

國立臺灣大學電機資訊學院電機工程學系



碩士論文

Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

社交型協作機器人基於情境的涵意提供適切的服務

Social Co-Robot for Just-Good Services Based on
Situational Context Perception

謝仲凱

Chung Kai Hsieh


指導教授：羅仁權 博士

Advisor: Ren C. Luo, Ph.D.

中華民國 106 年 7 月

July 2017

誌謝



時間過得很快,轉眼間兩年的碩士班光陰即將邁入尾聲,想起剛到台大的生活,對於陌生的環境、不熟悉的人事物,感到焦慮與擔心。幸虧實驗室的學長姊 都很親切,對待學弟妹都非常好。待在這實驗室的這段日子,真的比我預期的學到更多東西,不僅僅只是學術,還包含了各種規劃事情以及溝通上的技。謝謝這段時間以來家人的支持與鼓勵,因為研究的關係偶爾才能回家一次,但每次回家都是充電,謝謝我的家人總是鼓勵並支持我,讓我能無後顧之憂專注在研究上。很感謝我敬愛的指導老師羅仁權教授,提供我們豐富的資源以及指導。老師給予我們非常多的國際觀,帶領我們參加國際研討會,教誨我們待人處事的道理,無論哪個環節都是一生難得的體驗。

在國立臺灣大學智慧機器人及自動化國際研究中心 (NTU-iCeiRA) 兩年的研究生活中,我要感謝鏡文、瑋隆、東榕、繼棠、金成、昕映、旭佳、禮聰、志遠、獻章等博班學長。還要感謝碩班學長煒森、銘駿、建安、士紘、文謙、金博、建偉、冠志、柏宏、榮育,從優秀的學長姐們身上我學到很多,也常把他們當做努力的榜樣,期許自己也能跟學長姐們一樣厲害。更不能忘記同屆一起努力奮鬥的伙伴莉彤、長鈞、李晟、晴岡、靖霖、昱佑、達方、凱鈞、俊豪、孟勳、柏凱,和你們一起窩在實驗室,不論是做研究、忙比賽、耍廢還是吃宵夜,都是我碩班生涯最快樂的事情之一。謝謝積極認真又貼心的學弟妹們,何鑫、培淳、石巖、武昱、嵩詠、智堅、威辰、錦賢、育榕、育澤、名彥、展嘉、王昊、曾旻,幫忙大大小小的事務,一起打球運動。也要感謝默默在背後幫我們完成許多瑣事的雯雅 (Tracy)、煜倫 (Dornin)、姿伶 (Amy)、芳嫻 (Helen)、佩芸 (Winnie) 等助理們。

謝謝大家的支持與幫助,不分晝夜地討論研究,同甘共苦地參加比賽,非常開心這段日子能和大家一起奮鬥,熬夜拼比賽,相信你們都是我生活中的回憶。能完成這篇論文,我要感謝在我身邊的每個人。感謝你們。

謝仲凱 謹誌

民國一百零六年七月

中文摘要



本文的目的是提出一種社交型協作機器人應用情境的涵意，學習並預測人類的想法，進而提供「剛剛好的服務」。為了在人類社交環境中與人友善的互動，機器人應具有情境感知瞭解人類社交技巧的能力並且表現出得體的行為。

在本文中，情境式上下文著重在讓機器人感知他人是否需要幫助，根據預測的人類想法，機器人提供剛剛好的服務。剛剛好的概念來自鼎泰豐餐廳的董事長，他說：「服務不足，是怠慢；殷勤過頭，變成打擾，『剛剛好的服務』是鼎泰豐團隊努力追求的目標」。在服務業方面，當顧客需要幫助時，服務員主動提供服務是非常暖心的。換句話說，當顧客不需要幫助時，不去打擾他們是很體貼的。

我們提出兩個深度學習模型，作為機器人的情境式上下文感知，並從人機互動中觀察並學習判斷人類的意圖。基於深度學習模型，我們賦予機器人感知人的意向的能力。因此，機器人可以基於預測的人類心理狀態，做出適當的社交行為。實驗結果表明，與常規分類器相比，我們提出的深度學習模型可以使機器人顯著提高預測人類思維的準確性。此外，在判斷人是否需要幫忙的任務上，基於情境式上下文的預測結果與服務業人士的意見保持高度一致。

關鍵字：人機互動、深度學習、情境感知

Abstract

The objective of this thesis is to develop a social co-robot for provision of “just-good services” using situational context perception for learning and predicting human’s mentation. To interact with humans in Human Social Environments (HSEs), robots are expected to possess the ability of situational context perception and behave appropriately.

In this paper, we employ the concept of situational context to our work, which mainly focus on making robots perceive others’ needing assistance and provide “just-good service”. The just-good concept is stem from the owner of Din Tai Fung restaurant, and he says: “Inadequate service is neglecting; too diligent become disturbing, just-good service is the goal Ding Tai Fung team pursue.” In service industry, it is indeed friendly to help others as they need. In other words, it is actually considerate not to bother others when they don’t need help.

We propose two deep learning models, as situational context perception of robot, to learn from observations of human-robot interaction. Based on these models, we endow robot the capability of perceiving human’s mentation. Thus, the appropriate social behaviors can be performed by the robot with respect to human’s mental state. The experimental results demonstrate that robot can significantly improve the accuracy of predicting a person’s mentation through the proposed deep learning models by comparison with conventional classifiers. Furthermore, the prediction of our situational context perception keep highly consistent with the opinion made by people who work in service industry.

Keywords: Human-Robot Interaction, Deep Learning, Situational Context Perception

TABLE OF CONTENTS



誌謝.....	I
中文摘要.....	II
ABSTRACT	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES	VI
LIST OF TABLE.....	VIII
CHAPTER 1 INTRODUCTION.....	1
1.1 MOTIVATION	1
1.2 OBJECTIVE	3
1.3 LITERATURE REVIEW.....	4
1.4 THESIS STRUCTURE	5
CHAPTER 2 SYSTEM STRUCTURE.....	6
2.1 HARDWARE STRUCTURE	6
2.1.1 <i>RenBo-S Service Robot</i>	6
2.1.2 <i>Kinect RGB-D Camera</i>	7
2.2 SOFTWARE STRUCTURE	9
2.2.1 <i>Point Cloud Library (PCL)</i>	9
2.2.2 <i>Open Source Computer Vision Library (OpenCV)</i>	10
2.2.3 <i>Scikit-Learn</i>	12
2.2.4 <i>KERAS</i>	14
2.2.5 <i>API.AI</i>	16
2.2.6 <i>Robot Operating System (ROS)</i>	18
CHAPTER 3 BACKGROUND AND INITIAL WORK	23
3.1 UNDERSTANDING AND USING CONTEXT	23
3.1.1 <i>Definition of Context</i>	24
3.1.2 <i>Definition of Context-Aware</i>	24
3.2 JUST-GOOD SERVICES AND ROBOT’S APPROPRIATE BEHAVIORS	25
3.2.1 <i>Definition of Just-Good</i>	25
3.2.2 <i>Robot’s Appropriate Behaviors</i>	26
3.3 INITIAL WORK	27
3.3.1 <i>Data Collection</i>	27
CHAPTER 4 SITUATIONAL CONTEXT PERCEPTION FOR JUST-GOOD SERVICES.....	29
4.1 DEFINITION OF SITUATIONAL CONTEXT PERCEPTION	29
4.2 ANALYSIS AND TRAINING METHDOLOGY	30
4.3 FEATURE EXTRACTION	31

4.3.1	<i>Handcraft Feature</i>	31
4.3.2	<i>Convolutional Neural Networks Auto-encoder</i>	33
4.4	CLASSIFIER IMPLEMENTATION.....	50
4.4.1	<i>Deep Learning Based Classifiers</i>	50
4.4.2	<i>Conventional Classifiers</i>	55
4.5	SOCIAL CO-ROBOT VERSUS PEOPLE IN SERVICE INDUSTRY.....	56
CHAPTER 5	EXPERIMENTAL RESULTS	58
5.1	DEEP LEARNING MODELS EVALUATION.....	58
5.1.1	<i>K-fold Cross-Validation</i>	58
5.1.2	<i>Features Comparison</i>	58
5.1.3	<i>Classifier Appropriateness</i>	60
5.1.4	<i>Multi-feature Fusion</i>	62
5.1.5	<i>Deep Learning Models Comparison</i>	63
5.2	SITUATIONAL CONTEXT PERCEPTION EVALUATION.....	64
5.2.1	<i>Results and Discussion</i>	66
CHAPTER 6	CONCLUSION, CONTRIBUTIONS AND FUTURE WORKS	69
6.1	CONCLUSIONS.....	69
6.2	CONTRIBUTIONS.....	70
6.3	FUTURE WORKS.....	70
	REFERENCES	71
	VITA	76



LIST OF FIGURES

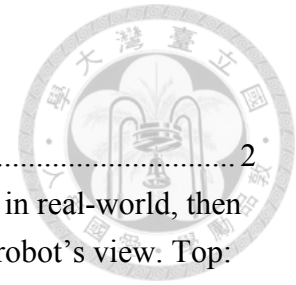


Figure 1-1 Ding Tai Fung team.....	2
Figure 1-2 A robot developed in our lab observes hundreds of human data in real-world, then asking whether they need help or not. Two images are shown at the robot’s view. Top: is a girl from France takes internship in our lab and seeking for instruments. Bottom: is a person just pass-by in the lobby at first floor of NTU research building.....	3
Figure 2-1 RenBo-S service robot.....	6
Figure 2-2 The Kinect sensor.....	7
Figure 2-3 Framework of PCL algorithms.....	10
Figure 2-4 OpenCV Overview: computer vision library.....	11
Figure 2-5 Scikit-learn: a machine learning software library.....	13
Figure 2-6 Keras: a high-level neural networks API.....	15
Figure 2-7 The character of API.AI.....	17
Figure 2-8 ROS logo.....	18
Figure 2-9 Robots which is ROS inside.....	21
Figure 2-10 The distribution of ROS.....	22
Figure 3-1 The Logo of DIN TAI FUNG.....	25
Figure 3-2 The amazing food, including steamed dumpling, fried rice and soup, supplied by DIN TAI FUNG.....	26
Figure 3-3 An overview of system architecture based on ROS for data collection via kinect sensor.....	27
Figure 4-1 present the scenario of human-robot interaction.....	29
Figure 4-2 An example of a convolutional neural network.....	30
Figure 4-3 A presentation of raw image and hog feature variation.....	31
Figure 4-4 An optical flow feature applies in human motion scenario.....	32
Figure 4-5 The architecture of an auto-encoder.....	33
Figure 4-6 The proposed convolutional auto-encoder. In our image auto-encoder architecture, the encoder part consists of three convolution layers and three max-pooling layers in stack. The decoder part is composed of three deconvolution layers and three unpooling layers in stack.....	34
Figure 4-7 The architecture of CNNs auto-encoder built upon Keras.....	37
Figure 4-8 Visualization of autoencoder results.....	38
Figure 4-9 An implementation of CNNs in category classification.....	41
Figure 4-10 The demonstration of convolution operator.....	42
Figure 4-11 A demonstration of max pooling.....	42
Figure 4-12 The role of an activation function in a neuron cell.....	43

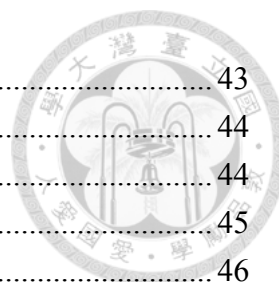


Figure 4-13 Presentation of linear function.....	43
Figure 4-14 Presentation of sigmoid function.....	44
Figure 4-15 Presentation of tanh function.....	44
Figure 4-16 Presentation of ReLU function.....	45
Figure 4-17 The presentation of SGD with/without momentum.	46
Figure 4-18 The early stopping epoch to prevent overfitting.....	49
Figure 4-19 Left one is standard NNs, Right one is after applying dropout NNs.....	49
Figure 4-20 The proposed LSTM-based classifier architecture. Where $f_k = (f_0, f_1, \dots, f_N)$ and p_k denote as sequence features and probability prediction of k-th observation to learning and predicting human's mentation by analyzing features, in sequence.....	52
Figure 4-21 The architecture of LSTM-based classifier implemented on Keras.....	52
Figure 4-22 Present the proposed CNNs followed by LSTM architecture	53
Figure 4-23 The CNNs followed by LSTM architecture implemented on Keras.....	54
Figure 4-24 Linear SVM classification. Blue points are one class and green points are another class. SVM tries to classify two class.....	55
Figure 4-25 The situational context perception apply upon Human-robot interaction.	56
Figure 4-26 The question is answered by people after observing the person's behavior.	57
Figure 5-1 Demonstrate two sequential human behaviors, which are observed by robot. Both social co-robot and people work in service industry evaluate the mentation of people via these data.....	64
Figure 5-2 This figure shows the distribution of robot's prediction and decision made by people in service industry with respect to ground truth.....	67
Figure 5-3 Three human's behaviors contain non-consistent opinion among robot's prediction, people voting and ground truth.....	67

LIST OF TABLE



Table 2-1 Documentation of Kinect.....	8
Table 3-1 Needing assistance distribution of observational data collected from Laboratory and Lobby.	28
Table 5-1 Results from features comparison by 5-fold cross-validation, applying LSTM architecture to learn from different features.	60
Table 5-2 Results from Experiment of classifier appropriateness, applying LSTM-RNN architecture, SVM and Gaussian Naive Bayes to classify needing assistance from aforementioned features.	61
Table 5-3 Results from multi-feature fusion, applying LSTM-RNN architecture, SVM and Gaussian Naive Bayes to classifier needing assistances from concatenated two kinds of aforementioned features.	62
Table 5-4 Results for perceiving a person's mentation by 5-fold cross-validation, applying CNNs followed by LSTM architecture to learn from different features.	63
Table 5-5 This table shows the accuracy of robot's prediction and decision made by voting among people in service industry with respect to ground truth.	65



Chapter 1 INTRODUCTION

The main topic of this thesis is to present a situational context perception for a social co-robot to learn and to predict a person's mentation by deep learning models. The person's mentation in our research focuses on whether a person need help or not. The robot's appropriate behavior is determined upon the prediction of a person's mentation. We believe that it is indeed friendly take the initiative to help others as they need. In other words, it is actually considerate not to bother others when they don't need help.

In this Chapter, the motivation of the thesis is elaborated in Section 1.1. In Section 1.2, introduces the clear objective of the research. In Section 1.3, the related work provides a brief description of the existing work on the determination of robot's appropriate behaviors. In Section 1.4, presents the contribution of this research. Eventually, the overall organization of this thesis and the relationship among all Chapters are illustrated in Section 1.5.

1.1 MOTIVATION

Along with the highly mature technologies of mobile robot's location and navigation [1][2][3][4], there will be more and more robots involved in human's lives. However, if robots want to truly take part in our livelihood, they need not only possess the ability of perception and cognition but also act properly. A kind of robots, defined as social co-robots [5], have the ability to realize a proxemic interaction with humans in Human Social

Environments (HSEs). Furthermore, robots can become more human friendly, lovely and attractive when they are capable of having awareness of situational context [6][7][8]. The situational context [9] describes the reason why something is occurring and the appropriate behavior and actions associated with the situation. Consequently, the situational context plays an essential role in research of social co-robots. The first approach employ this concept is proposed by Nigam et al. [10]. The interesting results inspire us to pursue another challenging task, perceiving others' needs to provide just-good service. The concept of 'just-good service' is come from the owner of Din Tai Fung restaurant, and he says: "Inadequate service is neglecting; too diligent become disturbing, 'just-good service' is the goal Ding Tai Fung team (as shown in Figure 1-1) pursue." In service industry, it is indeed friendly taking the initiative to help others as they need. In other words, it is actually considerate not to bother others when they don't need help. We believe that once robots are equipped with just-good concept, they will become more human friendly, lovely and attractive in human societies.



Figure 1-1 Ding Tai Fung team

1.2 OBJECTIVE

The objective of this paper is to develop a social co-robot for provision of “just-good service” using situational context based perception for perceiving human’s mentation. Human’s mentation in our study focuses on whether a person needs help or not. The overall observation scenario is shown in Figure 1-2 . We propose two supervised learning models for perceiving needing assistance. In order to make robot learn from the observations, we develop two deep learning models to learn from a sequence of features. Consequently, the robot could possess the ability to analyze sequential features which reveal human’s mentation.

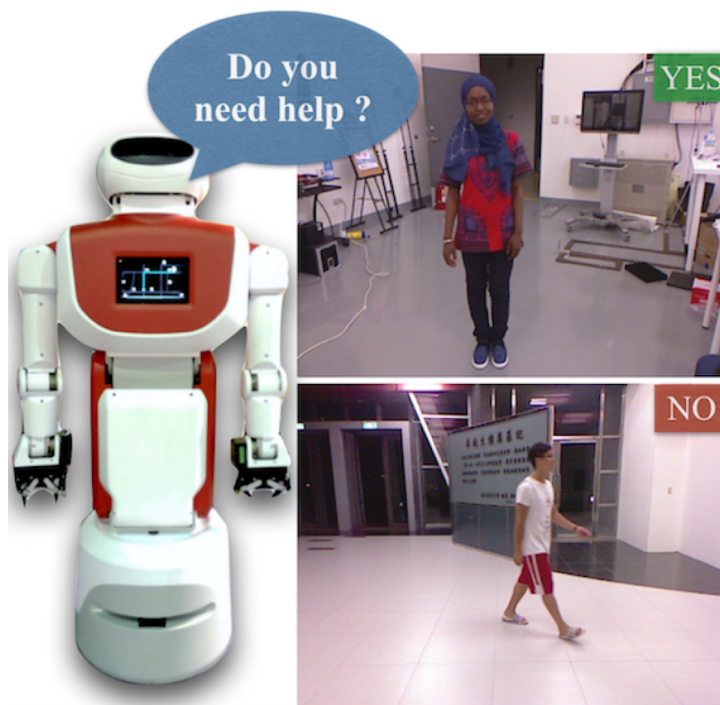
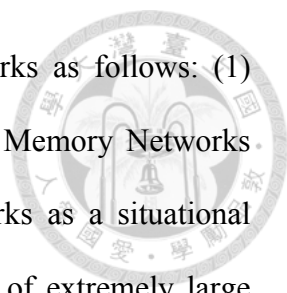


Figure 1-2 A robot developed in our lab observes hundreds of human data in real-world, then asking whether they need help or not. Two images are shown at the robot’s view. Top: is a girl from France takes internship in our lab and seeking for instruments. Bottom: is a person just pass-by in the lobby at first floor of NTU research building.



In our study, we utilize two kinds of artificial neural networks as follows: (1) Convolutional Neural Networks (CNNs) and (2) Long Short-term Memory Networks (LSTMs). The convolutional auto-encoder based upon CNNs works as a situational context compression for raw images input and tackle the problem of extremely large computation cost for analysis. The LSTM-based classifier is employed as learning human's mentation from descriptor, in sequence.

1.3 LITERATURE REVIEW

The deep learning model we proposed enable robot to learn from experience interacting with people, and determine robot's appropriate social behavior. This section demonstrates related works which contain the topic of human-robot interaction and corresponding appropriate behaviors.

The work by Qureshi et al [11] shows that the robot can determine whether it is appropriate to shake hands with people in a certain interaction period by utilizing deep reinforcement learning. The work in [12] tries to map a human's verbal behavior to a corresponding combined robot's verbal-nonverbal appropriate social behavior. The results present that individuals preferred more to interact with a robot that had the same personality with theirs. In [13], the author depicts that robots must display appropriate proxemic behavior — that is, follow societal norms in establishing their physical and psychological distancing with people. The results point out the participants who disliked the robot compensated for the increase in the robot's gaze by maintaining a greater physical distance from the robot. The research in [14] proposes that in a shopping scenario, the robot needs to understand which locations are appropriate for waiting as they are

waiting for users. The experimental results reveal that the user found robot, equipped with autonomous waiting skill, chose more appropriate location than a robot with random choice. The aforementioned works present robot's appropriate behaviors with respect to different human-robot interactions, and demonstrate interesting results.

So far, the deep learning methodology has been applied to areas, though include robotics, which have little to do with the domain of human-robot interaction. To the best knowledge of authors, we are the first team to let robot provide just-good service via making robot perceive the person's needs by deep learning methodology.

1.4 THESIS STRUCTURE

This thesis is organized as follows: In Chapter 1, we elaborate our motivation, objectives and the literature review of Human-Robot Interaction which is relevant to robot's appropriate behaviors. In Chapter 2, presents our system architecture, including hardware and software. In Chapter 3, introduces psychological background and the initial work. In Chapter 4, describes the proposed deep learning models for learning and predicting a person's need. In Chapter 5, demonstrates experimental results to evaluate our deep learning models. The conclusions, contributions and future works are described in Chapter 6.

Chapter 2 SYSTEM STRUCTURE



In this Chapter, the overall system, including hardware structure and software structure, is described. The hardware platform, we use a mobile-based service robot and RGB-D sensor, is covered in Section 2.1. The software platform, discussed in Section 2.2, briefly introduce as followed.

2.1 HARDWARE STRUCTURE

2.1.1 RenBo-S Service Robot

In this thesis, we utilize “RenBo-S”, as shown in Figure 2-1, domestic service robot as our platform to perceive and predict a person’s mentation. As for the sensor, the Kinect sensor is chosen to mounted on the head of RenBo-S for retrieving RGB and depth information to implement of our situational context perception. This robot is equipped with an RGB-D camera mounted on its head, a user interface panel in the chest, two 5 degrees of freedom manipulators, UTM-30LX laser range finder mounted at the front, and a mobile platform.

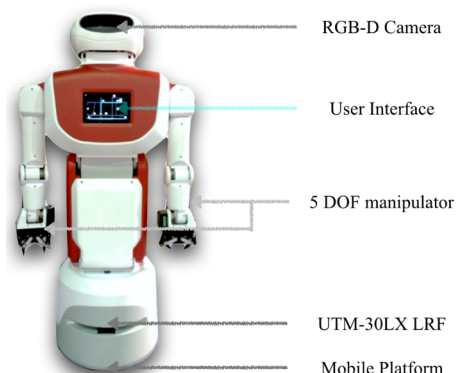


Figure 2-1 RenBo-S service robot.

2.1.2 Kinect RGB-D Camera

Kinect is a line of motion sensing devices developed by Microsoft for Xbox 360 and Microsoft Windows PCs. Based upon a webcam-style, Kinect enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface (UI) via using gestures and spoken commands. Through natural gesture and speech command, Kinect becomes a popular input device that can be applied in human robot interaction applications. Through the infrared and camera mounted on Kinect (see Figure 2-2, Table 2-1), RGB and depth information of the environment can be retrieved, and this supports us to implement more applications, such as human body detection, gesture detection, or 3D recognition and tracking. Since the price of Kinect is much cheaper than traditional 3D cameras, it has been utilized in areas.

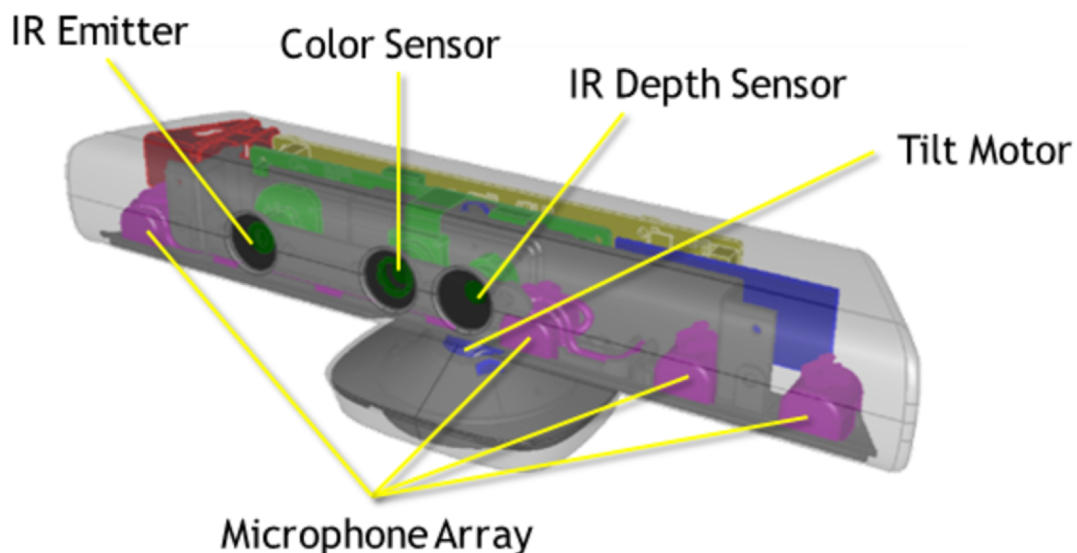


Figure 2-2 The Kinect sensor.



Kinect	Array Specifications
Viewing angle	43° vertical by 57° horizontal field of view
Vertical tilt range	±27°
Frame rate (depth and color stream)	30 frames per second (FPS)
Audio format	16-kHz, 24-bit mono pulse code modulation (PCM)
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression
Accelerometer characteristics	A 2G/4G/8G accelerometer configured for the 2G range, with a 1° accuracy upper limit.

Table 2-1 Documentation of Kinect.

2.2 SOFTWARE STRUCTURE



2.2.1 Point Cloud Library (PCL)

PCL [15] is an open-source library of algorithms for point cloud processing tasks and 3D geometry processing, such as occur in three-dimensional computer vision. The PCL framework contains numerous state-of-the-art algorithms including filtering, feature estimation, surface reconstruction, registration, model fitting and segmentation. These algorithms have been used, for example, for perception in robotics to filter outliers from noisy data, stitch 3D point clouds together, segment relevant parts of a scene, extract keypoints and compute descriptors to recognize objects in the world based on their geometric appearance, and create surfaces from point clouds and visualize them.

PCL is released under the terms of the 3-clause BSD license and is open source software. It is free for commercial and research use. PCL is cross-platform, and has been successfully compiled and deployed on Linux, MacOS, Windows, and Android/iOS. To simplify development, PCL is split into a series of smaller code libraries, that can be compiled separately. This modularity is important for distributing PCL on platforms with reduced computational or size constraints. Another way to think about PCL is as a graph of code libraries, similar to the Boost set of C++ libraries (see Figure 2-3). The development of the Point Cloud Library started in March 2010 at Willow Garage.

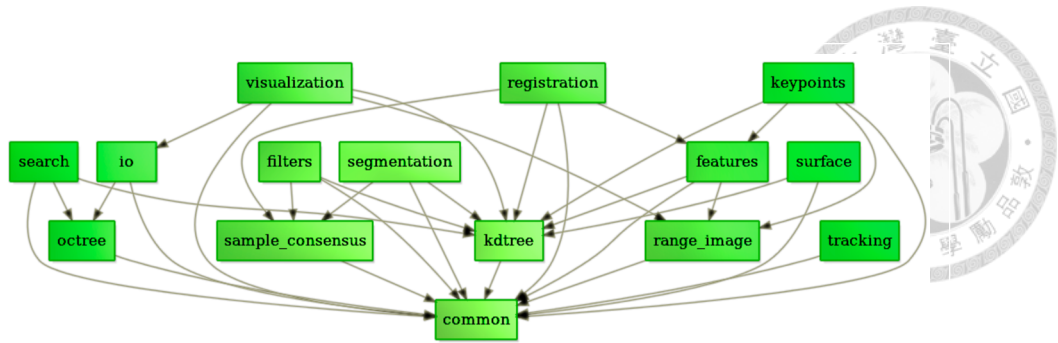


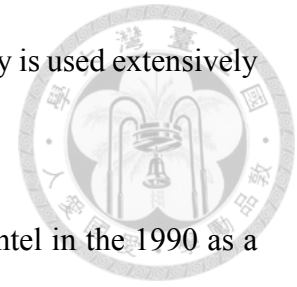
Figure 2-3 Framework of PCL algorithms.

2.2.2 Open Source Computer Vision Library (OpenCV)

OpenCV [16] is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for real-time computer vision applications and to accelerate the usage of machine perception in the commercial products. The library is free for use thanks to being under the open-source BSD license. Therefore, it is easy for businesses to utilize and modify the code.

The library contains more than 500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be applied in areas, such as: classify human actions in videos, detect and recognize faces, identify objects, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc (see Figure 2-4). OpenCV possess more than 47 thousand people of user community

and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.



From a historical point of view, OpenCV originally developed by Intel in the 1990 as a method to demonstrate how to accelerate certain algorithms in hardware. In 2000, Intel released OpenCV to the open source community as a beta version, followed by v1.0 in 2006. In 2008, Willow Garage took over to support and immediately released v1.1. Nowadays, OpenCV is maintained by Itseez.

OpenCV v2.0, released in 2009, contained many improvements and upgrades. Initially, OpenCV was primarily a C library. Subsequent versions of OpenCV added Python support, along with Windows, Linux, iOS and Android OS support, transforming OpenCV (currently at v3.2) into a cross-platform tool. OpenCV v3.2 contains more than 2500 supported functions.

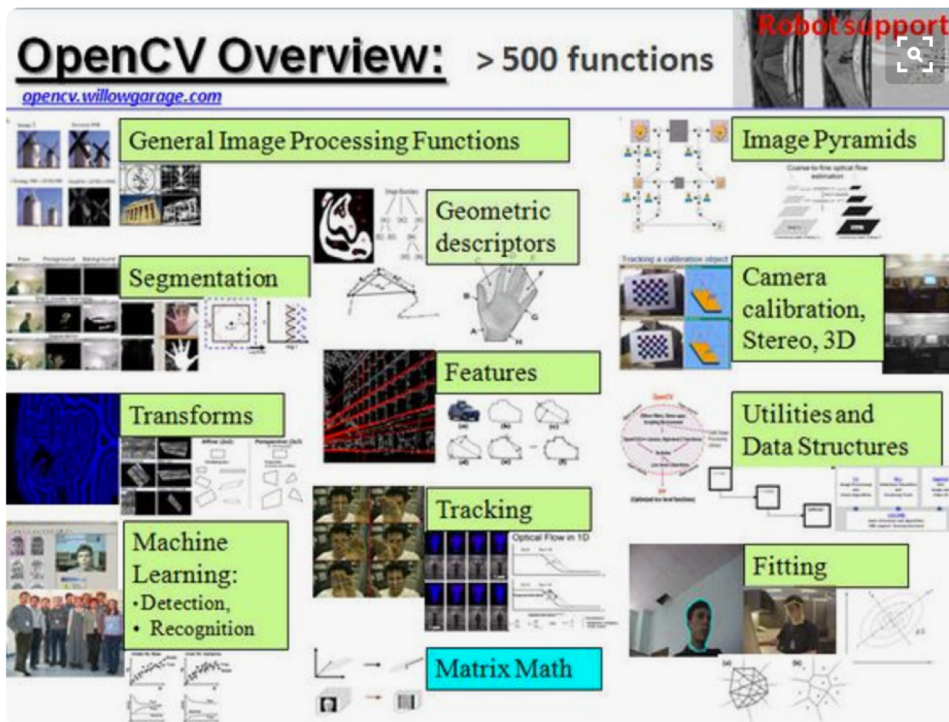


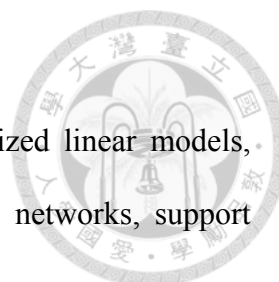
Figure 2-4 OpenCV Overview: computer vision library.

2.2.3 Scikit-Learn

Scikit-Learn [17] is a free software library for machine learning and support Python programming language. It contains various classification, regression, and clustering algorithms, such as Support Vector Machine (SVM), Random Forests, gradient boosting and k-means. Scikit-Learn is designed to interoperate with the Python numerical (NumPy) and scientific libraries (SciPy).

Scikit-Learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. Some popular groups of models (as shown in Figure 2-5) provided by Scikit-Learn include:

- **Clustering:** for grouping unlabeled data such as K-Means.
- **Cross Validation:** for estimating the performance of supervised models on unseen data.
- **Datasets:** for test datasets and for generating datasets with specific properties for investigating model behavior.
- **Dimensionality Reduction:** for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.
- **Ensemble methods:** for combining the predictions of multiple supervised models.
- **Feature extraction:** for defining attributes in image and text data.
- **Feature selection:** for identifying meaningful attributes from which to create supervised models.
- **Parameter Tuning:** for getting the most out of supervised models.
- **Manifold Learning:** For summarizing and depicting complex multi-



dimensional data.

- **Supervised Models:** a vast array not limited to generalized linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

From the historical point of view: Scikit-learn was initially developed by David Courneau as a Google summer of code project in 2007. Later, Matthieu Brucher joined the project and started to utilize it as a part of his thesis work. In 2010 INRIA got involved and the first public release (v0.1 beta) was published in late January 2010.

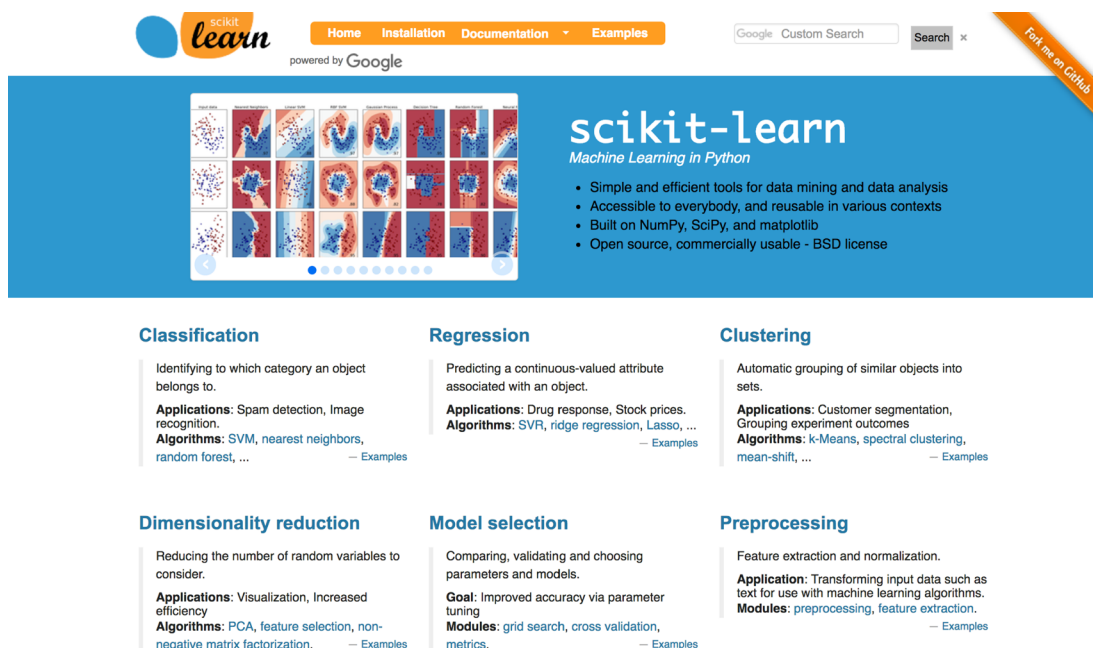


Figure 2-5 Scikit-learn: a machine learning software library.

2.2.4 KERAS

Keras [18] is an open source high-level neural networks Application Programming Interface (API), written in Python and capable of running on top of either TensorFlow, CNTK or Theano. The logo of Keras is shown in Figure 2-6. It was developed with a focus on enabling fast experimentation upon deep learning models. The core concept of Keras is that being able to go from idea to result with the least possible delay is key to doing good research.

The dominant characteristics of developing deep learning models via Keras are user friendliness, modularity, easy extensibility, and work with python. These features enable us to construct deep neural network (DNN), recurrent neural network (RNN) and convolutional neural network (CNN) more effortless than before. The detail properties is elaborated as followed:

- **User friendliness:** Keras is an API designed for human beings, not machines. It puts user experience (UX) front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.
- **Modularity:** A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as little restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that you can combine to create new models.
- **Easy extensibility:** New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new

modules allows for total expressiveness, making Keras suitable for advanced research.

- **Work with Python:** No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library. Chollet, the original author of Keras, explained that Keras was conceived to be an interface rather than an end-to-end machine-learning framework. It presents a higher-level, more intuitive set of abstractions that make it easy to configure neural networks regardless of the backend scientific computing library. Microsoft has been working to add a CNTK backend to Keras as well and the functionality is currently in beta release with CNTK v2.0.



Figure 2-6 Keras: a high-level neural networks API.

2.2.5 API.AI

Api.ai [19] (formerly Speaktoit) is a developer of human–computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant (by Speaktoit), a virtual buddy for Android, iOS, and Windows Phone smartphones that performs tasks and answers users' question in a natural language. Speaktoit has also created a natural language processing engine that incorporates conversation context like dialogue history, location and user preferences.

In May 2012, Speaktoit received a venture round (funding terms undisclosed) from Intel Capital. In July 2014, Speaktoit closed their Series B funding led by Motorola Solutions Venture Capital with participation from new investor Plug and Play Ventures and existing backers Intel Capital and Alpine Technology Fund. In September 2014, Speaktoit released api.ai (the voice-enabling engine that powers Assistant) to third-party developers, allowing the addition of voice interfaces to apps based on Android, iOS, HTML5, and Cordova. The SDK's contain voice recognition, natural language understanding, and text-to-speech. api.ai offers a web interface to build and test conversation scenarios. The platform is based on the natural language processing engine built by Speaktoit for its Assistant application. Api.ai allows Internet of Things developers to include natural language voice interfaces in their products. Assistant and Speaktoit's websites now redirect to api.ai's website. Google bought the company in September 2016 and it is now known as API.AI; it provides tools to developers building apps ("Actions") for the Google Assistant virtual assistant.

API.AI contains two major modules in dialogue system: (1) Natural Language Understanding (NLU) and Dialogue Management (DM). API.AI receives a query as input data. A query is either text in natural language or an event name sent to API.AI. API.AI matches the query to the most suitable intent based on information contained in the intent (examples, entities used for annotations, contexts, parameters, events) and the agent's machine learning model. API.AI transforms the query text into actionable data and returns output data as a JSON response object. The process of transforming natural language into actionable data is called Natural Language Understanding (NLU). Dialog management tools such as contexts and intent priorities allow developers to control the conversation flow.

The character of API.AI in the whole conversation between human and robot is shown in Figure 2-7. In the diagram, the green is provided by the API.AI platform. Your app / bot / device code provides the input and output methods and responds to actionable data. You can also provide an optional webhook implementation which API.AI uses to connect to your web service. Your web service can then perform business logic, call external APIs, or access data stores.

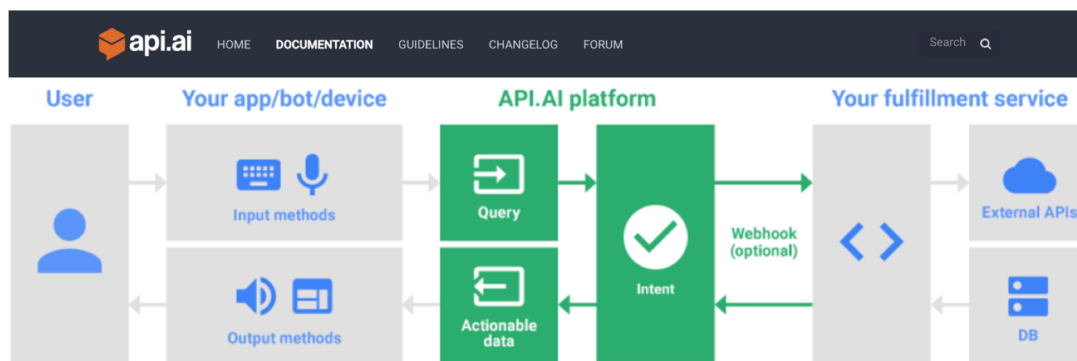


Figure 2-7 The character of API.AI

2.2.6 Robot Operating System (ROS)

ROS [20] is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. ROS, as shown in Figure 2-8, provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator and other messages.



Figure 2-8 ROS logo.

ROS was built from the ground up to encourage collaborative robotics software development. For example, one laboratory might have experts in mapping indoor environments, and could contribute a world-class system for producing maps. Another group might have experts at using maps to navigate, and yet another group might have discovered a computer vision approach that works well for recognizing small objects in clutter. ROS was designed specifically for groups like these to collaborate and build upon each other's work.

For the general concepts of ROS, it starts with the ROS Master. The Master allows all other ROS pieces of software (Nodes) to find and talk to each other. That way, we do not have to ever specifically state "Send this sensor data to that computer at 127.0.0.1". We can simply tell Node 1 to send messages to Node 2 as shown as Fig. 2-9. The Nodes do this by publishing and subscribing to Topics.

If we have a camera on our Robot. We want to be able to see the images from the camera, both on the Robot itself, and on another laptop. For instance, we can write a Camera Node that takes care of communication with the camera, a Image Processing Node on the robot that process image data, and a Image Display Node that displays

images on a screen as shown in Fig. 2-10. To start with, all Nodes have registered with the Master. Think of the Master as a lookup table where all the nodes go to find where exactly to send messages. In registering with the ROS Master, the Camera Node states that it will Publish a Topic called `/image_data`. Both of the other Nodes register that they are Subscribed to the Topic `/image_data`. Thus, once the Camera Node receives some data from the Camera, it sends the `/image_data` message directly to the other two nodes.

The main ROS client libraries (C++, Python, LISP) are geared toward a Unix-like system, primarily because of their dependence on large collections of open-source software dependencies. For these client libraries, Ubuntu Linux is listed as "Supported" while other variants such as Fedora Linux, Mac OS X, and Microsoft Windows are designated "Experimental" and are supported by the community. Being such a powerful tool, ROS has been used in many robot platforms as shown in Figure 2-9.



Software in the ROS Ecosystem can be separated into three groups:

- Language-and platform-independent tools used for building and distributing ROS-based software;
- ROS client library implementations such as roscpp, rospy, and roslisp;
- Packages containing application-related code which uses one or more ROS client libraries.

ROS was originally developed in 2007 under the name switchyard by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot STAIR project. From 2008 until 2013, development was performed primarily at Willow Garage, a robotics research institute/incubator. During that time, researchers at more than twenty institutions collaborated with Willow Garage engineers in a federated development model. In February 2013, ROS stewardship transitioned to the Open Source Robotics Foundation. In August 2013, a blog posting announced that Willow Garage would be absorbed by another company started by its founder, Suitable Technologies. The support responsibilities for the PR2 created by Willow Garage were also subsequently taken over by Clearpath Robotics. Until 2016, The ROS distribution has a variety of versioned set of ROS packages as shown in Figure 2-10.







































RBCAR	ROBOTIS MANIPULATOR	Shadow Hand	Summit-X
			
TurtleBot2	Thormang	Roch	RB-1
			
Tally	Ridgeback	Warthog	WheeledRobin
			
Fetch	Care-O-Bot	AUBO I-Series	AMIGO
			
ABB Manipulators	CoroBot	MPO-500	Mover
			
ADAS Development Vehicle Kit	Freight	Cyton Gamma	Guardian
			
Maggie	Pepper	nao	PR2
			
Mobility Base	Jazz	Pioneer LX Manipulator	PMB-2
			
HiroNXO	Grizzly	BIG-i	Panther
			

Figure 2-9 Robots which is ROS inside.















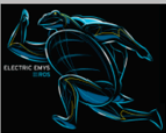

Distro	Release date	Poster	Turtle, turtle in tutorial	EOL date
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame (Recommended)	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015
ROS Groovy Galapagos	December 31, 2012			July, 2014
ROS Fuerte Turtle	April 23, 2012			--
ROS Electric Emys	August 30, 2011			--

Figure 2-10 The distribution of ROS.

Chapter 3 BACKGROUND and INITIAL WORK



In this chapter, the background of context is elaborated and how the context is utilized in Human-Robot Interaction. The definition of context, context-awareness application and situational context perception are also integral demonstrated. The overall scenario of this thesis, including situational context perception and corresponding appropriate robot's behavior, is intact defined. Lastly, the initial work for observation from robot's view is presented.

3.1 UNDERSTANDING and USING CONTEXT

Humans are good at ideas conveying to each other and react with an appropriate manner. We can understand others intention due to many factors, such as the background we shared, the richness language we utilized, and the implicit understanding of situations. During a conversation, humans are able to apply implicit situational information, or context, to increase the bandwidth of conversation. Unfortunately, this talent is not well comprehended by robot. In a human-robot interaction, robots still have impoverished information to naturally interact with humans. Thus, we should understand what the context is and how it can be employed to well determine what robot's context-aware behaviors to support user-friendly service application.

3.1.1 Definition of Context

While context definition varies across fields, the most general definition accepted by researchers is made by Dey. He defines context as: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. "

This definition simplifies the procedure to define an application scenario on enumerating the context. In a human-robot interaction, if a piece of information can be used to describe this interaction then we can call that information is context.

3.1.2 Definition of Context-Aware

In Dey's definition: "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task." He proposes there are three categories of features that an application equipped with context-aware can support. First, presentation of information and services to a user. Second, automatic execution of a service for a user. Lastly, tagging of context to information to support later retrieval.

The Dey's definition of context-aware can easily elaborate in human-robot interaction for service application. From author's point of view, if a robot is capable to understand a user's intent or mentation for provision of human-friendly services which user needed, we can call it as context-aware robot application. The robot possesses the concept of context-aware will make it more easier take part in human livelihood.

3.2 JUST-GOOD SERVICES and ROBOT'S APPROPRIATE BEHAVIORS



The definition of just-good and the corresponding robot's appropriate are demonstrated in this section. Along with the mobile robot localization and object pose grasping problems in a working environment have been central research and get well performance, there are an increasing number of service robots participate in service industry. Take restaurants as an example, if a restaurant manager wishes to create a successful robotics service restaurant model, one of the key components is that service is customer-oriented. Therefore, the robot must be equipped with situational context perception to understand human's intention for provision human-friendly services. In this thesis we believe that just-good service is one of the most human-friendly services.

3.2.1 Definition of Just-Good

The definition of just-good service is come from the owner of Din Tai Fung, as shown in Figure 3-1.



Figure 3-1 The Logo of DIN TAI FUNG

Din Tai Fung is one of the world-famous restaurants known for the amazing steamed dumplings as shown in Figure 3-2. The owner of Din Tai Fung restaurant says: “Inadequate service is neglecting; too diligent become disturbing, ‘just-good service’ is the goal Ding Tai Fung team pursue.” From author’s point of view, the definition of just-good is that social a co-robot can serve appropriately, neither neglecting nor disturbing, in service industry.

3.2.2 Robot’s Appropriate Behaviors

In this article, we focus on the event: “whether people need help or not?” At the Ji-Hua, Yang’s point of view, the magic of the service is that serve before customers ask. Therefore, we wish that a robot can identify a person needs help or not before he/she begins to ask. We believe that it’s a truly magic and playing a key role for robots to be involved in human’s lives. Thus, our scenario is clearly defined as a robot observes the procedure of a human entering in a social environment, and ask whether he/she needs help or not. Simultaneously, in order to achieve the target: ‘Just-good service.’, the robot begins to ask whether he/she needs help only when the person truly needs it, for the purpose of serving appropriately to prevent neglecting or disturbing.



Figure 3-2 The amazing food, including steamed dumpling, fried rice and soup, supplied by DIN TAI FUNG

3.3 INITIAL WORK

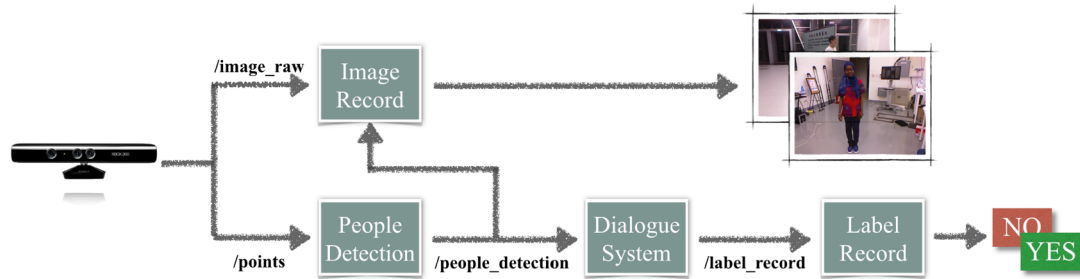
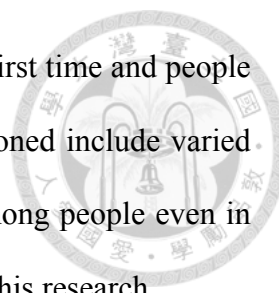


Figure 3-3 An overview of system architecture based on ROS for data collection via kinect sensor.

3.3.1 Data Collection

Our initial work aims at collecting context information to determine the robot's proper behavior. Take not only a human and a robot but also the environment into considerations, the perception and behaviors can interact synergistically via the environment. We suppose that only the raw images can fully represent the proxemics among humans, social environments and robots.

1) Data Collection: We collect data from real-world. In our ultimate goal, we wish the robots with situational context perception which can learn from naturalistic data and reflect to the real situation. Therefore, we choose one domestic place and one public place around Research Building at National Taiwan University which is located in our campus. The domestic one is the room 304 of our lab on the third floor. For the public one, we choose the lobby on the first floor. In the lab, two main contexts can occur, one is going to do research context and another is finding something or somebody context. Thus, the human in the finding something or somebody context may need help. In the lobby, there are multiple situational contexts. For example, going to work context, going to study context, joining the course of D-School on the fourth floor context, etc.



In the lobby case, students who join the course of D-School for the first time and people who go to the lavatory may need help. All the contexts aforementioned include varied levels of surrounding noise and having large different behaviors among people even in the same context. However, this is also the interesting part of doing this research.

The overall interaction architecture is shown in Figure 3-3. During the procedure of data collection, our robot is set to be as a counter staff, and stand still in front of a door to observe human behavior. The observation is triggered by the showing of a person and our robot would begin to greeting with he/she. After the five second observation, robot would ask the person whether he/she needs help or not and the person's answer is regard as label. To gather robust data set, we collect data across several days, and collection occurs during different times (morning, afternoon, evening). Totally, we collected 200 observations and labels. There are two responses, positive (the person need assistance) and negative (the person doesn't need assistance) as shown in Table 3-1.

	Positive Response	Negative Response
Laboratory	56	48
Lobby	41	55
Total	97	103

Table 3-1 Needing assistance distribution of observational data collected from Laboratory and Lobby.

Chapter 4 SITUATIONAL

CONTEXT

PERCEPTION for JUST-GOOD SERVICES



In chapter 4, we will mainly focus on demonstrating the situational context perception for provision of just-good service in human-robot interaction. The definition of situational context perception is described, and relevant technical methodologies are also presented.

4.1 DEFINITION OF SITUATIONAL CONTEXT

PERCEPTION

In this thesis, we focus on making robot perceive a person's needs, and provide just-good services. Therefore, the human and robot are the context in human-robot interaction, as shown in Figure 4-1, for service application. The contexts in our scenario includes the position, facing direction, duration of observation of our social co-robot and the walking trajectory, walking speed, and human body language of a person. In our setting, the definition of situational context perception is that our social co-robot possesses the capability to perceive a person's need via situational context we defined.



Figure 4-1 present the scenario of human-robot interaction.

4.2 ANALYSIS and TRAINING METHDOLOGY

Our situational context perception learning model, for perceiving a person's needs, consists of two parts. The first part (learning) is gathering observation to train our deep LSTM-RNN perceptual model. The second part (association) is putting testing data set to perceptual model for evaluation. Each observation is record with ten Hz during five seconds. Every piece of image is considered as one keyframes. Therefore, it is composed of totally 50 keyframes per observation. In order to learn sequential data, the first obstacle we have to face is computational cost. The dimension reduction mechanism should be included. Thus, the Convolutional auto-encoder based on Convolutional Neural Networks (CNNs), as shown in Figure 4-2, is utilized as an image encoder. In our opinion, the hidden layer information of CNNs auto-encoder may fully represent the original image and contains some hidden messages. The encoded images are taken as features, then we use LSTM-RNN to learn features in sequence.

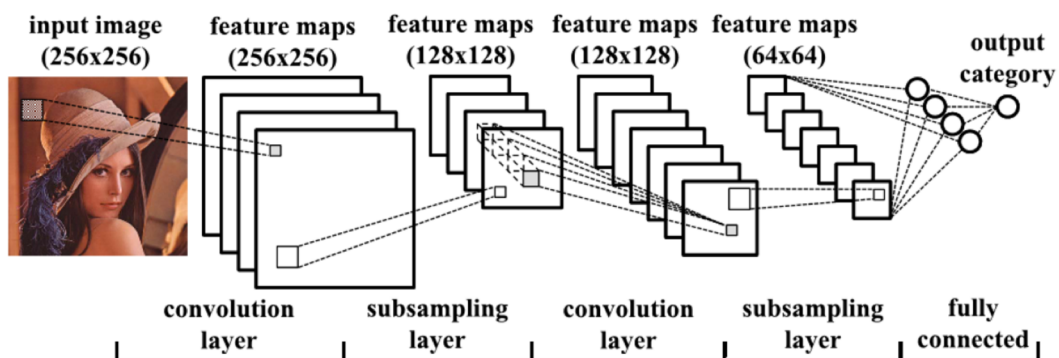


Figure 4-2 An example of a convolutional neural network.

4.3 FEATURE EXTRACTION



4.3.1 Handcraft Feature

For the purpose of perceiving a person's needs via applying situational context perception in a human-robot interaction, we should understand which key context is to pursue our goal, making robot provide just-good services. From author's point of view, human body language would be the key context in this scenario. Thus, Histogram of Oriented Gradient (HOG) and Optical Flow are employed as our handcraft features for comparison with encoded images.

- Histogram of Oriented Gradient (HOG)

The histogram of oriented gradients is a feature descriptor utilized in image process and computer vision. HOG feature is well-known for the application upon human detection [21] with support vector machine (SVM) classifier. An example of raw image and varies hog features are shown in Figure 4-3.



Figure 4-3 A presentation of raw image and hog feature variation.

- Optical Flow

Optical flow [22] is one of feature in computer vision for detecting motion of objects, surfaces, and edges in visual system. The motion is caused by the relative motion between an observer and a scene. The first introduce Optical flow is James J. Gibson, an American psychologist. Gibson emphasize that optical flow is important for affordance perception, the ability to discriminate the action within the environment. The observer's, such as a monitor system or a robot, perception of movement can be benefit from the optical flow feature. The advantage of Optical Flow feature is that it takes Spatio-Temporal factor into account and could be utilized in transition people motion analysis, as shown in Figure 4-4.

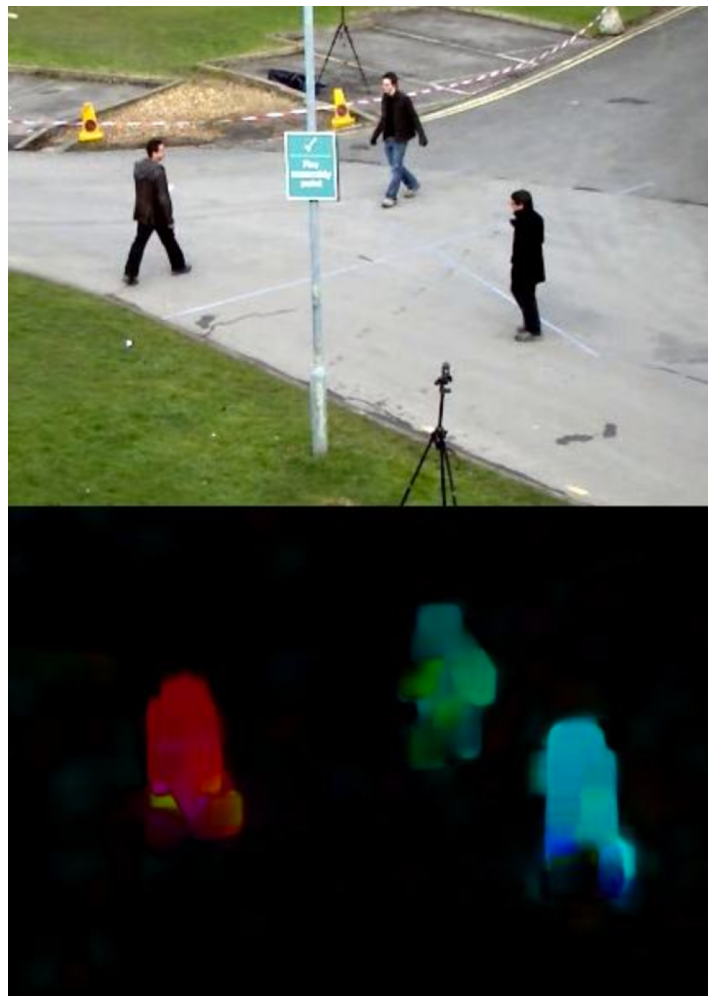


Figure 4-4 An optical flow feature applies in human motion scenario.

4.3.2 Convolutional Neural Networks Auto-encoder

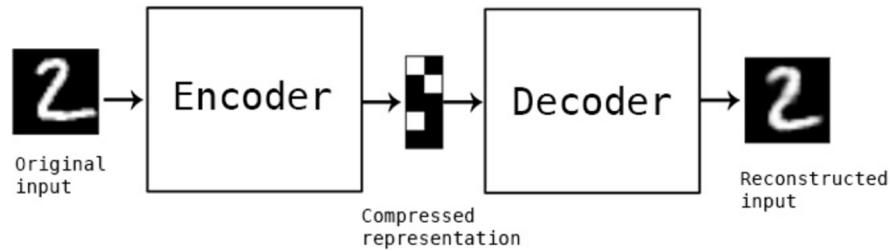


Figure 4-5 The architecture of an auto-encoder.

Autoencoding [23] is a data compression algorithm for dimension reduction. Auto-encoder, as shown in Figure 4-5, consists of two parts, compression and decompression, which are learned automatically from data instead of human engineering. To establish an auto-encoder, we should prepare three things: an encoding function, a decoding function, and a loss function to evaluate the similarity between output (reconstructed part) and the original input. The encoding and decoding parts would be parametric functions, especially neural networks, and to be differentiable with respect to loss function, hence the parameter of encoder/decoder functions can be optimized to reduce the error between reconstructed input and original input (i.e. minimize loss function).

Nowadays, auto-encoder technology is applied in two interesting practical applications, which are data denoising and dimensionality reduction for visual data. In this topic, we would like to implement auto-encoder in dimension reduction application. With proper dimensionality and sparsity constraints, auto-encoders can learn data projections which possess more interesting results than Principal Component Analysis (PCA) or other conventional technologies.

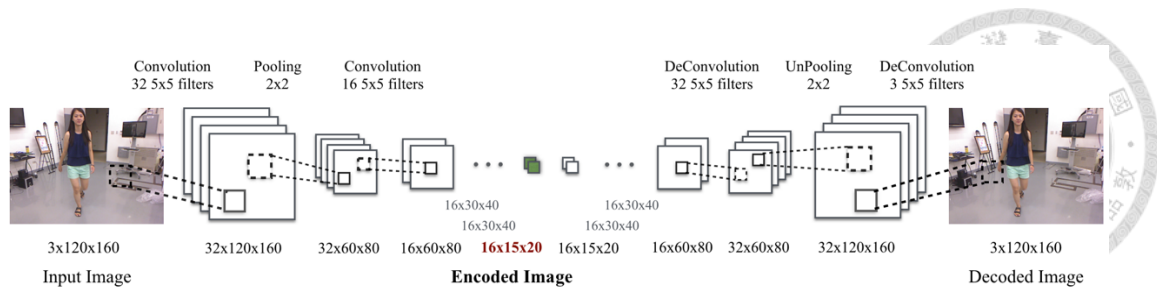
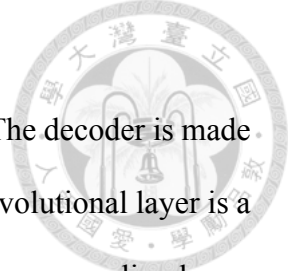


Figure 4-6 The proposed convolutional auto-encoder. In our image auto-encoder architecture, the encoder part consists of three convolution layers and three max-pooling layers in stack. The decoder part is composed of three deconvolution layers and three unpooling layers in stack.

For the purpose of resolving the computational cost and extracting meaningful features, we adopt convolutional auto-encoder to be descriptor extractor. In deep learning, auto-encoders are usually applied convolution neural networks as they are employed in image processing. CNNs auto-encoder is composed of an encoder and a decoder. The proposed convolutional auto-encoder is shown in Figure 4-6. The detail components to build an CNNs auto-encoder are elaborated as followed:

- Encoder

In our model, the encoder part is composed of convolutional layer and max-pooling layer in stack. Totally, there are three convolutional layers and three max-pooling layers. In a deep learning model, it makes sense to utilize convolutional layers in feature extraction. The role of max-pooling layers is employed for spatial down-sampling to satisfy our goal: dimensionality reduction for extracting meaningful descriptor. The activation function in convolutional layers, we choose, is Rectified Linear Unit (ReLU). In convolutional networks, it is more effectively to utilize ReLU than the widely used logistic sigmoid function.



- Decoder

The decoding function is in opposite site as encoding function. The decoder is made up with deconvolutional layer and up-sampling layer in stack. Deconvolutional layer is a reverse of convolutional layer, and up-sampling layer is a contrary to max-pooling layer. In practical, the deconvolutional layers is just as same as convolutional layer, hence the activation we choose is also ReLU.

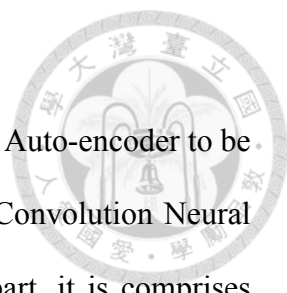
- Loss Function

A loss/cost function is a function that maps an event to a real number to represent some cost associated with the event. In an auto-encoder, as a self-supervised learning, we should define a loss/cost function to evaluate the model. The Loss function must be differentiable. In our proposed auto-encoder, the loss function represents the distance between the reconstructed image and the original input. Hence we define our loss function as Mean Square Error (MSE) as shown in equation (4.1). The MSE assesses the quality of a model. It is definitely non-zero, the value of MSE is better to be closer to zero.

In auto-encoder, \hat{y} is the original image for model trying to reconstruct. y means the reconstructed image made by auto-encoder. N is the total number of images in a batch size.

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{n=1}^N \|\hat{y}^n - y^n\|^2 \quad (4.1)$$

- Architecture built upon Keras



In order to extract meaningful features, we adopt Convolutional Auto-encoder to be descriptor extractor. In practice, auto-encoders are usually applied Convolution Neural Networks as they are employed in image processing. In encoder part, it is comprises convolutional layers and max- pooling layers in stack. The input raw images (3x120x160) are given to the first convolutional layer which convolves 32 filters of 5x5 with stride 1 followed by ReLU results in 32 feature maps and each of size is 120x160. Then feed the 32 feature maps into 2x2 max-pooling layer with the stride 1 for downsampling and the result become 32 feature maps with each size of 60x80 . Overall, the encoder part is three convolutional layers and max-pooling in stack. The last layer of encoder possesses 16 feature maps with size 15x20, and we call this layer encoded image which we take it as meaningful features. In decoder part, it is composed of deconvolutional layers and un-pooling in stack. The encoded images (16x15x20) is given to the first deconvolutional layer of decoder which convolves 16 filters of 5x5 filters with stride 1 followed by ReLU results in 16 feature maps and each of size is 15x20. Then feed the 16 feature maps into 2x2 un-pooling layer with stride 1 for upsampling and the result become 16 feature maps with each size of 30x40. Overall, the decoder part, the inverse of encoder part, is three deconvolutional layers and un-pooling in stack. The last layer of decoder is decoded image which possess the 3 feature maps with size 120x160 as same as original input image. The auto-encoder can learn automatically from data and the optimizer we chose is RMSprop. The auto-encoder system built upon Keras is present in Figure 4-7. We can see the detailed number of feature maps and corresponding feature size in the Keras structure.

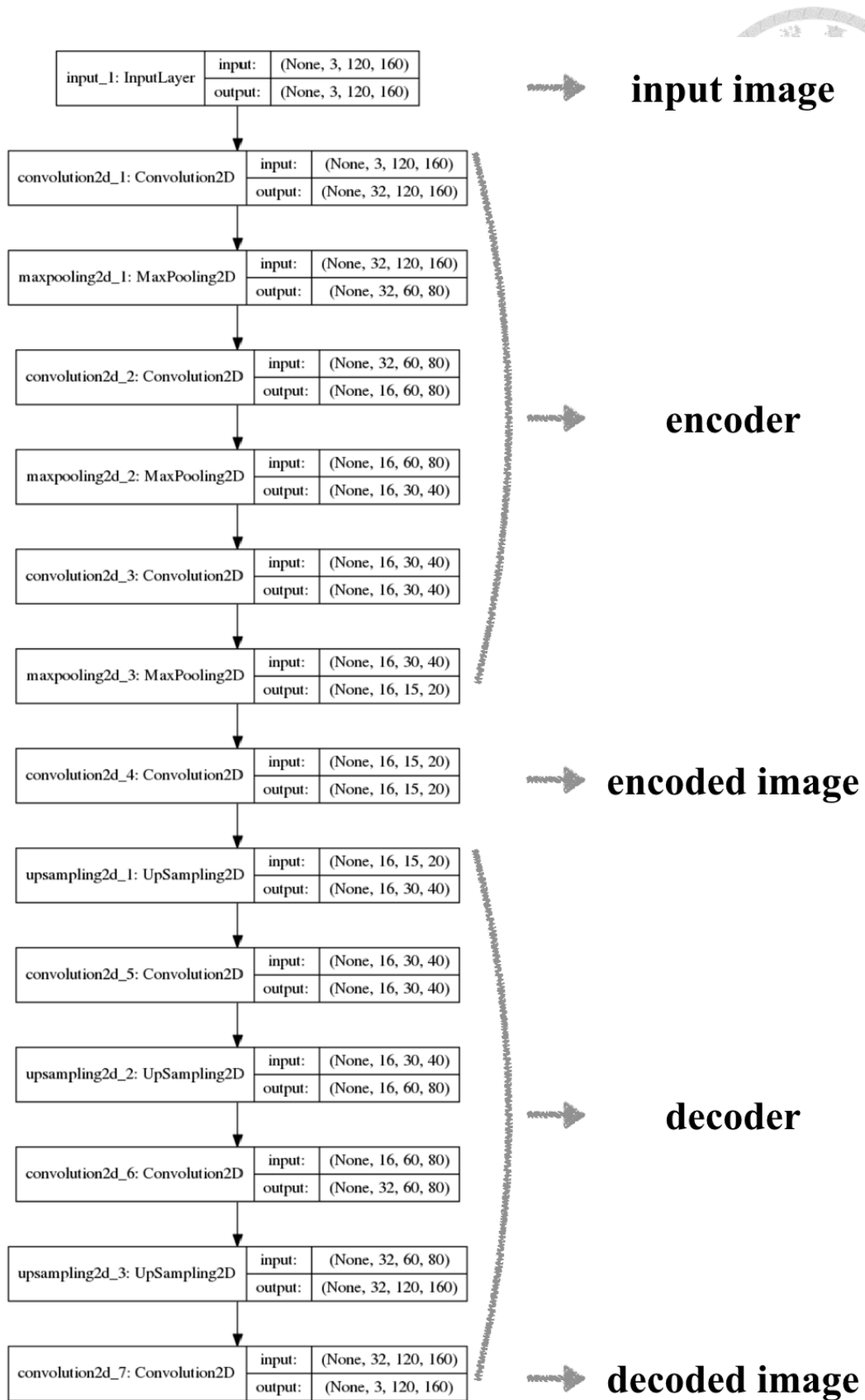


Figure 4-7 The architecture of CNNs auto-encoder built upon Keras.



- Visualization of Auto-encoder

The visualization of auto-encoder around encoded images and decoded images sample is presented in Figure 4-8. The first row is raw image, the second and third rows extract (1,3,5) and (2,4,6) filters of encoded image to be input of RGB value. The last row is decoded image. We would like to visualize the message hided in encoded images, and how the decoded images appearance. We may see that the decoded images possess the high similarity to the original images although there are some blurring due to the downsampling and upsampling. The results show that auto-encoder can project image into more meaningful low-dimension (i.e. encoded imaged), which we regard as features.

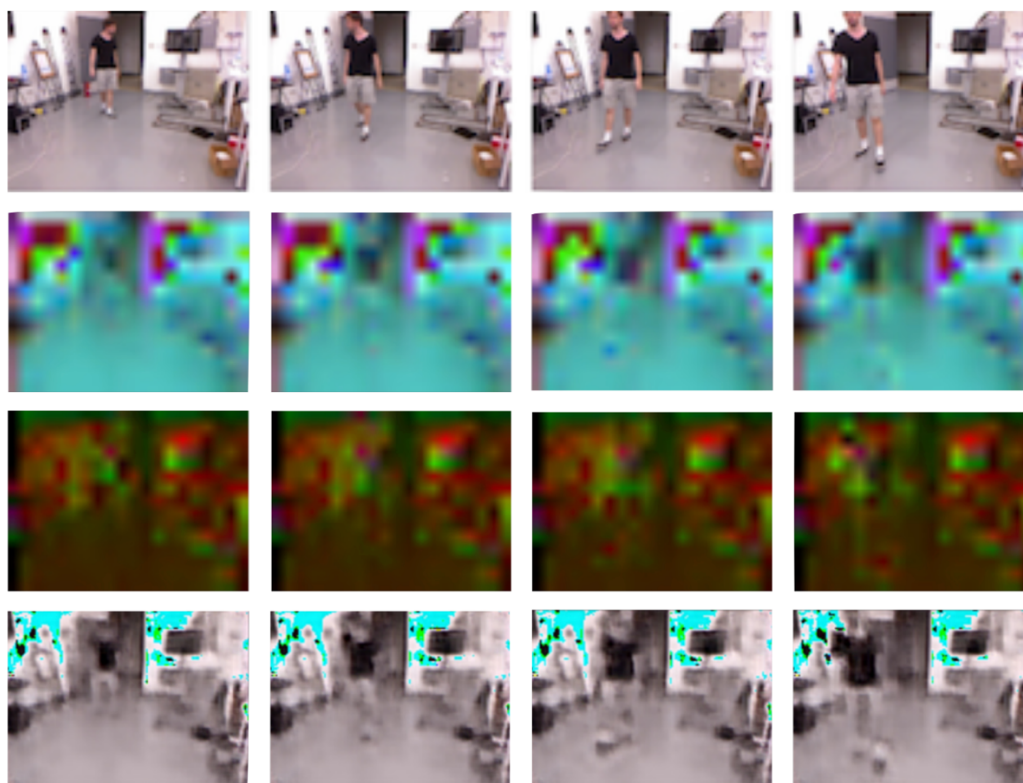
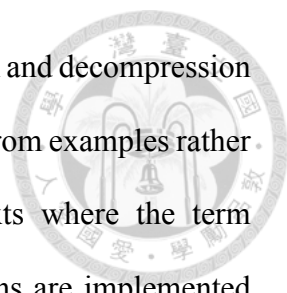


Figure 4-8 Visualization of autoencoder results.

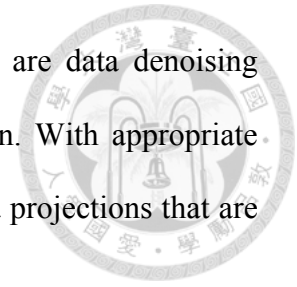


Autoencoding is a data compression algorithm where the compression and decompression functions are 1) data-specific, 2) lossy, and 3) learned automatically from examples rather than engineered by a human. Additionally, in almost all contexts where the term "autoencoder" is used, the compression and decompression functions are implemented with neural networks.

1. Autoencoders are data-specific, which means that they will only be able to compress data similar to what they have been trained on. An autoencoder trained on pictures of faces would do a rather poor job of compressing pictures of trees, because the features it would learn would be face-specific.
2. Autoencoders are lossy, which means that the decompressed outputs will be degraded compared to the original inputs (similar to MP3 or JPEG compression). This differs from lossless arithmetic compression.
3. Autoencoders are learned automatically from data examples, which is a useful property: it means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input. It doesn't require any new engineering, just appropriate training data.

To build an autoencoder, you need three things: an encoding function, a decoding function, and a distance function between the amount of information loss between the compressed representation of your data and the decompressed representation (i.e. a "loss" function). The encoder and decoder will be chosen to be parametric functions (typically neural networks), and to be differentiable with respect to the distance function, so the parameters of the encoding/decoding functions can be optimized to minimize the reconstruction loss, using Stochastic Gradient Descent.

Today two interesting practical applications of autoencoders are data denoising (which we feature later in this post), and dimensionality reduction. With appropriate dimensionality and sparsity constraints, autoencoders can learn data projections that are more interesting than PCA or other basic techniques.



One of the main reason why autoencoder have attracted so much research and attention is because they have long been thought to be a potential avenue for solving the problem of unsupervised learning, i.e. the learning of useful representations without the need for labels. Then again, autoencoders are not a true unsupervised learning technique (which would imply a different learning process altogether), they are a self-supervised technique, a specific instance of supervised learning where the targets are generated from the input data. In order to get self-supervised models to learn interesting features, you have to come up with an interesting synthetic target and loss function, and that's where problems arise: merely learning to reconstruct your input in minute detail might not be the right choice here. At this point there is significant evidence that focusing on the reconstruction of a picture at the pixel level, for instance, is not conducive to learning interesting, abstract features of the kind that label-supervised learning induces (where targets are fairly abstract concepts "invented" by humans such as "dog", "car"...). In fact, one may argue that the "best features" in this regard are those that are the worst at exact input reconstruction while achieving high performance on the main task that you are interested in (classification, localization, etc).



- Convolutional Layers

Convolutional layers (convolutional neural networks) are a category of feed-forward artificial neural network that have proven significantly effective in computer vision field such as image recognition and classification. CNNs have been successful in identifying faces, classifying facial expression, object recognition, etc. For example, CNNs are widely utilized in category classification [24] as shown in Figure 4-9. Today, convolutional neural networks are an essential tool for machine learning implementation.

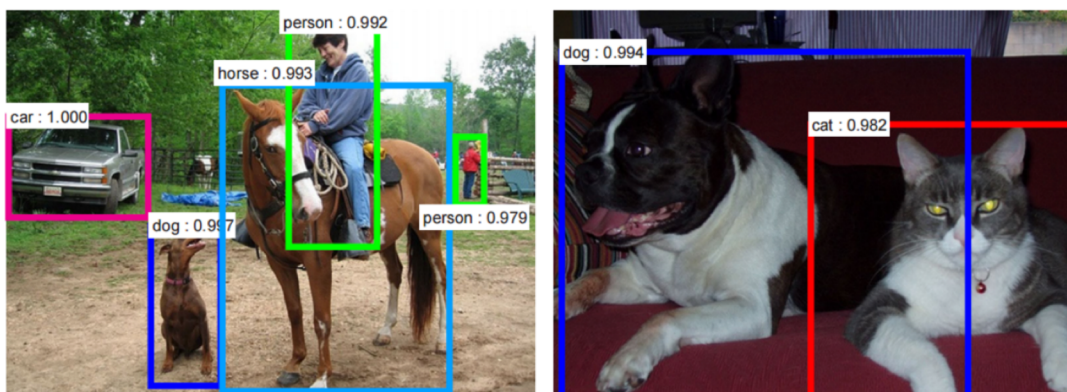
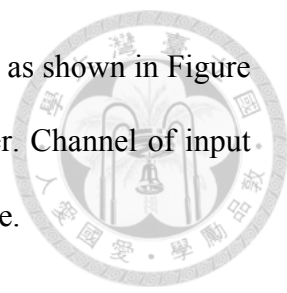


Figure 4-9 An implementation of CNNs in category classification.

The most function in CNNs is the convolution operator. The prime purpose of CNNs is to extract features from the input images. There are some terminologies in CNNs should be introduced first to elaborate convolution operator. Channel: A standard image has three channels-red, green, blue. A grayscale image has only one channel. Filter: also called kernel is a matrix to extract feature from image. Each kind of filter can extract different kind of feature while taking a same image input. For example, edge detection, sharpening, and blurring. Stride: is the number of pixels for our filter to slide over the input image. Zero-padding: pad the input image with zero around border mainly for the purpose of making feature map as same size as input image.



There is an example for demonstrating the convolution operator as shown in Figure 4-10. A see a 5x5 matrix as an input image, a 3x3 matrix as a filter. Channel of input image is one, stride of convolution step is 1, and zero-padding is none.

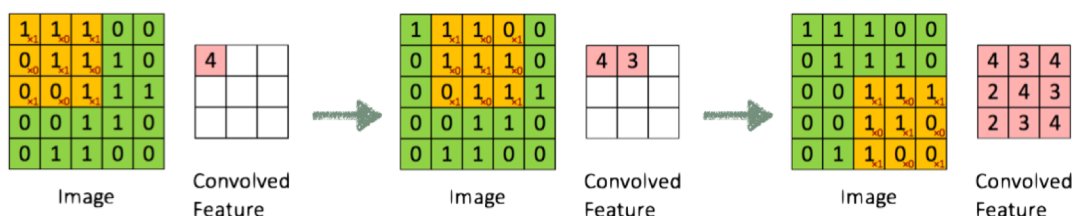


Figure 4-10 The demonstration of convolution operator.

- Pooling Layers

The pooling layers (also called subsampling) are designed for the dimensionality reduction of each feature map and retain the most important information in the windows. There are vary kind of pooling such as max-pooling, average-pooling, sum-pooling, etc. Take max-pooling as shown in Figure 4-11, we utilize in our proposed deep auto-encoder, for example. We can define a neighbor relationship between pixels, defined as 2x2 window, and extract max element from this window to be our most important information.

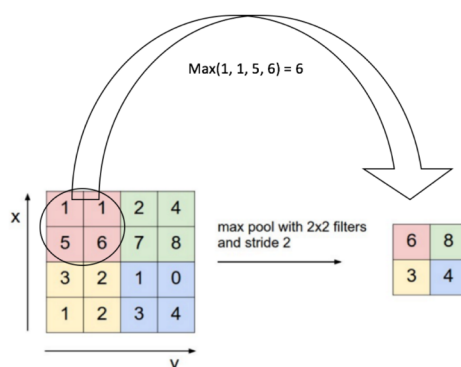
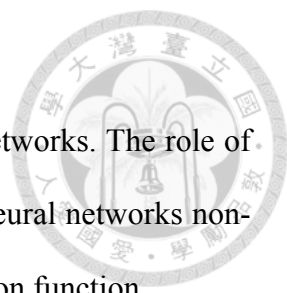


Figure 4-11 A demonstration of max pooling.



- Activation Function

There are vary activation function applies in artificial neural networks. The role of an activation function [25], as shown in Figure 4-12, is to make a neural networks non-linear. In this part we would like to introduce some common activation function.

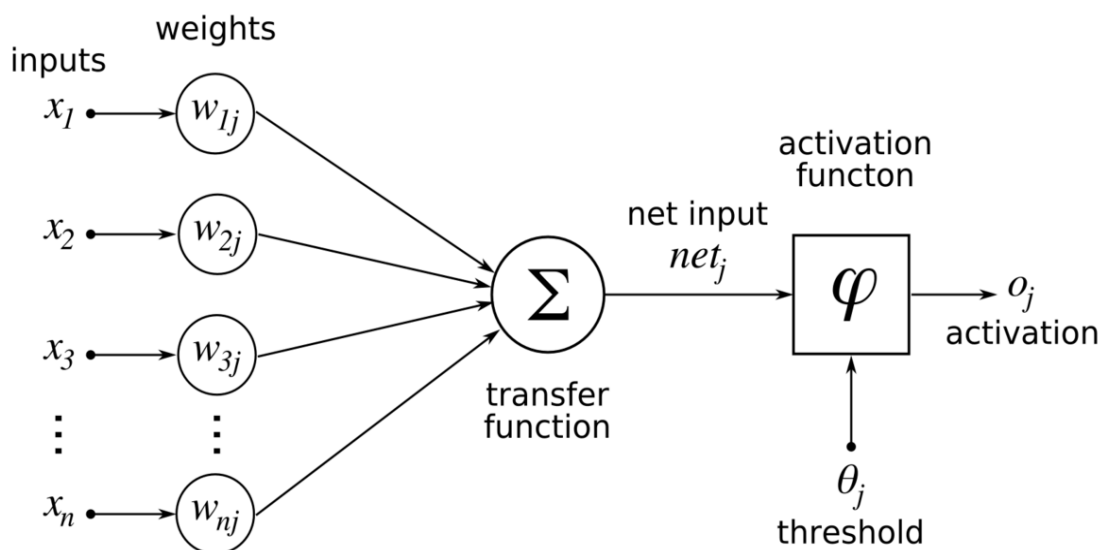


Figure 4-12 The role of an activation function in a neuron cell.

Linear

The linear activation function mathematical expression is shown in equation (4.2)

and the the mathematical image is shown in Figure 4-13.

$$f(x) = x \tag{ 4.2 }$$

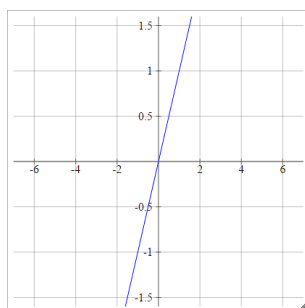
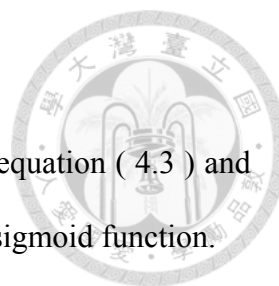


Figure 4-13 Presentation of linear function.



Sigmoid

The sigmoid activation function mathematical expression is in equation (4.3) and the the mathematical image is shown in Figure 4-14 Presentation of sigmoid function.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$

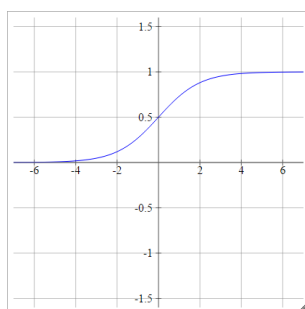


Figure 4-14 Presentation of sigmoid function.

Tanh

The linear activation function mathematical expression is in equation (4.4) and the the mathematical image is shown in Figure 4-15.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.4)$$

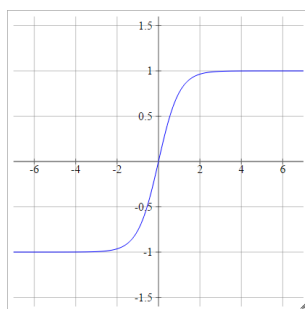
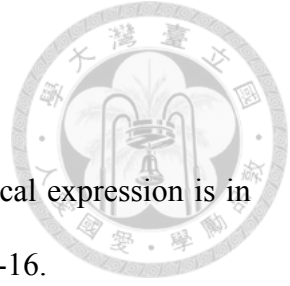


Figure 4-15 Presentation of tanh function.



ReLU

The rectified linear unit (ReLU) activation function mathematical expression is in equation (4.5) and the the mathematical image is shown in Figure 4-16.

$$f(x) = \max (0, x) \quad (4.5)$$

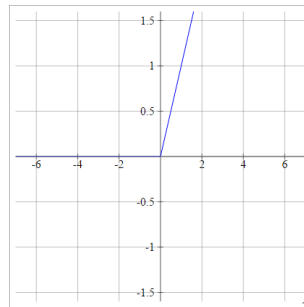


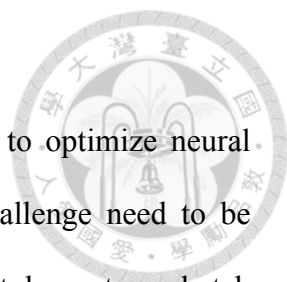
Figure 4-16 Presentation of ReLU function.

Softmax

The softmax activation function mathematical expression is shown in equation (4.6). The softmax activation function is also called normalized exponential. This activation is usually utilized in the output layer, due to it's value is depend on the other neurons of that layers. Usually, softmax activation us employed in categorical classification to represent the probability, since the sum of the value of output neurons would be 1.

$$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} \quad \forall j \in 1, \dots, K \quad (4.6)$$

Where K is the total number neurons of such layers.



- Gradient Descent Optimizer

Gradient descent [26] is the most common and popular way to optimize neural networks. There are three variant gradient descents and many challenge need to be resolved to optimize a neural network model. Three types of gradient descents are batch gradient descent, stochastic gradient descent (SGD), and mini-batch gradient descent. It depends on how many samples would be seen for a model to perform a parameter update. Stochastic is to do a parameter update upon a sample. Batch is to do update upon whole dataset. Mini-batch is between these two types. There are some challenges should be solved to execute a better optimization, such as choosing a proper learning rate is difficult, preventing stock on a saddle point is notoriously hard for SGD, etc. In this part we would like to introduce some methodologies help us to overcome aforementioned obstacles.

Momentum

Stochastic gradient descent is hard to convergence in the case of a two dimensions curve with the property which one dimension is more steeply than another. Hence if the momentum is utilized in SGD would help converge faster. The demonstration is shown in Figure 4-17.

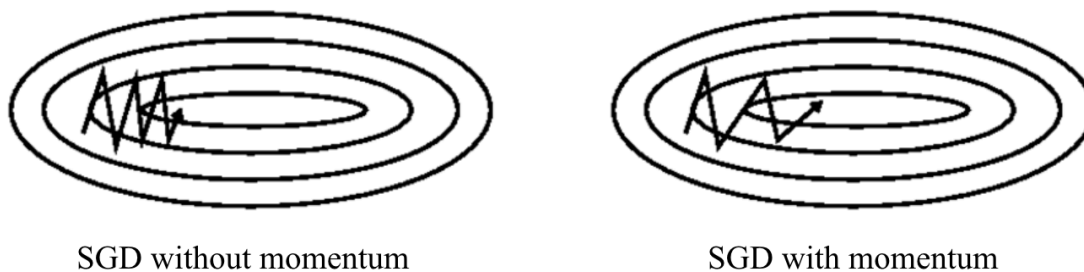
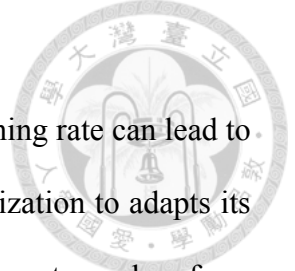


Figure 4-17 The presentation of SGD with/without momentum.



Adagrad

Choosing an appropriate learning rate is difficult. A proper learning rate can lead to a better result. The Adagrad is a algorithm for gradient-based optimization to adapts its learning rate. Adagrad perform larger update for the infrequently parameter and perform smaller update for the frequent parameter. Adagrad parameter update is shown in equation (4.7).

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i} \quad (4.7)$$

where $g_{t,i}$ is the gradient of the cost function w.r.t. the parameter θ_i at time the time step t . $G_t \in R^{d \times d}$ is a diagonal matrix with each diagonal element i , i is the sum of square of gradients with respect to θ_i and ϵ is a small number (usually 1×10^{-8}) to avoid division by zero.

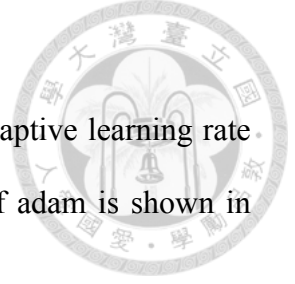
RMSprop

RMSProp is one of Adagrad deformation to resolve the adagrad radical diminish gradient problem. The adagrad radical diminish problem is stem from the accumulation sum of gradients. This cause the learning rate to shrink and become infinites small. The RMSprop mathematical expression is shown in equation (4.8)- (4.9).

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2 \quad (4.8)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (4.9)$$

Where γ is a constant parameter (usually set as 0.9) to define the past or current gradient which one is important.



Adam

Adam is one the gradient descent optimizer, which take the adaptive learning rate and momentum into consideration. The mathematical expression of adam is shown in equation (4.10) - (4.14).

$$m_t = B_1 m_{t-1} + (1 - B_1) g_t \quad (4.10)$$

$$v_t = B_2 v_{t-1} + (1 - B_2) g_t^2 \quad (4.11)$$

$$\hat{m}_t = m_t / (1 - B_1^t) \quad (4.12)$$

$$\hat{v}_t = v_t / (1 - B_2^t) \quad (4.13)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (4.14)$$

Where B_1, B_2 and ϵ are constant and also default as 0.9, 0.999 and 1×10^{-8} , respectively.

Early Stopping

In order to training neural networks for retrieving better performance, numerous decisions need to be made regarding the settings (parameters) utilized. Once the parameter is the training epochs: that is, how many turns the full data set (epochs) are been seen for training. Once the fewer epochs we use, the model would fall into underfitting (i.e. unable to learn everything from training set). If too epochs we utilize, the model would become overfitting (i.e. learn training set too detailed so the ‘noise’ of training set is also considered, however can’t use in general case and would conduct poor performance on testing set). Early stopping attempt to set this value automatically to prevent model from overfitting. The early stop epoch picture [27] is shown in Figure 4-18.

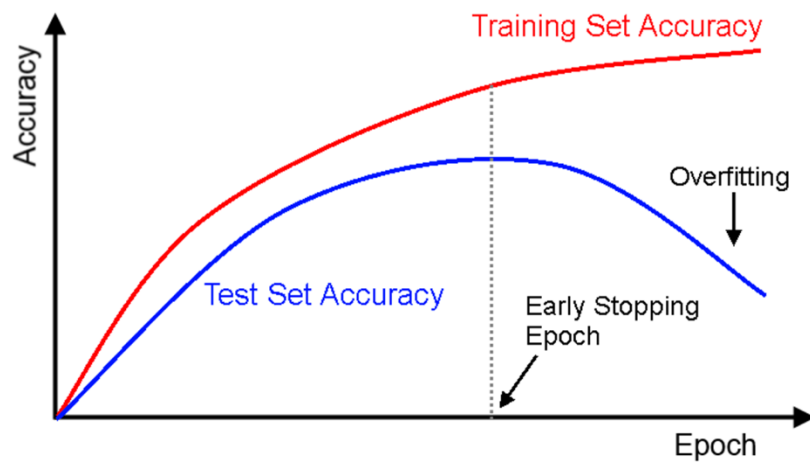


Figure 4-18 The early stopping epoch to prevent overfitting.

Dropout

To overcome the overfitting: that is, the model only learns training set to classifier and adapt itself to the training example instead of learning decision capable of classifying generic instances. The Dropout, as shown in Figure 4-19, is design to resolve overfitting by ensemble the deep learning models.

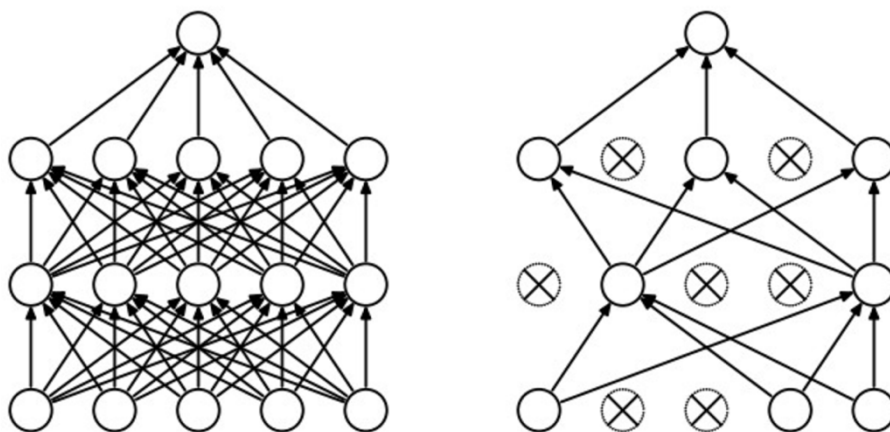


Figure 4-19 Left one is standard NNs, Right one is after applying dropout NNs.

4.4 CLASSIFIER IMPLEMENTATION



4.4.1 Deep Learning Based Classifiers

To analyze features in sequence, we apply Long Short Term Memory (LSTM), one type of recurrent neural networks, in our research.

- LSTM-based Classifier

Recurrent Neural Networks (RNNs) is the most renowned model for sequence learning [28] [29] [30]. Different from conventional feedforward neural networks, RNNs have cyclic connections making them be able to map from entire history of previous inputs to target vectors. In supervised learning, RNNs can be trained via Backpropagation Through Time (BPTT) with sequential input data and output target. However, the gradient exploding or vanishing problems during BPTT of model training obstruct the performance of RNNs. The phenomenon of gradient exploding or vanishing would cause the LSTM-based deep learning model hard to converge and perform poorly. It implies basic RNNs may not handle long term dependencies [31]. Consequently, Long Short-Term Memory networks (LSTMs) is an architecture which is proposed to prevent these problems [32]. One LSTM cell is composed of three gates and cell state, the equations of LSTM cell are shown in equations (4.15) - (4.19).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4.15)$$

$$f_t = \sigma(W_{xf} + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4.16)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4.17)$$

$$o_t = \sigma(W_{x_o}x_t + W_{h_o}h_{t-1} + W_{c_o}c_t + b_o) \quad (4.18)$$

$$h_t = o_t \tanh(c_t) \quad (4.19)$$

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{n=1}^N [y^n \log \hat{y}^n + (1 - y^n) \log(1 - \hat{y}^n)] \quad (4.20)$$



Where σ is the logistic sigmoid activation function, and i , f , o and c are the input gate, forget gate, output gate and cell state, respectively. In LSTMs, three gates are a way to optionally let information through. The input gate is to determine which new information is going to be stored in the cell state. The ratio of input is calculated in equation (4.15) and affect on the equation (4.17). The forget gate is to decide which information is going to be discarded from the cell state. The ratio of the previous memory is shown in the equation (4.16) and utilized in the equation (4.17). The output gate is to determine whether passing the output of memory cell or not. The equation (4.19) shows this process. By using LSTMs, the gradient vanishing and exploding problems would be resolved due to the three gates. That is the reason why we adopt LSTM- RNN architecture to learn our sequential data. Furthermore, Binary Cross-Entropy (CE) function, as shown in equation (4.20), holds sharper loss property for not falling into local minimum with relative to root mean square function. Thus we adopt it as loss function in the proposed LSTM-based classifier, as shown in Figure 4-20. LSTM-RNN classifier comprises a LSTM layer, with 32 neurons, followed by a Multiple Layer Perceptron (MLP) which is composed of three layers in each filled with 128, 256, 2 neurons, respectively. The activation function in LSTM layer is sigmoid function, in MLP layer is ReLU function and in the last layer is to be softmax function. The detailed architecture of LSTM-based classifier implemented upon Keras is shown in Figure 4-21.

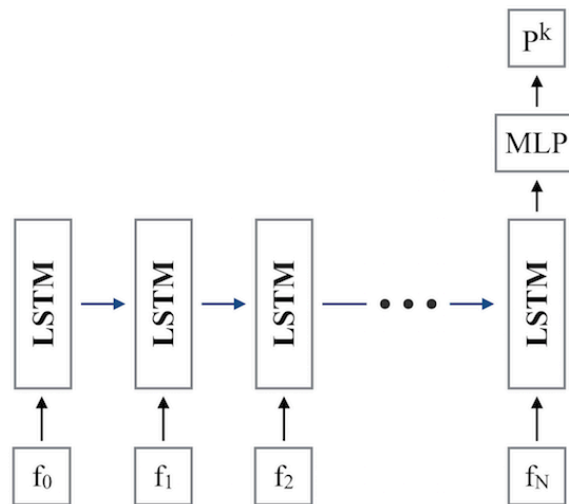


Figure 4-20 The proposed LSTM-based classifier architecture. Where $f^k = (f_0, f_1, \dots, f_N)$ and p^k denote as sequence features and probability prediction of k-th observation to learning and predicting human's mentation by analyzing features, in sequence.

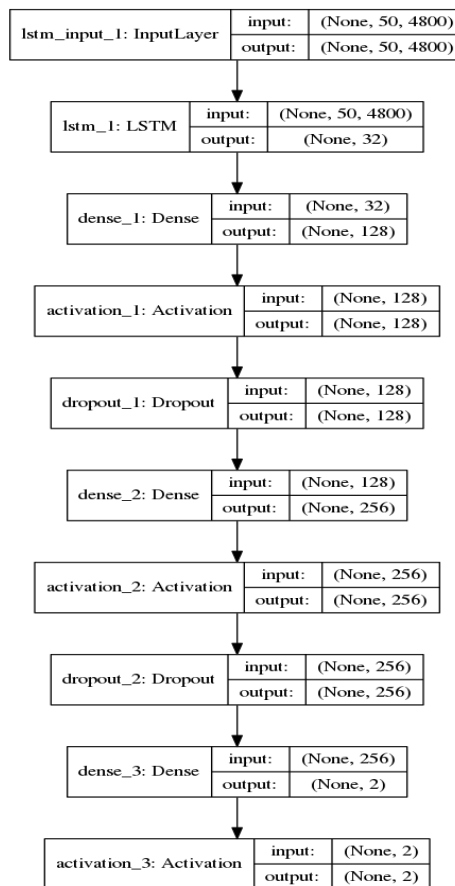
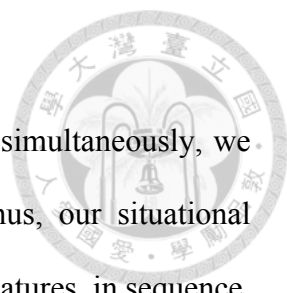


Figure 4-21 The architecture of LSTM-based classifier implemented on Keras.



- CNN Followed by LSTM Classifier

In order to take spatial and temporal factor into consideration simultaneously, we propose a convolutional neural networks followed by LSTM. Thus, our situational context perception can perceive a person's mentation by analyzing features, in sequence. In Figure 4-22, demonstrates the deep learning model we proposed which is a convolution neural networks followed by a recurrent neural networks. In terms of convolution neural networks we also employ convolution layer and max-pooling layer then flatten as one dimension to fit LSTM input. Last step is classified via MLP to predict whether the person needing assistances. The activation function in Convolutional Net and MLP is ReLU, in LSTM is sigmoid. The cost function is binary cross-entropy. The detailed architecture built upon Keras is shown in Figure 4-23.

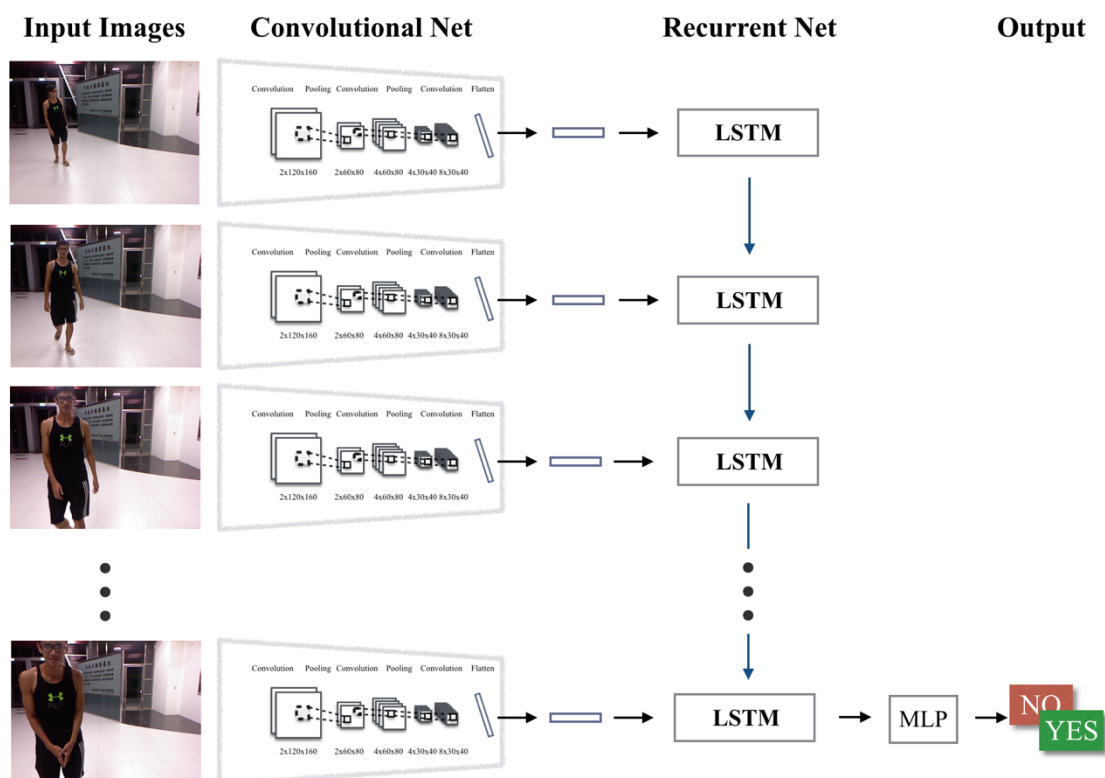


Figure 4-22 Present the proposed CNNs followed by LSTM architecture

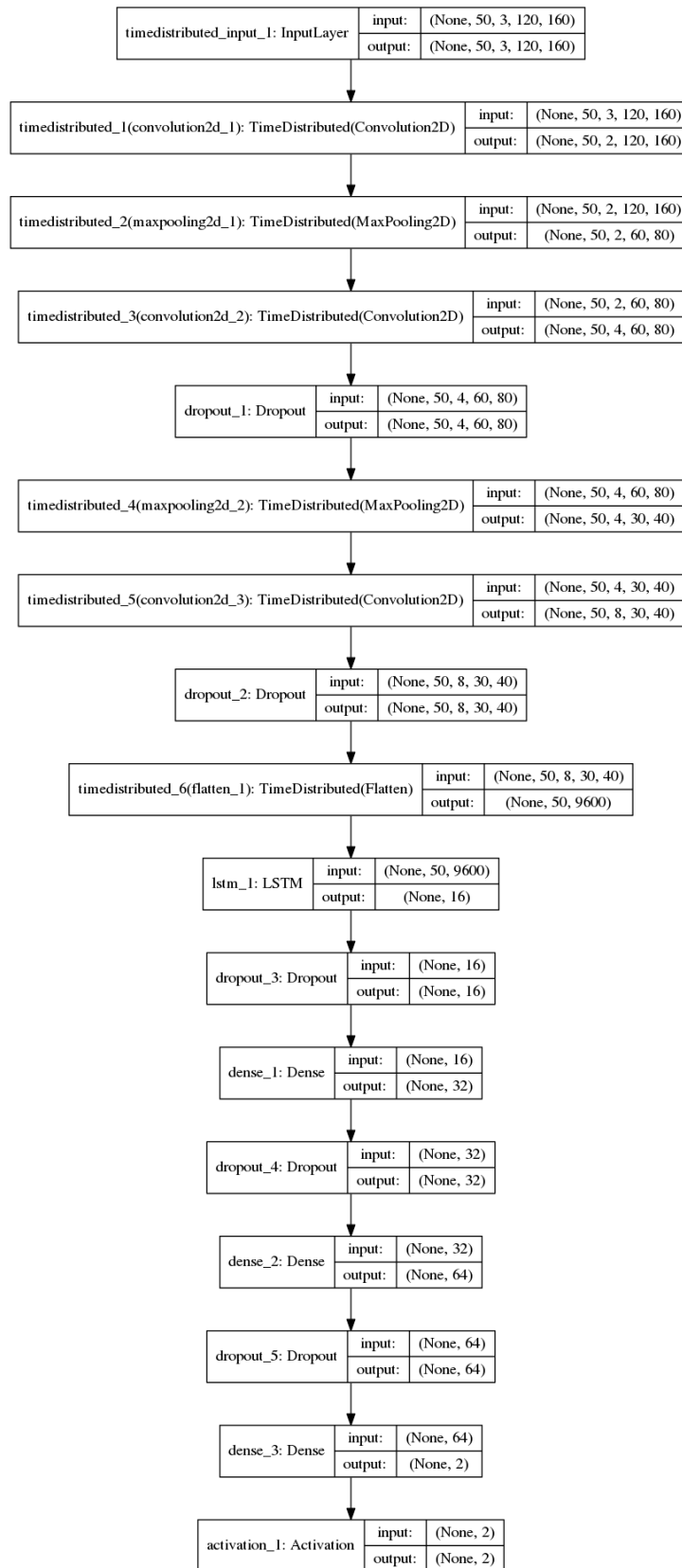


Figure 4-23 The CNNs followed by LSTM architecture implemented on Keras.

4.4.2 Conventional Classifiers



- Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning model. SVM is usually utilized in classification [33], as shown in Figure 4-24, and regression analysis. The SVM classifier is well-known for the pedestrian detection with HOG features.

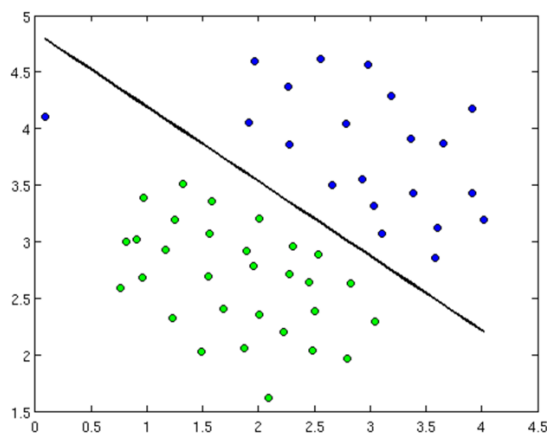


Figure 4-24 Linear SVM classification. Blue points are one class and green points are another class. SVM tries to classify two class.

- Gaussian Naïve Bayes

Naïve Bayes is one kind of simple probabilistic classifiers in machine learning which is based on employing Bayes' theorem with the assumptions of independent between features. As processing continuous data, it is classical assumption to regard the continuous value associated with each class are Gaussian distribution. Gaussian Naive Bayes is chosen in our research, since it's simple property and has advantage for performing on small training set.

4.5 SOCIAL CO-ROBOT VERSUS PEOPLE in SERVICE INDUSTRY



In this section, we would like to analysis how the similarity between prediction of social robot and decision made by people in service industry. Therefore, we make the deep learning model (CNNs followed by LSTM) run online, thus robot can observe a person's behaviors and predict him/her mentation in real-time. The overall scenario of interaction is shown in Figure 4-25.

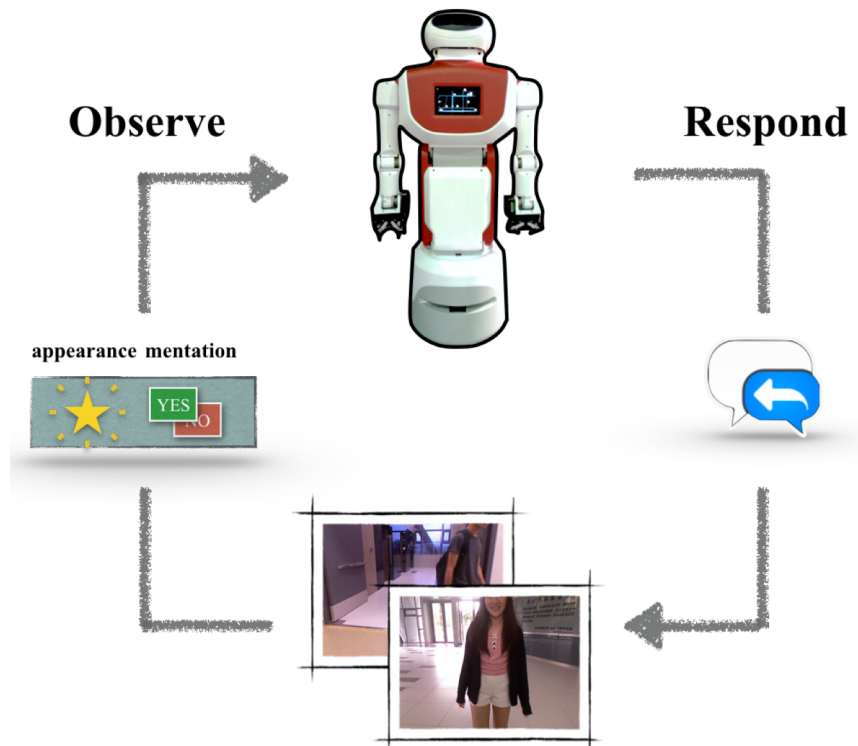


Figure 4-25 The situational context perception apply upon Human-robot interaction.

As a person appear, robot begins to greeting at first and start to observe the person's behavior around five seconds. Then, robot will respond appropriately with respect to prediction of human's mentation. If only if prediction of human's mentation is True (i.e. people may need some help), robot will take the initiative to polite say "may I help you?". Otherwise, robot will keep silence to prevent bother the person.

Eventually, there are 29 human-robot interaction are observed in total. Then, these observations are also evaluated by 17 service workers which include employee of Ding Tai Feng, chef and employee in restaurant, restaurant owner, intern in hospital, etc. The service employees are asked observe the person's behavior at our social co-robot's point of view and answer one question which is shown in Figure 4-26 for each observation. The final decision whether people will courteous ask "may I help you?" in each observation are made by voting. We make people serve in Ding Tai Feng has three votes each person, and the others has one vote in each.

From your personal experience, will you begin to say "May I help you?" in this context.

Yes, I will. No, I won't.

Figure 4-26 The question is answered by people after observing the person's behavior.

Chapter 5 EXPERIMENTAL RESULTS



In chapter 5, we will show some experimental results which divided into two parts. In the first part, the evaluation of proposed deep learning models by 5-fold cross-validation is present. The features comparison, classifiers appropriateness, Multi-feature fusion, and deep learning models comparison (Mentioned in Section 4.1-4.4) are elaborated. In the second part, the evaluation of situational context perception is exhibited to analyze the similarity between prediction of social co-robot and decision made by people in service industry (Mentioned in Section 4.5).

5.1 DEEP LEARNING MODELS EVALUATION

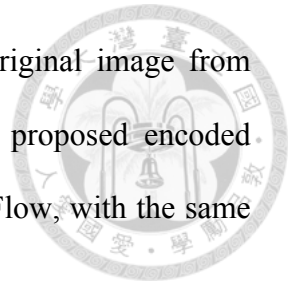
5.1.1 K-fold Cross-Validation

K-fold cross-validation is to partition labeled data into K equal size subsamples (folds). Among the K folds, a single fold would be left as the testing set and the retained K-1 folds are utilized as training set. The process of cross-validation is then repeat k times with each k fold is used exactly once as the testing set. The K results can be averaged to produce a single estimation.

5.1.2 Features Comparison

We utilize CNN auto-encoder to extract feature from raw image and employed encoded image as feature. Employing encoded image provides two advantages. One is that it reduces the dimension of raw image which dimension is 57600 ($3 \times 120 \times 160$) via encoder the image turns into a more meaningful space which holds only 4800 dimension.

The other is that encoded image has potentiality to reconstruct original image from relative reduced dimension. In validation stage, we compare the proposed encoded images to two other kinds of handcraft features, HOG and Optical Flow, with the same LSTM classifier.

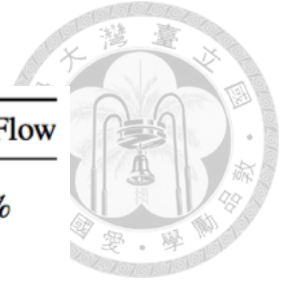


- Handcraft Feature Selection

Regards to the situational context. We consider that human body language would be the key factor in this scenario. Thus, HOG and Optical Flow are employed as our handcraft features for comparison. HOG feature is commonly applied in human detection and raise well results especially in pedestrian detection [34]. And the advantage of Optical Flow feature is that it takes Spatio-Temporal factor into account and could be utilized in transition people motion analysis [35]. As a result of variation of dimension across features. Principal Component Analysis (PCA) is employed for reducing both HOG and Optical Flow feature to the dimension of 4800 to be as same as encoded image.

- Results and Discussion

The LSTM architecture is employed as classifier and we apply some training strategy as follows: epoch 30, batch size 8, initial learning rate 0.001, gradient descent optimization algorithms Adam [36] which contains the concept of adapting learning rate and momentum. The initial parameters are set as he_normal distribution which is normal distribution centered on 0 with standard deviation equals to square root of $(2/fan_in)$ where fan_in is the input units in the weight tensor. Dropout is employed to prevent overfitting. The experimental results are shown in Table 5-1. There are two interesting aspects to be discussed. First, the three proposed features yield significantly high accuracy



	Encoded Image	HOG	Optical Flow
Training Accuracy	86%	99.37%	97.5%
Testing Accuracy	73%	69%	63%

Table 5-1 Results from features comparison by 5-fold cross-validation, applying LSTM architecture to learn from different features.

on training set. It represents that our hypothesis is truthful, the human body language plays a key role to our target, perceiving the needing assistance for providing heartwarming services. Second, due to the variation among people, there must exist noise from person to person. It may easily raise poor performance on testing set. However, encoded image, HOG feature, and Optical Flow feature yield 73%, 69% and 63% accuracy (chance = 50%), respectively. That means our LSTM-based classifier successfully learned variety of body language from sequential data. Next step, we would like to examine that the sequence characteristic is really crucial or not.

5.1.3 Classifier Appropriateness

In this experiment, we would like to figure out whether take the sequential information into consideration would perform better. Thus, we compare our LSTM-based classifier to two prevalent classifiers, SVM and Naive Bayes classifiers. In each observation, the input dimension of LSTM-based classifier would be 50×4800 (a sequence of 50 keyframes, 4800 feature dimension) and in SVM and Naive Bayes dimension would be 24000.

	LSTM-RNN (Our Approach)			SVM			Gaussian Naive Bayes		
	Encoded Image	HOG	Optical Flow	Encoded Image	HOG	Optical Flow	Encoded Image	HOG	Optical Flow
Training Accuracy	86%	99.37%	97.5%	53.38%	54.25%	95.13%	98.63%	99.88%	68.25%
Testing Accuracy	73%	69%	63%	49.5%	49.5%	66%	64%	60%	57.5%

Table 5-2 Results from Experiment of classifier appropriateness, applying LSTM-RNN architecture, SVM and Gaussian Naive Bayes to classify needing assistance from aforementioned features.

- Classifier Selection

Gaussian Naive Bayes is chosen, since it's simple property and has advantage for performing on small training set. And the reason why we chose SVM is that it yields high accuracy, nice theoretical guarantees regarding overfitting and often used in people detection especially detection of pedestrians.

- Results and Discussion

In these experiments, LSTM-based, SVM, and Gaussian Naive Bayes classifiers are employed on three kinds of features as shown in Table 5-2. Based on the results in previous experiment, the discussion here mainly focuses on testing accuracy to evaluate the classifier performance. By using encoded image and HOG as our features, we could see that LSTM-based classifier perform more outstanding than the other two classifiers. On the encoded image aspect, LSTM-based classifier raises more 23.5% and 7% accuracy compared to SVM and Gaussian Naive Bayes classifiers, respectively. On the HOG aspect, LSTM-based raises more 19.5% and 9% accuracy relative to SVM and Gaussian Naive Bayes classifiers, respectively. However, at the view of Optical Flow feature, LSTM drops a little in terms of accuracy less than SVM with about 3%. From our perspective, Optical Flow feature already takes the transition of two image into consideration. Hence, this feature is just a little suitable on SVM classifier. The results

	LSTM-RNN (Our Approach)			SVM			Gaussian Naive Bayes		
	Encoded Image + HOG	Encoded Image + Optical Flow	Optical Flow + HOG	Encoded Image + HOG	Encoded Image + Optical Flow	Optical Flow + HOG	Encoded Image + HOG	Encoded Image + Optical Flow	Optical Flow + HOG
Training Accuracy	94.75%	88.74%	98.12%	53.25%	91%	91.125%	96.625%	84%	74.375%
Testing Accuracy	75%	66.5%	64%	45.5%	68.5%	67.5%	64%	63%	60%

Table 5-3 Results from multi-feature fusion, applying LSTM-RNN architecture, SVM and Gaussian Naive Bayes to classifier needing assistances from concatenated two kinds of aforementioned features.

shown in this experiment meets our hypothesis that the perception of needing assistance is not an impulse trigger, however a sequence of features will considerably raise accuracy.

5.1.4 Multi-feature Fusion

In Experiment of multi-features fusion, we would like to acknowledge whether concatenated two kinds of aforementioned features may achieve better performance in each classifier. The results are shown in Table 5-3.

- Results and Discussion

Previous experimental results show that encoded image and HOG are presented better performance on LSTM-based classifier and Optical Flow is shown to be a little suitable by utilizing SVM classifier. The results in Experiment 3 show that encoded image + HOG via LSTM-based classifier enhance 2% accuracy, comparison to encoded image only. In terms of Optical flow, the accuracy slightly raises 2.5% and 1.5% via SVM by concatenating with encoded image and HOG, respectively. From our perspective, we take these three kinds of features represent as human body language, thus concatenated these features may have a limit benefit on accuracy. To us mind, next time we would like to take another kind of feature into consideration such as facial expression, maybe it would enhance our performance by fusing human body language with facial expression.

5.1.5 Deep Learning Models Comparison



	Raw Image	HOG	Optical Flow
Training Accuracy	50.5%	48.0%	92.13%
Testing Accuracy	46%	47.5%	78%

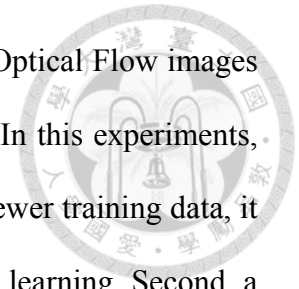
Table 5-4 Results for perceiving a person’s mentation by 5-fold cross-validation, applying CNNs followed by LSTM architecture to learn from different features.

In previous experiments, we learn spatial and temporal factors separately, which means we extract spatial features then apply LSTMs to learn in sequence. In this experiment, we would like to make deep learning model automatically learn from spatio-temporal feature in one model. The proposed CNNs followed by LSTM architecture as shown in Figure 4-22 may potentially keep the capability to uplift the accuracy. Therefore, the enhanced learning architecture is fed with raw images, HOG images, and Optical Flow images as input. The experimental results are shown in Table 5-4.

- Results and Discussion

In order to overcome computational cost, we reduce largely the numbers of filter in each convolutional layers. To us surprise, the raw images and HOG images input yields poor accuracy, it even learns nothing. from our perspective, there are two reason, one is that these two kinds of input may contains too much noise, we prefer to determine the person needing assistances via analyzing his/her sequential behaviors. Another is that we possess fewer observations for training model, it may have not enough data to learn such complex model. However, if we apply some preprocess step, such as extract Optical Flow feature beforehand, the training accuracy significant raise to 92.13%. It means model truly

learn something. In terms of testing accuracy, we may see that the Optical Flow images can reach 78% accuracy which beats aforementioned experiments. In this experiments, we come up with two conclusions. First, we know that if there are fewer training data, it may be a good idea to exact some simple handcraft features before learning. Second, a deep learning model which is composed CNNs followed by LSTMs contain more potentiality than two separate learning models.



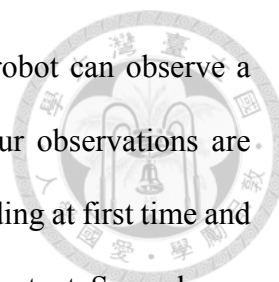
5.2 SITUATIONAL CONTEXT PERCEPTION

EVALUATION



Figure 5-1 Demonstrate two sequential human behaviors, which are observed by robot. Both social co-robot and people work in service industry evaluate the mentation of people via these data.

In this experiment, we would like to analysis how the similarity between prediction of social robot and decision made by people in service industry. Therefore, we make the

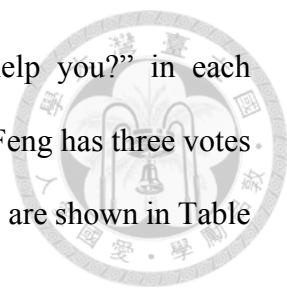


deep learning model (CNNs followed by LSTM) run online, thus robot can observe a person's behaviors and predict him/her mentation in real-time. Four observations are shown in Figure 5-1. First row: the woman come to our research building at first time and it is an appropriate behavior of robot to ask "may I help you" in this context. Second row: the man just pass by aisle and doesn't need any help. Third row: We go to W Hotel to gather situational context of hotel. The man come for booking and needs some help. Fourth row: The situational context occurs at Taipei main station. The man is hand around and does not need any assistance.

	Social Co-Robot	People in Service Industry
Accuracy (%)	91.67%	91.67%
TP rate (%)	52.08%	50%
FP rate (%)	8.33%	6.25%
TN rate (%)	39.59%	41.67%
FN rate (%)	0%	2.08%

Table 5-5 This table shows the accuracy of robot's prediction and decision made by voting among people in service industry with respect to ground truth.

The scenario is that as a person appear, robot begins to greeting at first and start to observe the person's behavior around five seconds. Then, robot will respond appropriately with respect to prediction of human's mentation. If only if prediction of human's mentation is True (i.e. people may need some help), robot will take the initiative to polite say "may I help you?". Otherwise, robot will keep silence to prevent bother the person. Eventually, there are 48 human-robot interaction are observed in total. Then, these observations are also evaluated by 17 service workers which include employee of Ding Tai Feng, chef and employee in restaurant, restaurant owner, intern in hospital, etc. The service employees are asked observe at robot's point of view and answer one question for each observation.



The final decision whether people will courteous ask “may I help you?” in each observation are made by voting. We make people serve in Ding Tai Feng has three votes each person, and the others has one vote in each. The accuracy results are shown in Table 5-5.

5.2.1 Results and Discussion

In accordance with experimental results, it is very exciting that both social robot and people in service industry obtain over 90% accuracy.

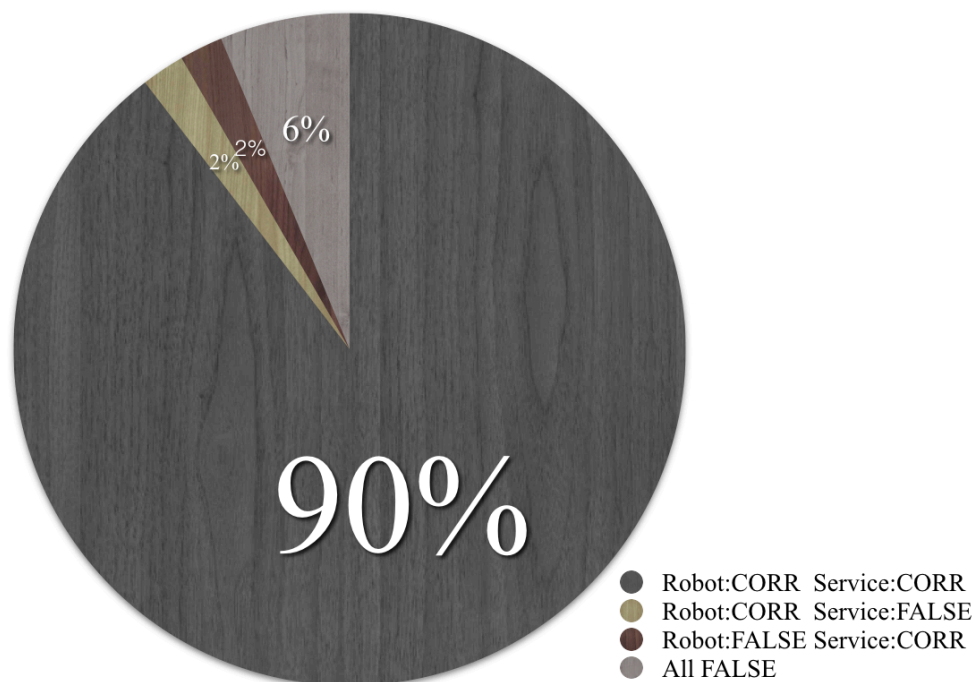


Figure 5-2 This figure shows the distribution of robot’s prediction and decision made by people in service industry with respect to ground truth.

According to this interesting results, we would like to go deeper to figure out where are the errors come from. The distribution among prediction of robot, decision made by people and ground truth is demonstrated in Figure 5-2. There are 90% consistency among robot, people and ground truth. The second part occupies around 6% demonstrate that

robot and people are consistent, however the answer is not fit with ground truth. The 2% shows that robot's prediction is true, but people's decision is false with respect to ground truth. The another last 2% shows that robot's prediction is false, but the people's decision is true with respect to ground truth. Three case of non-consistent among robot, people and ground truth are shown in Figure 5-3. In the 6% case, ground truth is False (i.e. no need help), but the answer made by both robot and human is True (i.e determine the person may need help).

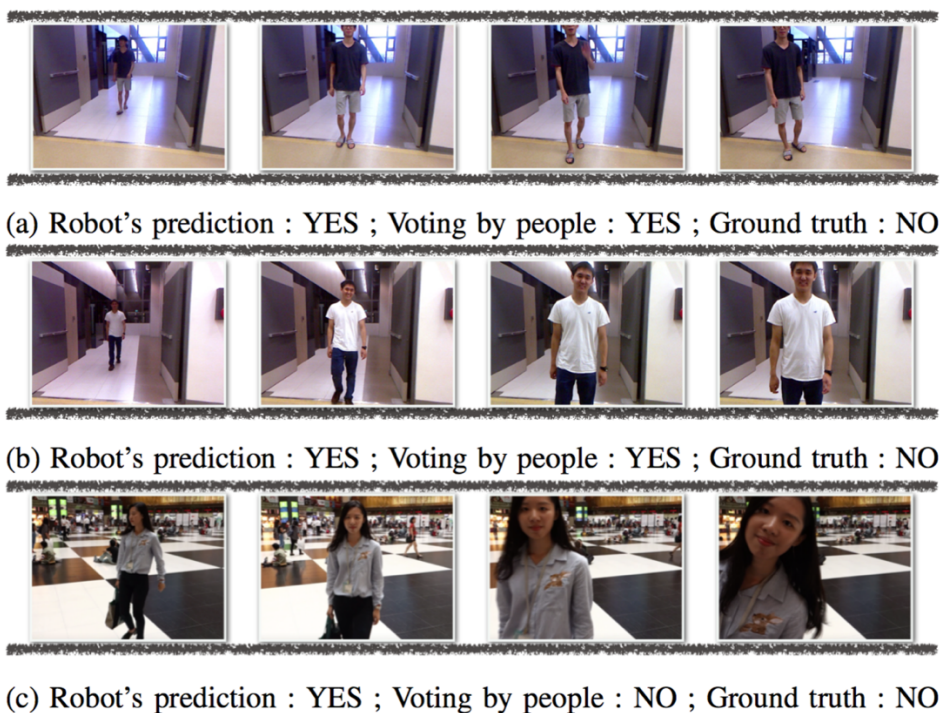
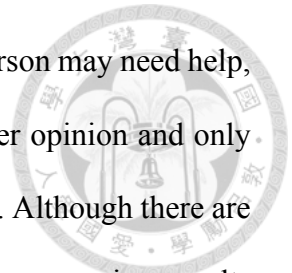


Figure 5-3 Three human's behaviors contain non-consistent opinion among robot's prediction, people voting and ground truth.

The people shown in Figure 5-3 (a) and (b) tell me that they indeed don't need any help in that contexts, but just feel curious and would like to know what the robot will respond to them. In the 2% case, prediction of robot is True, but the decision of people in service industry and ground truth is False. The girl shown in Figure 5-3 (c) would like to

see the robot for a glimpse but need no help. Our robot predict this person may need help, however decision made by people in service industry possess another opinion and only two votes show that they will begin to ask whether she will need help. Although there are few interesting exceptions occur in human-robot interaction, the encouraging results show that prediction made by social robot present 96% as same as people in service industry.



Chapter 6 CONCLUSION, CONTRIBUTIONS and FUTURE WORKS



6.1 CONCLUSIONS

In this thesis, we present an exciting result that robots have potential capability to learn from situational context for providing “just-good services”. The situational context perception based on state-of-the-art deep learning models we proposed allow robot to successfully learn from sequential human behavior for identifying whether a person needs assistance. Thus, the robot can provide greater appropriate service with respect to person’s mentation, more friendly and more considerate.

In the experimental results, we find that human body language reveals the messages of human’s mentation. Moreover, we retrieve a significant improvement on identifying needing assistance via taking spatio-temporal factor into consideration. With regard to encoded images feature, it only yields 49.5% accuracy by SVM classifier. However, the sequential encoded images classified through LSTM-based classifier yields a significant improved accuracy to 73%. The proposed CNNs followed by LSTM architecture, which considers spatial and temporal factors simultaneously, perform 78% accuracy with optical flow feature. Lastly, we implement the situational context perception on social co-robot to perceive a person’s mentation in real-time. The identification of robot is compared with the decision made by people who work in service industry on the task of providing “just-good services”. The result show that there is 96% consistent opinion between social co-robot and people in service industry.

6.2 CONTRIBUTIONS

In this thesis, we proposed the deep learning based situational context perception upon perceiving a person's mentation which focuses on needing assistance. Therefore, social co-robot can behave appropriately according to the person's mentation in service industry.

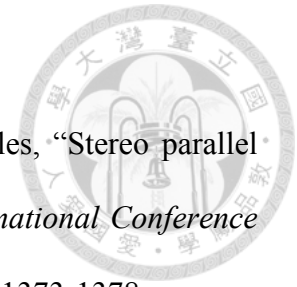
The contributions of our research can be summarized as follows:

1. The proposed deep learning models perform significant improvement on identifying whether a person needs assistance by taking spatio-temporal factor into consideration.
2. The deep learning based situational context perception can implement on robot and analyze features in real-time.
3. The social co-robot, equipped with the proposed situational context perception, possess highly consistent opinion with people who work in service industry upon perceiving a person's mentation.


6.3 FUTURE WORKS

From our perspective, data acquisition is the most challenge in our human-robot interaction scenario. Thus, a well-defined data collection mechanism for extending the knowledge of situational context perception is demand. There are three issue we plan to conduct for our proposed perception to adapt into varied fields in service industry. First, establish a reward function for people who interact with robot to score the performance. Second, employ reinforcement learning to make robot adapt into various situational context. Last, select a service for robot to fulfill and evaluate overall service process.

References



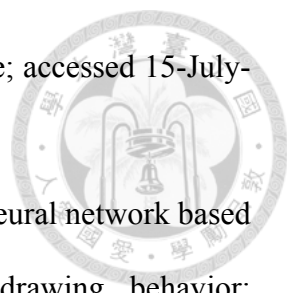
- [1] T. Pire, T. Fischer, J. Civera, P. De Cristforis and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 1373-1378.
- [2] K. Qiu, F. Zhang and M. Liu, "Visible Light Communication-based indoor localization using Gaussian Process," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 3125-3130.
- [3] R. C. Luo, V. W. S. Ee and C. K. Hsieh, "3D point cloud based indoor mobile robot in 6-DoF pose localization using Fast Scene Recognition and Alignment approach," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Baden-Baden, Germany, 2016, pp. 470-475.
- [4] H. Kikkeri, G. Parent, M. Jalobeanu and S. Birchfield, "An inexpensive method for evaluating the localization performance of a mobile robot navigation system," *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 4100-4107.
- [5] R. L. Riek, "The Social Co-Robotics Problem Space: Six Key Challenges," *Robotics Challenges and Vision (RCV2013)*, 2014.
- [6] C. R. Raymundo, C. G. Johnson and P. A. Vargas, "An architecture for emotional and context-aware associative learning for robot companions," *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Kobe, 2015, pp. 31-36.


- 
- [7] P. Lison, C. Ehrler and G. J. M. Kruijff, “Belief modelling for situation awareness in human-robot interaction,” *International Symposium in Robot and Human Interactive Communication*, Viareggio, 2010, pp. 138- 143.
- [8] S. H. Tseng, J. H. Hua, S. P. Ma and L. e. Fu, “Human awareness based robot performance learning in a social environment,” *IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013, pp. 4291- 4296.
- [9] Situational Context, <https://www.alleydog.com/glossary/psychology-glossary.php>
[Online; accessed 1-March-2017]
- [10] Nigam and L. D. Riek, “Social context perception for mobile robots,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 3621-3627.
- [11] H. Qureshi, Y. Nakamura, Y. Yoshikawa and H. Ishiguro, “Robot gains social intelligence through multimodal deep reinforcement learning,” *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Cancun, 2016, pp. 745-751.
- [12] Aly and A. Tapus, “A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction,” *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Tokyo, 2013, pp. 325-332.
- [13] J.Mumm and B.Mutlu, “Human-robotproxemics : Physical and psychological distancing in human-robot interaction,” *ACM/IEEE International Conference on*

Human-Robot Interaction (HRI), Lausanne, 2011, pp. 331-338.



- [14] T. Kitade, S. Satake, T. Kanda and M. Imai, “Understanding suitable locations for waiting,” *ACM/IEEE International Conference on Human- Robot Interaction (HRI)*, Tokyo, 2013, pp. 57-64.
- [15] PCL, <http://pointclouds.org/> [Online; accessed 15-July-2017]
- [16] OpenCV, <http://opencv.org/> [Online; accessed 15-July-2017]
- [17] Scikit-Learn, <http://scikit-learn.org/stable/> [Online; accessed 15-July-2017]
- [18] Keras, <https://keras.io/> [Online; accessed 15-July-2017]
- [19] API.AI, <https://api.ai/> [Online; accessed 15-July-2017]
- [20] ROS, <http://www.ros.org/> [Online; accessed 15-July-2017]
- [21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1.
- [22] Optical Flow, http://docs.opencv.org/trunk/d7/d8b/tutorial_py_lucas_kanade.html [Online; accessed 15-July-2017]
- [23] Autoencoders, <https://blog.keras.io/building-autoencoders-in-keras.html> [Online; accessed 15-July-2017]
- [24] Category classification by CNNs, <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/> [Online; accessed 15-July-2017]
- [25] Activation function, https://en.wikibooks.org/wiki/Artificial_Neural_Networks/-Print_Version [Online; accessed 15-July-2017]
- [26] Gradient descent, <http://sebastianruder.com/optimizing-gradient-descent/> [Online; accessed 15-July-2017]

- 
- [27] Early stopping, <https://deeplearning4j.org/earlystopping> [Online; accessed 15-July-2017]
- [28] K. Sasaki, H. Tjandra, K. Noda, K. Takahashi and T. Ogata, “Neural network based model for visual-motor integration learning of robot’s drawing behavior: Association of a drawing motion from a drawn image,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 2736-2741.
- [29] V. Veeriah, N. Zhuang and G. J. Qi, “Differential Recurrent Neural Networks for Action Recognition,” *IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 4041-4049.
- [30] Q.Li,X.ZhaoandK.Huang,“Learningtemporallycorrelatedrepresentations using lstms for visual tracking,” *IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 1614-1618.
- [31] Y. Bengio, P. Simard and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994.
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [33] SVM, <http://digdata.in/post/94066544971/support-vector-machine-without-tears> [Online; accessed 15-July-2017]
- [34] X. Yuan, L. Cai-nian, X. Xiao-liang, J. Mei and Z. Jian-guo, “A two- stage hog feature extraction processor embedded with SVM for pedes- trian detection,” *IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, 2015, pp. 3452-3455.

- 
- [35] Y. Benabbas, N. Ihaddadene, T. Yahiaoui, T. Urruty and C. Djeraba, “Spatio-Temporal Optical Flow Analysis for People Counting,” *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Boston, MA, 2010, pp. 212-217.
- [36] J. Ba and D. Kingma, “Adam: a Method for Stochastic Optimization,” *International Conference on Learning Representations*, San Diego, 2015.

VITA



姓名：謝仲凱

性別：男

生日：03.30.1993

籍貫：台中市

學歷：

1. 民國 106 年 國立台灣大學電機工程學研究所畢業
2. 民國 104 年 國立清華大學電機工程學系畢業
3. 民國 100 年 國立臺中第一高級中學畢業

發表著作：

1. Ren C. Luo, Chung-Kai Hsieh, "Robotic Sensory Perception on Human Mentation for Offering Proper Services", 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2017), Daegu, Korea, November 16-18, 2017, submitted (EI).
2. Ren C. Luo, Chung-Kai Hsieh, "Deep Learning Based Social Context Perception on Human-Robot Interaction", 49th International Symposium on Robotics (ISR 2017Asia), Shanghai, China, July 5-8, 2017.
3. Ren C. Luo, Vincent Wei Sen Ee, Chung-Kai Hsieh, "3D Point Cloud Based Indoor Mobile Robot in 6-DoF Pose Localization Using Fast Scene Recognition and Alignment Approach", 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016), Baden-Baden, Germany, September 19-21, 2016. (EI)

榮譽事績：

2017 年 6 月 「2017 智慧對話機器人」榮獲 竹間智能創意獎

2016 年 9 月 「2016 機器人創意競賽」榮獲 冠軍

2015 年 11 月 「2015 IROHCS 籃球賽機器人競賽」榮獲 亞軍