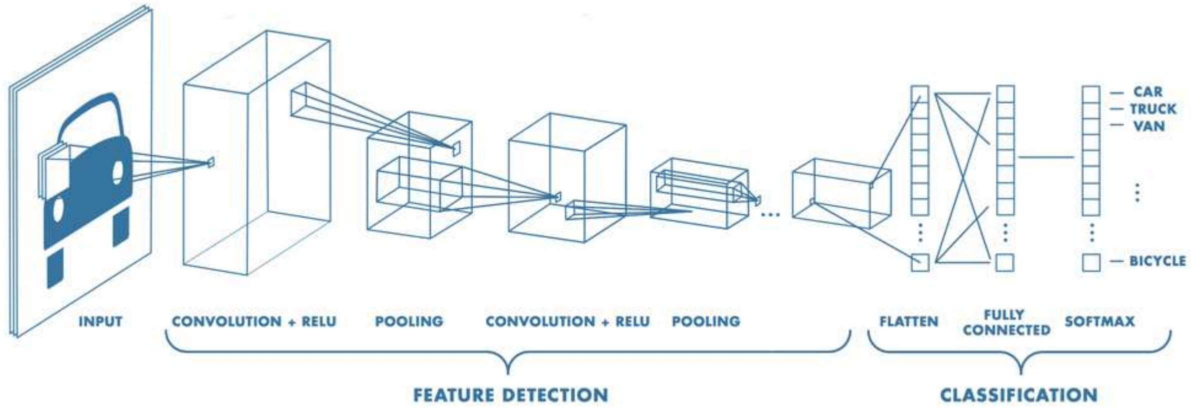


Figure 5.1.3 An example of CNN

A CNN usually consists of three main layers: convolutional layer, pooling layer and fully connected layer [114], [127], [129], [131], [132]. Convolutional layer locates at the beginning of the network and transforms input images to feature maps highlighting specific features by functional filters. A rectified linear unit (ReLU) layer is normally added after convolutional layer to perform an element-wise function that transforms negative values to zero so as to achieve accelerated training with more effective performance. Pooling layer decreases the size of spatial measurement for the representation of data so that the parameters for the network to learning can be reduced. Fully connected layer that commonly integrates with softmax function at the end of a CNN plays the similar role in the common neural network to calculate the probabilities of each class.

Depending on the complexity of input target in regard to image contents, CNN can perform with input containing a single object, and even can detect every object in images involving complex scenarios in the way of giving labels on the pixel level, which is normally denoted as fully convolutional networks [134]. The difference between common convolutional neural network and fully convolutional network referrers to the design of the last layer, that is to say, fully model converts the fully connected layer in the form of convolutional layer [135]. Currently, there are many models for CNN in which more famous ones are LeNet, AlexNet, ZF Net, GoogLeNet, VGGNet and ResNet, and they have proved the effectiveness of CNN [131].

5.2 State-of-the-art of CNN based stereo corresponding algorithms

Since CNN was introduced for enhancing the artificial intelligent learning, there have been more and more practical applications that have started to joint the family of performance improvement with ideal outcomes by exploiting the benefits of this architecture simulating the human brain, e.g.: face recognition, action and activity recognition, object detection, and human pose estimation [136]. Furthermore, the power of CNN impulses the advancement of machine/computer vision in respect of autonomous intelligence like drones, autonomous robots and cars, and visual-based medical diagnosis [126], [131]. By the same token, stereo vision as the most popular visual perception technologies combined with CNN is getting popular for implementing tasks requiring more intelligent capability, like pedestrian detection [137] and robot following person [138]. Many researchers have proposed diverse methodologies exploiting CNN to deal with the corresponding problem between stereo visual pairs for the purpose of disparity enhancement from multifarious aspects.

The paper [139] gives demonstrations of depth estimation with CNN that using the AlexNet and fully convolutional network to learn with given ground truth, and the results show the effectiveness of CNN producing disparity values for RGB images. Some approaches involve patches comparisons with respect to the level of similarity rather than entire images processing. The works of [140] (expanded in [141] to further evaluate) and [142] present a work performing efficient comparisons between patches from raw stereo images directly on the basis of learning the similarity through the structures of CNN according to the principle of Siamese network.

The problem of ill-posed areas in the disparity map can also be improved by applying stereo matching algorithms with CNN. An approach introduced by [143] improves the accuracy of disparity map in a way of estimating the output disparity maps form different traditional algorithms of stereo matching, which this estimation is implemented by a CNN, moreover, it can advance performance on occlusions. In order to tackle ill-posed issues, the research of [142] divides disparity generation into two steps, and each step is accomplished by a CNN that first one is designed for producing initial disparity map and the second one refines the initial map.

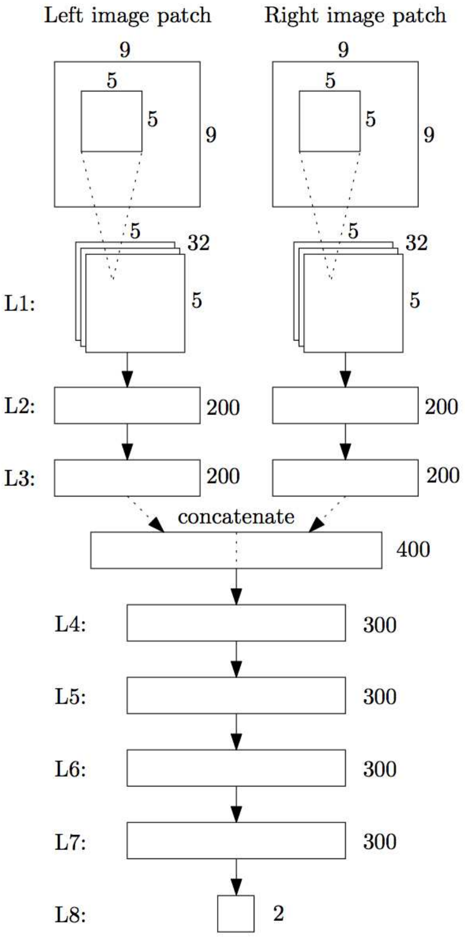


Figure 5.2.1 CNN proposed by state-of-the-art

In stereo matching algorithms, CNN can take the place of any stage among the entire classical pipeline. The first method that has been widely employed performs the replacement at the stage of matching cost computation with CNN [91]. The algorithm proposed by [91] leads the research direction of exploring the cooperation between CNN and matching cost optimization, subsequently, inspires many studies latterly, moreover, the design inspiration f CNN system for our study is also drawn from it. The pipeline of [91] starts from computing matching cost with a designed CNN as the schematic in Figure 5.2.1, and following with aggregation and cost refinement approaches for disparities calculation accomplished by WTA scheme, and at last refines the disparity map with a series of post-processing techniques. As depicted in Figure 5.2.1, eight layers constitute the network. The input of the CNN consists of two image patches from left and right images, which the centre pixel of right patch is the potential corresponding pixel regarding the reference pixel in the centre of left patch, moreover, the two patches are all grey scale.

Layer L1 is the convolutional layer that convolutes patches with 32 filters in the form of 5x5x1 dimensionality. Layers L2 to L8 are fully connected layers. At layers L2 and L3, 200 neurons connect to the output of each section from previous layers. The two groups of output from L3 are conjoined into one vector following by four layers from L4 to L7 containing 300 neurons in each layer. At the end of the CNN, two neurons layer L8 combining with softmax function classify the output into two categories to indicate a good or bad match. While pooling layer is not adopted in this architecture. For further investigation, they also suggest that convolutional layers could substitute layers from L4 to L8 such like the expansion work of them in [144]. The pattern of pipeline presented in [91] and [144] have laid a foundation on such area, following in time, their ideas are expanded by researchers in different methodologies by modifying the structure of CNN [145], [146], [147], [148]. They extend the use of [91] and [144] in the way of creating diverse models of CNN in the form of parallel and hierarchical combinations which can gain an advanced level of performance. Most recently, methodologies containing more complex architecture introduced by [149] involves cost aggregation computed by CNN models.

The principle of deep learning reveals its great vantage to cope intricate matter requiring a deep scale of analysis which makes such technique considerably suitable for topics such like computer vision, moreover, CNN outperforms among a variety of algorithms in the community of stereo depth estimation [148]. The scheme of our system will be presented in the next section.

5.3 Creation of designed convolutional neural network

The convolutional neural network has drawn a great attention to implement applications in the family of computer vision by providing profit such like hierarchical analysis mentioned in the previous section, moreover, has achieved great grades. Dense stereo corresponding approach as one of typical computer vision task requires correspondence matching at the pixel level, in other terms, in a stereo images pair, each pixel in an image has to be matched with a pixel in the other image, which leads to a quantity of information analysis. Therefore, regarding to these properties, the convolutional neural network was utilised in our research to perform stereo correspondence estimation.

5.3.1 Architecture design

The inspiration for constructing a convolutional neural network for our study was derived from the paper [91]. [91] proposed a neural network to compute matching cost for a pipeline in relation to the process of local stereo corresponding algorithm, while, as mentioned in the previous section, a CNN can replace any stage of local algorithm based pipeline.

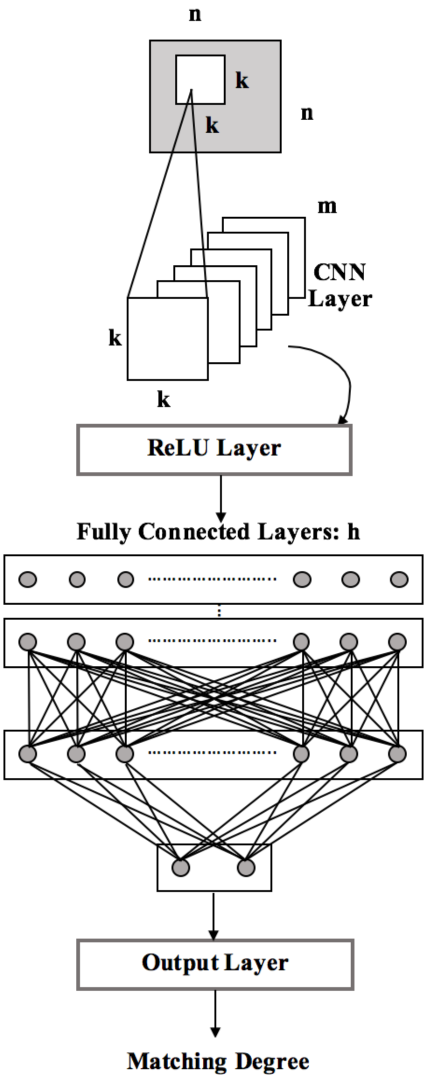


Figure 5.3.1.1 Basic model of b-CNN

In our research, we adopted a CNN to estimate the level of matching rather than matching cost computation which performs the same function as SNN and d-Multiple NNs introduced in Chapter 4. In the light of CNN architecture in [91], a basic model of our CNN (hereby, we call it b-CNN) was created, where the detailed structure is presented by the schematic shown in Figure 5.3.1.1. With the view of the CNN introduced by [91] given in Figure 5.2.1, we designed b-CNN in order to produce matching scores, which includes four main layers in the architecture of b-CNN: convolutional layer, ReLU layer, a set of fully connected layers, and a final output layer. [91] employs two intensity patches from left and right images as the input of the CNN, while one image patch containing three differential features that have been used in SNN and d-Multiple NNs are adopted in the b-CNN based system.

An input image patch is made up of three differential features that locate on three channels as illustrated in Figure 5.3.1.2, where the differences are obtained as absolute intensity difference, absolute magnitude difference, and absolute orientation difference. Each channel of image patch includes one differential feature window in relation to a pair of stereo pixels, accordingly, the size for an input patch is n×n×3. These difference features are computed on the basis of Equation 3.11 - 3.13 given in Chapter 3 Section 3.4.2.

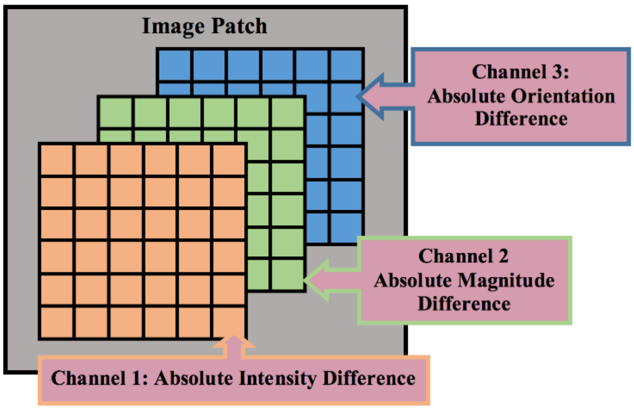


Image Patch
Channel 3: Absolute Orientation Difference
Channel 2 Absolute Magnitude Difference
Channel 1: Absolute Intensity Difference

Figure 5.3.1.2 Input image patch of b-CNN

5.3.2 Layer algorithms

Convolutional layer produces feature maps representing the partial properties of an image patch in the way of convoluting the patch with filters with certain window size k×k as shown in Figure 5.3.1, and m in this figure denotes the number of filters. A filter represents a weight regarding all sub-patches to the convolutional layer, in other terms, convolutional layer adopts shared weight for one input patch in the form of a filter containing different values in a window, moreover, each filter cooperates with one bias.

The computation at the convolutional layer involves moving filters over image patches, the movement is carried out on the basis of pixel level, for this reason, the step of sliding filter refers to how many pixels should be in the gap between current and previous filter window, in convolutional neural network, one parameter called stride defines how man steps from the current pixel to the next one.

Moreover, on account of during the convolutional process, a filter requires a selected size to extract a feature window, which affects the spatial structure of feature maps in order to control the size of output volume, in this case, an input image patch is generally padded with zeroes around the boundary.

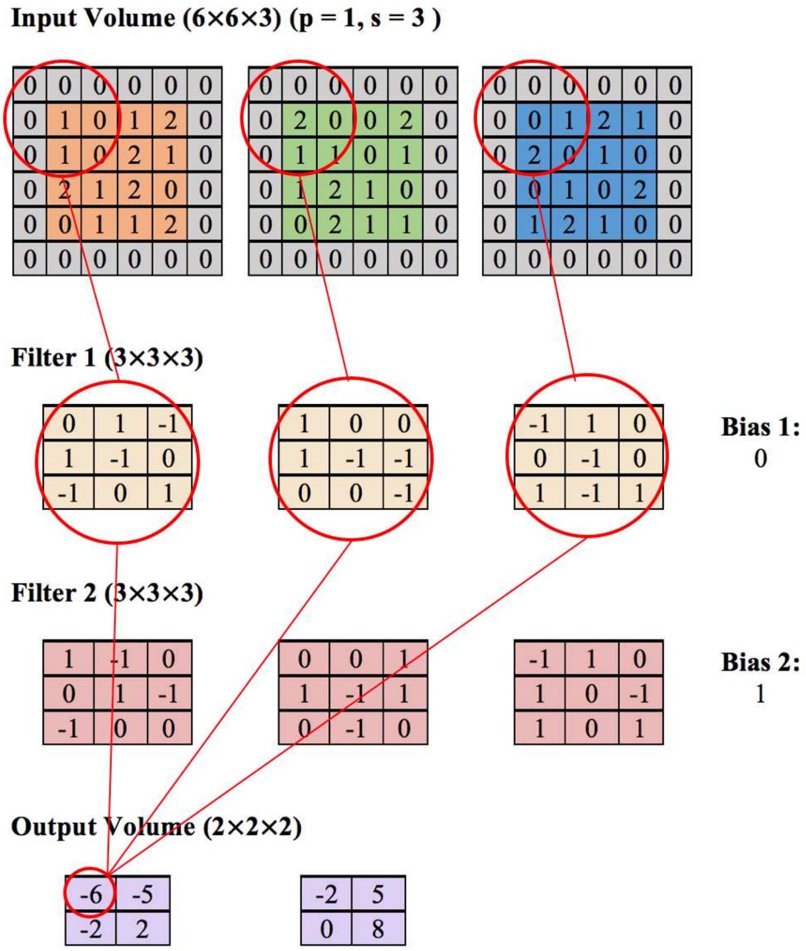


Figure 5.3.2.1 Computational process example of convolutional layer

Accordingly, the presentation and size of output volume at convolutional layer depend on values of filters and biases, and four parameters: size of input image size, number of filters, filter window size, stride and padding settings [103], [114], [131]. Figure 5.3.2.1 presents an example of computational theory at the convolutional layer. Multiplication of matrix between the filter and selected feature window from input patch plays the main role in the computational process. Summation of each output of matrix multiplication becomes one value of a feature map in relation to a filter. If the input image patch contains more than one channels, a filter will have the same number of channels accordingly, thereupon, one convolutional value (OCV) in a feature map is multiplication output summation of all channels, in addition, plus with a bias, which the process can be denoted as Equation 5.1 - 5.2.

$$C_w = \sum_{c=1}^{n} M_{ic} M_{fc} \tag{5.1}$$

$$OCV = \sum_{v=1}^{m} C_w + b \tag{5.2}$$

Where M_{ic} and M_{fc} are the matrixes of feature and filter windows, n is the number of channels, and m is the number of values in a convolutional window C_w. b denotes the bias. The practical example given in Figure 5.3.3 illustrated the actual computation, where the specifications of this example are listed as follows.

- Input patch size: 4×4×3
- Number of filters: 2
- Filter window size: 3×3
- Stride number: 3
- Padding number: 1
- Bias: 0

By selecting the first feature window as framed in the red circles on three channels of input patch, a convolutional value is computed in the way of multiplying matrixes, where the detailed examples are given as following:

$$C_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix} \times \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -2 & 0 \end{bmatrix}$$

$$C_{\mathrm{overall}} = C_1 + C_2 + C_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & -2 & -1 \end{bmatrix}$$

$$OCV_1 = 0+0+0+0-3+0+0-2-1+0 = -6$$

Where C_1, C_2, and C_3 are the matrix multiplication results for three channels and C_{overall} is the overall convolutional matrix of the input patch. The final convolutional value for this first window is represented by OCV_1 . After applying the same procedure to the entire patch, feature maps are output by convolutional layer. The size of output volume ($OPVS_x$, $OPVS_y$, $OPVS_D$) = (2, 2, 2) for this input patch is then computed as below:

$$OPVS_x = OPVS_y = \frac{4-3+2\times1}{3}+1 = 2; OPVS_D = 2$$

By denoting the size of the input image patch as ($X_{in}\times Y_{in}\times D_{in}$) and output volume as ($X_{out}\times Y_{out}\times D_{out}$), the dimensional values of output volume can be generally formulated as Equation 5.3 - 5.5.

$$X_{out} = \frac{X_{in}-F_C+2P}{S}+1 \tag{5.3}$$

$$Y_{out} = \frac{Y_{in}-F_C+2P}{S}+1 \tag{5.4}$$

$$D_{out} = M \tag{5.5}$$

Where F_C is the number of channels of each filter. P is the number of padding. S is the number of stride size. M denotes the number of filters.

In the architecture of b-CNN, a ReLU layer follows after convolutional layer to rectify the convoluted volume in the way of modifying values with a threshold. This layer performs max value determination between zero and each element and keeps the volume size unchanged. If the element is bigger or equal to zero, then it remains in the volume, otherwise, a zero replaces the element in the feature map, where the presentation is given below:

$$f(v) \begin{cases} v, & v \geq 0 \\ 0, & v < 0 \end{cases}$$

After the ReLU layer, a set of fully connected layers (illustrated in Figure 5.3.1, where h denotes layer numbers) performs feature combination analysis by flattening all the attributes from feature maps into one vector of neurons in every fully connected layer, and all the neurons connect to each other from one layer to the other layer to construct a fully connected network so as to classify the categorize for input image patch.

A structure of fully connected layers forms one classic neural network typically a multilayer perceptron network introduced in Chapter 4. Normally the number of neurons of last fully connected layer has the same numbers as classes, in our case, there are two classes which are matched and unmatched classes, accordingly, the last fully connected layer in b-CNN contains two neurons which outputs scores for the two classes.

At the output layer, the scores generated from the last fully connected layer are converted into class probabilities by Softmax function. This function computes the probability of each class over all classes, where the range of probability is between zero to one, moreover, the sum of all class probabilities equals one. The probabilities indicate which class the input patch belongs to, in other terms, the class with the highest probability is assigned as categorising result. The formula of Softmax function is defined by Equation 5.6, that calculates the exponential of an output value over the summation of all exponential of output values from the last fully connected layer, where x denotes a class score produced by fully connected layers and n is the number of scores in relation to all the classes, j ∈ (1, 2, …, n).

$$f(x_j) = \frac{\exp^{x_j}}{\sum_{i=1}^{n} \exp^{x_i}} \tag{5.6}$$

Softmax function normally cooperates with a loss function which is called as cross-entropy [103], [114], [131] to estimate the difference between target and output distributions, in other words, the error rate referring to the true class, moreover, the computation theory can be formulated by Equation 5.7.

$$E(T, O) = -\sum_{j=1}^{n} T_j \log O_j \tag{5.7}$$

Where $O_j = f(x_j)$, T_j and O_j indicate the correct distribution and output distribution estimated by the convolutional neural network for the j^{th} class. The error or distance of distributions between real classes and estimated classes are then represented by $E(T, O)$. The purpose of training procedure involves minimising this loss function by the means of finding out the best weight and bias that minimise the distribution divergence between true and output class.

The most commonly used training function in the community of CNN regards to Stochastic Gradient Descent (SGD) [131]. Gradient descent (GD) algorithm trains the neural network by updating the parameters with all samples in the input dataset. SGD adopts the same training theory as GD formulated by Equation 4.2 in Chapter 4, while calculates gradient and update weights and bias with every training instance, which can accelerate the speed of learning procedure. In the training of a CNN, SGD commonly performs training with mini batches that consists of a certain number of samples split from dataset instead of a single sample. With this method, the update operation occurs after analysing every mini batch. Moreover, SGD cooperating with mini batch can generate a smoother appearance of learning trend rather than individual instance based training, consequently, outperforms originally method that advancing learning ability with every case.

SGD algorithm sometimes may occur oscillate in finding the optimum following the route of steepest descent, thereupon, a term of momentum participates in to help reduce the oscillate so as to accelerate the computational speed, based on Equation 4.2, the computational theory can be formulated as Equation 5.8 [116]. SGD with momentum (SGDM) was considered as the main training function in all experiments for b-CNN in our research.

$$x_{k+1} = x_k - \mu\nabla E_k + \gamma(x_k - x_{k-1}) \qquad (5.8)$$

Where γ represents the contribution from the previous to current step, and x_{k-1} is the previous vector of current step k. Pooling layer is not included in the design of b-CNN, on account of adequacy information matter, as this layer downsampling feature maps which leads to the loosing of attributes, considering the input modality refers to an image patch containing a restricted amount of information within a certain window.

In consideration of the diversified characteristics of every layer introduced in b-CNN, experiments were implemented for the purpose of advancing and optimizing this model in respect to different parameter settings so as to maximize the performance of b-CNN. Following sections will present relative experiments and estimations with the results to reveal an optimal model for b-CNN.

5.4 Experimental optimizations and evaluations for designed CNN

In this section, experiments and evaluations of optimizing the performance of b-CNN will be presented in order to improve the classification accuracy in respect to categories of matched and unmatched pairs. The performance was calculated based on classification accuracy as given in Equation 5.9, which considers the rate of correctly assigned samples over the whole dataset. All the experiments were estimated with 10-Fold cross-validation. Where TC and OC are the target class and output class, and n is the total number of instances. $\delta = 0$ denotes the threshold to determine if a sample is correctly assigned, when the absolute difference value equal to zero, the output class is correct otherwise wrong category.

$$\text{AccuracyRate} = 1 - \frac{1}{n}\sum_{i=1}^{n}(|TC_i\text{-}OC_i| = \delta) \qquad (5.9)$$

A deep learning toolbox provided by [103] was adopted to accomplish the experiment, where the training function was SGDM (defined in Equation 5.8) setting.

5.4.1 Fundamental relationship estimation for convolutional layer parameter

In the design of b-CNN, the input is a patch containing difference features of a pixel pair rather than the entire image for the purpose of pixel level based corresponding estimation, under this circumstances, the size of patch window plays a significant role in the aspect of information efficiency.

Convolutional layer obtains values of feature maps with filters in a certain box, if filter box is big and patch window is small, the useful attributes that can be retrieved for a feature map then could be insufficient. Furthermore, the number of filters determines how many feature maps produced at the output stage of the convolutional layer, adequate maps could make the CNN perform the ability of hierarchical analysis efficiently. On account of these conditions, we implemented experiments to optimize the relationship between patch and filter window sizes, numbers of filters so as to maximize the performance of b-CNN.

In order to evaluate an optimal model for the relationship of patch and filter window sizes, numbers of filters, we designed several sets of value combinations with these four parameters on the basis of observing settings used in research [91].

Experimental values for each parameter increased with a certain step. The number of patch size was determined as four, and the incremental step for filter size was two, and the step for filter number was set to eight. This experiment utilised two fully connected layers for the purpose of observing convolutional parameters combination effect: the first one was experimented with different numbers of neurons in an incremental step - five, and the second one is the class score layer that had two neurons. The detailed setting specifications are listed as following:

- Input image window size set: (5×5, 9×9, 13×13, 17×17)
- Filter size set: (3×3, 5×5, 7×7, 9×9)
- Filter number set: (8, 16, 24, 32, 40, 48)
- Fully connected size set: (5:5:50)

Each filter size had one corresponding input image window size, where the value of filter size was the value of cantered pixel index of the input window. All the filter numbers were estimated with every set of patch window and filter, moreover, each such experiment was tested with every setting of neuron number at the first fully connected layer. Experimental results are given in Figure 5.4.1.1 – Figure 5.4.1.4.

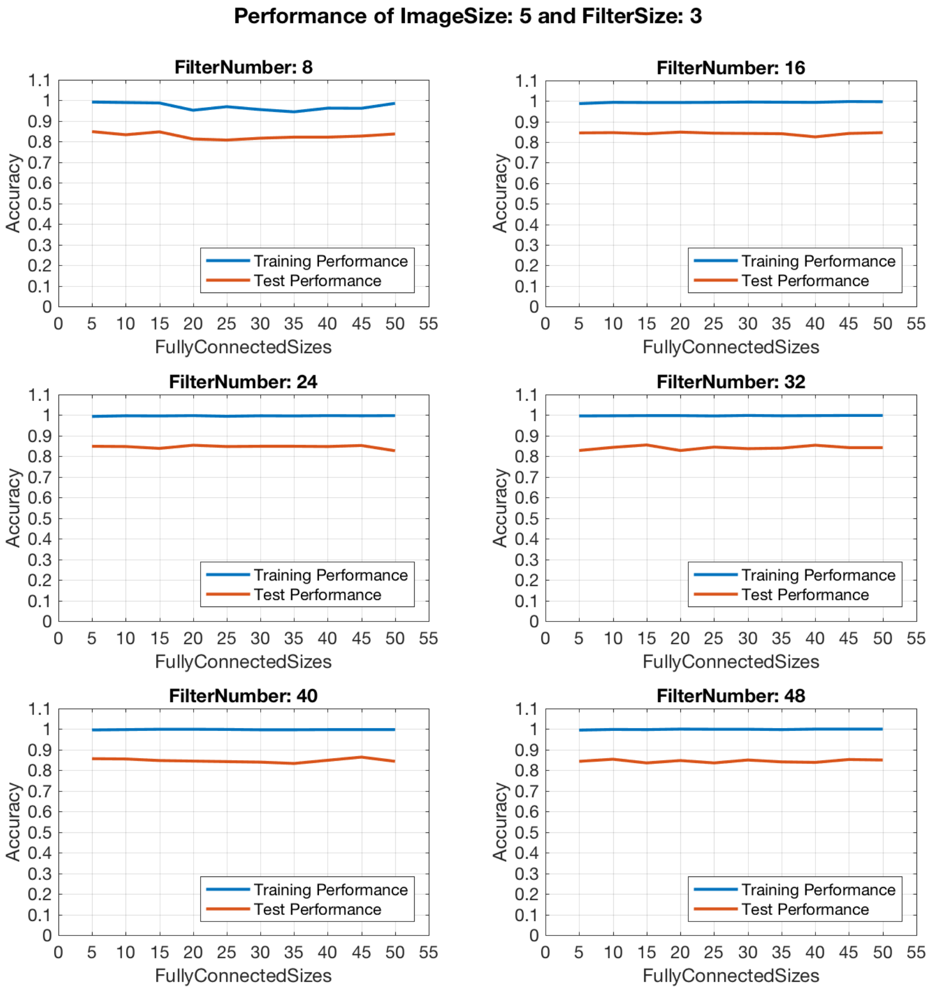


Figure 5.4.1.1 Parameter relationship estimation with Image Size 5 and Filter Size 3

Experimental performances were plotted based on each combination of input image size and filter size. Each combination has 6 subplots, which shows performances of experiments carried out with all values of filter numbers and fully connected sizes. Each subplot in these figure presents the performance of one set of patch and filter selection trained with one filter number in respect to all neuron numbers settings at the first fully connected layer. As shown in Figure 5.4.1.1 with input patch size equals to five and filter size three, the performance with every filter number can produce a good performance for the setting of every neuron value at the first fully connected layer. During these results, filter number 8 produced the most unstable performance as the lines for both training and test performance go up and down as shown in the first subplot.

Performances with other filter numbers in Figure 5.4.1.1 have generated approximately similar and relative accuracies. This closer observation reveals that the small number of filter for this combination of patch and filter size cannot perform an efficient analysis. Moreover, these subplots present that these combinations with different neuron values made barely changes to the level of accuracy in relation to each filter number setting respectively.

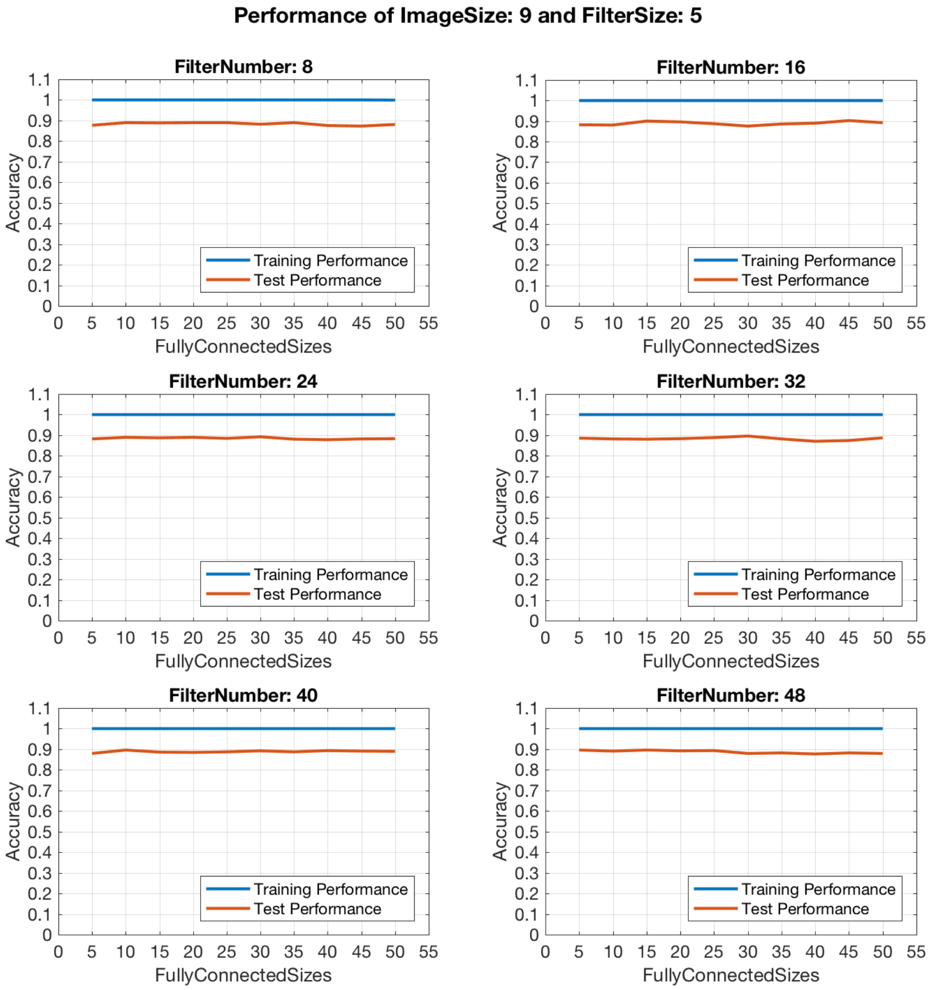


Figure 5.4.1.2 Parameter relationship estimation with Image Patch Size 9 and Filter Size 5

Performances with Image Patch Size 9 and Filter Size 5 presented by Figure 5.4.1.2 shows that the accuracies produced with all numbers of filters had similar levels, in addition, the subplots illustrate that both training and test performances keep approximately stable trend with all fully connected neuron values. These appearances imply that using this combination of input patch and filter size settings, the filter number and neuron value settings cannot make a large influence on training process, in other terms, this set of input patch and filter size can make b-CNN implement stable learning capability.

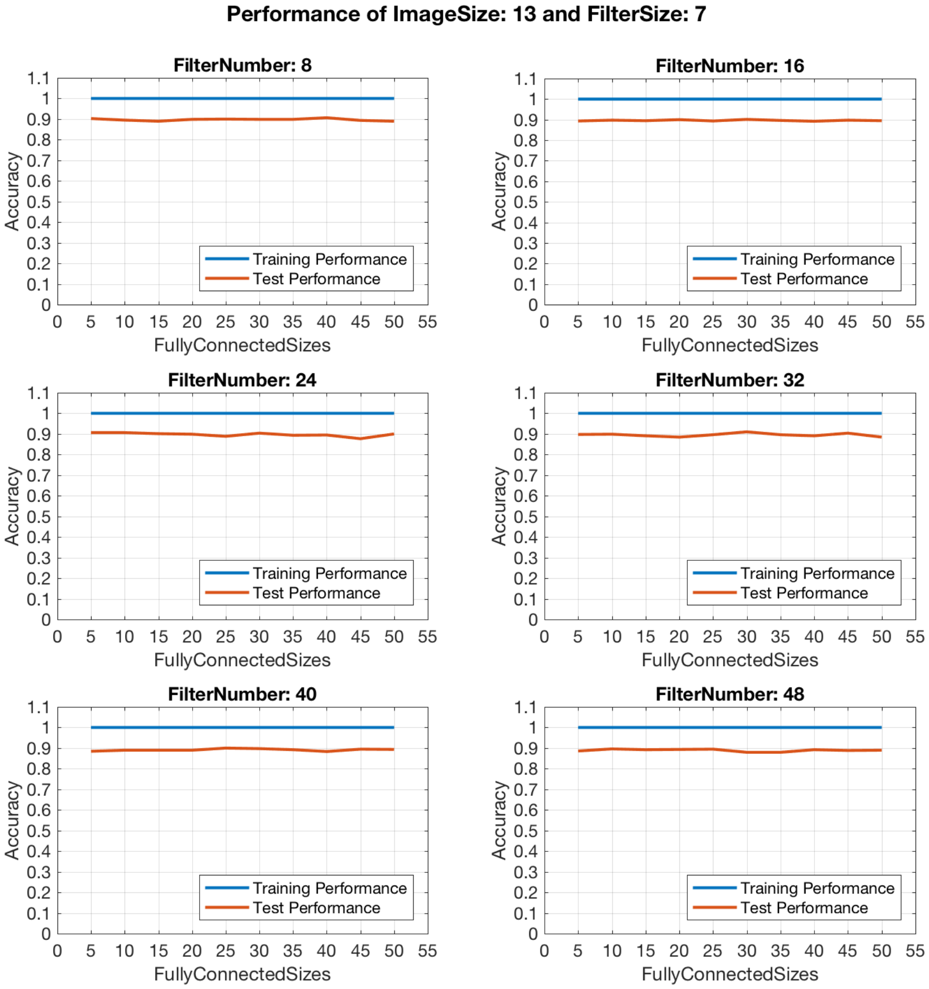


Figure 5.4.1.3 Parameter relationship estimation with Image Patch Size 13 and Filter Size 7

Experimental results generated by the combination of Image Patch Size 13 and Filter Size 7 indicates that from small to large values of filter number all have produced similar performance with different neuron numbers without specific changes on the lines shown in all subplots of

Figure 5.4.1.3, that is to say, this relationship between parameters can reach to a relatively balanced stage.

The parameters combination referring to Figure 5.4.1.4 produced stable results in respect to the number of filters did not make significant variations on the first five values. The last filter number setting with 48 appeared extrusive indeterminateness with neuron number 30 in contrast to other subplots that presents all neuron values had the similar accuracy, which means, filter number 48 can not efficiently cooperate with other parameters in this combination model.

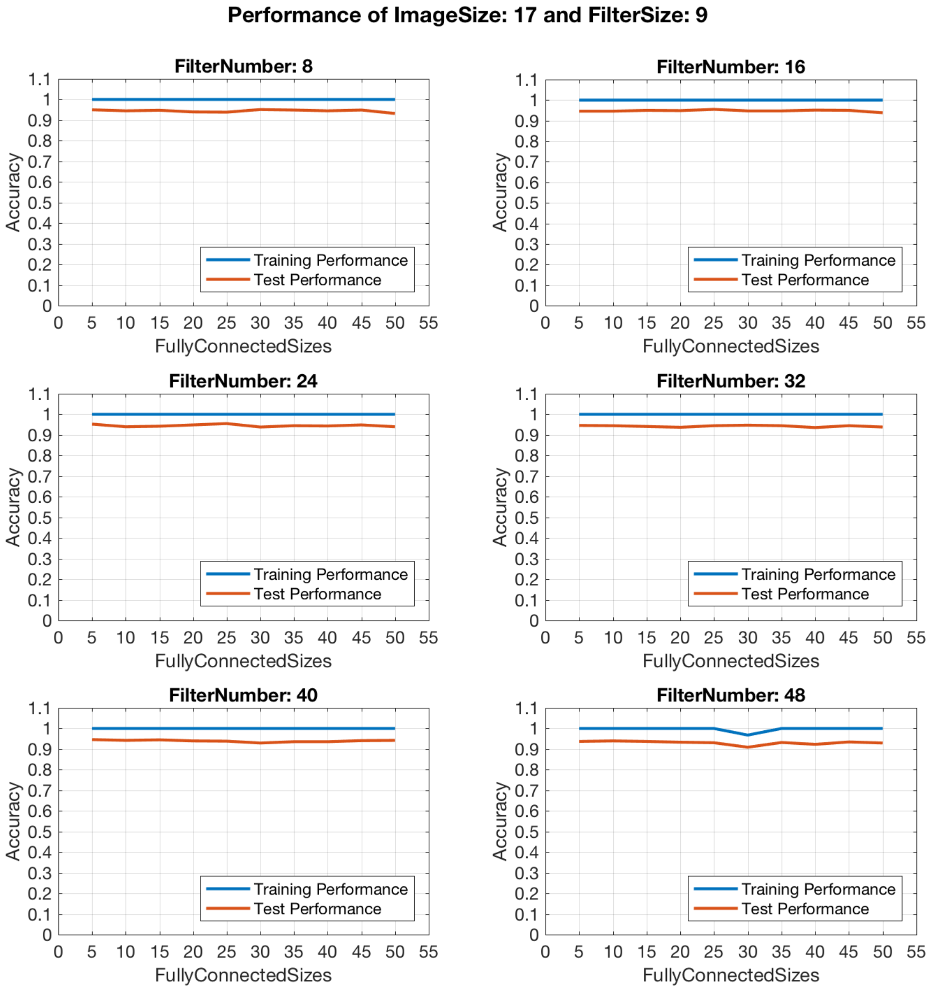


Figure 5.4.1.4 Parameter relationship estimation with Image Patch Size 17 and Filter Size 9

On one hand, b-CNN with all four combinations of parameters generated relative stable and high level of accuracies as the results shown in these four figures. This implies that the inside relationship of each combination could all achieve a relative balance between each other, which means the way of selecting values in respect to compatibility among these parameters were suitable for each combination respectively. However, from the observations of these results, 8 and 48 were excluded from the optimal model on account of the unstable performance produced in the first and fourth combinations shown in Figure 5.4.1.1 and Figure 5.4.1.4. Therefore, we considered such method for setting parameter values as the selectable solution for the convolutional process that involving selection of filter size in accordance with input patch size with filter number between 16 and 40.

Combination Set	**Patch Size 17 and Filter Size 9**	
	32 Filters	40 Filters
Average Accuracies	0.9426	0.9396

Table 5.4.1.1 Average accuracies with 32 and 40 filters for the combination of Patch Size 17 and Filter Size 9

On the other hand, comparing performances of four combinations, with the increased size of input image patch, the level of accuracy improved for test set, from the four figures we can see, the red line representing the test performance gradually moves up (in other words, accuracy increased) along with incremental number settings for patch window size accompanying with corresponding filter size. In the view of this phenomenon, b-CNN requires more information to reach to the higher level of learning ability so as to produce an advanced performance. For this reason, Image Patch 17 and Filter Size 9 were chosen as parameter settings for the convolutional layer of b-CNN.

Furthermore, the accuracies that produced with filter number 16 and 24 appear to drop down when increasing the number of neurons for the combination presented in Figure 5.4.1.4, so that these two choices were removed from the optimal selection of filter number setting. Moreover, according to the average accuracies (mean accuracy overall estimations in respect to x-axis direction in Figure 5.4.1.4) for implementations with filter number 32 and 40 listed in Table 5.4.1.1, filter number set with 32 outperformed the comparisons. For this reason, we considered 32 as the final optimal choice for the filter number at the convolutional layer.

5.4.2 Model selection for fully connected layers of b-CNN

As shown in Figure 5.3.1.1, we adopted a series of fully connected layers to combine attributes from feature maps between ReLU and two neurons layers. The number of these fully connected layers and their neurons size affect the efficiency of a network functioning as hidden layers as described in Chapter 4. The ideal combination of parameters can advance the learning ability of a CNN. A diverse number of hidden layers with different neuron sizes were estimated referring to the classification accuracy for the purpose of finding a better model for constructing fully connected layers.

In order to observe the impact of the number of fully connected layers, our experiments estimated the accuracy along with the increasing number of hidden layers. There were four sets of fully connected layer sizes used in the specific experiment. The parameter selections for convolutional layer employed the set that was investigated from the previous section as an optimal combination. The detailed descriptions of b-CNN settings are listed as follow:

- Input image window size: 17×17
- Filter size: 9×9
- Filter number: 32
- Fully connected size set: (2, 3, 4, 5)
- Total neurons for each fully connected layer set: (20, 30, 40, 50)

The number of layers in each set increased one layer from the previous set by starting with two fully connected layers. Table 5.4.2.1 presents the classification accuracies produced by trained b-CNN integrating parameters as given above in this experiment, where Set 1 with two layers, Set 2 with three layers, Set 3with four layers, and Set 4 with five layers. For each fully connected layer of each set, neuron values were set to 10 equally on every hidden layer.

Fully Connected Set	**Set 1 (2 layers)**	**Set 2 (3 layers)**	**Set 3 (4 layers)**	**Set 4 (5 layers)**
Accuracies	0.9232	0.9321	0.9345	0.6667

Table 5.4.2.1 Accuracies with different fully connected layer sizes

Along with the increasing number of fully connected layers, the classification accuracies increased step-by-step until reach to five layers. This implies that the growing number of layers can improve the performance of learning, whereas this does not mean the optimal resolution should refer to construct this layers as unlimited large as possible, as shown in Table 5.4.2.1, there is a limitation occurred during the experiments that the accuracy significantly decreased with five hidden layers.

According to this matter, every model should cooperate with one suitable designed fully connected layers specifically rather than selecting a big value for layer number casually. Among the accuracies provided by Table 5.4.2.1, the best performance was produced by network integrating with Set 3 utilising four hidden layers, which indicates that this setting can be fit for the devised b-CNN. The characteristic training processes for training with different numbers of layers are illustrated in the following figures.

Training·Process·with·10×2·Fully·Connected·Layers¶

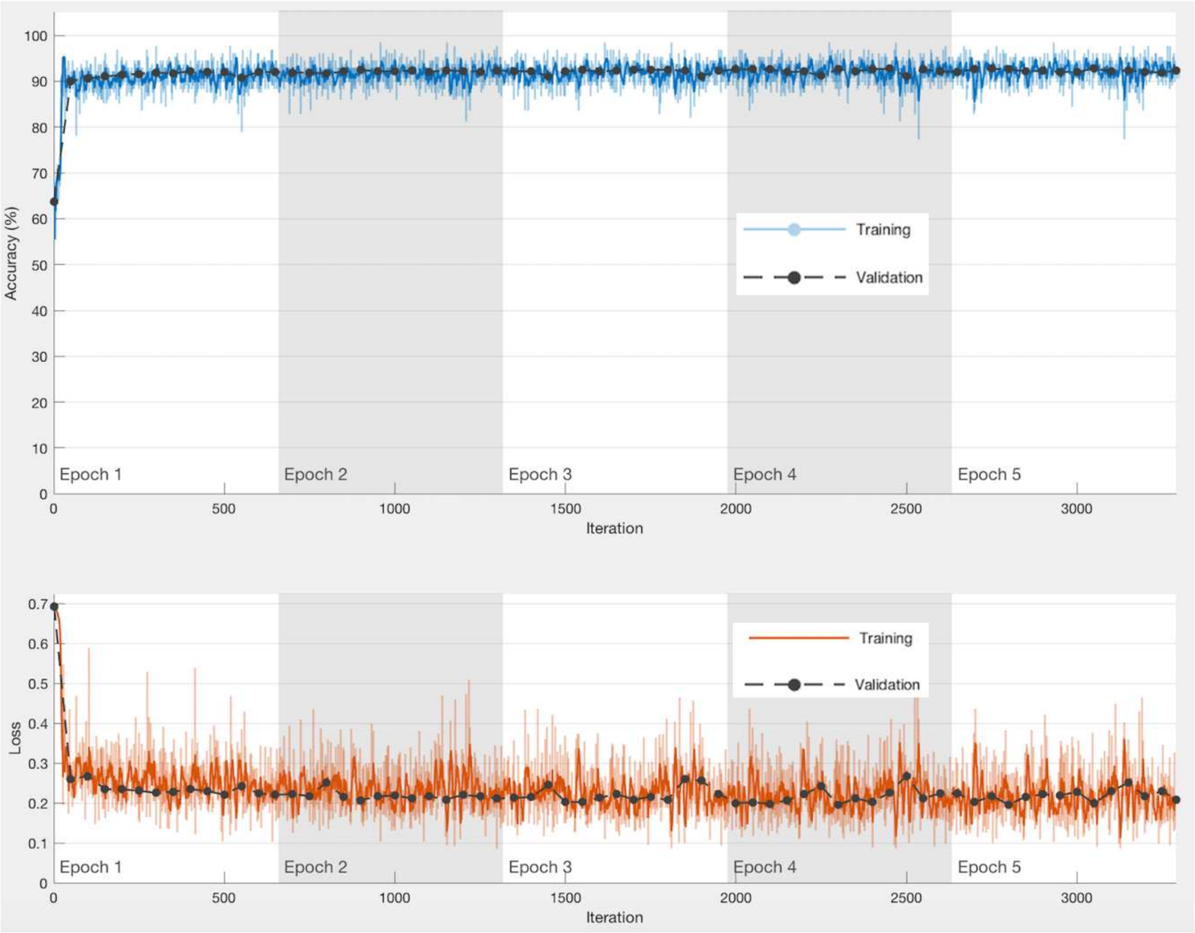


Figure 5.4.2.1 Training process with Set 1 (10×2) fully connected layers

There are two subplots in training process figures in accordance with two performance estimation method. The first subplots present the training process in relation to the classification accuracies on the basis of Equation 5.9, and the second subplots show the error rate computed with the loss function formulated by Equation 5.7. The trends of accuracy and loss go inverse direction, where the higher accuracy denotes lower loss in respect of the difference between correct class and output class.

The blue and red lines represent the training accuracy and loss rate, and the black dash line with dots signifies test performance for both measurements respectively in these figures. The x-axis indicates the iterations based on the size of mini batch, and the grey column with number states the number of epochs in training. All the training processes were stopped before the significant drop of accuracies for training and test which indicates the point of excessive training that causes network losing the ability of correct analysis with given data.

Training Process with 10×3 Fully Connected Layers

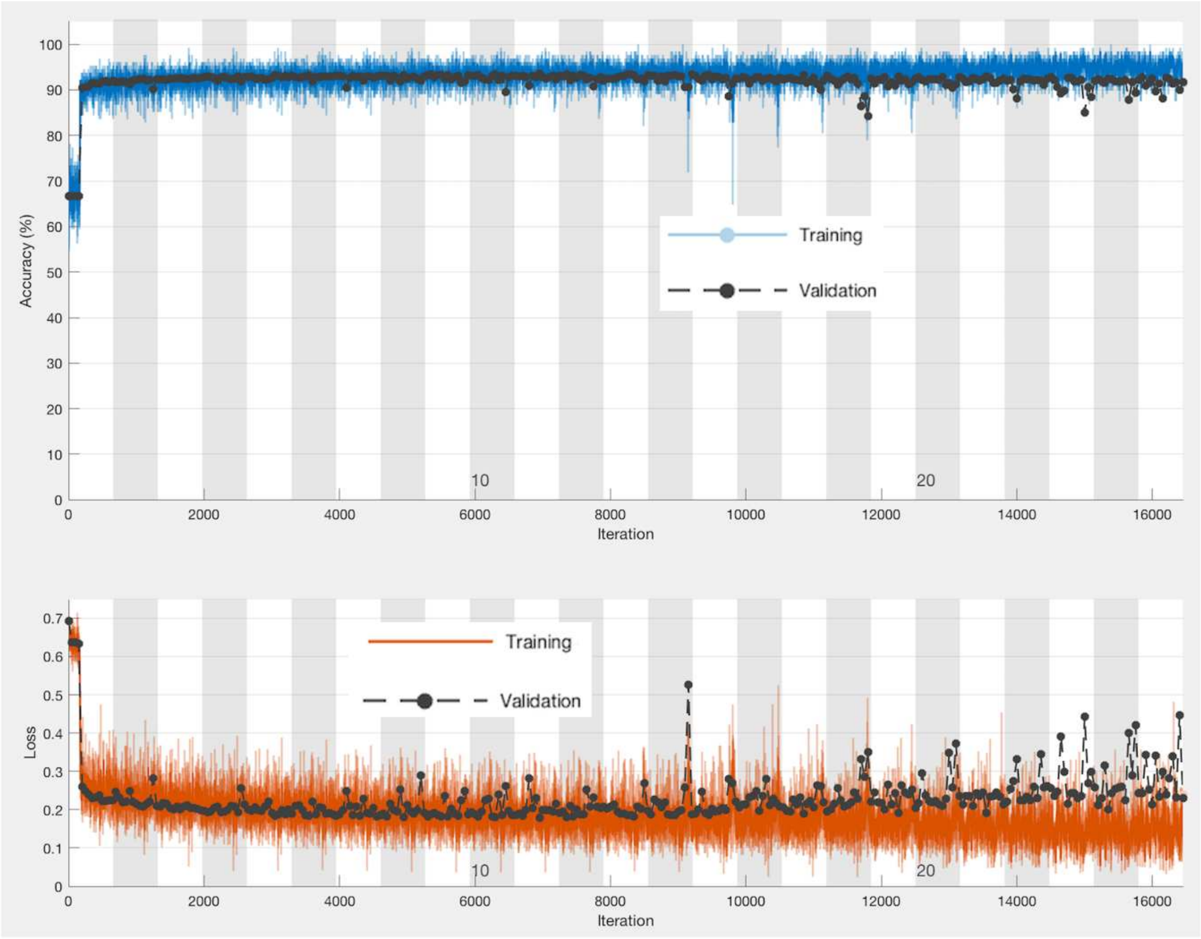


Figure 5.4.2.2 Training process with Set 1 (10×3) fully connected layers

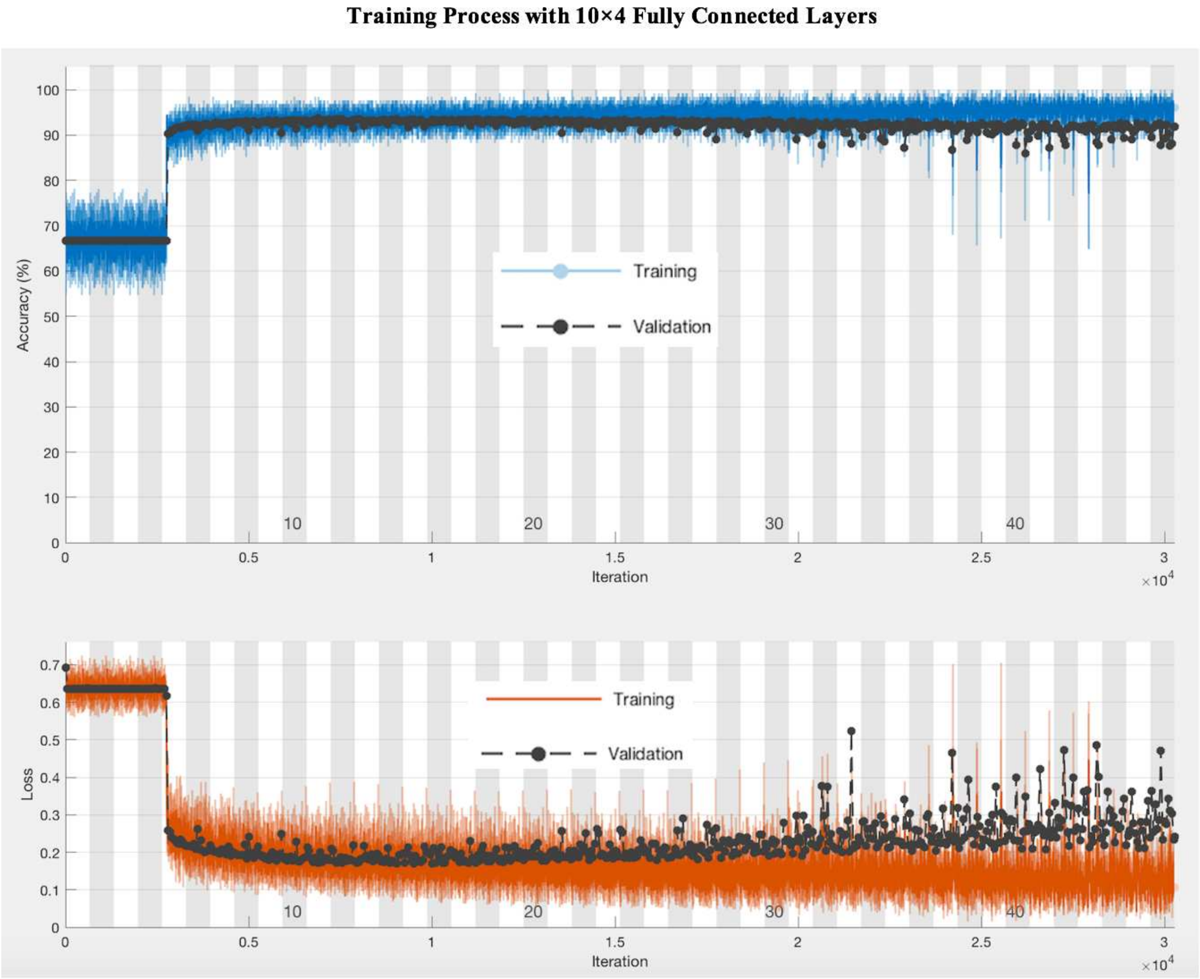


Figure 5.4.2.3 Training process with Set 1 (10×4) fully connected layers

Figure 5.4.2.1 to Figure 5.4.2.4 show the training processes of b-CNN using four sets of fully connected layers, where Figure 5.4.2.1 for two layers, Figure 5.4.2.2 for three layers, Figure 5.4.2.3 for four layers and the last Figure 5.4.2.4 for five layers. By employing two and three layers, the first improvement of accuracies from initial learning happened at the first epoch, which very early iteration for two layers and one third for three layers. When the number of layers increased to four, accuracies rose in the fifth epoch. The training with four layers took a longer period than with two and three layers to reach to the first level of adaptive learning. In consideration of this phenomenon, one state can be found that large size of layers could lead to learning procedure requiring more epochs to achieve better performance.

As subplots for loss estimation shown in Figure 5.4.2.1 to Figure 5.4.2.3, while layer size increased from two to four, the overfitting issue of training appears to raise before training procedure stops, moreover, the more training, the more unstable performance produces. With two layers, overfitting barely happened before the stop stage, whereas processes with three and four layers gradually overtrained the network and the trends of training and test went towards opposite tendency after reach to the best test point, moreover, this matter occurred more in the process with four layers. These clues can prove that a network with a small number of layers cannot learn properly to find out a convergence point.

Training Process with 10×5 Fully Connected Layers

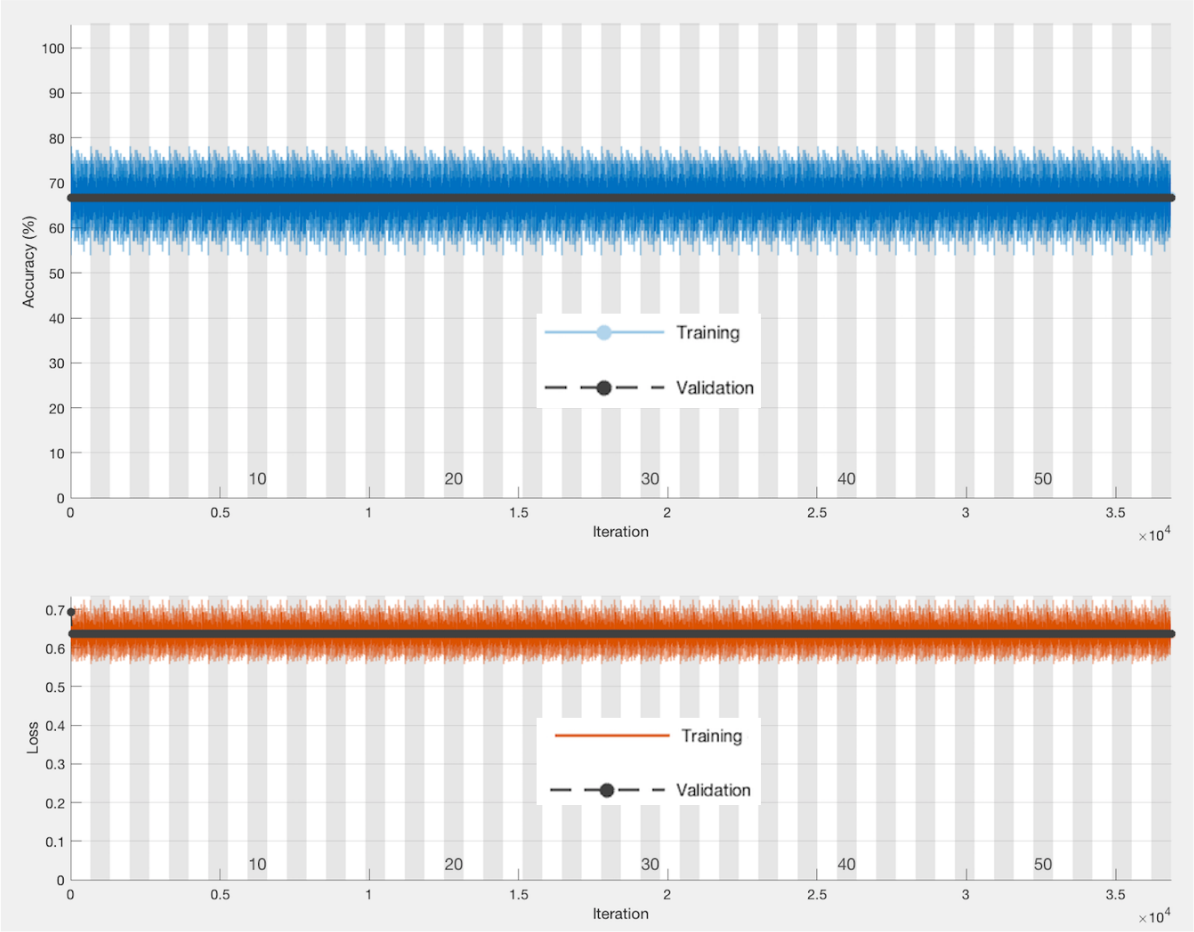


Figure 5.4.2.4 Training process with Set 1 (10×5) fully connected layers

Figure 5.4.2.4 presents the situation when the network composed with the structure of fully connected layers that beyond the capability of the entire model, which caused corrupt training process. The whole training process with five layers shown in Figure 5.4.2.4 keeps low performance both for accuracy and loos measurement all the time, which means five hidden layers can not suitable for b-CNN with current parameter settings at the present stage of network construction and training. So far, training with four hidden layers have proved to be the ideal solution for constituting fully connected layers as the part of b-CNN.

Accuracies with 20×4 Fully Connected Layers	Accuracies with 30×4 Fully Connected Layers
0.9351	0.9328

Table 5.4.2.2 Performance of different incremental neurons with four fully connected layers

In consideration of maximizing the performance of b-CNN, we increased the number of neurons at each hidden layer to explore the possible influence. On the basis of previous layer number estimation, the implementation of experiment adopted methodologies that the neuron values added a step value at each hidden layer for structure integrating with four layers, where the step value was set to ten. Table 5.4.2.2 present the experimental outputs for this investigation.

Training Process with 20×4 Fully Connected Layers

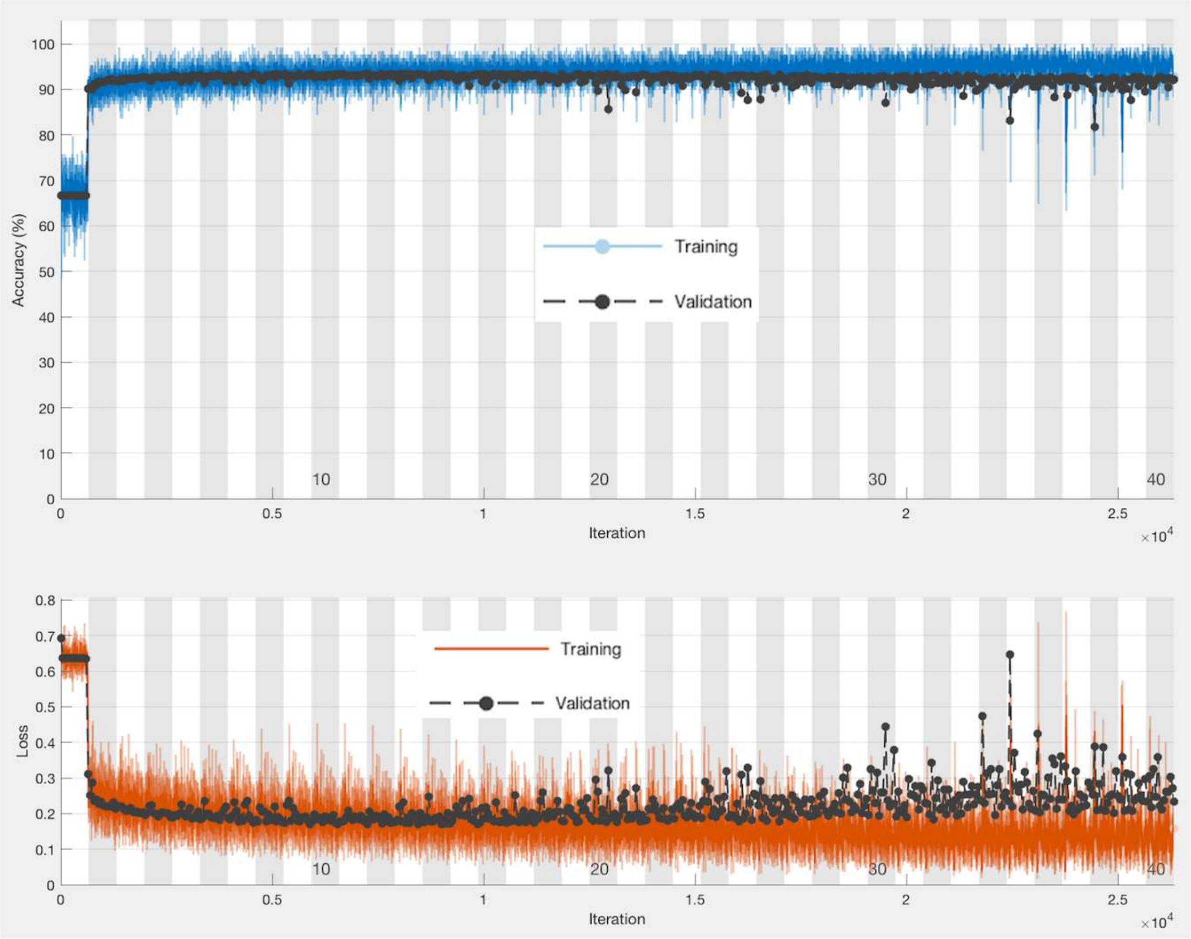


Figure 5.4.2.5 Training process with Set 1 (20×4) fully connected layers

Table 5.4.2.2 lists the classification accuracies for the network utilising twenty neurons and thirty neurons for four hidden layers. Comparing with the accuracy produced by four layers with ten neurons in Table 5.4.2.1, the accuracy improved with twenty neurons and reduced when with thirty neurons. It can be seen that increasing the number of neurons can advance the performance of b-CNN, however, the advancement as well as comes with a limitation, since when using thirty neurons, the accuracy decreases as the results given in Table 5.4.2.2.

A representative training process with twenty neurons at every layer for the four-layer structure is illustrated by Figure 5.4.2.5. From the figure we can see, the first stage of accuracy jumping from low to high level happens at the end of the first epoch, whereas this situation occurs at the fifth epoch for training with ten neurons. Moreover, the stopping point (where the performance becomes significant low) of adopting twenty neurons appeared at an earlier stage, which was six epochs smaller than ten neurons structure. Furthermore, the overfitting appearance reduced in the training process of twenty neurons framework in contrast to network with every hidden layer integrating with ten neurons.

In the view of these circumstances, the way of adding neuron numbers can not only improve the speed of training procedure, but also advance the learning capability of b-CNN in consequence with better performance. From all these observations, the structure consists of four hidden layers with twenty neurons at every layer was considered as an optimal model for building up fully connected layers in b-CNN.

5.4.3 Mini batch size selection

Stochastic Gradient Descent algorithm trains by updating learnable parameters for the network with every instance as introduced in Section 5.3.2, however, in practical situation, training data for a CNN sometimes may involve large dataset, in this case, the dataset normally is divide into many mini sets which is so called mini batches so as to update parameters on the basis of each mini batch. Since mini batch refers to the subset of the whole dataset, one parameter is in need of consideration, which comes to the size of mini batch. This parameter decides the speed of the whole training process, in other terms, the processing speed depends on how many iterations in every epoch. The term of iteration represents the number of mini bathes used in training, where one epoch contains iterations for updates with all mini batches. Furthermore, by updating with different numbers of samples at every iteration, the performance can also be varied accordingly, in addition, the convergence characteristics of learning capability also get influenced by this factor.

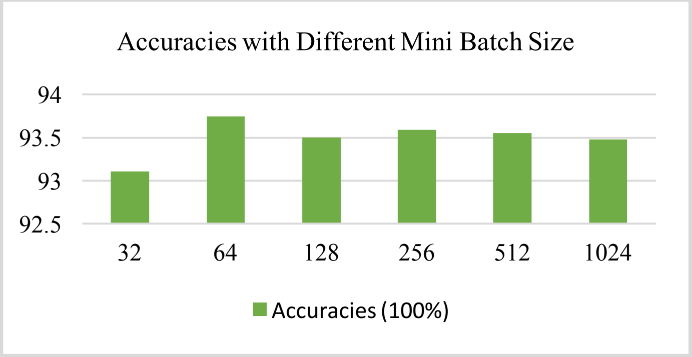


Figure 5.4.3.1 Performance produced with different Mini batch sizes

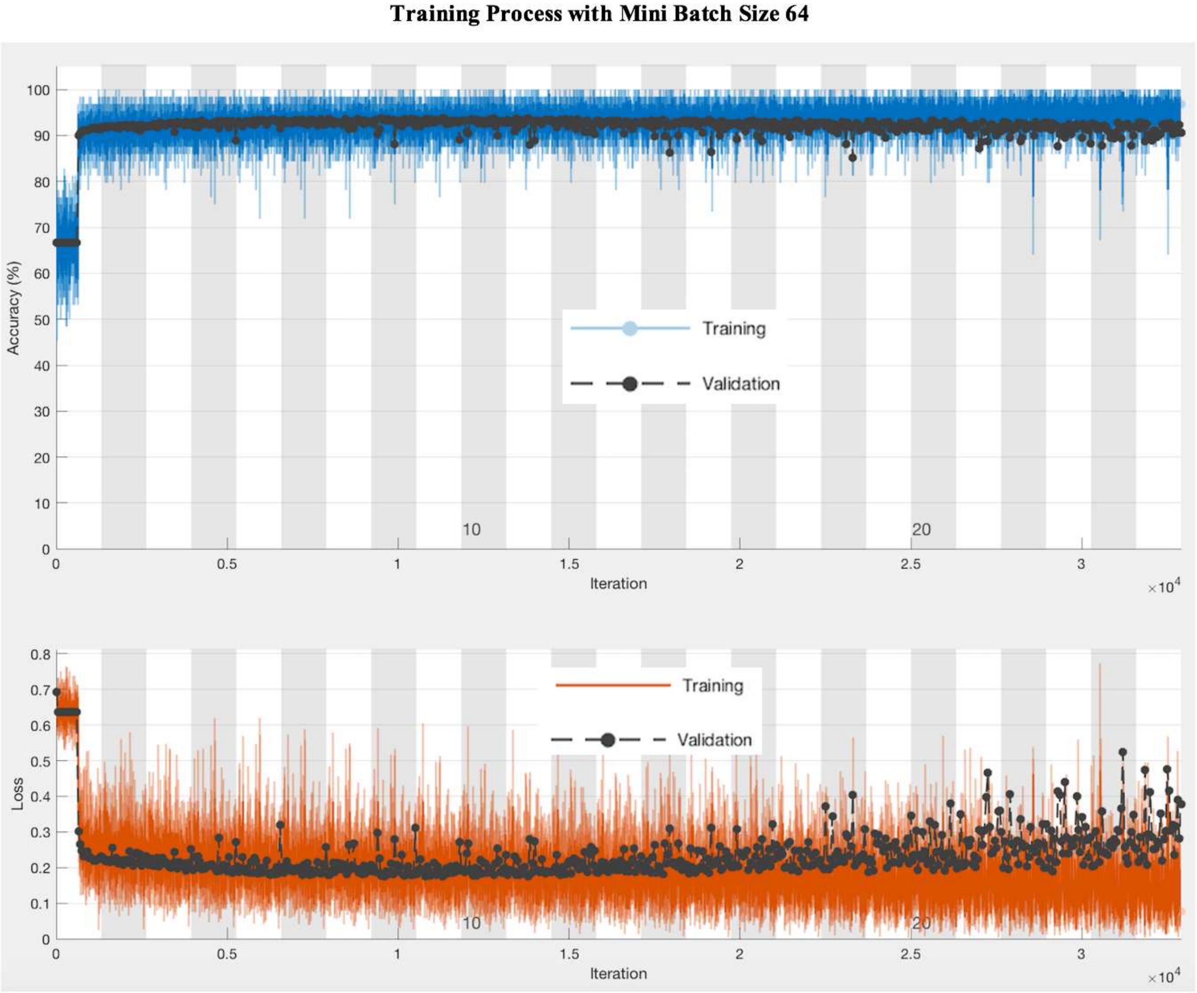


Figure 5.4.3.2 Training process with 64 mini batches

Training Process with Mini Batch Size 256

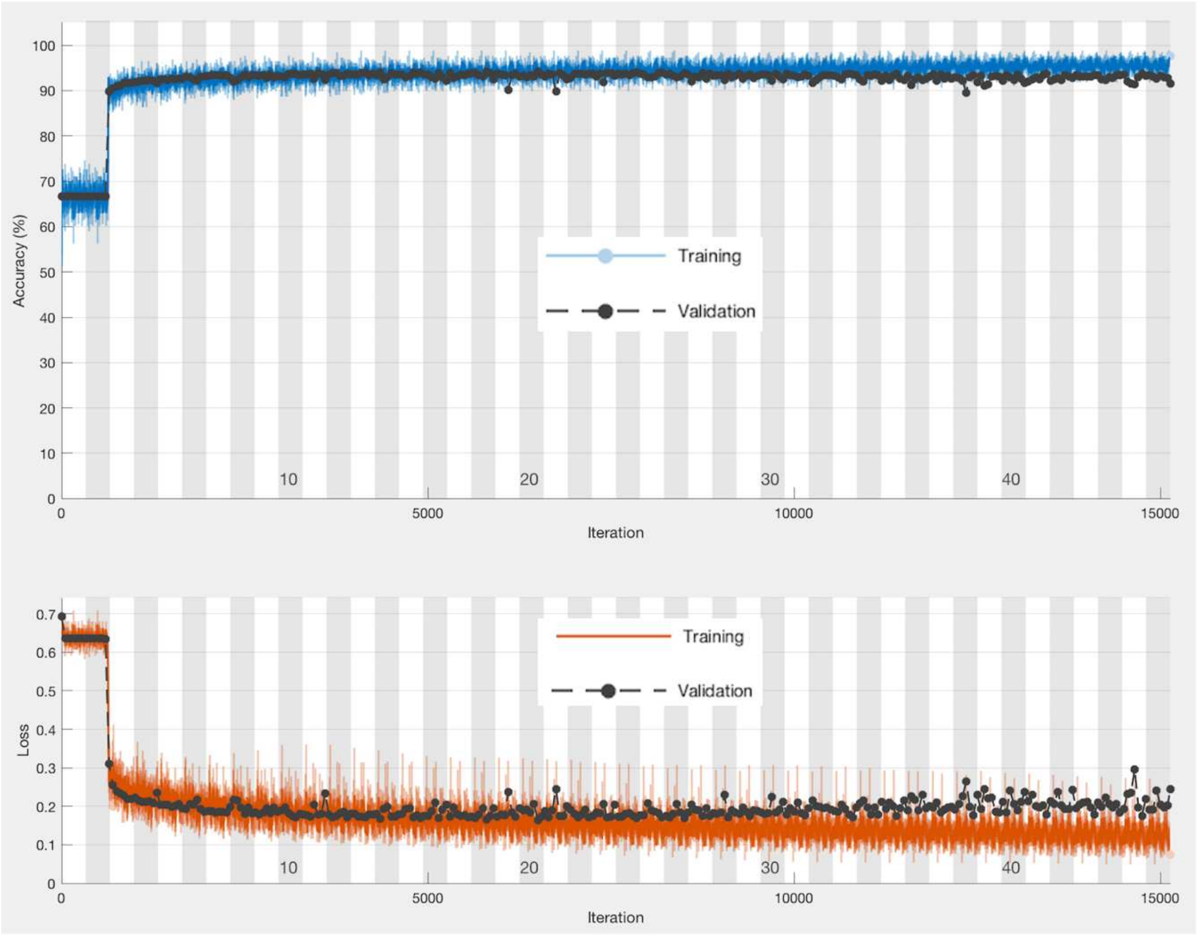


Figure 5.4.3.3 Training process with 256 mini batches

In accordance with the impact of mini batch size, experiments were carried out to research the contribution of mini batch in optimizing b-CNN. Model of b-CNN was trained with different numbers of mini batch, and the generated accuracies are presented in Figure 5.4.3.1. This investigation adopted parameters that were derived from experiments in previous sections and were deemed to be optimal choices for convolutional and fully connected layers at the current point, where the details are given below:

- Input image window size: 17×17
- Filter size: 9×9
- Filter number: 32
- Fully connected layers setting before two neurons layer:
  4 layers, 20 neurons at each hidden layer
- Mini batch sizes: (32, 64, 128, 256, 512, 1024)

There were six values used in experimental work based on computer memory constraints which range from not too small to not too large scope in consideration of the memory capability. In Figure 5.4.3.1, the performance is represented by classification accuracies, the higher bar indicates higher accuracy, and the x-axis denotes the experiment with mini batch sizes. The results show that performances improved from 32 to 64 mini batch size, and dropped with size 128, and then increased back a bit with size 256, after this point, accuracies kept decreasing with size 512 and 1024 slightly. As the changes between accuracies illustrated by Figure 5.4.3.1, the performance can be affected by a different number of division rule for mini batch from the entire dataset, thus, for the purpose of maximizing the performance of the network, a CNN model should integrate with an adaptive size.

Furthermore, training with size 64 and 256 produced top two performance, and their characteristic training process are shown in Figure 5.4.3.2 and Figure 5.4.3.3. As the training process presented, larger mini batch size can shorten the number of iterations in every epoch, however, take more epochs to reach to the first adaptive learning level and stop point as mentioned in the previous section. Moreover, the trend of training process can be smoothed with the larger size of mini batch on account of parameter update with less number of iterations. Nevertheless, although size 256 can perform training with a smoothed tendency in short iterations in total, the generated performance cannot prevail over the training with size 64. The purpose of parameter selection for model optimization focused on performance advancement in our research, in addition, training procedure can be implemented offline which would not affect the speed of disparity map computation in real time application, therefore, division size 64 for creating mini batches were deemed to be a better resolution for b-CNN optimization.

5.4.4 Learning rate optimization

Experimental model optimization of our study as well as investigated optimal parameters in relation to learning rate to improve performance for b-CNN as one factor plays a role in affecting the efficiency of updating learnable parameters for the network as explicated in Chapter 4 Section 4.4.4.1. An adaptive learning rate can advance the performance of the learning procedure for a model. Generally speaking, a CNN with SGD algorithm can perform the training process with constant learning rate from the beginning to the end. While sometimes with a constant value, the training may not be able to find the optimal convergence point, in this case, learning rate normally is reduced with a factor at a certain point so as to achieve performance as higher as possible.

For the purpose of finding out a learning rate and its corresponding drop factor that can advance the accuracy of network, relative experiments were carried out. This exploration found on the basis of the model with parameter settings obtained from previous sections for every layer in b-CNN. An initial learning rate was firstly estimated, and then an optimal drop factor for the selected learning rate was discovered for the purpose of further performance improvement. In consideration of the value of learning rate should not be too small or too big in respect that one may take a long time to train and the other one could cause poor performance as presented in Section 4.4.4.1, the first initial learning rate was set to 0.01 which is the most commonly used initial learning rate, and then increased to 0.02. Table 5.4.4.1 gives comparisons of output classification accuracies between training with learning rate 0.01 and 0.02.

Accuracies with Learning Rate 0.01	**Accuracies with Learning Rate 0.02**
0.9375	0.9315

Table 5.4.4.1 Accuracy comparisons for learning rate 0.01 and 0.02 for b-CNN

The comparisons listed in Table 5.4.4.1 indicates that when learning rate increases to 0.02, the accuracy reduces in contrast to training with learning rate 0.01. This implies that 0.01 as widely adopted learning rate value was suitable for b-CNN combining with other selected parameter settings referring to convolutional and fully connected layers, moreover the mini batch size. According to this observation, learning rate with 0.01 was then experimented with different drop factor in order to further develop the ability of network classifying matched and unmatched classes for pixel pairs. A drop factor reduces a learning rate in the way of multiplying the previous rate so that the rate decreases step-by-step. If drop factor is too big, the learning rate would drop too fast which can lead to the learning process may miss and can not find the optimal convergence point, conversely, small drop factor can have a higher possibility of minimizing the error rate whereas with more processing epochs to reach to the optimal rate. Taking this matter into consideration, drop factors employed in the experiment were 0.1, 0.2 and 0.3 as shown in Figure 5.4.4.1.

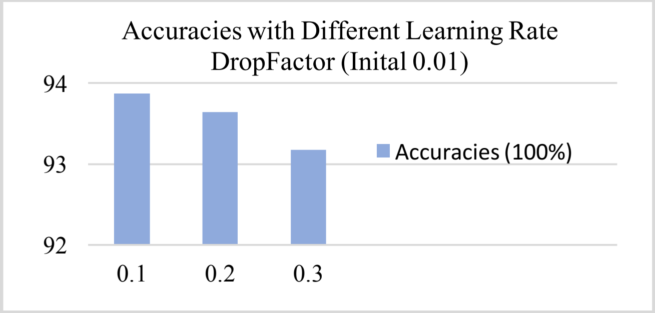


Figure 5.4.4.1 Performance of different learning rate drop factor from 0.01 for b-CNN

As accuracies with different drop factors provided by Figure 5.4.4.1, factor 0.1 outperforms three values, which means 0.1 can be fit with b-CNN rather than larger values, in other words, the model of b-CNN did not require a larger step of learning rate drop in respect to the relative stage of optimization. Therefore, learning rate 0.01 and drop factor 0.1 were considered as the optimum selections for b-CNN training. So far, form all the observations and evaluations of experiments for b-CNN optimization, b-CNN have proved the capability of performing projects referring to stereo pixel matching.

5.5 Chapter summary

This Chapter has firstly introduced deep learning in computer vision, and then state-of-the-art referring to stereo correspondence algorithm based on convolutional neural network was explicated to present the circumstance of such research field.

Mainly, this Chapter presented the novel stereo corresponding approach found on convolutional neural network, which the innovative creation of designed convolutional neural network (b-CNN) that was built by this project was interpreted which involving the architecture design and the methodologies for every layer. Moreover, experimental model optimizations and evaluations that were carried out by the project were as well as included in this Chapter, specifically from aspects relate to parameter combinations at the convolutional layer, construction of fully connected layers, mini batch size and learning rate with drop factor in detail. The generated performances have shown the effectiveness of implementing b-CNN with stereo corresponding estimation.

Chapter 6: Disparity map computation and evaluation

In computer vision, the 3D reconstruction mainly refers to retrieve spatial dimensional value which is known as vertical (x-axis), horizontal (y-axis) and depth (z-axis) measurement. For a captured stereo views by cameras, vertical and horizontal information can be easily obtained in accordance with the height and width of the image, however, the depth information requires specific algorithms to extract, which normally comes down to disparity map computation that can be obtained with stereo correspondence technique. The quality of disparity map relies on whether corresponding points can be correctly matched, for this reason, a disparity map plays a role in straightforward evaluation for the performance of stereo correspondence algorithm.

This Chapter introduces a series of approaches from the novel optimization method created and designed by this project for the computational speed of disparity map generation to design a certain procedure of post processing to refine the raw disparity map in the proposed pipeline, which the innovative methodologies of this optimization approach and refinement algorithms in these aspects are presented in detail. Moreover, the performances of created networks are compared and evaluated with the state-of-the-art based on a benchmark, and further comparison and evaluation between created networks are as well as presented.

6.1 Fundamental computation of disparity map

Disparity computation refers to find the correspondence between pixels. A disparity at a pixel coordinate is derived from the difference between x coordinates of a paired stereo pixels based on triangulation principle in the epipolar space. Assigning disparity for every pixel forms a dense disparity map.

6.1.1 Disparity with triangulation theory

The term of stereo disparity involved in epipolar geometry that is one of coefficients constituting the computational theory of triangulation. Figure 6.1.1.1 shows the geometry of triangulation theory for depth estimation [150]. Equation 6.1 presents this geometry in the form of the mathematical principle. Suppose a stereo image pair is in a condition that has been rectified to a common image plane containing parallel epipolar lines for every pair of corresponding points. The depth (Z) then can be computed according to the triangulation principle formulated as Equation 6.1. As denoted by the Equation 6.1, the computation of 3D depth involves three parameters: baseline (b), focal length (f) and disparity (d) of matching points [7], [8], [45], [53], [150]:

$$Z = \frac{b\,f}{d} \tag{6.1}$$

Where d is the disparity between x coordinates of left (x_L) and right (x_R) point, that are determined with one matched corresponding pixel pair:

$$d = x_L - x_R \tag{6.2}$$

And the values for X and Y coordinates are formulated as follows:

$$X = \frac{x_L\,Z}{f} \quad \text{or} \quad b + \frac{x_R\,Z}{f} \tag{6.3}$$

$$Y = \frac{y_L\,Z}{f} \quad \text{or} \quad \frac{y_R\,Z}{f} \tag{6.4}$$

Where y_L and y_R are the vertical coordinates of matching points, and b denotes the baseline. At this point, the (X, Y, Z) coordinate of a point can be obtained.

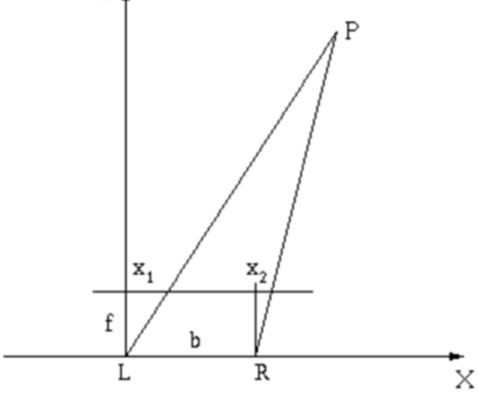


Figure 6.1.1.1 Geometry of Triangulation theory

There is one specific rule between distance and disparity, as the inner connection decides the representation from disparity to z coordinate. The relationship between depth and disparity is nonlinear as given in Figure 6.1.1.2 [7]. The statistical diagram shows that distance trend goes the opposite direction with disparity. The lower values of disparity, the farther the distances are. In other words, small disparity indicates a long distance, on the other hand, large disparity means distance is in a short range. In the consequence of this principle, the resolutions of closer objects are higher than long-distance ones. A disparity map representing the depth information contains disparities for matched points. In a grey scale disparity map, pixels with lighter colour present the object locates in a closer position in respect to the camera, conversely, the further object has darker pixel colour in the disparity map.

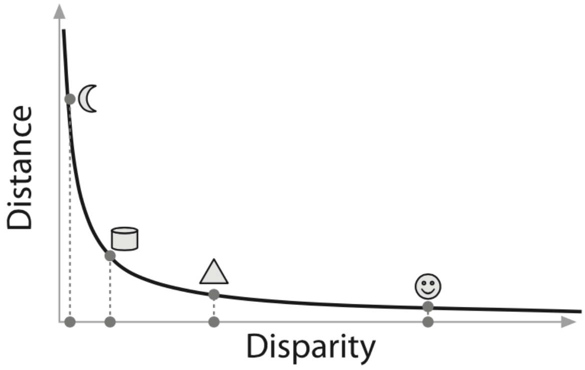


Figure 6.1.1.2 Distance relationship with disparity

6.1.2 The principle of disparity computation

Dense stereo corresponding algorithms estimate correspondence and produce disparity for every pixel in the reference image. On the basis of epipolar principle, one pair of corresponding pixels will lie on the same epiploar line in the common image plane. Moreover, on account of stereo vision produces a difference between the same point in left and right images caused by view shift, corresponding searching is carried out in a disparity range. That is to say, a term of disparity range indicates a scope where the matching pixel for a selected reference pixel can possibly locate at. In our research, the reference image refers to left image in a stereo image pair. A schematic example given by Figure 6.1.2.1 illustrates the process of dense disparity map computation. The example shows that a correspondence is found by matching candidate pixel in disparity range with its reference pixel along with their epipolar line.

In Figure 6.1.2.1, left image is selected as the reference image, where P_L^W represents one of the reference pixel with a feature window W. The feature window is extracted with pixels P_L locating in the centre and its neighbouring pixels within a given row and column sizes as expounded in Chapter 3 Section 3.4.1. P_{Rj}^W indicates potential corresponding pixel P_{Rj} with feature window in respect to P_L lies in a disparity range DR at the j^{th} column in right image.

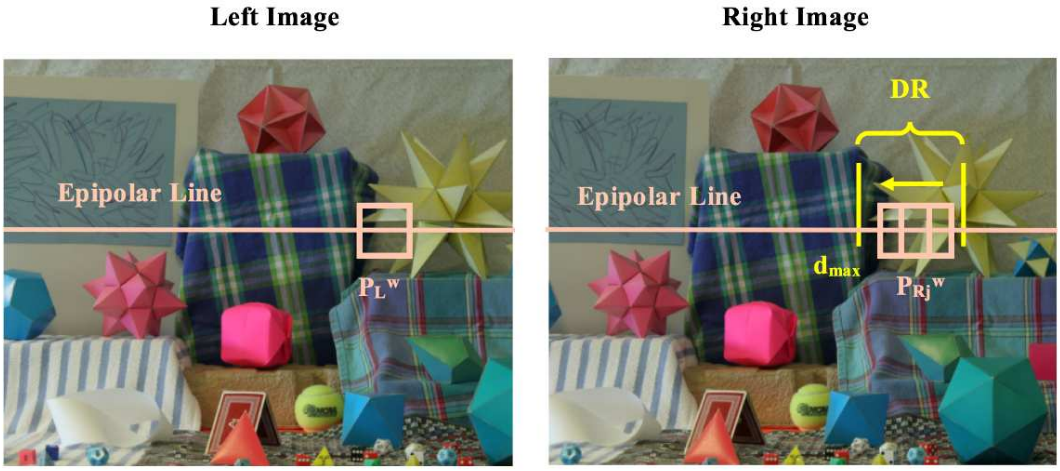


Figure 6.1.2.1 Dense disparity map computation example

All candidate pixels locate on a common epipolar line of P_L and P_{Rj} that represents the corresponding characteristics in stereo vision. The searching procedure starts from P_{Rj} that has the same coordinate as P_L in right image and then follows the direction from right to left till the pixel at the maximum disparity step d_{max}. In our methodologies, a reference pixel grouping with every candidate pixel in DR to compute matching degree which implies a pair of pixels whether corresponding with each other or not. The higher matching degree represents the higher possibility of corresponding. Thereupon, by adopting winner-take-all method (WTA, introduced in Chapter 2), the disparity is determined with candidate pixel that produces the maximum matching degree (in other terms, the minimum aggregated cost) which as given by Equation 6.5, and then this pixel becomes the corresponding pixel. Where $d \in (0 \ldots d_{max}\text{-}1)$. Accordingly, the disparity at P_L equals to the difference value of column numbers ($c - j$, vertical coordinates difference) between P_L and matched P_{Rj} according to Equation 6.2, where $j \in (c, c\text{-}1, \ldots\ldots, c\text{-}d_{max}\text{-}1)$, c is the column number of P_L.

$$D(P_L) = \underset{d}{\operatorname{argmax}} M(P_L, P_R|d) \tag{6.5}$$

The computational procedure of dense disparity map normally occurs time-consuming issue owing to stereo correspondence analysis at the pixel level within a DR that leading to a quantity of data processing matter, therefore, there is need of an approach to accelerate the speed of computation. Next section will present the optimization of dense disparity map generation.

6.2 Computational speed optimization for disparities based on stereo correspondence

Computational speed has been an issue in computer vision as there are always a huge amount of data for analyzing during the entire process. In particular, dense stereo correspondence scheme estimates matching level pixel-by-pixel so as to compute pixel position disparities often has to deal with a vast of input units, in this case, an optimization procedure comes in handy to help an implementation velocity.

6.2.1 Overview of the time-consuming issue in computer vision

Algorithms and hardware work in cooperation with each other effect the processing time of a system, on account of this matter, the processing actions of computer vision requires an advanced capability of hardware resources to support for achieving a smooth performance. However, currently, it appears a hard-pressed situation that the development of hardware has been difficult to keep up the same level along with the growth of visual technologies. The cooperation between algorithms and hardware resources is an essential issue for exploring algorithms of stereo matching especially when comes to estimate disparity map on the basis of dense approaches which implementing on pixel level. Methods relate to pixel-by-pixel analysis that increasing computational quantity are significantly in need of aid on such solutions. For this reason, real-time stereo algorithms have been studied for a long period from different aspects in such a state of affairs.

There are various approaches have been discovered, some can produce results in semi-real-time processing, and some can achieve real-time performance. In order to reach to real-time level, some investigations of algorithms adopt the way of cooperation with specific hardware, in other words, that is to seek a compatible way between the mathematical system and a shortage of hardware resources. The common hardware resources for implementing algorithms improvement are GPU (graphics processing unit), FPGA (a field programming gate array), DSP (a digital signal processor) and ASIC (an application specific integrated circuit) [151]. However, no matter which hardware is in use, the efficiency of algorithms still plays a qualitative role in cooperating with hardware resources. Apart from the design of algorithms, by considering from the root level that communicating with hardware, how to utilise and devise the pattern of programming effectively becomes an important factor, as programming is the straightforward representation for algorithms to interact with hardware.

Normally, one way is to efficiently code in respect to free processor memory as suggested by [152] that as well as provides a very handful explanation for such area with Matlab programming. [153] and [154] introduce a technique called Look-up Table for improving the speed of hardware processing in the form of creating connections between programming patterns by compiling reference tables. These approaches as fundamental solutions make a great effort in aid of accelerating the duration of algorithms progressing at the root phase. We adopted such strategy to design approach so as to achieve speed optimization.

6.2.2 Designed speed optimization approach

In the research of [90], the procedure of disparity map computation performs matching degree estimation with a neural network in the way of calling the neural network repeatedly for every pair of stereo pixels, in other words, the neural network is only given one pair to produce one matching degree at every time. The method of [90] requires extremely long processing time to complete generation of one disparity map from two stereo images, since pairs for correspondence matching normally involves a quantity of number on the total amount between an image pair. Each reference pixel is paired with a range of candidate pixels based on the maximum disparity range from the other image which leads to a large number of pixel pairs, in this case, applying neural network for every pair every time appears inefficiently. In this case, we designed an optimization approach to efficiently utilize neural networks to generate matching degrees for the purpose of accelerating the computational speed.

Methodologies of designed optimization method:

According to the characteristics of ANN, the core concept refers to input the whole sets of pairs extracted from left and right images to networks at one time rather than single estimation for one pair. In order to achieve this task, we adopted an index operation method to optimize the speed. There are two main steps in relation to stages of disparity map computation: feature extraction from all pairs of reference pixels and its potential corresponding pixels, retrieving disparities for each reference pixel from computed matching degree vector.

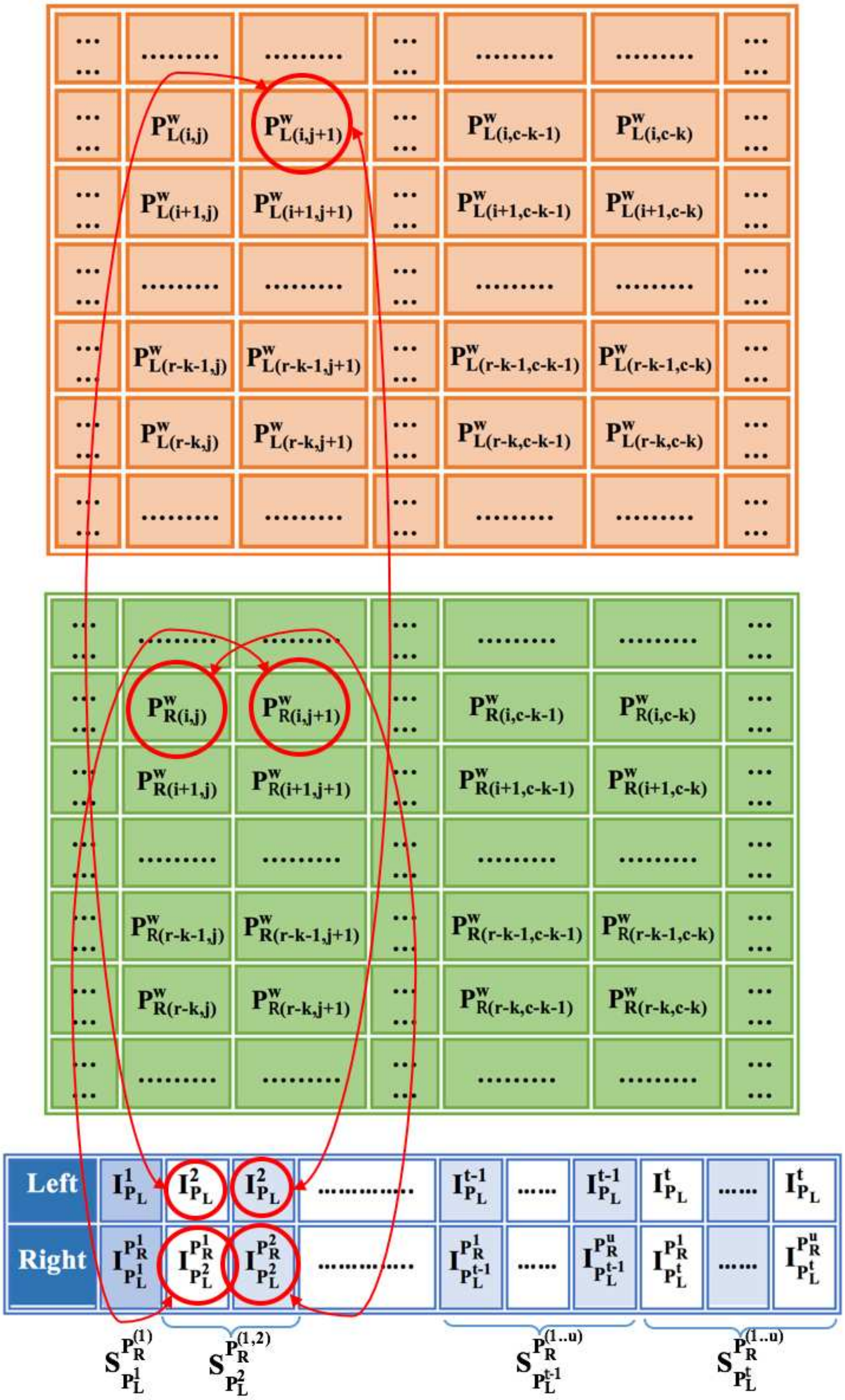


Figure 6.2.2.1 Speed optimization method for entire feature extraction

Designed optimization methodology performing feature extraction aims to solve difference feature vector construction for every pair once and for all, where are then used to produce matching degree with networks built in our research. In this optimization step, two tables are created for the purpose of storing feature values and indices of all stereo pairs. Theory of obtaining features for all pairs principally is presented in Figure 6.2.2.1. In Figure 6.2.2.1, the first table stores feature windows for every pixel from the left image and the second table contains right pixels with corresponding feature windows form the right image, where each window in both tables locates at the same coordinates as the row and column indices of centre pixels of these windows in left and right images.

P_L^w and P_R^w represents feature windows for pixel P_L and P_R in left and right images. The indicators of i and j denote the number of row and column, moreover, r and c are the total number of rows and columns which also refers to the size of stereo images - (r × c). The distance from boundary to the initial pixels that can have required window with a certain size is indicated by k which can be computed as Equation 6.6. The range of i and j are defined as given: $i \in (1+k, 2+k, 3+k, \ldots\ldots, r-k-1, r-k)$ and $j \in (1+k, 2+k, 3+k, \ldots\ldots, c-k-1, c-k)$.

$$k = \frac{N-1}{2} \tag{6.6}$$

Where N is the size of the required feature window.

Each reference pixel has a set of candidate pixels within maximum disparity range in the right image, which constitutes a set of pixel pairs in relation to a reference pixel, where such set is denoted as $S_{P_L^t}^{P_R^{(1..u)}}$. All the indices of paired pixels are kept in the third table in sequence in terms of reference pixels with potential matching pixels from the first to the last one. In this case, the order is determined by two directions: reference pixel direction and direction of candidate pixels for every reference pixel.

The direction for reference pixels follows the path of rows, and a candidate pixel route directs from right to left over the maximum disparity range starting from the same column coordinate as the reference pixel. As the example shown in Figure 6.2.2.1, the fist stored indices set in respect of the first reference pixel and its potential corresponding pixels are $P_{L(k,k)}$ and $P_{R(k,k)}$, and the following set consist of $P_{L(k,k+1)}$ and ($P_{R(k,k+1)}, P_{R(k,k)}$). $I_{P_L^t}^{P_R^u}$ indicates the index of one candidate pixel P_R^u regarding its reference pixel P_L^t that lies on the location $I_{P_L}^t$, where u is the number of potential pixels and t is the number of reference pixels with the scope: u ∈ (1, 2,……, d_{max}) and t ∈ (1, 2, ……, T), and T is the total amount of pixels that can have demanded feature windows as defined by Equation 6.7. According to the structure of feature tables given in Figure 6.2.2.1, tables storing feature windows for intensity, magnitude and orientation attributes are generated for the difference attribute computation respectively.

$$T = r\,c - (2\,k\,r - 4\,k^2 + 2\,k\,c\,) \qquad (6.7)$$

Once the method establishes feature window tables, the differential feature windows can be retrieved by calling the indices of all pixel pairs stored in the indices table to locate corresponding features windows back into the features tables as the red direction lines shown in Figure 6.2.2.1, and then computing the subtractions between left and right pixels of all pairs at o. At this point, after formatting the difference feature windows into vectors, the feature dataset for the stereo images can be obtained.

Following the construction of the feature dataset for left and right images, the next step involves matching degree estimation. With the optimization approach, at this stage, a network inputs a dataset containing all feature vectors for every pair, and then calculates matching degree for every pair of stereo pixels. The output of the network becomes one vector stores all matching degrees in a certain order according to the same sequence as presented by the indices table in Figure 6.2.2.1. The disparity at each reference pixel location is derived from every set of matching degrees in relation to the reference pixel and its candidate pixels with the WTA method.

The detailed process of disparity extraction from the whole matching degree vector for all pairs falls into steps as follows:

1) Finding the index of the maximum matching degree from each interval in respect to every set of reference and candidate pixels stored in the whole matching degree vector. supposing the start index of an interval is S_v and the end index is E_v, where v denotes the number of intervals in the entire vector, thus the maximum value of v equals to the

total amount of reference pixels used for a stereo image pair, that is to say, v ∈ (1, 2, ……, T), the calculation of S_v and E_v refers to the process below:

- If a reference pixel can not have the same number of candidate pixels as the maximum disparity range (MDR), then the values for S_v can be found as below, where v starts from two to MDR-1 with $S_1 = 1$ and $E_1 = 1$:

$$S_v = S_{v-1} + step_o \tag{6.8}$$

$$E_v = S_v + step_o \tag{6.9}$$

Where o ∈ (1, 2, 3, ……, MDR-2).

- If a reference pixel can have the number of pairs regarding potential pixels which is the same as MDR, then v should start from MDR to T, then:

$$S_{v=MDR} = \frac{MDR^2 - MDR}{2} + 1 \tag{6.10}$$

Once $S_{v=MDR}$ is obtained, its corresponding end index and the start and end indices of rest sets can be calculated as following equations:

$$S_v = S_{v-1} + MDR \tag{6.11}$$

$$E_v = S_v + MDR - 1 \tag{6.12}$$

According to the S_v and E_v, the coordinates of paired pixels in an interval can be retrieved with the indices table (given in Figure 6.2.2.1) based on the number of interval and the order of candidate pixels in this interval.

2) The second step computes the disparity for each reference pixel. Firstly, maximum matching degrees in all intervals that can be obtained as explicated in the previous step are found, then candidate pixels of a reference pixel in right image which can produce

the largest matching degrees are considered as the corresponding pixels for the reference pixel. Secondly, disparities are calculated by subtracting the column indices of reference pixels and matched candidate pixels.

3) At last, generated disparities are located in a map row by row so as to form an initial/raw disparity map. The procedure performs disparity assignment in accordance with the coordinates of their reference pixels that can be found with the method in step one in order to represent the depth clue for every reference pixel in the left image. A raw disparity map is then refined with post-processing approach in order to improve the map quality.

Experimental results evaluation:

The designed optimization approach can significantly improve the computational speed of disparity map generation in contrast to the method introduced by [90]. Table 6.2.2.1 presents the experimental comparisons between single matching degree computation method (SMD) and our approach that using network computes all matching degrees for the whole amount of pairs at one time (WMD).

	Elapsed Time of Disparity Map Computation
SMD	13.9480 *days*
WMD	54.9668 *seconds*

Table 6.2.2.1 Comparison of computational speed between SMD and WMD

In the experiment for Table 6.2.2.1, the adopted stereo image was Mountain set introduced in Chapter 3 Section 3.2. The number of pairs was computed with the multiplication of the value of maximum disparity range and number of pixels that can produce feature windows with a certain size and can be calculated by Equation 6.5. The specifications of this experiment are listed below in detail:

- Image size: 408×589
- Network: SNN with Scaled Conjugate Gradient algorithm
- Feature window size: 7×7

- Max disparity range: 20
- Numbers of Pairs: 4687320
- Hardware:
  Processor - 2.8 GHz Intel Core i7; Memory - 16 GB 1600 MHz DDR2

As the results are shown by Table 6.2.2.1, WMD approach produced disparity map within a short processing time, moreover, WMD outperformed SMD, while SMD took a very long period to accomplish the gain of the disparity map. This implies that the idea of network analyzing the entire pair dataset treads on the right path, furthermore, our approach can make a conducive impact on accelerating the elapsed time of the entire process for disparity map generation, in particular, the formation of pixel level based dense disparity map can obtain advantage from WMD method in respect to relative speedy calculation.

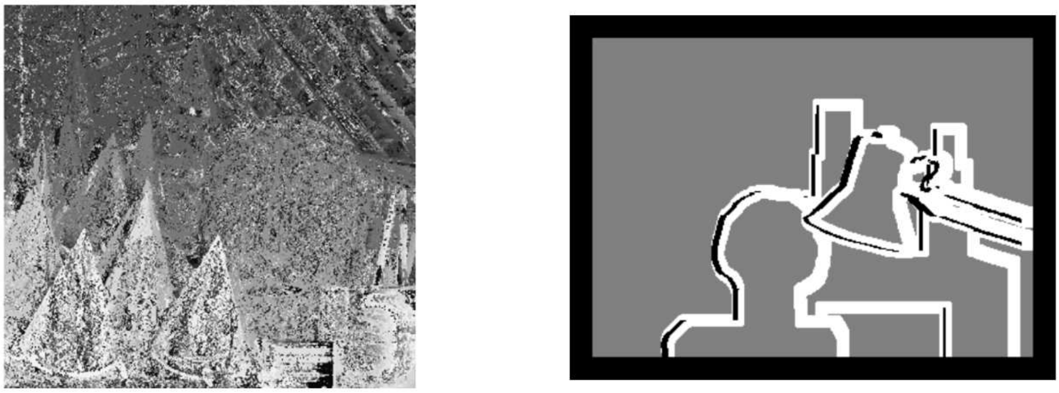
6.3 Post-processing for disparity map

One procedure commonly follows after generation of a disparity map, which refers to refine the disparity map so as to further improve the representation provided by the map. In other terms, such extension plays a favourable role in clarifying the produced disparity map at the last stage of the entire estimation flow.

6.3.1 Overview for disparity map refinement

As explicated in previous sections, a map containing differences between corresponding points generated by stereo correspondence algorithms is designated by the name of disparity map. Depth value can then be computed based on this formation according to triangulation formula. Generally, a standard disparity map is a grayscale plot. Such map implies the distance of each pixel in respect to the real world. The smaller the disparity is, the farther the depth/distance is.

A map that is directly output from stereo matching algorithms generally comes in the raw state. In this case, the produced initial map requires refinement techniques to enhance the definition on its plot so as to improve the accuracy of computed disparities. This step is known as post-processing for the purpose of quality enhancement. Common approaches involve sub-pixel improvement, noises removal, occlusion filling and discontinue enhancement [57].



(a) Disparity map with noises (b) Occlusion and discontinues

Figure 6.3.1.1 Noises, Occlusion and discontinues regions

Sub-pixel refinement that is widely applied normally refers to curve fitting with neighbour cost [57], [155]. Noises on the map as the schematic of Figure 6.3.1.1.(a) [156] can be reduced by filter-based method such like median filter[57], [157] that clear messy elements on images. An occlusion indicates a missing region between stereo images as the black area (excluding the black boundary) shown in Figure 6.3.1.1.(b) [57] caused by the variance of perspective angle. For predicting occlusion region, the most applied straightforward approach is cross-check/left-right check, which checks the uniformity of pixels between left and right disparity maps [158]. After occlusion detected, the following step performs a filling method like area disparity estimation [159]. Discontinue regions appear where disparities disjoint involves the term of edge enhancement [160], which as the white area revealed in Figure 6.3.1.1. (b), such areas normally relate to the boundary of objects.

6.3.2 Adopted post processing approach

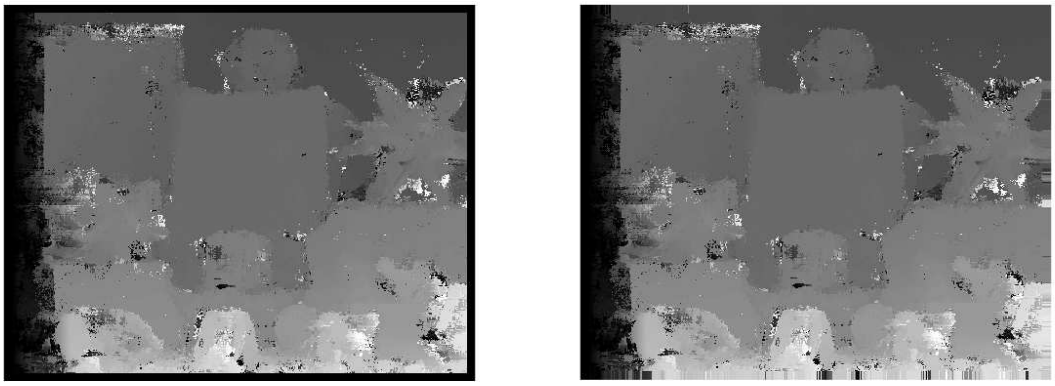
Our research aims to perform a compact post-processing approach for the purpose of implementation efficiency of systems, in the consideration of more algorithms require more elapsed time to complete one update of an initial disparity map. According to objective of research is to implement estimation in general level, moreover, on account of a raw disparity maps sometimes occur low resolution condition with noises on the appearing plot, in this case, the produced initial map requires refinement techniques to enhance the definition on its plot so as to improve the accuracy of computed disparities. The starting point of the refinement scheme determined by our research involves cleaning up jumbled factors that occur in maps presenting differences of the vertical axis between corresponding pixels generated from stereo matching algorithms so as to clarify the overall appearance of disparity maps.

On the basis of such motivation, the adopted post-processing scheme was found on the purpose of noise removal from raw disparity map in our study. The entire refinement procedure can be divided into three steps as follows:

1) Map size recovery:

   The stereo correspondence algorithm performs the corresponding estimation by measuring the level of matching between a pair of pixels with differential feature window obtained from two feature windows using these two pixels as the centre. For this reason, pixels around the boundary of images can not have feature window so that are excluded from computation, consequently, the output disparity map thus presents black colour on the border for those missing pixels as shown in Figure 6.3.2.1. (a).

   This step recovers the black boundary in the way of replicating border values next to the black border, and locates these values in the black region, which an example outcome is presented by Figure 6.3.2.1. (b).



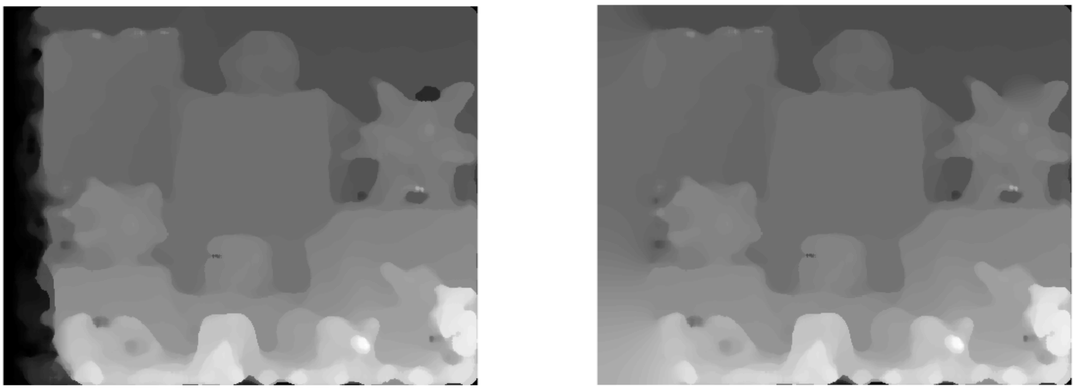
(a) Output disparity map (b) Border recovered disparity map

Figure 6.3.2.1 Border recovery for disparity map

2) General noise clean:

   In the field of noise removal, the most widely used algorithm in image processing refers to Median Filter that possesses the advantage of traits on keeping edge and cleaning up impulse noise [161]. The theory of Median Filter algorithm involves discovering a median value within a given mask that normally consists of odd numbers. Firstly, a mask extracts a set of values and sort these values from the smallest one to the largest one. Secondly, the median value from the sorted set with values in descending order is selected to replace the value of the center pixel in the mask. The mask can be formed with a variety of shapes such as square, cross, circular and one dimension. Applying with a median filter, the representation of the disparity map can be improved from the black-border retrieved map as presented by Figure 6.3.2.2. (a).

3) Further regions padding:
   As we can see, after applying with a median filter, there still are some areas mismatching in the disparity map, which are represented by very dark noisy areas. Therefore, this stage further improves the quality of map by filling these regions according to the values of pixels locating on the outer boundary of these areas, where a given mask points out these regions in the map that should be refined [162]. Figure 6.3.2.2(b) gives the result of a disparity map with region filling in accordance with Figure 6.3.2.2.(a).



(a) Disparity map with median filter (b) Region filled disparity map

Figure 6.3.2.2 Disparity map applied with median filter and region padding

After the implementation of three steps, the final disparity map with an improved level of clarity can be obtained. Experiments were carried out to evaluate the effectiveness of implied post-processing approach, and results are illustrated in Figure 6.3.2.3. The computed disparity maps in respect to performances of Figure 6.3.2.3 was derived from Moebius image set. The evaluation was based on bad pixel percentage (details explicated in Section 6.4).

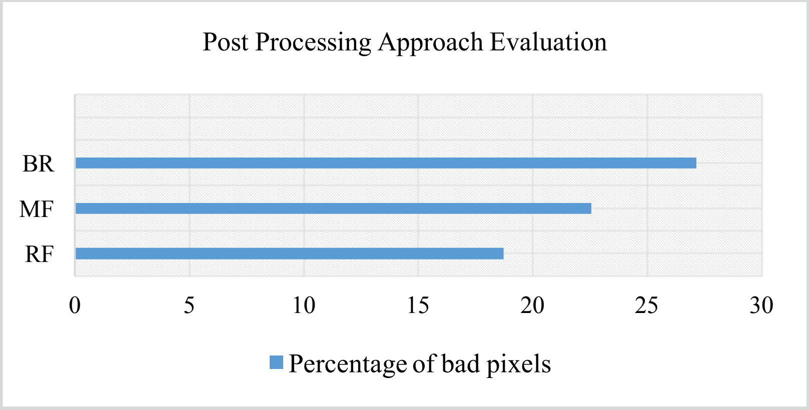


Figure 6.3.2.3 Designed post-processing approach evaluation

In Figure 6.3.2.3, BR, MF and RF denote the post-processing step with border recovery, overall noise reduction with a median filter, and final stage performing region interpolation. The shorter bar represents a lower percentage of incorrect disparities which were caused by incorrectly matched pixel pairs. As shown in Figure 6.3.2.3, the percentage of bad pixels decreased along with the enhancement procedure step-by-step, which implies the usability for disparity map refinement. The results show that this post-processing approach cannot only clarify the appearance and advance the definition of the raw disparity map, but also improve the accuracy of the disparity map.

6.4 Evaluation with disparity map

The final usage of output from stereo correspondence estimation involves a calculation stage for disparity recovery to form a map containing depth cues. On account of this matter, disparity map provides direct estimation for the performance of stereo matching algorithms. In dense disparity map, each pixel value representing the disparity of object observed in stereo images at this pixel location, whereas the value is derived from matched pixel pair in respect to the current location in disparity map, accordingly, the effectiveness of stereo matching procedure affects the qualitative properties straight away. This section will evaluate the systems integrating three networks as explicated in Chapter 4 and 5 with produced disparity maps.

6.4.1 Comparison and evaluation with state-of-the-art approaches

For the purpose of implementing quantitative evaluation with state-of-the-art, we adopted the Middlebury benchmark to perform the task. Three systems were used to estimate corresponding pixel (in right image) for every reference pixel (in left image) to generate dense disparity maps, in accordance with this characteristics, evaluations then relate to measure the quantity of correct disparity values in disparity maps produced by matched pairs. The evaluation method in respect to such measurement normally refers to one quality metrics which is known as percentage of bad matching pixels. The computational process for this performance estimation can be formulated as Equation 6.13.

$$\mathrm{PBM} = \frac{1}{K}\sum_{k=1}^{K}\left(\left|(d^{o}_{(x_k,y_k)} - d^{t}_{(x_k,y_k)}\right| > \delta_d \right) \tag{6.13}$$

Where k represents the number of pixels in total, $d^{o}_{(x_k,y_k)}$ and $d^{t}_{(x_k,y_k)}$ denotes the output and target disparity values, and (x, y) indicate locations of generated values in the disparity map. δ_d is the threshold for determining whether an accurate disparity occurring, where $\delta_d = 1$ in our evaluations. According to the equation, the fundamental theory performs measurement on the difference between computed disparity and correct disparity of every pixel. If a difference value between $d^{o}_{(x_k,y_k)}$ and $d^{t}_{(x_k,y_k)}$ is bigger than δ_d, one error is count in the total number of pixels with wrong disparities, once the number of all bad pixels is obtained, PBM calculates the percentage of incorrect disparities over all pixels.

The main pipeline of our methodologies stars from feature extraction, matching degree computation, and disparity map computation. In accordance with the standard process of stereo corresponding algorithms, the first step corresponds with the stage of matching cost computation, the second step refers to the aggregation of cost, the third step performs the function of disparity optimization, accordingly, such process builds up the pipeline of the local stereo corresponding algorithm. On the basis of this traits, we carried out comparisons with state-of-the-art methods that performing stereo approaches in relation to local algorithms.

The algorithms that were adopted in our comparison were from the Middlebury Stereo Evaluation Version 2, moreover, stereo image sets for our evaluation involved in the benchmark contained Teddy and Cones. Table 6.4.1.1 lists the comparison of PBM (that was computed with the mask of the all-scheme according to the benchmark, which measured the PBM over the entire disparity map while excluding few unknown regions) between three networks built in our thesis (SNN, d-Multiple NNs, and b-CNN) and sate of the art approaches.

	Teddy	Cones	Average
CCRADAR [163]	10.6	7.37	8.99
LM3C [164]	10.9	7.59	9.25
LAMC-DSM [165]	10.4	8.31	9.36
HistoAggr2 [166]	11.3	7.78	9.54
DTAggr-P [167]	11.5	7.82	9.66
HCFilter [168]	11.5	8.07	9.79
MSWLinRegr [169]	11.0	8.76	9.88
ConfSuppWin [170]	11.4	8.60	10.00
CostFilter [171]	11.8	8.24	10.02
TF_ASW [172]	11.8	8.32	10.06
GradAdaptWgt [173]	13.1	7.67	10.39
RealtimeHD [174]	10.7	10.1	10.40
iFBS [175]	12.8	8.73	10.77
VSW [176]	13.3	8.85	11.08
RTAdaptWgt [177]	13.3	9.34	11.32
d-Multiple NNs	13.25	10.80	12.03
b-CNN	13.81	10.54	12.18
SNN	13.84	10.87	12.36
VariableCross [59]	15.1	12.7	13.90
RINCensus [178]	17.3	16.2	16.75
SSD+MF [57]	24.8	19.8	22.30

Table 6.4.1.1 Comparisons between state-of-the-art and three networks based on percentage of bad matching pixels

We used the optimised model for SNN, d-Multiple NNs, and b-CNN obtained from previous Chapters to produce disparity maps for evaluations. The specifications of the three networks are listed in detail below:

➢ **SNN:**

- Input layer: 147 neurons
- Hidden layer: one layer, 49 neurons
- Output layer: 1 neuron
- Training function: Scaled Conjugate Gradient

➢ **d-Multiple NNs:**

- Input layer: 49 neurons
- Hidden layer: two layers, [49, 28] neurons
- Output layer: 1 neuron
- Training function: Scaled Conjugate Gradient

➢ **b-CNN:**

- Input image window size: 17×17
- Filter size: 9×9
- Filter number: 32
- Fully connected layers:

  4 layers with 20 neurons at each hidden layer, one layer with two neuron
- Mini batch sizes: 64
- Learning rate: 0.01
- Drop factor of learning rate: 0.1
- Training function: Stochastic Gradient Descent

In the Middlebury Stereo Evaluation Version 2, the best performing network in the Teddy dataset is ranked at place 100 out of the 166 algorithms submitting results to the repository over seventeen years (1999 to 2015). Correspondingly for the Cones dataset the best performing of our networks is ranked 108th out of 166. Further evaluation with the common local related schemes is presented by Table 6.4.1.1. Table 6.4.1.1 gives the error rates of disparity maps for Teddy and Cones according to PBM in 100% unit with given ground truths from the benchmark, which represent the overall error rates of computed disparity maps. The column of average measurement indicates the mean PBM of output disparity maps for both Teddy and Cones together in respect to each approach in the table. The disparity maps produced by the three networks for Teddy and Cones images are shown in Table 6.4.2.1.

In general, all state-of-the-art approaches produced better performance for Cones than Teddy images owing to Teddy image has more areas with less textured scenarios which is the typical issue of the local method as mentioned in Chapter 2, moreover, methods with three networks appeared the same phenomenon as others. Such appearance indicates that thee-networks based systems possess the characteristics of stereo matching algorithms on the basis of the local scheme, that is to say, they can be capable of local stereo correspondence estimation. Furthermore, in accordance with this observation, the improvement of systems integrating with three architectures of neural networks presented by our study can be investigated from the aspect referring to advance local methods.

As shown in Table 6.4.1.1, the percentage range of PBM for Teddy is from 10.4% to 24.8%, and 7.37% to 19.8% for Cones, which can be roughly rounded into 10% to 25% for Teddy estimation and 7% to 20% for Cones estimation. SNN, d-Multiple NNs and b-CNN generated PBM around the percentage of 13 to 14 for Teddy disparities, and 11% for Cones disparities, where representative rates of accuracies located in the similar range of PBM for both Teddy and Cones disparity maps, which means these three systems can achieve stable performance among a variety of methodologies.

State-of-the-art methods listed in Table 6.4.1.1 were created on the basis of common local-stereo techniques as algorithms introduced in Chapter 2 Section 2.5.2, in this case, their performances represent the characteristics of general algorithms for the corresponding estimation between stereo images. The PBM referring to average error rate over Teddy and Cones for each method that are shown by Table 6.4.1.1 can be rounded from 9% to 22%, moreover, most of state-of-the-art approaches had error rates between 9% and 11% in accordance with the PBM, and SNN, d-Multiple NNs and b-CNN produced about 12% bad matching pixels.

The comparison based upon mean PBM measurement implies that, along with the development of stereo correspondence technologies, the general performance of well-investigated methodologies (which found on the common local strategy of stereo matching) for finding corresponding pixel pairs can be considered to converge at about 10% which represents the average percentage of 9% to 11%. SNN, d-Multiple NNs and b-CNN as novel approaches in contrast to traditional stereo matching techniques have achieved about two percentages far from general PBM. All these reveals indicate that although systems with three networks have not reached to the average performance that common approaches attained, they are getting very close to the accuracy levels achieved by those conventional algorithms. Therefore, as burgeoning technologies, SNN, d-Multiple NNs and b-CNN have the ability and potential capability of effectually matching stereo pixels. In other terms, such stereo correspondence approaches that integrating with these three networks possesses great prospects on account of advantages in respect to the flexibility and intelligent functionality.

6.4.2 Comparison and evaluation between three networks

This thesis has presented three artificial neural networks (SNN, d-Multiple NNs and b-CNN) for stereo matching implementation by matching pixels with the computation of matching degrees for reference pixel with its candidate corresponding pixels to finally construct disparity map. They consist of diversified architecture designs which their own effect respectively, accordingly, different outcomes representing the specific capabilities of the three networks are then generated from these three systems.

It is very helpful to explore the characteristics of performance in respect to each individual network based stereo matching system so as to understand each network is good at which field, even more, to discover clues for system advancement in the further step from different aspects in the future by starting from the current stage of designed systems.

The quality of a disparity map involving the column differences between two corresponding pixels denotes the performance of stereo correspondence algorithms, in other words, by estimating the error rate of a computed disparity map, the efficiency of a stereo matching method can be revealed perspicuously.

On one hand, quantitative accuracies can be obtained directly in the way of counting the number of inaccurate disparities caused by falsely matched pixels as defined by Equation 6.13. On the other hand, with the benefit of visible measurement provided by the disparity map, the qualitative analysis of performance with regard to three networks can be investigated in a straight way of visualised observation. By the combination of this two strategies, we carried out effectiveness analysis for SNN, d-Multiple NNs and b-CNN based on produced disparity maps of stereo images from datasets which were introduced in Chapter 3 Section 3.5.

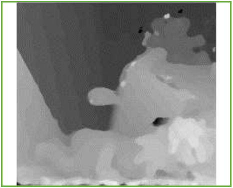
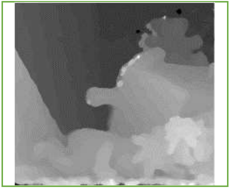
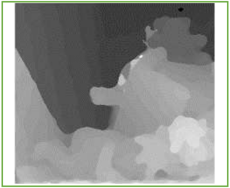
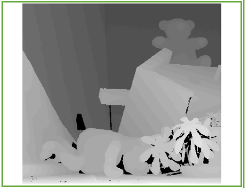
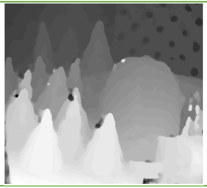
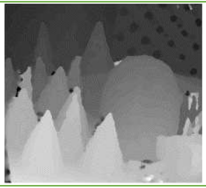
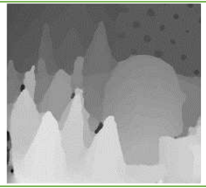
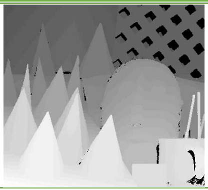
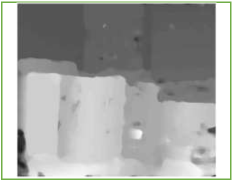
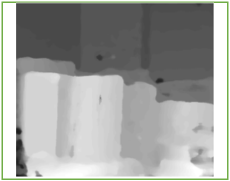
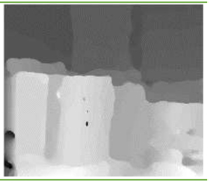
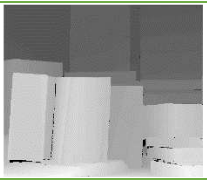
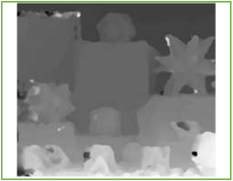
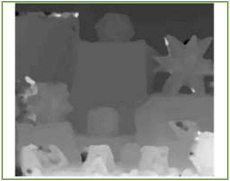
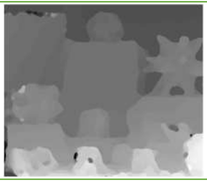
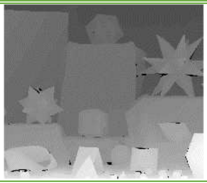
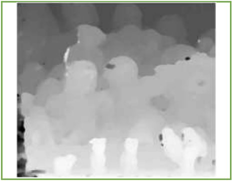
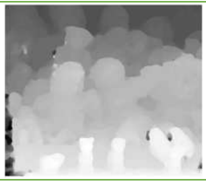
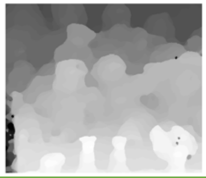
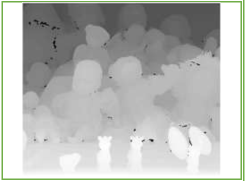
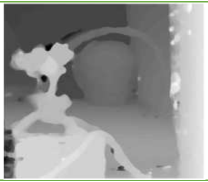
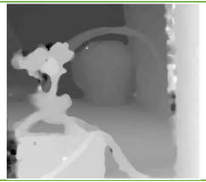
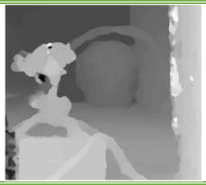
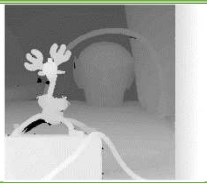
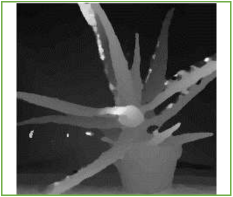
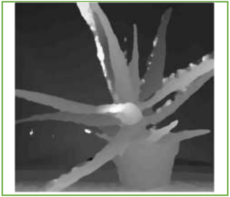
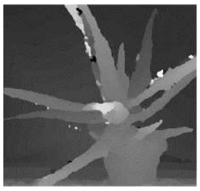
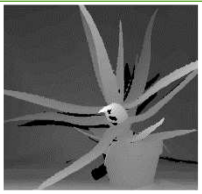
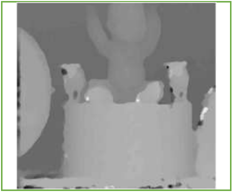
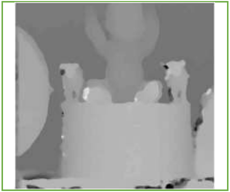
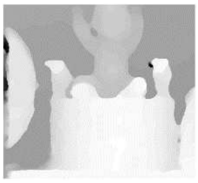
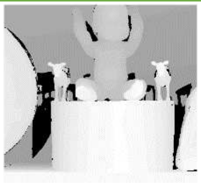
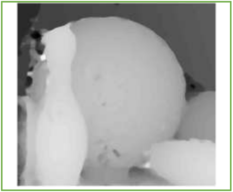
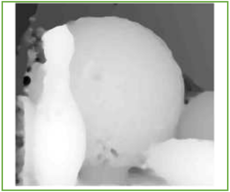
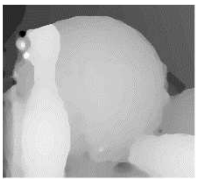
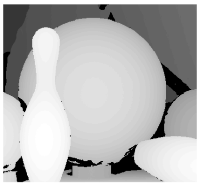
	SNN	d-Multiple NNs	b-CNN	Ground Truth
Teddy				
Cones				
Book				
Moebius				
Dolls				
Reindeer				
Aloe				
Baby 3				
Bowling 2				

Table 6.4.2.1 Disparity maps generated with three networks

Table 6.4.2.1 illustrates disparity maps computed by systems with SNN, d-Multiple NNs and b-CNN and corresponding ground truth for each image set, where the nine stereo images utilised for producing maps in Table 6.4.2.1 were Teddy, Cones, Book, Moebius, Dolls, Reindeer, Aloe, Baby 3, and Bowling 2. For the networks settings, we used the same optimal models for the three networks as listed in the previous section to generate these maps.

Occlusions over all images that appear at the left side of objects with brighter colour as shown in these plots occur less in SNN maps comparing with maps generated by d-Multiple NNs and b-CNN. Nevertheless, as the figures presented in the table, disparity maps produced with SNN appears to have more noise regions on the surface than the other two networks, especially, when objects with large flat surface such as Teddy and Book, where d-Multiple NNs lines in the second position that has less noises than SNN but more than b-CNN on such surface. This observation implies that b-CNN possesses the trait of less sensitive to large flat areas with low texture in contrast to SNN and d-Multiple NNs, conversely, the performance of SNN is very easy to be affected by such areas.

Disparity maps generated by b-CNN have the smoothest and clearest appearance for objects, however, tend to blur the outliner of objects, if the gap between objects is small, objects are merged together such like the legs of the small toy in Baby 3. Moreover, the objects with narrow or small shapes cannot be detected by b-CNN, for instance, objects in a mug contained in Cones ground truth are disappeared in the b-CNN map, where SNN maintains the most shape of these small objects. That is to say, b-CNN can perform better estimation with large shape objects, and SNN is good at small shape analysis, where d-Multiple NNs has the middle capability between them, in other words, d-Multiple NNs can balance the ability of shape detection with different sizes. By observing over these disparity maps, d-Multiple NNs keep the sharpest boundaries for objects, which can distinguish the depth changes better than the other two networks.

In the view of these observations, SNN, d-Multiple NNs and b-CNN all have their own abilities to deal with different tasks. Accordingly, with specific strengths, three networks can produce the different formation of appearance for disparity maps. Overall, in contrast to ground truths, three networks can produce disparity maps closer to the correct presentation in terms of basic scenarios in images can be presented by these maps. In order to further explore the performance between three networks, PBM was calculated for computed disparity maps listed in Table 6.4.2.1.

	SNN	d-Multiple NNs	b-CNN
Book	17.34	14.39	14.64
Moebius	14.47	14.45	14.19
Dolls	11.14	12.01	11.80
Reindeer	16.94	15.97	16.95
Aloe	13.81	14.88	14.15
Baby3	14.75	15.58	15.19
Bowling2	16.62	16.46	16.99
Average	15.01	14.82	14.84

Table 6.4.2.2 Percentage of bad pixels for disparity maps generated with three networks

Corresponding PBMs for disparity maps that were produced with 2005 and 2006 Datasets in Table 6.4.2.1 is listed in Table 6.4.2.2. In the Table 6.4.2.2, the overall error rates for all disparity maps produced by each network are denoted by average PBM. As shown by Table 6.4.2.2, moreover with the performance of Table 6.4.1.1, each network can produce outstanding performance with different images, moreover, for the same image set, the three networks generally have the difference of error rate around one to three percentages of bad matching pixels between them. For some image sets, SNN can obtain better performance, while some image sets can get good quality disparity map with d-Multiple NNs, and b-CNN can produce higher accuracy for some image sets. This phenomenon corresponds to properties that each network possesses diverse strengths as the previous analysis from the appearance of disparity maps given in Table 6.4.2.1.

The best performance in Table 6.4.2.2 all referrers to Dolls image set, which has a relatively large number of objects among adopted stereo images for disparity map computation. This means, these three networks can be available for images containing multiple objects, which is a good sign as a scene form real word normally include many objects.

From the overall aspects, according to the results of average PBM, d-Multiple NNs had the lowest average error rate among three networks, and b-CNN attained the second performance after d-Multiple NNs, where the lowest accuracy was produced by SNN, in this case, system integrating with d-Multiple NNs outperformed SNN and b-CNN based systems. Nevertheless, error rates of b-CNN and d- Multiple NNs only differed with each other in a very small value especially as shown in Table 6.4.2.2 in accordance with the mean error rates, it can reveal that b-CNN can achieve a very similar level of accuracy as d-Multiple NNs. Furthermore, in consideration of the structure design for each network, b-CNN possesses the most flexible architecture in three networks, where d-Multiple NNs consists of adjustable network layout at the second order in accordance with the flexibility, and SNN has the most difficult design to be modified. Therefore, comparing with SNN, d-Multiple NNs and b-CNN can not only reach to the higher level of accuracy, but also have the more potential capability for further advancement of performance.

6.5 Chapter summary

Firstly, this Chapter explained the fundamental theory for disparity recovery from stereo images in accordance with the triangulation principle. Secondly, the computation process of dense disparity map on the basis of stereo corresponding pixels retrieved from stereo matching algorithms was presented in detail.

Next, this Chapter expatiated a novel approach designed for optimising the computational speed of generating a disparity map which was created by this project from the innovative methodologies to experiments and results evaluation, which the result showed this approach was able to improve the computational speed effectively. A post-processing scheme designed in the pipeline which referring to noise reduction was described step-by-step following with experimental estimation, which can reduce a certain amount of noise in a raw disparity map. Moreover, evaluations and comparisons for three networks (SNN, d-Multiple NNs and b-CNN) based on computed disparity maps were presented from two aspects: with state-of-the-art approaches and among three networks, which revealed the effectiveness of SNN, d-Multiple NNs and b-CNN for stereo correspondence estimation.

Chapter 7: Conclusion and future work

So far, this thesis has presented the approach of estimating dense stereo correspondence with three types of artificial neural networks: simple neural network, multiple neural networks and convolutional neural network. This Chapter contains concluding discussions for the work presented previously in detail, and it will introduce possible areas for further study in the future.

7.1 Conclusion

Achieving accurate and reliable visual perception is a complex but necessary functionality for a number of different application areas including robotics and autonomous cars. Among methodologies developed to achieve this, stereo vision is the one most often adopted due to its similarity to the operation of human vision. In contrast to other vision systems, such as those based on monocular vision, stereo perception can represent depth information in a more precise way and easily be integrated with most general applications.

As discussed previously, stereo correspondence, an essential part of stereo perception algorithms, is accordingly a significant research field in computer vision domain. Furthermore, comparing to sparse approaches, dense stereo matching can produce disparity maps with higher resolution although sometimes the process is computationally intensive and time-consuming due to dependence on pixel-by-pixel search. To address this a range of methods are introduced with good performance in real-time processing, including local algorithms, which can be an optimal choice when efficiency is of essence.

The pipeline for our stereo matching algorithms using artificial neural network systems presented in this thesis (SNN, d-Multiple NNs, b-CNN in terms of simple neural network, multiple neural networks, and deep learning techniques) is illustrated by the flowchart in Figure 7.1.1. The process first starts with *extracting features* from rectified stereo images, where one part involves training data construction for networks learning stage, and another part implements feature preparation from stereo image pair for selecting matching pixels. After feature extraction, all different types of networks designed were first trained with training data to learn matched and unmatched classes to produce matching degrees for a pixel pair. This represents the level of correspondence between two pixels. Once the training stage is finished, input feature vectors for reference pixels in the left image, along the corresponding features for their candidate pixel pair in the right image, are used to obtain matching degrees. In the next step a raw disparity map is formed in accordance with the selected matching degrees using a speed optimization approach. For each reference pixel, the candidate pixel with the largest matching degree is selected as the corresponding pixel for the forming the correspondence pair. Accordingly, the disparity computation procedure assigns disparity at the location of reference pixels with the x-axis difference between two matched pixels. At the last stage, post-processing refines initial maps to reduce noises the final disparity map.

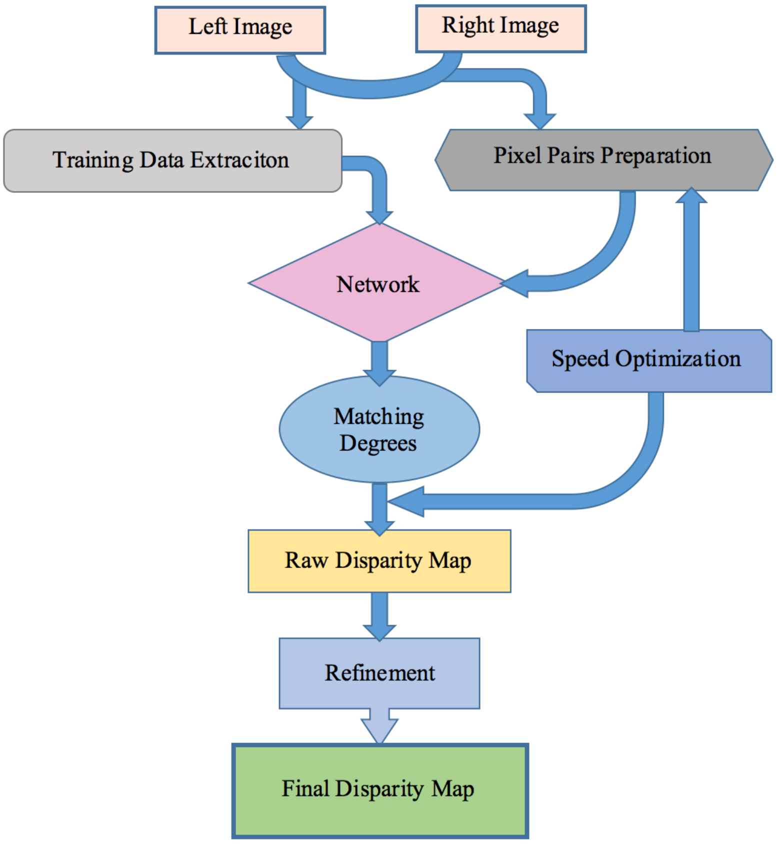


Figure 7.1.1 Flowchart of proposed algorithm pipeline

Regarding contributions made with respect to *Feature Extraction* and *Feature Types Impact*, this thesis has presented the analysis of feature engineering for systems using three different artificial neural network types to perform stereo correspondence estimation. Feature engineering is the starting point for such systems, which play a fundamental role in machine learning algorithms. From our initial results, the basic approach of feature engineering has been established. In consideration of the characteristics of stereo vision, in our methodology, training data was selected based on a designed scheme that selects instances fitting epipolar constraints with respect to matched and unmatched categories. Our experimental estimation showed that this method can outperform approaches utilising random selection. Moreover, the investigation of effect on performance with different types of features based on basic feature design found out the optimal modality of attributes referring to balanced numerical data.

As mentioned previously, there were three different types of neural network systems presented in the thesis for implementing stereo correspondence procedure. The complexity of three networks increased from neural network with simple design (SNN) to combination structure consisted of multiple networks (d-Multiple NNs), and the most complex one that adopted deep learning theory (b-CNN), where d-Multiple NNs was created on the basis of SNN. These investigations led to optimal architecture definitions for each of the three types of NNs and form the *Network Structure Design* contribution of this thesis.

Following with the optimisation of network structure, further contributions of this work focused on *Network Layers and Parameter and Model Optimizations* (Section 1.3). The function of these three networks referred to compute matching level between given pixel pairs extracted from stereo images. For the purpose of maximising the performance of networks, model optimizations were carried out to explore the relationship between parameter settings which involved learning rate related aspect, training functions, association of input patch size and filter size, filter number, mini batch size, moreover the constructions for hidden/fully connected layers. The experiments and evaluations revealed capability of three networks learning stereo properties and possible strategies for discovering the optimal combinatory of network layers and various parameters for advancing the performance in respect to each network based system as high as possible.

A contribution on *Speed Improvement for Disparity Map Computation* was made in relation to the design of an optimisation approach for accelerating the computation of disparity maps, which can shorten the computational period from days to minutes by improving feature extraction and disparity estimation processing. Refinement of raw disparity maps was also achieved by noise decrease resulting in that the final map can have better quality than the initial one (this was referred previously as the *Refinement of Raw Disparity Map* contribution of this thesis).

Evaluations with disparity maps were carried out from two aspects: comparison with state-of-the-art methods, and between the three networks, to evaluate the performance quantitatively as well as qualitatively. Details can be found in the *Comparisons* contribution in Section 1.3. In summarising the outcomes of these investigations, it was shown that all three network systems are capable of effectively estimating dense stereo correspondence. Moreover, d-Multiple NNs and b-CNN have very similar accuracy with each other, and they both outperform SNN. Finally, it was illustrated that they have greater potential for advanced performance due to their more flexible architectures and scalability.

7.2 Future work

Dense stereo correspondence estimation with neural network systems presented in this thesis have shown long-term potential space of development with a great possible prospect in the future along with carrying out further investigations. There are some future works can be implemented to advance the level of performance, in particular for d-Multiple NNs and b-CNN based correspondence matching systems.

In our research, the primary goal has focused on the implementation of matching corresponding pixels for producing disparity map in general phase. In consideration of this aspect, one exploration for both d-Multiple NNs and b-CNN systems can be referred to consider improvement of accuracy in respect to regions with occlusions and discontinues cooperating with complementary refinement, by reason that the real scenario contains intricate contents interacting with each other, which leads to such areas normally appear to occupy a certain percentage of a scene from the real world.

The correspondence search was performed on the pixel level presented in our thesis, however, in a practical circumstance, objects in an image sometimes consist of partial pixels instead of full pixels, therefore, another future work can involve matching procedure with sub-pixel grade based estimation so as to improve the resolution of disparity map.

Taking account of the characteristic that ANN possesses the dynamic and flexible ability that networks can be modified can constructed with different forms of architectures including diversified parameter settings, moreover, d-Multiple NNs and b-CNN particular were designed with flexible models, so that further improvement on the model designs for d-Multiple NNs and b-CNN can be investigated so as to further improve the performance. On the basis of network properties, possible works can involve different combinations among the diversity of layers following with relationships optimization with a variety of parameters

We have used datasets from Middlebury benchmark in our studies, where the images are mainly made up of different objects such as books, teddy bears, aloes, and so on. On account of the capability in relation to generalization analysis for real circumstance in the world, the designed systems integrated with d-Multiple NNs and b-CNN can be trained with more types of datasets, for example, one well-known dataset called the KITTI dataset [179] providing images captured with road environments.

By applying with such possible explorations, the designed systems can be further improved to the next level in the future.

Appendix A:

List of acronyms for algorithms taken from the Middlebury benchmark [96]

CCRADAR	Combined Cost Remaining Artifacts Detection and Refinement	[163]
LM3C	Local Method Three Census	[164]
LAMC-DSM	Local Adaptive Matching Cost - Dense Stereo Matching	[165]
HistoAggr2	Histogram Aggregation Two	[166]
DTAggr-P	Domain Transformation Aggregation - Pixel	[167]
HCFilter	Hierarchical Clustering Filtering	[168]
MSWLinRegr	Multiscale Weber Linear Regression	[169]
ConfSuppWin	Confidence Support Window	[170]
CostFilter	Cost Filtering	[171]
TF_ASW	Trilateral Filter _ Adaptive Support Window	[172]
GradAdaptWgt	Gradient Adaptive Weight	[173]
RealtimeHD	Real-time High Decision	[174]
iFBS	Iterative Fast Block Support	[175]
VSW	Virtual Support Window	[176]
RTAdaptWgt	Real-Time Adaptive Weight	[177]
IterAdaptWgt	Iterative Adaptive Weight	[180]
VariableCross	Variable Cross	[59]
RINCensus	Refined Intensity Neighborhood Census	[178]
SSD+MF	Sum-of-Squared-Differences + Min-Filter	[57]

References:

[1] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2011.

[2] Mercedes-Benz. (2013). *Mercedes-Benz, S-Class S 500 INTELLIGENT DRIVE - Networked sensor systems of the S 500 INTELLIGENT DRIVE research vehicle*. Available: http://media.daimler.com/marsMediaSite

[3] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, 2006, pp. 1161-1168.

[4] Boredom. (2017). *Boredom ProjectsL: Robot Navigation using Stereo Vision*. Available: https://boredomprojects.net/index.php/projects/robot-navigation-using-stereo-vision

[5] D. Kumari and K. Kaur, "A survey on stereo matching techniques for 3D vision in image processing," *Int. J. Eng. Manuf,* vol. 4, pp. 40-49, 2016.

[6] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*, Second ed. Prentice Hall Professional Technical Reference, 2012.

[7] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[8] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision Second Edition," *Cambridge University Press,* 2004.

[9] NASA. (2017). *Rover*. Available: https://www.nasa.gov/audience/forstudents/k-4/dictionary/Rover.html

[10] A. Bhatti, *Current advancements in stereo vision*. InTech, 2012.

[11] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, vol. 1, pp. I-I: IEEE.

[12] R. Furukawa, R. Sagawa, and H. Kawasaki, "Depth estimation using structured light flow--analysis of projected pattern flow on an object's surface," *arXiv preprint arXiv:1710.00513,* 2017.

[13] A. Wittmann, A. Al-Nuaimi, E. G. Steinbach, and G. Schroth, "Enhanced Depth Estimation using a Combination of Structured Light Sensing and Stereo Reconstruction," in *VISIGRAPP (3: VISAPP)*, 2016, pp. 512-523.

[14] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-quality scanning using time-of-flight depth superresolution," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, 2008, pp. 1-7: IEEE.

[15] M. Balki, Y. Lee, S. Halpern, and J. C. Carvalho, "Ultrasound imaging of the lumbar spine in the transverse plane: the correlation between estimated and actual depth to the epidural space in obese parturients," *Anesthesia & Analgesia,* vol. 108, no. 6, pp. 1876-1881, 2009.

[16] Q. Li, M. Biswas, M. R. Pickering, and M. R. Frater, "Accurate depth estimation using structured light and passive stereo disparity estimation," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 969-972: IEEE.

[17] O. Kaller, L. Bolecek, L. Polak, and T. Kratochvil, "Depth Map Improvement by Combining Passive and Active Scanning Methods," *Radioengineering,* vol. 25, no. 3, p. 537, 2016.

[18] J. Flores-Delgado, L. Martínez-Santos, R. Lozano, I. Gonzalez-Hernandez, and D. Mercado, "Embedded control using monocular vision: Face tracking," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, 2017, pp. 1285-1291: IEEE.

[19] M. N. A. Bakar and A. R. M. Saad, "A monocular vision-based specific person detection system for mobile robot applications," *Procedia Engineering,* vol. 41, pp. 22-31, 2012.

[20] C. Villanueva-Escudero, J. Villegas-Cortez, A. Zúñiga-López, and C. Avilés-Cruz, "Monocular visual odometry based navigation for a differential mobile robot with android OS," in *Mexican International Conference on Artificial Intelligence*, 2014, pp. 281-292: Springer.

[21] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 593-600: ACM.

[22] K. Cherry. (2018). *What Are Monocular Cues?* Available: https://www.verywellmind.com/what-are-monocular-cues-2795829

[23] Y. Noori. (2015). *Sensation and perception bba lect 4*. Available: https://www.slideshare.net/YahyaNoori/sensation-and-perception-bba-lect-4

[24] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1-8: IEEE.

[25] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *International journal of computer vision,* vol. 76, no. 1, pp. 53-69, 2008.

[26] Y. Salih, A. S. Malik, and Z. May, "Depth estimation using monocular cues from single image," in *National Postgraduate Conference (NPC), 2011*, 2011, pp. 1-4: IEEE.

[27] T. Nagai, T. Naruse, M. Ikehara, and A. Kurematsu, "Hmm-based surface reconstruction from single images," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, vol. 2, pp. II-II: IEEE.

[28] T. Nagai, M. Ikehara, and A. Kurematsu, "HMM-based surface reconstruction from single images," *Systems and Computers in Japan,* vol. 38, no. 11, pp. 80-89, 2007.

[29] Y. Oktar, "Depth Estimation from Single Image using Sparse Representations," *arXiv preprint arXiv:1606.08315,* 2016.

[30] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366-2374.

[31] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650-2658.

[32] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, "Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 873-878: IEEE.

[33] S. Hong, M. Li, M. Liao, and P. van Beek, "Real-time mobile robot navigation based on stereo vision and low-cost GPS," *Electronic Imaging,* vol. 2017, no. 9, pp. 10-15, 2017.

[34] I. Kostavelis, E. Boukas, L. Nalpantidis, and A. Gasteratos, "Stereo-based visual odometry for autonomous robot navigation," *International Journal of Advanced Robotic Systems,* vol. 13, no. 1, p. 21, 2016.

[35] M. Havlena, K.-K. Maninis, D. Bouget, E. Vander Poorten, and L. Van Gool, "3D Reconstruction of the Retinal Surface for Robot-Assisted Eye Surgery," in *Proceedings of the 6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery (CRAS 2016)*, 2016, pp. 112-113.

[36] S. R. Fanello *et al.*, "3D stereo estimation and fully automated learning of eye-hand coordination in humanoid robots," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, 2014, pp. 1028-1035: IEEE.

[37] H. C. Radhakrishnamurthy, P. Murugesapandian, N. Ramachandran, and S. Yaacob, "Stereo vision system for a bin picking adept robot," *Malaysian Journal of Computer Science,* vol. 20, no. 1, pp. 91-98, 2017.

[38] T. Schwarze and Z. Zhong, "Stair detection and tracking from egocentric stereo vision," in *Image Processing (ICIP), 2015 IEEE International Conference on*, 2015, pp. 2690-2694: IEEE.

[39] R. Bhola. (2006). *Binocular Vision*. Available: http://webeye.ophth.uiowa.edu/eyeforum/tutorials/Bhola-BinocularVision.htm

[40] A. T. Smith, "Binocular vision: joining up the eyes," *Current Biology,* vol. 25, no. 15, pp. R661-R663, 2015.

[41] LumenLearning and OpenStax. *Anatomy and Physiology I Module 16: The Brain and Cranial Nerves*. Available: https://courses.lumenlearning.com/suny-ap1/chapter/central-processing/

[42] Q. Liu, R. Li, H. Hu, and D. Gu, "Extracting semantic information from visual data: A survey," *Robotics,* vol. 5, no. 1, p. 8, 2016.

[43] M.-H. Chiang, H.-T. Lin, and C.-L. Hou, "Development of a stereo vision measurement system for a 3D three-axial pneumatic parallel mechanism robot arm," *Sensors,* vol. 11, no. 2, pp. 2257-2281, 2011.

[44] N. Q. Ann, M. H. Achmad, L. Bayuaji, M. R. Daud, and D. Pebrianti, "Study on 3D scene reconstruction in robot navigation using stereo vision," in *Automatic Control and Intelligent Systems (I2CACIS), IEEE International Conference on*, 2016, pp. 72-77: IEEE.

[45] S. Mattoccia, "Stereo vision: algorithms and applications," *DEIS, University Of Bologna,* 2015.

[46] A. Rajeswari, B. Bhuvaneshwari, and V. G. Priyaa, "Depth Measurement and 3D Reconstruction of Stereo Images," in *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*, 2012, pp. 161-166: IEEE.

[47] MathWorks, *Computer Vision System Toolbox™ User's Guide*. The MathWorks,Inc., 2018.

[48] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[49] W. Burger, "Zhang’s camera calibration algorithm: in-depth tutorial and implementation," *month,* 2016.

[50] B. Cyganek and J. P. Siebert, *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2009.

[51] P. Sturm, "Pinhole camera model," in *Computer Vision*: Springer, 2014, pp. 610-613.

[52] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE transactions on pattern analysis and machine intelligence,* vol. 25, no. 8, pp. 993-1008, 2003.

[53] UW-CSE-vision-faculty, "Lecture 16: Stereo and 3D Vision," p. 53

[54] A. Mordvintsev and K. Abid. (2013). *OpenCVPythonTutorials: Introduction to SIFT (Scale-Invariant Feature Transform)*. Available: http://opencv-python-tutroals.readthedocs.io/

[55] A. Mordvintsev and K. Abid. (2013). *OpenCVPythonTutorials: Introduction to SURF (Speeded-Up Robust Features)*. Available: http://opencv-python-tutroals.readthedocs.io/

[56] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," in *Computer Graphics Forum*, 2011, vol. 30, no. 6, pp. 1681-1707: Wiley Online Library.

[57] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision,* vol. 47, no. 1-3, pp. 7-42, 2002.

[58] L. Nalpantidis and A. Gasteratos, "Stereo vision depth estimation methods for robotic applications," *Depth Map and 3D Imaging Applications: Algorithms and Technologies,* pp. 397-417, 2011.

[59] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE transactions on circuits and systems for video technology,* vol. 19, no. 7, pp. 1073-1079, 2009.

[60] S.-H. Ok, Y.-H. Lee, J. H. Shim, S. K. Lim, and B. Moon, "The Impact of 3D Stacking and Technology Scaling on the Power and Area of Stereo Matching Processors," *Sensors,* vol. 17, no. 2, p. 426, 2017.

[61] C. Cigla and A. A. Alatan, "Information permeability for stereo matching," *Signal Processing: Image Communication,* vol. 28, no. 9, pp. 1072-1088, 2013.

[62] F. Zhao, Q. Huang, and W. Gao, "Image matching by normalized cross-correlation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2006, vol. 2, pp. II-II: IEEE.

[63] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007, pp. 1-8: IEEE.

[64] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE transactions on pattern analysis and machine intelligence,* vol. 31, no. 9, pp. 1582-1599, 2009.

[65] N. Einecke and J. Eggert, "A two-stage correlation method for stereoscopic depth estimation," in *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, 2010, pp. 227-234: IEEE.

[66] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European conference on computer vision*, 1994, pp. 151-158: Springer.

[67] J. Lee, D. Jun, C. Eem, and H. Hong, "Improved census transform for noise robust stereo matching," *Optical Engineering,* vol. 55, no. 6, p. 063107, 2016.

[68] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8: IEEE.

[69] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, and G. Karras, "On accurate dense stereo-matching using a local adaptive multi-cost approach," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 91, pp. 29-49, 2014.

[70] F. Tombari, S. Mattoccia, and L. Di Stefano, "Segmentation-based adaptive support for accurate stereo correspondence," in *Pacific-Rim Symposium on Image and Video Technology*, 2007, pp. 427-438: Springer.

[71] T. Liu, P. Zhang, and L. Luo, "Dense stereo correspondence with contrast context histogram, segmentation-based two-pass aggregation and occlusion handling," in *Pacific-Rim Symposium on Image and Video Technology*, 2009, pp. 449-461: Springer.

[72] Z. Gu, X. Su, Y. Liu, and Q. Zhang, "Local stereo matching with adaptive support-weight, rank transform and disparity calibration," *Pattern Recognition Letters,* vol. 29, no. 9, pp. 1230-1235, 2008.

[73] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 4, pp. 650-656, 2006.

[74] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 839-846: IEEE.

[75] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, no. 6, pp. 1397-1409, 2013.

[76] C. Cigla, "Recursive edge-aware filters for stereo matching," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 27-34: IEEE.

[77] J. Kim, "Visual correspondence using energy minimization and mutual information," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 1033-1040: IEEE.

[78] R. Szeliski *et al.*, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE transactions on pattern analysis and machine intelligence,* vol. 30, no. 6, pp. 1068-1080, 2008.

[79] B. Thai, M. Al-nasrawi, G. Deng, R. Ross, and P. Huynh, "Constrained Smoothness Cost in Markov Random Field Based Stereo Matching," in *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*, 2016, pp. 1-5: IEEE.

[80] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 25, no. 7, pp. 787-800, 2003.

[81] M. G. Mozerov and J. van de Weijer, "Accurate stereo matching by two-step energy minimization," *IEEE Transactions on Image Processing,* vol. 24, no. 3, pp. 1153-1163, 2015.

[82] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 1, pp. I-I: IEEE.

[83] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-quality real-time stereo using adaptive cost aggregation and dynamic programming," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, 2006, pp. 798-805: IEEE.

[84] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?," *IEEE transactions on pattern analysis and machine intelligence,* vol. 26, no. 2, pp. 147-159, 2004.

[85] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 30, no. 2, pp. 328-341, 2008.

[86] S. Y. Park, S. H. Lee, and N. I. Cho, "Segmentation based disparity estimation using color and depth information," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, 2004, vol. 5, pp. 3275-3278: IEEE.

[87] E. T. Psota, J. Kowalczuk, M. Mittek, and L. C. Perez, "Map disparity estimation using hidden markov trees," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2219-2227.

[88] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007, pp. 1-8: IEEE.

[89] Y. Li and D. P. Huttenlocher, "Learning for stereo vision using the structured support vector machine," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8: IEEE.

[90] N. Baha and S. Larabi, "Neural disparity map estimation from stereo image," *parameters,* vol. 6, no. 7, pp. 17-23, 2009.

[91] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1592-1599.

[92] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications,* vol. 12, no. 1, pp. 16-22, 2000.

[93] I. Brilakis, H. Fathi, and A. Rashidi, "Progressive 3D reconstruction of infrastructure with videogrammetry," *Automation in Construction,* vol. 20, no. 7, pp. 884-895, 2011.

[94] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 1, pp. 125-131: IEEE.

[95] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of database systems,* pp. 1-7, 2016.

[96] D. Scharstein, R. Szeliski, and H. Hirschmüller. Stereo [Online]. Available: http://vision.middlebury.edu/stereo/

[97] C. Stergiou and D. Siganos, "Neural Networks," *Imperial College of London Surprise 96 Journal,* vol. 4.

[98] C. Gershenson, "Artificial Neural Networks for Beginners," C.Gershenson@sussex.ac.uk.

[99] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer,* vol. 29, no. 3, pp. 31-44, 1996.

[100] M. Sarker, S. Noor, and U. K. Acharjee, *Basic Application and Study of Artificial Neural Networks*. 2017, pp. 1-12.

[101] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods,* vol. 43, no. 1, pp. 3-31, 2000.

[102] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[103] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox™ User's Guide*. The MathWorks,Inc., 2018.

[104] P. Dev, "Perception of depth surfaces in random-dot stereograms: a neural model," *International Journal of Man-Machine Studies,* vol. 7, no. 4, pp. 511-528, 1975.

[105] Y. T. Zhou and R. Chellappa, "Stereo matching using a neural network," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, 1988, pp. 940-943: IEEE.

[106] M. Vanetti, I. Gallo, and E. Binaghi, "Dense two-frame stereo correspondence by self-organizing neural network," in *International Conference on Image Analysis and Processing*, 2009, pp. 1035-1042: Springer.

[107] Ł. Laskowski, "A novel hybrid-maximum neural network in stereo-matching process," *Neural Computing and Applications,* vol. 23, no. 7-8, pp. 2435-2450, 2013.

[108] Ł. Laskowski, J. Jelonkiewicz, and Y. Hayashi, "Extensions of hopfield neural networks for solving of stereo-matching problem," in *International Conference on Artificial Intelligence and Soft Computing*, 2015, pp. 59-71: Springer.

[109] N. Baha and S. Larabi, "Accurate real-time neural disparity MAP estimation with FPGA," *Pattern Recognition,* vol. 45, no. 3, pp. 1195-1204, 2012.

[110] J.-H. Wang and C.-P. Hsiao, "On disparity matching in stereo vision via a neural network framework," *PROCEEDINGS-NATIONAL SCIENCE COUNCIL REPUBLIC OF CHINA PART A PHYSICAL SCIENCE AND ENGINEERING,* vol. 23, pp. 665-677, 1999.

[111] N. Baha and S. Larabi, "Disparity map estimation with neural network," *Proceedings of the IEEE ICMWI,* pp. 282-285, 2010.

[112] M. Cilimkovic, "Neural networks and back propagation algorithm," *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin,* vol. 15, 2015.

[113] E. Ilgun, E. Mekić, and E. Mekić, "Application of Ann in Australian Credit Card Approval," *European researcher. Series A,* no. 2-2, pp. 334-342, 2014.

[114] A. Karpathy. The Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition [Online]. Available: http://cs231n.github.io/

[115] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox™ getting started guide*. The MathWorks,Inc., 2018.

[116] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Deep learning Toolbox™ User's Guide*. The MathWorks,Inc., 2018.

[117] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks,* vol. 6, no. 4, pp. 525-533, 1993.

[118] C. Peterson, T. Rögnvaldsson, and L. Lönnblad, "JETNET 3.0—A versatile artificial neural network package," *Computer Physics Communications,* vol. 81, no. 1-2, pp. 185-220, 1994.

[119] A. Schmidt, "A modular neural network architecture with additional generalization abilities for high dimensional input vectors," *Manchester Metropolitan University, Department of Computing,* 1996.

[120] M. N. Almasri and J. J. Kaluarachchi, "Modular neural networks to predict the nitrate distribution in ground water using the on-ground nitrogen loading and recharge data," *Environmental Modelling & Software,* vol. 20, no. 7, pp. 851-871, 2005.

[121] K. Chen, "Deep and modular neural networks," in *Springer Handbook of Computational Intelligence*: Springer, 2015, pp. 473-494.

[122] M. W. Shields and M. C. Casey, "A theoretical framework for multiple neural network systems," *Neurocomputing,* vol. 71, no. 7-9, pp. 1462-1476, 2008.

[123] S.-B. Cho and J. H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 25, no. 2, pp. 380-384, 1995.

[124] C. P. Lim and R. F. Harrison, "Online pattern classification with multiple neural network systems: an experimental study," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 33, no. 2, pp. 235-247, 2003.

[125] T. Kiatkaiwansiri and S. Sinthupinyo, "Combining nodes in multiple neural network on large datasets," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on*, 2014, pp. 28-30: IEEE.

[126] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, p. 436, 2015.

[127] R. K. Sinha, R. Pandey, and R. Pattnaik, "Deep Learning For Computer Vision Tasks: A review," *arXiv preprint arXiv:1804.03928,* 2018.

[128] M. T. Jones, "Deep learning architectures: The rise of artificial intelligence," IBM developerWorks, 2017.

[129] MathWorks, *Introducing Deep Learning with MATLAB*. The MathWorks,Inc., 2017.

[130] W. Xing and D. Du, "Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention," *Journal of Educational Computing Research,* p. 0735633118757015, 2018.

[131] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach*. " O'Reilly Media, Inc.", 2017.

[132] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458,* 2015.

[133] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 2010, pp. 253-256: IEEE.

[134] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. Kruthiventi, and R. V. Babu, "A taxonomy of deep convolutional neural nets for computer vision," *Frontiers in Robotics and AI,* vol. 2, p. 36, 2016.

[135] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.

[136] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience,* vol. 2018, 2018.

[137] D. O. Pop, A. Rogozan, F. Nashashibi, and A. Bensrhair, "Fusion of stereo vision for pedestrian recognition using convolutional neural networks," in *ESANN 2017-25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2017.

[138] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot," in *International Conference on Computer Vision Systems*, 2017, pp. 300-313: Springer.

[139] M. Vitelli and S. Dasgupta, "DeepStereo Dense Depth Estimation from Stereo Image Pairs using Convolutional Neural Networks," 2015.

[140] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015, pp. 4353-4361: IEEE.

[141] S. Zagoruyko and N. Komodakis, "Deep compare: A study on using convolutional neural networks to compare image patches," *Computer Vision and Image Understanding,* vol. 164, pp. 38-55, 2017.

[142] J. Pang, W. Sun, J. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *International Conf. on Computer Vision-Workshop on Geometry Meets Deep Learning (ICCVW 2017)*, 2017, vol. 3, no. 9.

[143] M. Poggi and S. Mattoccia, "Deep stereo fusion: combining multiple disparity hypotheses with deep-learning," in *3D Vision (3DV), 2016 Fourth International Conference on*, 2016, pp. 138-147: IEEE.

[144] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research,* vol. 17, no. 1-32, p. 2, 2016.

[145] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 972-980.

[146] J. Chen and C. Yuan, "Convolutional neural network using multi-scale information for stereo matching cost computation," in *Image Processing (ICIP), 2016 IEEE International Conference on*, 2016, pp. 3424-3428: IEEE.

[147] M. Yang, Y. Liu, and Z. You, "The Euclidean embedding learning based on convolutional neural network for stereo matching," *Neurocomputing,* vol. 267, pp. 195-200, 2017.

[148] J. Park and J.-H. Lee, "A cost effective estimation of depth from stereo image pairs using shallow siamese convolutional networks," in *Robotics and Intelligent Sensors (IRIS), 2017 IEEE International Symposium on*, 2017, pp. 213-217: IEEE.

[149] H. Lu, H. Xu, L. Zhang, and Y. Zhao, "Cascaded multi-scale and multi-dimension convolutional neural network for stereo matching," *arXiv preprint arXiv:1803.09437,* 2018.

[150] L. Iocchi. (1998). *Stereo Vision: Triangulation*. Available: http://www.dis.uniroma1.it/~iocchi/stereo/triang.html

[151] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *Journal of Real-Time Image Processing,* vol. 11, no. 1, pp. 5-25, 2016.

[152] S. McGarrity and MathWorks. *Programming Patterns: Maximizing Code Performance by Optimizing Memory Access*. Available: https://uk.mathworks.com/company/newsletters/articles/programming-patterns-maximizing-code-performance-by-optimizing-memory-access.html

[153] C. Moreno. *Look-Up Tables (LUT) Operations in C++*. Available: https://www.mochima.com/articles/LUT/LUT.html

[154] B. Howison. (2015). *Bob's Imaging Fundamentals #1: Look-Up Tables*. Available: http://possibility.teledynedalsa.com/imaging-fundamentals-lut/

[155] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *Computer Vision, Graphics, and Image Processing,* vol. 35, no. 2, pp. 220-233, 1986.

[156] C. Georgoulas, L. Kotoulas, G. C. Sirakoulis, I. Andreadis, and A. Gasteratos, "Real-time disparity map computation module," *Microprocessors and Microsystems,* vol. 32, no. 3, pp. 159-170, 2008.

[157] M. A. Schulze. (2001). *What are the mean and median filters?* Available: https://www.markschulze.net/java/meanmed.html

[158] K. Wegner and O. Stankiewicz, "Depth Estimation using Modified Cost Function for Occlusion Handling," *arXiv preprint arXiv:1703.00919,* 2017.

[159] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8: IEEE.

[160] G. A. Kordelas, D. S. Alexiadis, P. Daras, and E. Izquierdo, "Enhanced disparity estimation in stereo images," *Image and Vision Computing,* vol. 35, pp. 31-49, 2015.

[161] Y. Zhu and C. Huang, "An improved median filtering algorithm for image noise reduction," *Physics Procedia,* vol. 25, pp. 609-616, 2012.

[162] MathWorks, *Image Processing Toolbox™ User's Guide*. The MathWorks,Inc., 2018.

[163] J. Jiao, R. Wang, W. Wang, S. Dong, Z. Wang, and W. Gao, "Local stereo matching with improved matching cost and disparity refinement," *IEEE MultiMedia,* vol. 21, no. 4, pp. 16-27, 2014.

[164] Z. Lee, J. Juang, and T. Q. Nguyen, "Local disparity estimation with three-moded cross census and advanced support weight," *IEEE Transactions on Multimedia,* vol. 15, no. 8, pp. 1855-1864, 2013.

[165] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, E. Petsa, and G. Karras, "A local adaptive approach for dense stereo matching in architectural scene reconstruction," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 5, p. W1, 2013.

[166] D. Min, J. Lu, and M. N. Do, "Joint histogram-based cost aggregation for stereo matching," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, no. 10, pp. 2539-2545, 2013.

[167] C. C. Pham and J. W. Jeon, "Domain transformation-based efficient cost aggregation for local stereo matching," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 23, no. 7, pp. 1119-1130, 2013.

[168] Y. Lin, N. Lu, X. Lou, F. Zou, Y. Yao, and Z. Du, "Matching cost filtering for dense stereo correspondence," *Mathematical Problems in Engineering,* vol. 2013, 2013.

[169] T. Liu, X. Dai, Z. Huo, X. Zhu, and L. Luo, "A cost construction via MSW and linear regression for stereo matching," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 914-917: IEEE.

[170] C. Shi, G. Wang, X. Pei, B. He, and X. Lin, "Stereo matching using local plane fitting in confidence-based support window," *IEICE TRANSACTIONS on Information and Systems,* vol. 95, no. 2, pp. 699-702, 2012.

[171] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 2, pp. 504-511, 2013.

[172] D. Chen, M. Ardabilian, and L. Chen, "A Novel Trilateral Filter based Adaptive Support Weight Method for Stereo Matching," in *BMVC*, 2013, vol. 318.

[173] L. De-Maeztu, A. Villanueva, and R. Cabeza, "Stereo matching using gradient similarity and locally adaptive support-weight," *Pattern Recognition Letters,* vol. 32, no. 13, pp. 1643-1651, 2011.

[174] V. Drazic and N. Sabater, "A precise real-time stereo algorithm," in *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, 2012, pp. 138-143: ACM.

[175] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, "Efficient aggregation via iterative block-based adapting support-weights," in *3D Imaging (IC3D), 2011 International Conference on*, 2011, pp. 1-5: IEEE.

[176] W. Hu, K. Zhang, L. Sun, J. Li, Y. Li, and S. Yang, "Virtual support window for adaptive-weight stereo matching," in *Visual Communications and Image Processing (VCIP), 2011 IEEE*, 2011, pp. 1-4: IEEE.

[177] J. Kowalczuk, E. T. Psota, and L. C. Perez, "Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences," *IEEE transactions on circuits and systems for video technology,* vol. 23, no. 1, pp. 94-104, 2013.

[178] L. Ma, J. Li, J. Ma, and H. Zhang, "A modified census transform based on the neighborhood information for stereo matching algorithm," in *Image and Graphics (ICIG), 2013 Seventh International Conference on*, 2013, pp. 533-538: IEEE.

[179] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research,* vol. 32, no. 11, pp. 1231-1237, 2013.

[180] E. T. Psota, J. Kowalczuk, J. Carlson, and L. C. Pérez, "A local iterative refinement method for adaptive support-weight stereo matching," in *International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 2011, vol. 14.