Visual Object Detection from Lifelogs using Visual Non-lifelog Data

TengQi Ye

B.Sc. (hons)

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (Ph.D.)

to

School of Computing

Dublin City University

Supervisors: Dr. Cathal Gurrin and Prof. Alan F. Smeaton

January 10, 2018

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

Student ID No.: 13212487

Date:

Contents

Acknowledgements

I would first like to thank my mother, Hua Ye, who raised me by herself. My grandmother, Yumei Zuo, went to Heaven when I was writing this thesis. But I will never forget her warm hugs when I was down.

Thanks to my supervisors in Northeastern University (China), Prof. Ying Liu and Prof. Guoqi Liu. I started my research with Prof. Ying Liu since 2010. Her hard push benefits me, even until now. Thanks also to Mr. Qi Wang who displayed how interesting about mathematical modeling.

Thanks to my supervisors in Dublin City University (Ireland), Dr. Cathal Gurrin and Prof. Alan F. Smeaton. They provided an ideal research platform for me and assisted me greatly in the preparation of this thesis. They helped me with my English writing and background work of lifelogs. The research was supported by the Irish Research Council (IRCSET) under Grant Number GOIPG/2013/330. The author wishes to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. Sincere thanks to my external examiner Prof. Stefan Rüger and Dr. Alessandra Mileo. Prof. Rüger is so serious about his duty that he even pointed out errors in my formula. Dr. Mileo introduced new perspective for me.

I have to give special thanks to four of my friends who gave me enormous help of my thesis. Mr. Tianchun Wang introduced me to how to research machine learning. We kept the twice a month discussion for several months until I can do it on my own. The idea of transfer learning was inspired by him because his research interest is transfer learning. Dr. Kevin McGuinness is an extraordinary researcher who was always willing to help. He introduced me to deep learning. I would also like to thank Dr. Liangyou Li and Dr. Feiyan Hu. Dr. Liangyou Li helped me a lot in my deep learning experiments. Dr. Feiyan Hu helped me a lot in research of lifelogs. Both of them provide also provide considerable guidance in my life.

I am also grateful to my friends who cheered me up at different periods of my study, Longyue Wang, Dongyun Nie, Joseph Antony and Eva Mohedano. I extend my sincere

gratitude to my excellent colleagues Dian Zhang, Jian Zhang, Jinghua Du, Shu Chen and Xiaofeng Wu.

List of Figures

List of Tables

Abstract

Limited by the challenge of insufficient training data, research into lifelog analysis, especially visual lifelogging, has not progressed as fast as expected. To advance research on object detection on visual lifelogs, this thesis builds a deep learning model to enhance visual lifelogs by utilizing other sources of visual (non-lifelog) data which is more readily available.

By theoretical analysis and empirical validation, the first step of the thesis identifies the close connection and relation between lifelog images and non-lifelog images. Following that, the second phase employs a domain-adversarial convolutional neural network to transfer knowledge from the domain of visual non-lifelog data to the domain of visual lifelogs. In the end, the third section of this work considers the task of visual object detection of lifelog, which could be easily extended to other related lifelog tasks.

One intended outcome of the study, on a theoretical level of lifelog research, is to identify the relationship between visual non-lifelog data and visual lifelog data from the perspective of computer vision. On a practical point of view, a second intended outcome of the research is to demonstrate how to apply domain adaptation to enhance learning on visual lifelogs by transferring knowledge from visual non-lifelogs. Specifically, the thesis utilizes variants of convolutional neural networks. Furthermore, a third intended outcome contributes to the release of the corresponding visual non-lifelog dataset which corresponds to an existing visual lifelog one. Finally, another output from this research is the suggestion that visual object detection from lifelogs could be seamlessly used in other tasks on visual lifelogging.

Source code can be found at https://github.com/tengerye.

Chapter 1

Introduction

Humans have always had a desire to log and record their life experience. The phenomenon could be traced back as early as prehistory period when our ancestors created cave paintings to convey their impression and understanding of their surroundings. After written language becomes widespread, the diary became a common way for people to document everyday life and their personal emotion and analysis. In modern society, social networks take over, not only because of convenience, but also because of the social interaction functionality. The recent advances in digital technologies mean that it is now becoming possible to automatically log (record) much of life experience, whether for personal (diary) use or as a source of data for social sharing. This automatic logging of life activity is called lifelogging and is the focus of this research.

Lifelogging is the ambient, digital capture of several possible data sources which log the ordinary day to day activities of a person in daily life [62]. The person that performs lifelogging is referred as lifelogger and the outcome of lifelogging are data archives called lifelogs. Depending on the means of data collection, lifelogs could be in the form of numerical data, multimedia data, documents; most sources of digital data about the individual could be included. Visual data, in the form of images or videos, is one of the most important data streams. When the visual signal is the only one recorded, it is referred to as visual lifelogging [15] and most of this research is based on the idea that visual lifelogging will produce vast archives of visual data that needs to be processed and enriched in order to

support higher-level access. In this work, we distinguish between images or videos that are captured automatically by lifelogging devices (e.g., wearable cameras) which we refer to as visual lifelogs and images or video that are captured from non-wearable devices, which we call visual non-lifelogs.

Apparently, not all visual non-lifelogs data is beneficial to learning tasks on visual lifelog data. For example, medical images or satellite images are unlikely useful for recognizing objects from lifelog images. The visual non-lifelog data in the rest of the thesis is the data that is helpful by default. Therefore, I created a useful visual non-lifelog dataset which is detailed in Appendix A. This terminology will be used throughout this dissertation.

The last several centuries have seen an unprecedented explosion in the vision capture development, including devices that collect visual lifelog data. The visual lifelog data, in the form of images or videos, receives much welcome and attention for its expressiveness. People could generate digital diaries automatically by segmenting visual lifelog streams into events [115]. Such visual lifelog data is also used for the study of social interaction [41] and analyzing attention patterns [103].

The development of computer vision could greatly help the visual lifelogging, since it provides a means to extract semantic value from the visual data. Computer vision deals with how computers can be made to achieve high-level understanding from digital images or videos. From the aspect of engineering, it seeks to automate tasks that the human visual system can do [5]. Recent progress reports that attributes such as hand appearance, object attributes, local hand motion and camera ego-motion are important for characterizing the real-world actions of lifelogger [81, 99].

Visual object detection, finding and localizing objects from an image, on visual lifelogs is a promising research topic and is a necessary building-block for many other lifelog studies. For example, content-based lifelog image retrieval [79] needs to know what objects are in the lifelog images and uses the objects as the vocabulary for images. Another example is scene understanding [20], which also needs the information of the content of the lifelog images. The scene understanding labels the scene type based on the kinds of objects in the images and the conditional probability of the objects and the scenes. The object detec-

tion is well-defined and is well-studied in the long history of computer vision [51, 97, 117], however, there are not enough studies [16] of the object detection on the lifelog images.

1.1 Motivation

From its beginning, lifelog research was held back by two types of handicaps [8, 19]: hardware and software. Visual lifelogging is not an exception. Because visual lifelog data unavoidably takes tremendous space, massive storage has to be small enough to carry and cheap to purchase. Batteries are expected to enable digital cameras never stop taking photos or even videos and can run all day. The more sensors the lifelog devices have, the better they will be. Fortunately, the continuous development of electronics has relieved the problem of hardware. Some lifelog devices, such as Autographer wearable cameras and Google Glass, can capture images continuously for hours (reference Section 2.1.2 for more details). It seems we get closer to the total capture—capture everything we do and see [106].

However, in reality, modern people are suffering from information explosion. Collecting huge amounts of personal data — even large numbers of pictures — simply creates a problem instead of solving one.

On the one hand, people have limited inspiration to practice lifelogging if they can't see much benefit from it (reference more details in Section 2.2.1). For example, lifelog researchers hope that lifelogging could help people monitor their health (like weight, blood pressure, etc.), hence people would improve their health by more frequent exercising. But how can you expect people, who find it hard to exercise for their health, to collect data? It takes great effort to collect so much data, to analyze it, to maintain and curate it.

Another obvious obstacle rises from the great concern for privacy. Not too many lifeloggers are willing to share their lifelog images, even if those who are willing to do so, they share only a subset of their images. Therefore, the amount of lifelog data accessible for research purposes is much less than that has been collected, which will be explained later why it is a serious problem for further analysis. Bystanders, who are not lifeloggers, may be worried the lifeloggers also may violate the privacy of bystanders when they visually

record their surroundings [127], which further poses challenges for scholarly research.

Just as the hardware of lifelogging (e.g., cameras, storage) depends on the developments in the field of electronics engineering, the software of lifelogging (e.g., visual content automatically understanding, GPS trajectories analysis) depends on the advancement of artificial intelligence. In recent years, we witness the fast improvement of machine learning, especially in the field of deep learning. Its success benefits from a series of factors, such as effective training approaches, a large number of hyper-parameters, and a necessary context — the birth of big data. A famous milestone of deep learning is the deep convolutional neural network [71], which reported considerably better results for object recognition than the previous state-of-the-art on ImageNet. The ImageNet (2016 version), one of the biggest visual non-lifelog dataset, with 22 thousand categories and 14 million images. It describes each category by an average of 650 images collected from the Internet and verified by multiple humans [34].

Another important reason for the success of deep learning is big data. Predictive error has two components, and while more powerful learners reduce one (bias) they increase the other (variance) [37]. Machine learning models with more hyper-parameters have a better chance to challenge the problems with larger feature spaces, but they need more data at the same time. The tasks with large feature spaces but with small amounts of training data poses a big challenge [14, 38]. Smaller training set provides less information. Decreasing the size of the data set increases the over-fitting problem.

Due to the free motion of the camera and the passive acquisition of lifelogging data, objects in the lifelog images mostly fluctuate and their appearance may vary broadly. The freedom of positions and shapes make the tasks in visual lifelogging more difficult than for visual non-lifelog data because the visual lifelog has larger feature space (reference a similar study case of digit recognition, as the first example in Chapter 12 of [14]). Moreover, the occurrences of objects in lifelog images have much higher variance than that of non-lifelog images in most datasets, e.g., ImageNet. The objects of non-lifelog image databases usually have the close frequency for each kind for better training, but the frequency of lifelog images depends on real-life collecting. We can conclude that for some objects in

lifelog images, their available training examples are small.

Therefore, current state-of-the-art tasks of objects in lifelog images have a lot of restrictions. Some work constrains the desired objects to be within specific positions (for example, objects on the hand [99]); others need extra manual effort to aid their tasks (for example, manually select object candidates [16]).

1.2 Aim and Scope

Given the abundant visual non-lifelog data, such as ImageNet, in this research we pose the question; *is it possible to transfer prior knowledge from sufficient visual non-lifelog data to insufficient visual lifelog data?* As mentioned in above, the lifelog images have larger feature space than non-lifelog images. While in the meantime, because for some objects, we see them much less than other objects, there are not as many training examples as the former ones. It is intuitive to explore the ways to transfer the knowledge from non-lifelog images to lifelog images.

Inspired by the fact that human intelligence has the ability to store knowledge obtained while solving a task and applying it to a different yet similar problem. For example, the human can recognize a new object by reading its linguistic description without seeing the object in real life beforehand. The thesis set its paramount aim as *enhancing the visual object detection from lifelog data with the help of sufficient volumes of visual non-lifelog data*.

Limits to the research are noted from the start. The thesis does not suggest a panacea that could apply to all tasks of visual lifelogging. Instead, the thesis explores the solution to challenges that are caused by a deficiency in the volume of visual lifelog data and the task focuses on visual object detection of lifelogs because the task is well-defined and could be investigated using state-of-the-art approaches.

The following three research questions intend to display the checkpoints of the thesis and corresponding answers are cornerstones of the thesis. Each of them forms a core chapter:

Research Question 1. *What is the relation between visual non-lifelog data and visual lifelog data for machines?*

The research question (corresponding to Chapter 3) starts the study. There are three potential answers to the research questions: they are totally different, which means visual non-lifelogs can not help visual lifelogs; they are identical from the perspective of the machine, which means no extra effort is required to supplement visual lifelog using visual non-lifelog; they are similar to some extent, and knowledge from visual non-lifelog could help tasks on visual lifelog. Whatever the answer to the research question is, it will create new knowledge in the field of visual lifelog.

Intuitively, we may guess visual lifelog and visual non-lifelog are similar. But that is far from enough. We need rational analysis to show the similarity comes from some obvious reasons and it is not accidental. Furthermore, experiments are necessary to support the intuition and validate the analysis.

Research Question 2. *How could we transfer useful knowledge from visual non-lifelogs to help tasks on visual lifelogs?*

As mentioned above, the question (corresponding to chapter 4) depends on the answer of Research Question 1. Not until we confirm the visual lifelogs and visual non-lifelogs are similar but not identical, we can set off to explore this research question. Therefore, Research Question 2 actually seeks for a solution or a strategy that could transfer the knowledge from visual non-lifelog to visual lifelog, namely, machine learning models could jointly be trained on the visual non-lifelog and performs equally well on visual lifelog.

This research question aims to explore an effective transfer learning algorithm for the object detection task on visual lifelogs. An expected challenge is that the non-lifelog image typically contains exactly one pre-defined object while the lifelog image contains arbitrary objects. The lifelog images can not be used directly for the object recognition. In this sense, we can crop objects from the lifelog images and explore the research question in the task of the object detection. Experiments are expected to show the effectiveness of the

corresponding transfer learning algorithm.

Research Question 3. *How could we convert the object recognition task to the object detection for the lifelog images?*

This research question follows the success of the Research Question 2. If there is no effective way to transfer the knowledge from the objects of the visual non-lifelog to the objects of the visual non-lifelog, it is infeasible to address the object detection task of lifelog images. The challenge of the Research Question 3 is that most current transfer learning aims at the object recognition task. Analysis and experiments are required to show the effectiveness. This work is discussed in Chapter 5.

1.3 Contributions

There are four proposed contributions of the thesis. One intended outcome of the study, on an impacting theoretical level of lifelog research, is to identify the relation between visual non-lifelogs and visual lifelogs from the perspective of computer vision. On a practical level, a second intended outcome of the study is to demonstrate how to apply domain adaptation to enhance learning on visual lifelogs by transferring knowledge from visual non-lifelog. Specifically, the thesis focuses on the model of the convolutional neural networks. Further, although the non-lifelog images are widely available, there is no ad-hoc dataset of visual non-lifelog. A third contribute is the release of the corresponding visual non-lifelog dataset which corresponds to a visual lifelog one (EDUB). Finally, a further concern in the research is the result of visual object detection of lifelog could be seamlessly used in other tasks on visual lifelog.

1.4 Overview of Thesis

This thesis consists of five more chapters, the middle three (Chapters 3 to 5) of which are the main body of the work.

Background (Chapter 2)

Chapter 2 situates the current study in the related literature of lifelogging. This includes a critical review of the historical context of lifelog research, the up-to-date situation of visual lifelog research, and current practice of visual object detection of lifelog research. Chapter 2 argues for the need of research on small examples of lifelog data and the significant challenge of visual object detection of lifelog. Based on that, the most pressing gaps in the literature are identified and research questions are posed accordingly. To my best knowledge, Chapter 2 is the first literature review of lifelog research from the perspective of artificial intelligence.

Visual Lifelogs are on Different Domain from Visual Non-lifelogs (Chapter 3)

Chapter 3 is the entrance of the whole study and is fundamental for the next two chapters (Chapter 4 and Chapter 5). It proposes two hypotheses that visual non-lifelogs and visual lifelogs are of different domains. The chapter further reviews the deep convolutional neural networks and training tricks for the sake of repeatable implementation. It performs three experiments to underpin its proposals: trains model on the lifelog images and test on the lifelog images; trains model on the non-lifelog images and test on the non-lifelog images; and trains the model on the non-lifelog and evaluates on the lifelog data. Along with the experiments, Chapter 3 also releases a dataset of non-lifelog images as a contribution.

Enhancing Visual Lifelogs for Object Recognition with Visual Non-lifelogs (Chapter 4)

Based on the findings in Chapter 3, Chapter 4 starts with the rigorous definition of the problem and deals with applying domain adaptation to enhance visual lifelogs by transferring knowledge from visual non-lifelogs. This chapter recaps the research field of domain adaptation and analyzes how the domain-adversarial training of deep convolutional neural network could work on visual object recognition task. The experiments display the effectiveness of the proposed approach. Both Chapter 3 and Chapter 4 focus on the task of object

recognition.

Object Detection in Visual Lifelogs (Chapter 5)

This chapter aims to employ the results from the object recognition task (from above two chapters, Chapter 3 and Chapter 4) and applies it to the object detection task. Object detection usually comprises of two parts: the region proposal and the object recognition. This chapter has to either prove the region proposal will not affect the transfer learning or adjust the region proposal. Overall performance (from original lifelog images to object regions) will also be shown in the experiments. Potential extensions to other lifelogging tasks are discussed in the chapter.

Conclusion (Chapter 6)

Finally, Chapter 6 contains the conclusions and reflective evaluation of the study and suggests a future research agenda.

Chapter 2

Background to Lifelogging

In this chapter, we turn to a more exhaustive background study of lifelogging. Firstly, the history of work in the area of lifelogging will be outlined and summarised. This is presented as a series of three "ages" for research and development in lifelogging, covering digitization, the emergence of a variety of applications, and the current interest in mining and using data derived from lifelogs. The second topic covered in the chapter is the introduction and discussion of several important related research topics. Finally, the third contribution of the chapter is a review of related challenges and research developments.

2.1 History of Lifelog Research

Based on the popularity of different research topics, I believe the history of lifelog research can be categorized into roughly three ages namely the early vision and device improvement, the emergence of lifelog applications and finally the resurgence of interest in data mining on lifelog data.

Before we examine each of these three "ages" in turn, we first look at how lifelogging and diary recording became "digital lifelogging".

2.1.1 From Life-log to Digital Lifelogs

The term lifelog is literally based on the combination of "life" and "log", a form of compound or portmanteau which naturally refers to logging or recording of everyday life. In my opinion, humans have always had a natural inclination to record what is happening in their lives. The phenomenon can be traced back as early as to cave paintings. Cave paintings (also known as "parietal art") are painted drawings on cave walls or ceilings, mainly of prehistoric origin, created some 40,000 years ago in both Asia and Europe. One guess from scholars in this field is that the cave paintings carry the records of knowledge about animals and the everyday lives of ancient people.

The original cave painting was ultimately replaced by the diary when written languages were invented and became accessible to many people. A diary is a record (handwritten or digital) arranged by date reporting on what has happened over the course of a day, which may include a person's experiences, thoughts or feelings. It is my belief that writing a diary as a regular activity became a popular activity or pastime for many people, creating a personalized archive which could be used for reflection or other purposes.

With advances in digital technology, the form of lifelogging or the creation of personal diaries has changed. Social media and social networking services (e.g., Facebook) now provide multiple ways of digital lifelogging, though normally people do not share details of many everyday experiences by posting on Facebook[1] or tweeting on Twitter[2]. Instead, people post or tweet about their more unusual or self-enhancing experiences. Blogging or Vlogging is a type of on-line based recording of experiences/memories where the recordings of everyday activities are intended for sharing and reliving. While the traditional diary has typically been private and intended just for the use of its author for personal reflection, blogging is the opposite in that the blog or diary equivalent is usually open to the public and the primary purpose is for sharing ones experiences, feelings, opinions, comments, etc [118]. More recently the idea of an automated digital diary based on lifelogging has been proposed [95].

[1] https://www.facebook.com
[2] https://twitter.com

So as we can see, the lifelog is a broad and adaptive concept which is not new and reflects the nature of human beings' desire to capture, record, and share our everyday experiences.

The idea of electronic lifelogging was first proposed by Vannevar Bush in his paper "As we may think" [19], proposing that digital devices and technology could be employed to record life experiences. Since then, the term "lifelog" has gradually expanded to refer to a diverse range of technologies including capture [13], processing [96], storage and retrieval [3], summarisation [55], and so on. The term "lifelog" now commonly refers to a digital lifelog by default.

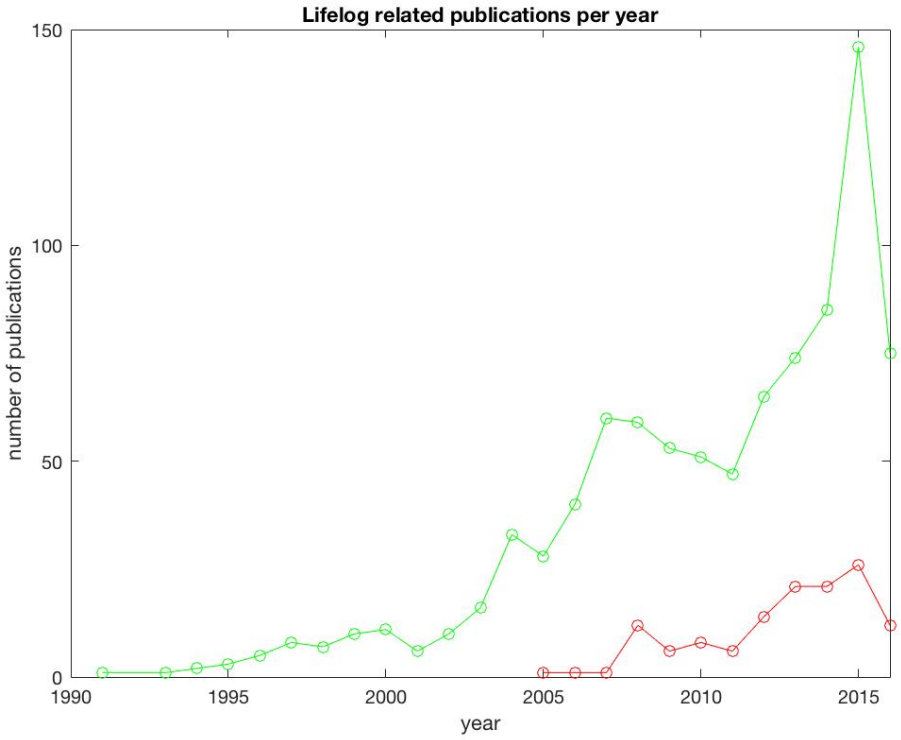


Figure 2.1: Number of lifelog related publications in the Web of Science, per year.

Figure 2.1 shows the number of lifelog related peer-reviewed publications filtered by the subject of computer science and recorded in the Web of Science, per year up to the present date. The red line indicates the number of articles retrieved based on a search using keywords "lifelogging" or "lifelog", while the green line indicates the number of articles

based on using “egocentric” or “first-person”. Because we limit the search to papers in the subject area of computer science, the keywords from the latter search are closer to computer vision (which is called visual lifelogging in the thesis), and in particular about egocentric or first-person (computer) vision. The publication and interest trends for lifelogs and visual lifelogs are similar to an increasing number of publications appearing that are related to lifelogging. Moreover, the increasing interest in first person video analysis [13] and in articles based on using photos or video cameras as a source of imagery [15] are similar to that in figure 2.1.

Lifelogging has witnessed three different ages of lifelog since Vannevar Bush’s milestone paper more than 70 years ago. The following three ages of lifelog research (including visual lifelog) have no clear boundary but are each triggered by some important papers. This chapter has no intention to cover all of the literature, but provides a summary and comments for better understanding the developments and latent motivations for each age.

2.1.2 First Age: Early Vision and Device Improvement

Vannevar Bush, in 1945, imagined a device called “memex” in which we could potentially store an individual’s lifetime media consumption, including all of his or her books read, records captured, and communications with others such as letters [19]. Bush’s work was innovative and important, but even during that time, it was realized that the most important aspect of lifelogging is how to make the first step and make a lifelog device to collect data.

One of the key challenges to lifelogging today is how to miniaturize devices that are used to capture everyday activities, which usually means miniaturizing wearable computing devices. This has required the development of specialized hardware used to enhance the contents of the lifelog. Steve Mann, as a famous example, developed wearable head-mounted displays, cameras, and wireless communications, which enable computer-assisted forms of interaction in ordinary everyday situations in 1997 [82]. Subsequent to that work, Kiyoharu Aizawa noticed that some widely available computing devices, like TVs, phones or glass-type wearable CCD cameras, could be used for capturing and even accessing a lifelog [4].

Steve Mann [82], as one of the most famous early pioneers, was the first to develop customised lifelogging and ubiquitous computing hardware, which he called EyeTap[3]. This EyeTap enables interaction for lifeloggers by allowing "the user's eye to operate as both a monitor and a camera as the EyeTap intakes the world around it and augments the image the user sees." Despite its innovation, Eyetap is not produced or available to purchase. The GoPro[4] is a type of action cameras and it is available on the market since 2005. The camera can be used in various hostile environmental conditions, even in deep water. The SenseCam (2005) is another wearable camera designed to capture a digital record of the lifelogger's life, by recording a series of images and a log of sensor data [59]. It was introduced to the market after 2007. It is a wearable digital camera that is designed to take photographs passively, without user's intervention, while it is being worn around the neck[5]. A similar device, the Autographer[6] based on the principles from the SenseCam, is also a hands-free, wearable digital camera, which also embeds multiple sensors to record activities. More recently, Google Glass[7], designed in the shape of a pair of eyeglasses, even provides a platform for developers to change its facility of interaction with the wearer [127]. The Google Glass has a similar goal to the EyeTap but was openly sold in the market.

Figure 2.2 shows a range of currently available commercial wearable cameras, most of them with embedded sensors. Such devices can be grouped into three categories [13]:

- Smart Glasses: These have several sensors, their own processing capabilities, and a heads-up display screen, making them ideal to develop real-time applications and to improve the interaction between the user and its device. In addition, smart glasses are nowadays seen as the starting point for an augmented reality system. The survey published in [13] is of the view that it is difficult to make such products mature, but the reality is proving this assertion wrong. An example of a wearable Smart Glass is Google Glass (shown in figure 2.2d).

[3] https://en.wikipedia.org/wiki/EyeTap#cite_note-19
[4] https://gopro.com/
[5] http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/default.htm
[6] http://autographer.com/
[7] https://en.wikipedia.org/wiki/Google_Glass

(a) GoPro (2005), image is from Official website of GoPro.

(b) Autographer (2013), image is from a blog about Autographer.

(c) Narrative (2013), image is from Official website of Narrative.

(d) Google Glass (2013), image is from Wikipedia of Google Glass.

Figure 2.2: Four examples of wearable cameras available on the market today and shown chronologically: GoPro, Autographer, Narrative, and Google Glass. The sources of the images are given in the captions.

- Lifelogging Cameras: Wearable cameras have been available for over a decade since the release of the original Microsoft SenseCam, which was a small wearable camera with a wide-angle (fish-eye) lens which captured images from the viewpoint of the wearer at VGA resolution in an automated manner at a number per minute. There have been a number of wearable cameras brought to market since then (e.g. OMG Autographer and Narrative Clip) which operate on the same basic principle.

- Action Cameras: These wearable cameras are popular among sportsmen and among lifeloggers. The lifelogging research community has been using them as a capture device to develop methods and algorithms, while expecting the commercial availability of the smart glass in future years. Anecdotally, more people prefer smart glasses to action cameras even though action cameras are becoming cheaper and are starting to

exhibit their own (albeit still somewhat limited) processing capabilities. An example of an Action Camera is the GoPro (shown in figure 2.2a).

- Eye Trackers: Wearable eye trackers have been successfully applied to analyze consumer behaviors in commercial environments [56] and [13]. Prototype eye trackers are available mainly for research purposes and in the form of spectacles or eye glasses, where multiple applications have been proposed in conjunction with first-person video. Although they are potentially useful, they are too expensive to be popular. Moreover, detecting where the users' eye is directed is not an easy job.

In addition to wearable cameras as a source for lifelogs, there are also cases where surveillance cameras (a.k.a. CCTV) are used to record activities taking place at a specific place. This creates a third-person perspective on the events taking place instead of the first-person perspective we get from a wearable camera. CCTV is widely used in a variety of conditions, including crime prevention, traffic monitoring, and sports events, though it is also used for applications like ambient monitoring of homes especially for the elder in order to detect falls or other emergency situations [45], and so it does form a class of (visual) lifelogging device, though not wearable.

Figure 2.3 lists different kinds of sensors used in some commercial wearable lifelogging devices [13]. The capability of these devices decreases in the figure from top to bottom.

Since the capture and devices are not the focus of the thesis, we will not examine them in a lot of detail. However, readers could reference the Section 2.1 of a thesis [62] for more details of the development of lifelogging devices.

Without data, it is infeasible to study lifelog research. At the present time, with new data sources (sensors) and opportunities for large-scale data storage and processing, researchers are addressing challenges which were previously held back because of the devices. This brings us to the point of this work, the urgent problem of making lifelog data suseful.

	Camera	Eye Tracking	Microphone	GPS	Accelerometer	Gyroscope	Magnetometer	Altitude	Light Sensor	Proximity Sensor	Body-Heat Detector	Temperature Sensor	Head-Up Display
Google Glasses	✓		✓	✓	✓	✓	✓		✓	✓			✓
Epson Moverio	✓		✓	✓	✓	✓	✓						✓
Recon Jet	✓		✓	✓	✓	✓	✓					✓	✓
Vuzix M100	✓		✓		✓	✓	✓		✓	✓			✓
GlassUp	✓		✓		✓	✓	✓		✓				✓
Meta	✓		✓	✓	✓	✓							✓
Optinvent Ora-s	✓		✓	✓	✓	✓	✓		✓				✓
SenseCam	✓		✓		✓			✓	✓		✓	✓	
Lumus	✓		✓		✓	✓	✓						✓
Pivothead	✓		✓										
GoPro	✓		✓						✓				
Looxcie camera	✓		✓										
Epiphany Eyewear	✓		✓										
SMI Eye tracking Glasses	✓	✓	✓										
Tobii	✓	✓	✓										

Figure 2.3: Commercial Approaches to Wearable Devices with First Person View Video Recording Capabilities [13].

2.1.3 Second Age: Lifelog Applications and Human-Computer Interaction

Very little attention from the research community had been paid to how to use lifelogs to improve or impact on people's lives, e.g., aiding memory for an aging population [22] or allowing a review of one's own life for reminiscence as in MyLifeBits [49], which fulfilled the Memex vision until years ago. In 2010, Sellen and Whittaker proposed a viewpoint that research into lifelog systems should focus on an explicit description of their potential value for users instead of the focus being on the technology behind lifelogging [106]. With this perspective, they advocate "the five Rs" for lifelogging — recollecting, reminiscing, retrieving, reflecting and remembering intentions.

The work by Sellen and Whittaker [106] triggered a chain of subsequent research. To better understand how and why different types of lifelog system could aid memory, one piece of work [69] extended lifelogging to include location information. In this work, streams of lifelog images with geographic data were augmented to examine how differ-

ent types of data (e.g., visual, location) might affect memory. It turned out that visual cues promote detailed memories (akin to recollection), while location information allows participants to reconstruct habits in their behavior.

Some very recent similar research investigated the incorporation of lifelogging technology into a therapeutic approach aimed to support people with dementia by using the Case Study method, an exploratory and descriptive approach. This research offers several suggestions from an individual person's visual lifelog as part of a form of Reminiscence Therapy. On a weekly basis, subjects with early-stage dementia can review selected images from their own personal lifelog, and these images are used as a trigger for reminiscence and dialogue of recent events. This seems to be one of the most useful applications of lifelogging technology for use with people with dementia [92].

Another study [95] introduced a lifelogging system which received several sensors' data collected through detecting contexts, segmenting events, generating narratives and representing results to users. During analysis of multimodal lifelog data, this study revealed that events could be effectively segmented by detecting the changes in sensor data. A meaningful textual narrative that accurately represents an event can be generated automatically and different access devices can benefit from different representations of lifelog data.

Because lifelog data amasses quickly, addressing the problem of how to manage a vast collection of personal data (especially images) [36] has been broken into four steps: identifying distinct events, retrieving similar events from a given event, ranking those events based on their importance, and augmenting images with higher quality from the Internet. An effort by Caprani [21] explores the importance of technology to aid social connection, the interest of the elder in learning to use computers, the practical difficulty of household devices, and a general understanding of the devices for underpinning daily tasks and frequently use.

In the second age, lifelog research received attention from the discipline of human-computer interaction (HCI). From the insight of everyday recording devices, a paper [84] studied recording technologies focusing on those situations in which people might or might not know whether such recording exists. This work [84] contributes lifelogging by not only

evaluating grounded reactions to current technologies but also using these responses to perform research and design considerations for new technologies. Another paper [29] argues that fixed infrastructure cameras like the CCTV mentioned earlier in sub-section 2.1.2, can be a useful source for lifelog capture which offers a complementary or alternative approach to existing wearable devices. Notably, such an approach can allow a fully-descriptive third-person view of human experiences rather than the more restricted first-person views.

2.1.4 Third Age: The Resurgence of Data Mining

During the second age of lifelog research, a series of publications contributed feedback, suggestions, and insights into lifelog applications and products.

This brief discussion of the third age of lifelog research will highlight three different aspects: non-sequential data mining, sequential data mining, and computer vision.

A necessary step in interpreting lifelogs is to perform some forms of semantic enrichment. To perform semantic interpretation of events in lifelog, a thesis [118] and some subsequent publications analyze the data based on segmenting each day's lifelog data into discrete and non-overlapping events corresponding to activities in the wearer's life. It addressed the problem of methods to select semantic concepts for indexing and representing events and proposed a semantic, density-based algorithm to cope with concept selection issues for lifelog data. Activity detection is also applied to classify everyday activities by employing the selected concepts as high-level semantic features. In the end, the activity from the lifelog is modeled by multi-context representations and enriched by semantic web technologies. With such semantic representation of lifelogs, this opens the possibility to mine the semantic representations for patterns, highlights, summaries, etc. The most recent advance in the automatic content analysis of visual lifelog data employs a training-free algorithm for enhancing semantic indexing of visual media based solely on concept detection results. In this work, an initial assignment of concepts with probabilities, is made to lifelog images and then these are collectively refined and improved based on statistical distribution of image concepts across the lifelog, making the refinement of initial concept detection based on semantic enhancement, practical and flexible [120].

Focusing on mining information from longitudinal lifelog data instead of concentrating on lifelog data over a short period of time, another recent Ph.D. thesis [62] studied the detection of periodicity or regularly repeating patterns in lifelog data, and the corresponding applications for detection of such periodicities. This work is the first in lifelog research that discovers and identifies periodicity in lifelog data. In addition, the work further proposed several metrics to capture periodicity intensity [63] which represents part of the frontier in current lifelog research.

Another important piece of work [70] presented a complete computational framework for discovering human actions and modeling human activities from videos, to enable intelligent computer systems to effectively recognize human activities. This was the first work that robustly recognized overlapped human activities using a syntactic framework. It contributed a bimodal learning approach that used both motion and visual context without the use of prior scene knowledge, whereas previous work used only motion or relied on *a priori* knowledge of the appearance of objects or actors. It also created a new unsupervised algorithm for learning syntactic structure from data with a lot of noise (potentially all negative examples), whereas previous work on grammatical induction used a training set of positive examples.

The mining of useful information from lifelog data initially occurred a long time ago, but for most of this work, data mining operates on the lifelogs of others and the useful information is not generally made available in an advantageous way, to the originator of the lifelog. There are many such examples, including accelerometer data, used to recognize activity [96] and [78]; visual lifelog or egocentric data, used to recognize handled objects [99], etc. Up to the year 2017, many of the most important lifelogging device manufacturers have stopped selling lifelog products including Autographer (2013-2016), Google Glass[8] (2013-2015). In addition, GoPro is reported to be struggling over its camera business[9].

One possible explanation for the apparent downturn in interest in the manufacture of lifelog devices is that people can not yet effectively process and then take benefit from their

[8] Although Google Glass has received a lot of negative press, it is worth noting that Google Glass version 2 has been released to market with a focus on industrial and assistive applications, rather than as a wearable technology for widespread market adaption.

[9] www.theverge.com/2016/11/30/13792014/go-pro-job-cuts-restructuring-november-2016

own lifelog data as expected, due to the currently available techniques. The hype cycle[10] is a well-known phenomenon in the development and uptake of many kinds of technologies and provides a potential approach to explain the concession. According to the Gartner hype cycle, a technology's life cycle can be divided into 5 phases: technology trigger, the peak of inflated expectations, the trough of disillusionment, the slope of enlightenment, and plateau of productivity. If the hype cycle is applied to the development of lifelogging devices, one opinion is that the devices can be said to be in the trough of disillusionment phase currently, but we can expect it to emerge with a defined set of hardware and services with beneficial use-cases.

2.2 Three Important Research Topics

The previous sections of this chapter reviewed the history of lifelogging based on time, but there are some topics that are explored and investigated but under other larger research fields. Regardless of whether they are mentioned in the previous section (Section 2.1), this section will discuss important research topics in lifelog data capture, lifelog content retrieval, and visual lifelog analysis.

2.2.1 Lifelog Data Capture

Lifelogging cameras were discussed in detail in Section 2.1. Here we will investigate other aspects of capture, including non-camera based data collection, annotation, the difficulties that lifelog wearers encounter and ethical issues.

The capture of a lifelog (a.k.a., data collection or acquisition) has different features according to the specific device and the task. Wearable devices for lifelogging, like the Google glasses with a camera, are usually designed to be free from other sensors considering their weights. As a comparison, a common non-camera based collector of lifelog data is the smartphone [95]. The mobile smartphone has not been regarded as a collector of images or videos for lifelogs because wearers need to hold the phone to take a picture or video, or to

[10] http://www.gartner.com/newsroom/id/3412017

wear it all the time. However, the smartphone is ideal as a device to collect data from its other on-board sensors like GPS, light meter, compass or accelerometer. This makes it a perfect complement to wearable lifelog cameras.

There are two ways to capture a lifelog for research purposes. The first is passively performing lifelog activities (an example is [109]), where a researcher supervises and observes the participant during the set of activities and this corresponds to the data collecting process where the researcher annotates while observing. This is only useful in lab-type settings or other controlled environments where the aim is not to do ambient lifelogging of a subject in ordinary settings. The second way is to use manual annotation protocols on automatically collected data, such as [62]. In this approach, a vocabulary of labels for annotation is created in advance. Then the subject could subsequently look over their data, perform the data annotation process using an annotation tool, and annotate the meta-data based on the controlled vocabulary. The second way of capture is easier and more popular (examples are [16], [68] and [107]).

Digital lifelogging is greatly beneficial for some situations (e.g., diabetics who need to know their recent activities and their blood sugar levels, so lifelogging and health tracking may be an important aspect of health and wellness for some), which reflects the popularity of the quantified-self movement [85], but this is still the domain of the interested few and early-adapters. In some cases,lifeloggers and enthusiasts have stopped tracking altogether. Take for example Chris Anderson, who recently found his many-years self-tracking as pointless and stopped it in 2016[11]. One of the reasons for this is that using current data mining techniques it is hard to extract something important from a lifelog which cannot be done in a simpler way. Other possible reasons could be that lifelogging takes the trouble, even slightly, and both lifeloggers and bystanders can take lifelogging as a threat to privacy[11].

Lifeloggers stop lifelogging maybe because they think the reward not worth the effort. At least, lifeloggers have to remember to charge, to carry, and to copy data from their devices. However, not everyone may think it is necessary to record so much information

[11] https://www.technologyreview.com/s/601300/life-logging-is-dead-long-live-life-logging/

from themselves ("If a device could capture every moment in life for your easy recall later, would you want it to? There are plenty of things I'd rather forget."[12] by Rachel Metz, 2014).

The psychological burden from lifelogging is for both lifeloggers ("What one-eyed *** ** * ***** decided to measure everything anyway?"[11] by Antonio Regalado, 2016) and by-standers who are captured in the lifelog data [127]. Since the camera is quite noticeable when worn, many by-standers may inquire about it, and some will try to remain out of view from it once they learn that it is a camera taking pictures. For some lifeloggers there were also some embarrassing situations in which other people felt uncomfortable to be standing in front the lifelogger while they were wearing a SenseCam, and required the lifelogger to turn it off, or even to delete the images that had recently been captured, containing the person in question. For lifeloggers, they have to remember to carry the device, turn it on, charge it and take data from the device.

Privacy is also an important issue in lifelogging, especially with visual lifelogging [64]. Privacy concerns involve both the lifeloggers and third parties,such as the by-standers. This first point related to privacy is quite obvious as the behavior of the lifeloggers is recorded digitally and can be replayed many times. If their data is leaked, information which they may wish to keep private could be exposed to others. As for third party privacy, examples of data that are relevant include emails, text messages or conversations between another person and the lifelogger. Some of this information is only expected to be seen by the two of them. Therefore, the leaking of lifelog data, or if the lifelog data was given to others, would violate the privacy of others also [127].

Lifelog technology could also reduce the privacy of people besides the lifelogger. Reductions of privacy caused by active lifeloggers to others have distinctive importance. The lifelogger could have decided autonomously that, despite the negative consequences of using lifelogs, recording information was in his best interests. A non-lifelogger may not have chosen to have his privacy reduced (or may not have had a choice at all) and, therefore, reductions of his privacy are unlikely to be the result of an autonomous decision by him. If

[12] https://www.technologyreview.com/s/528076/my-life-logged/

the choice to lifelog and reduce one's privacy are the result of an autonomous choice, then respect for autonomy can become a *prima facie* principle *in favor of* lifelogging. In case the reduction or violation of privacy was not the result of an autonomous choice, the principle of respect for autonomy could become a *prima facie* principle *against* lifelogging [64].

As can be seen, there are a number of restrictive challenges that lifelogging faces, and the work reported in this thesis is not meant to alleviate the effort needed for lifelogging, but instead dedicates itself to leverage the values of lifelogging.

2.2.2 Context-based Lifelog Retrieval

Retrieving images from a large personal database allows us to browse, search or find images of previously seen objects or places and thereby has the potential to solve a broad range of problems in egocentric vision, including searching for elements (Have I seen this before?); navigating (How often do I visit this place?); and understanding the environment (Where am I right now?) [15]. In these cases, the situation is that a user wants to find desired scenes efficiently from a potentially vast quantity of lifelog videos or images. The retrieval agent provides a convenient interface for browsing and retrieving the lifelog images efficiently. Of course, the retrieval agent has the general functions of a standard media player software, for example, "play, stop, pause, fast-forward, and fast-rewind".

Following these premises, a recent Ph.D. thesis [118] built a system for content-based searching and browsing that starts by splitting the stored data into segments and extracting three kinds of information: time and other relevant attributes; low visual features; and audio features. Then, in the retrieval step, they applied time-based filtering by comparing the time attributes of the images in the database with a query introduced by the user. A clustering step then extracts a representative clip from each cluster; and finally, the user can provide one or more query images for the system to refine the search based on visual features and improve the query result.

With this work, several open issues remain: in many situations, it is difficult to recall the time or location where a photo we are looking at was taken. Visual features are too simple to capture real object shape and texture differences and furthermore, when we are

using video then audio features are not captured by the majority of wearable lifelogging devices. A more up-to-date extension of the work [119] proposes an activity classification method for visual lifelogs based on Fisher kernels. It extracts discriminative embeddings from Hidden Markov Models of occurrences of semantic concepts. By using the gradients as features, the resulting classifiers can better distinguish different activities and from that we can make inferences about human behavior. Work in [24] represented millions of egocentric images on a sparse graph. The authors represented each image as a node in the graph, and added an edge between two nodes when they belonged to the same bag in a Bag of Words (BoW) representation. Relying on this representation, they showed that local density clustering is more suitable than global clustering methods, considering the high redundancy that lifelogging data inherently possess. Other work [2] proposed to retrieve novel scenes and actions with respect to a previously acquired egocentric dataset by using a set of "alignment" sequences, and matching them with a new "query" sequence by using dynamic time warping.

Additionally, experiences from multimedia retrieval [67,89]highlight the importance of object detection to the successful provision of multimedia retrieval facilities. The relation of multimedia retrieval and object detection will be further discussed in Section 3.3.

2.2.3 Visual Lifelog Data Discussion

Visual lifelog research (a.k.a., first-person [81] or egocentric [33] vision) studies common tasks such as object recognition [98], object detection [16], or unique tasks such as activity recognition [81], summarization [33] on images or videos [109] and [129] captured by lifeloggers. As with all research on multimedia data analytics and retrieval, there is an underlying necessity to ensure that appropriate and accessible data is employed. This subsection will briefly discuss visual lifelog data from three aspects: passive vs. active; first-person vs. third person; and total capture vs. situation-specific capture. Afterwards, the sub-section will detail the object related tasks of visual lifelog research.

- Passive vs. Active

  Most current lifelogging activities involve passive, continuous, and non-intrusive cap-

ture. This type of logging is usually carried out by using small-sized, wearable devices attached to or carried by the lifelogger and software running on computing devices. This is an example of passive capture, where the wearer does not need to actively trigger data capture; it is automatic once the device is being worn. Actively captured "lifelogs" can be traced back to diary keeping, in which people record their experiences and thoughts through writing. In the current age, although many people still write regular diaries, they are not the main forms of "active lifelog". At present, the prevalence of digital cameras, camcorders and micro blogs such as Twitter are making the active form of lifelogging increasingly popular.

- First person vs. Third person

  Lifelogs can be recorded or generated from the first-persons perspective, e.g. recording what is in front of a persons view (assuming that the recording is what the person sees) or what one hears, as a digital copy of the information one encounters in the physical world. Third-person perspective recording captures scenes where the lifelogger is present but not from their viewpoint. For example, this type of third person lifelogging may include photos taken by others which include the lifelogger, or CCTV recordings containing images of the lifelogger.

- Total capture vs. Situation-specific captures

  Total capture lifelogging aims to digitally capture as many aspects of a persons life as possible. In total capture lifelogs, multi-modal methods (e.g. a combination of visual, audio, textual) are usually used to continually capture a rich digital trace of life activities [55]. Situation-specific lifelogging captures only certain aspects or specific moments of a persons life. Examples include video recordings of meetings, diet monitoring to record times when food or drink is consumed, sport or exercise monitoring, and the monitoring of work or project progress in the office to improve work efficiency. Situation-specific lifelogging is limited in scope compared to the total capture type, although people also try to make the capturing process as automatic and complete as possible for these specific aspects or situations. In the context of this

thesis, we use the term lifelogging to refer to the total-capture type of lifelogging. Lifelogging thus refers to the activity of electronically capturing and storing every possible piece of information that a person (lifelogger) has encountered during the capturing period, and details of context as well as their experiences.

Before completing our coverage of the topic of lifelog data, we must mention NTCIR-Lifelog [13], which is a core task of the NTCIR-13 conference [54]. This core task aims to improve the state-of-the-art research in lifelogging as an application of information retrieval. The methodology employed for the lifelog task at NTCIR-13 is based on the successfully deployed methodology from NTCIR-12 and previous editions of this annual benchmarking activity, which released initial datasets [53]. This benchmarking activity has most recently created a large dataset of 90 days of multimodal lifelog data and has set a number of challenges (sub-tasks) for participants to participate in. These four sub-tasks are the lifelog semantic access task [104] and [105], a lifelog event segmentation task, a lifelog insight generation task, and lifelog visual annotation task. Once the NTCIR-Lifelog task at NRCIR-13 is completed later in 2017, it will provide, for the first time, a sharable lifelog collection with tasks and ground truth data, which will open up lifelog research to researchers who do not have themselves, have direct access to lifelog data.

[13] ntcir-lifelog.computing.dcu.ie

Chapter 3

Lifelogs: A New Domain for Visual Image Processing

The chapter will introduce a range of techniques used to analyze and then index visual lifelogs in terms of their semantic concepts appearing in the images. Before the hypotheses, we examine the fundamental concepts of transfer learning and domain adaptation. They are included in order to set out the problem of object discovery within a visual lifelog from the perspective of machine learning where a model learned on one domain, is then transferred to another domain (the lifelog domain). Afterwards, we will discuss the idea of indexing images and videos by semantic "tags", which is popular in major search services such as Google and Facebook. The remainder of the chapter is devoted to proving the hypotheses by analysis and experiment.

3.1 Background to Transfer Learning

Most research in machine learning, theoretical or empirical, assumes that models are trained and tested using data drawn from the same feature space and the same distribution of features. In many cases, however, we have a plentiful supply of labeled training data from a *source* domain but we wish to perform the same or similar machine learning tasks (e.g., classification, regression) on a related domain — a *target* domain — with a different feature

space or different distribution of features and with little or no labeled training data [91].

Cross-language text classification is an example of transfer learning from the natural language processing [7, 94] area, and it refers to the challenges that address content-based tasks (e.g., spam filtering, topic categorization, sentiment classification) on a language with few training samples while the training examples on another language are sufficient. It is widely acknowledged that collecting, annotating and maintaining corpus data is costly in natural language processing [7] hence the motivation for trying to use transfer learning.

Computer vision has its own challenges of transfer learning [72, 121]. A real-world application considered in [123], is an autonomous agriculture application that manages the growth of grapes in a vineyard. Robots are developed to take images of the crop throughout the growing season (generating example data) and the product is weighed at harvest at the end of each season (generating labels), and the task is to predict yield for each vine. The challenge introduced in [123] is that farmers would like to know their yield in advance so they can make better decisions on selling the produce or nurturing its growth. Acquiring training labels early in the season is very expensive because it requires a human to go out and manually estimate the yield.

Transfer learning can be used to take a step forward in other popular machine learning topics by simply alternating or augmenting some settings. Multi-task learning is a form of transfer learning which tries to learn multiple different machine learning tasks simultaneously [6] while more conventional transfer learning techniques try to transfer knowledge from one label space to another. An assumption made about the connection between the source and the target domains is, for the same class label, the conditional distributions of examples of that label are identical between the two domains. Note, however, that the label distributions (the difference is referred as the *class imbalance*) of the two domains are not guaranteed to be the same, though the domains are of the same label space [66].

A more widely studied and practically useful setting in this area is known as *domain adaptation*. This assumes that the machine learning task is identical (i.e. the source domain and the target domain share the same label space), or the features spaces of two domains are the same, but the marginal probability distributions of observations of class label occurrence

vary between the two domains [9, 91].

Among the two examples aforementioned to illustrate transfer learning, the example of autonomous agriculture is a domain adaptation problem, because the task (estimating the harvest) is the same and the feature space (the collection of images) is the same. Nevertheless, images of a vineyard taken during different seasons do look different because of the presence of foliage (leaves) and grapes in summer, and the vines are bare in winter. In the case of cross-language learning, it is not a domain adaptation problem in the perspective of machine learning since different languages own a unique set of vocabulary items which leads to different feature spaces. As an exception explicitly mentioned, domain adaptation follows the above definition and perspective of machine learning and has priority for terminology conflict[1].

3.2 Hypotheses

The section of the thesis introduces two hypotheses. The first hypothesis asks why machine learning algorithms for identifying within-image content tend to perform much worse on lifelog images than on the non-lifelog images. This then motivates the second hypothesis which states that transferring domain knowledge from non-lifelog images to lifelog images can be considered as a domain adaptation task. Illustrations and arguments follow the two hypotheses individually, and the experiments later in the chapter will support them.

Hypothesis 1. *For an object recognition task, a visual lifelog desires more effective information (training data) than can be extracted from an image taken from the non-lifelog domain.*

Generally speaking, for machine learning applications, we can say that the more data we have, the better. Intuitively, more data provides more information, providing more opportunity for models to learn. A simple example is the law of large numbers from the statistics field. According to the law, the average of the observers collected from a large number of

[1] In natural language processing, "domain adaptation" can be used as terminology for a cross-language problem in the way that, sometimes, a language is mentioned as a "domain" in history.

experiments should be near to the expected value and will tend to become closer as more trials are performed. The statement also agrees on some common senses of machine learning, for example, a dumb algorithm with lots and lots of data beats a clever one with modest amounts of it [38].

Theoretically, a visual lifelog needs much more effective information (training data) than a visual collection from a non-lifelog domain, for the same model to reach similar performance.

Note that lifelog images permit and support more tasks than a standard collection of non-lifelog images. For example, in figure 3.1, the non-lifelog images are on the left column and lifelog images on the right. Compared to the lifelog domain, there is no need for non-lifelog images to be used to support many kinds of machine learning task except perhaps object recognition (what is the object in the image?) while lifelog images could support many more tasks, such as activity recognition (what was I doing?) or scene recognition (where was I?). Because each task has a set of labels, therefore the label space for a visual lifelog is more complex (bigger) than that for a collection of non-lifelog images.

The hypothesis 1 points out that a visual lifelog collection calls for more labeled data than a non-lifelog visual collection. The current reality, however, is that a visual lifelog will has less available labeled data. There are two reasons for this: lifelog images have limited sources and the effective information contained within them is less. Capture of visual lifelogs is currently only performed among a small sub-population of lifeloggers and is not that convenient as discussed in Section 2.2.1. It is unlikely to be feasible for everyone to be a lifelogger for three reasons: visual lifelog devices can be awkward to use (inconvenient); not much valuable information can be extracted at present (techniques); visual lifelogging records the surroundings of the wearer without the permission of others (privacy). Visual lifelog capture risks more interference because the capture is usually performed during a variety of activities, i.e., running, driving, etc. Moreover, because people can be stationary while performing some tasks like attending meetings or driving, we can get hundreds of lifelog images which may be similar or even identical when a simple activity lasts several hours, like working. In this case, those images have the same information as each other.

In the context of lifelog computer vision tasks, even if lifelog images have good quality, cropping an object from a lifelog image is unsuitable for generating training data as it is usually too small or similar to its background. For instance, the persons appearing in figure 3.1d and figure 3.1b are barely distinguishable while figure 3.1a is an image taken for a more natural case where the persons are clearer.

The first hypothesis (hypothesis 1) lays the challenge for the whole thesis. Following this, we will focus on more specific tasks, namely object recognition and object detection.

Hypothesis 2. *Visual lifelogs and visual non-lifelogs are essentially two separate domains. Visual non-lifelogs can be regarded as a source domain while visual lifelogs can be regarded as the target domain.*

The first hypothesis (hypothesis 1), to some extent, also hints at the second hypothesis (hypothesis 2), because if the visual lifelog and visual non-lifelog come from the same distribution, a machine learning algorithm should have similar performance on both of them. However, according to Section 3.1, visual lifelogs and visual non-lifelogs could be considered as having different distributions because they are not sampled (collected) under the same conditions. Moreover, any difference could be observed directly from the comparison between lifelog images and non-lifelog images, such as in figure 3.1 and in the first figure of [15]. Furthermore, affected by the environment's interference and shortcomings of devices, the lifelog images look distinct from non-lifelog images (as in figure 3.1), i.e., the marginal distributions of images are different.

Since we admit that the visual lifelog domain has limited training examples (the target domain) while the visual non-lifelog domain has an abundance of training examples (source domain), then we want to exploit information from the sufficient source domain and apply this in the target domain. Since the images can surely be used for the same task, the tasks between the source domain and target domain are the same.

(a) A non-lifelog example of a person.

(b) A lifelog image contains person and lamp.

(c) A non-lifelog example of a lamp

(d) A lifelog image contains car, person and lamp.

(e) A non-lifelog example of a car

(f) A lifelog image contains car and lamp.

Figure 3.1: Comparison of visual non-lifelog (left column) and visual lifelog (right column). An obvious distinction is lifelog images have distortion because of various interferences. A lifelog image usually composes several objects, thus it is difficult to train models directly on visual lifelog.

3.3 Indexing Images By Semantic Concepts

This section studies the case of image indexing to explain the importance of object recognition and manually labeled data. In this context, we will also introduce content-based image retrieval and show the difference between manually labeling and automatic image annotation as the basis for image retrieval.

3.3.1 Semantic Labels are Necessary for Image Retrieval

The invention of the digital camera and the ease with which digital photographs are taken, stored and shared, has given ordinary people the opportunity to capture their world in pictures, and to conveniently share them with others. One can today generate huge volumes of images with content as diverse as family get-togethers and national park visits. Because of low-cost storage and easy web hosting, the common people were a passive consumer of photography in the past. However, they are current-day active producer nowadays.

It is almost impossible to reconstruct a perfect image solely from the literal description of the image. But it may be easier to find such a picture by looking through the collection and making unconscious "matches" with the one drawn by imagination, than to use literal descriptions that fail to capture the very essence of perfection. However, in the real-world image search, people expect to search desired images by literal keywords (description).

Manual indexing of text documents for retrieval which was the initial way that information retrieval operated, is essentially where each book or text item in a library or a collection has been individually indexed or reviewed to determine its contents [12]. This is quite different from image retrieval. In the very early pre-web days of text-based information retrieval, automatic indexing (or using algorithms/software to extract terms for indexing) has been proposed to cope with the large volume of data existing in the digital world. Instead of reading and understanding the content manually, automatic indexing extracts information mainly via document purification (making decisions of what information to be indexed) and term extraction (which terms to be used to represent documents). Some successful vector space models, e.g., latent semantic analysis [32], benefit from the foundation of the

term-document organization.

Information extraction in image indexing is much harder than in text document indexing. Text index construction usually adopts some linguistic pre-processing of tokens [83]: tokenization, normalization, stemming, etc. The reason that the procedure is even more difficult is the basic units of text documents are words (in English or some other language) while basic units of images are pixels. Words on their own each provides simple semantic information to represent the content of text documents while it is unlikely that pixels can provide any direct semantic information.

A straightforward approach for content-based image retrieval is to assign some semantic labels [31, 89]. For example, if every image has labels of objects (the objects in the image), it would then be rather easy to build a search engine with keywords or terms from an object vocabulary.

Today, searchable image data exist with extremely diverse visual and semantic content, spanning geographically disparate locations, and these are rapidly growing. Because of all these factors, researchers have to consider innumerable possibilities for real-world image search system. Despite the efforts made by image retrieval research, it is hardly believable that we may create a universally acceptable algorithmic by characterizing human vision, more specifically in the context of interpreting images. Owing the digital representation of an image is an array of pixel values, it corresponds poorly to our visual sense, let alone semantic understanding of the image.

There are two approaches to automatically or semi-automatically learning semantic information from images: unsupervised and supervised (this terminology can be found in [14]). Unsupervised approaches include clustering with similarity and zero-shot learning. The clustering approach [31] derives its idea from the fact that with a proper similarity measure, images of the same content will be automatically clustered together. Zero-shot learning, which is to classify instances of an unseen visual class, is based on the idea that visual classes that have a similar look will have similar semantic meaning [108], e.g., dog and cat, or will have similar attributes [74]. An inspiring example of the semi-supervised approach is active learning [87]. Supervised learning is the biggest part of the visual recog-

nition technologies, and it can be regarded as a classification task [14]. The area of the study of classification as a research topic pre-dates even from the invention of the first computer. For supervised learning of classes, one of the downsides is that manual labels are necessary, and we will explore labeling in the next section.

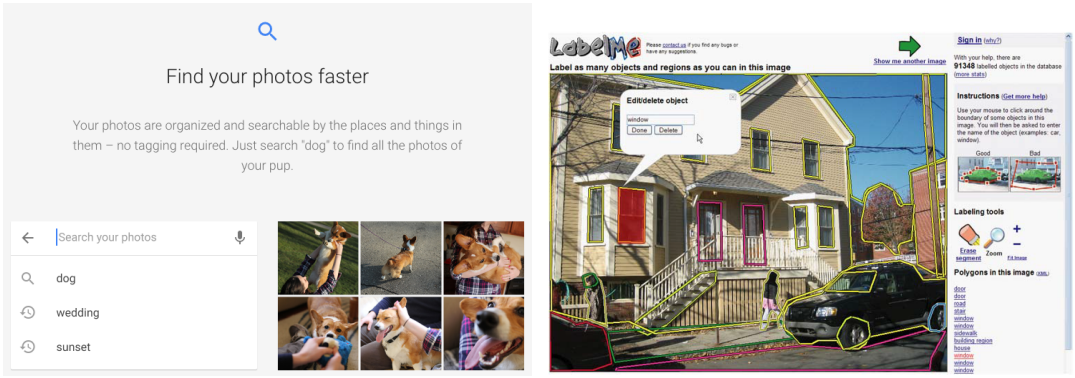
3.3.2 Manually Labelling vs Automatic Image Annotation

It is notably that the term "automatic image annotation" has a different meaning for labeling in the machine learning context. In machine learning, we usually use the word "prediction" instead of "automatic annotation" while the latter one ("labeling") is, by default, referring to manually labeling of images. In this sense, the labels produced from automatic image annotation cannot be used as ground truth labels because they will include errors and when building on such annotation data which has errors, the subsequent predictions will surely make further mistakes. Note that in this thesis, the term annotation can be considered as labeling unless it appears in the term "automatic image annotation". The reason is important and repeated in data selection as we shall see in Appendix A. The difference could also be recognized in online systems. Automatic image annotation systems currently available include Google Photos[2] and the Instagram Search Engine[3]. A famous example of labeling system is LabelMe [102]. Labelling has now become a paid service on Amazon's Mechanical Turk [110].

Figure 3.2 illustrates the interfaces for two applications. Online content-based image search engines have to index a lot of images. Such systems have to employ automatic image annotation to extract information about images, nevertheless the information extracted is not always absolutely correct. Labelling tools aim to provide reliable labels for supervised learning, hence the labels have to be totally correct.

Manual labeling of some images is an important requirement in supervised learning for automatic image annotation because performance results in terms of accuracy are expected to reach the levels of human judgment. Moreover, the more complex or more hyper-

[2] https://www.google.com/photos/about/
[3] https://mulpix.com/



(a) Google Photos, an online content-based image search engine with automatic image annotation. Source: https://www.google.com/photos/about/

(b) LabelMe, an online manually labelling tool [102].

Figure 3.2: Online tools for content-based image search and manual labelling.

parameters a model for learning or classification has, the more annotation data it needs. A very recent success in image annotation is the great improvement in accuracy brought about by the use of deep learning in image recognition [71]. Though the use of the deep learning network model is the key to the breakthrough, this could not have been accomplished without the ImageNet collection of image annotations. Without this large amount of labeled data, the 9 layers neural network with 60 million parameters and 650,000 neurons would have over-fitted. Pairing with the parameters in a deep convolutional neural network, the ImageNet based deep learning classifier has more than 22,000 categories and 14 million images [34]. The rationale for requiring so many parameters for this process comes from the fact that the human brain itself has devoted more than half of its neurons for visual recognition [43]), so this level of numbers of parameters is no real surprise.

In the next section, we look at how the ability to assign semantic concepts or tags to images, can be applied to images from visual lifelogs, using a principle called *"transfer learning"*.

3.4 Models for Experiments

The section will introduce the fully-connected neural networks and convolutional neural networks, and the training methods used to train these machine learning implementations.

Following that, the structures of the models employed in the experiments will be detailed. Finally, several tricks which help in the training process will be discussed. All the numbers, vectors and matrices in the section are computed in the real set by default.

3.4.1 Fully-connected Neural Network and Convolutional Neural Network

A fully-connected neural network, the basic neural network model, can be described a series of functional transformations [14]. Figure 3.3 depicts the data flow of a neuron in a fully-connected neural network where $z_i^{(l)}$ is the output value of the i^{th} neuron on the l^{th} layer. An arbitrary neuron j^{th} on the $l+1^{th}$ layer takes the weighted summation of previous layers $a_j^{(l+1)} = \sum_i w_{i,j}^{(l)} z_i^{(l)} + b_j^{(l+1)}$, and then output $z_j^{(l+1)} = h(a_j^{(l+1)})$, where $h(\cdot)$ is an activation function, $w_{i,j}^{(l)}$ is a weight, $b_j^{(l+1)}$ is a bias.

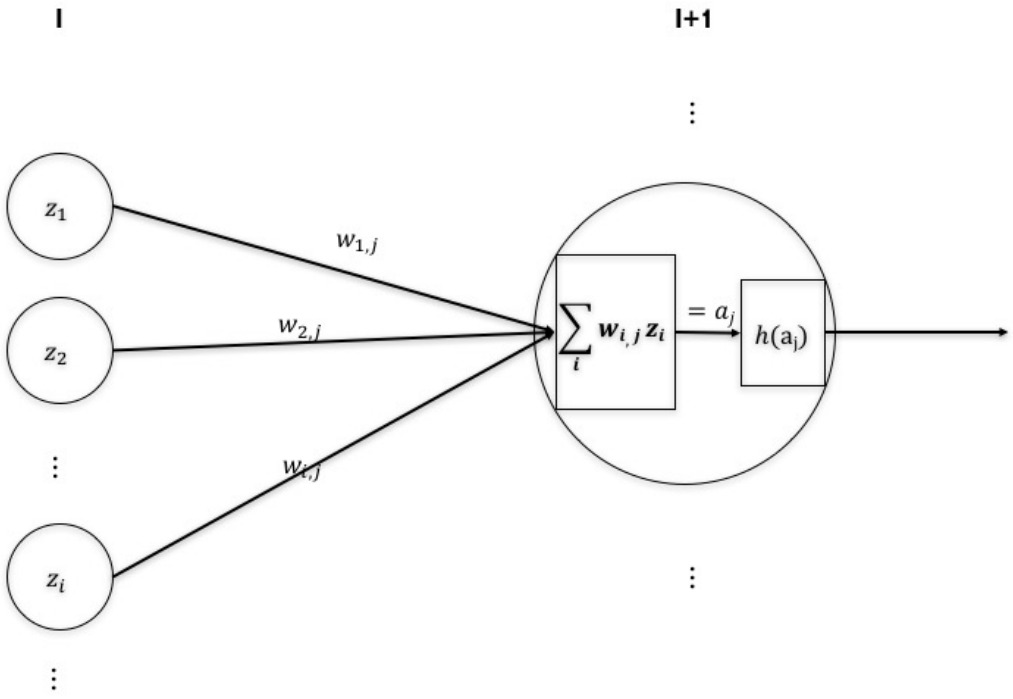


Figure 3.3: Data flow in a fully-connected neural network where a circle represents a neuron.

Here, despite previous deduction of single symbols [14, 101], we adopt the matrix notation to represent a feed-forward neural network, because it is not only computationally

expressive, but it could be fitted into a computational graph [1, 11].

For a fully-connected neural network which has L layers:

$$\mathbf{A}^{(l+1)} = \mathbf{W}^{(l)}\mathbf{Z}^{(l)} \tag{3.1}$$

$$\mathbf{Z}^{(l+1)} = h(\mathbf{A}^{(l+1)}) \tag{3.2}$$

where we combine the weights with biases as $\mathbf{W}^{(l)}$ of the layer l, since $\sum_i w_{i,j}^{(l)} z_i^{(l)} + b_j^{(l+1)} = \sum_i w_{i,j}^{(l)} z_i^{(l)} + b_j^{(l+1)} \cdot 1 = [w_{1,j}^{(l)}, \cdots, w_{i,j}^{(l)}, \cdots, b_j^{(l+1)}][z_i^{(l)}, \cdots, z_1^{(l)}, \cdots, 1]^\top$[4].

The activation function $h(\cdot)$ is performed element-wise, which could be linear or non-linear, continuous or discrete. For input $\mathbf{X}$, a column is an instance i and a row represents a feature j:

$$\mathbf{Z}^{(1)} = \mathbf{X} \tag{3.3}$$

As a consequence, for output $\mathbf{Y}$, every column is an instance i and every row is a feature k:

$$\mathbf{Y} = \mathbf{Z}^{(L)} \tag{3.4}$$

The square loss is usually adopted for a feed-forward neural network [14]:

$$E(W) = \frac{1}{2} Tr((\mathbf{Y} - \mathbf{T})(\mathbf{Y} - \mathbf{T})^\top) \tag{3.5}$$

where $\mathbf{Y}$ is the predicted result while $\mathbf{T}$ is the ground-truth. Obviously, $\mathbf{T}$ has the same size as $\mathbf{Y}$.

Based on equation 3.5 and the chain rule of back-propagation, it is easy to deduce the back-propagation formulas:

$$\frac{\partial E(W)}{\partial \mathbf{Z}^{(l)}} = (\mathbf{W}^{(l)})^\top \frac{\partial E(W)}{\partial \mathbf{A}^{(l+1)}} \tag{3.6}$$

and

[4] Matrix $\mathbf{M}^\top$ is the transpose of matrix $\mathbf{M}$

$$\frac{\partial \mathbf{Z}^{(l+1)}}{\partial \mathbf{A}^{(l+1)}} = \frac{\partial h(\mathbf{A}^{(l+1)})}{\partial \mathbf{A}^{(l+1)}} \tag{3.7}$$

Therefore, the equation to update weights is:

$$\frac{\partial E(W)}{\partial \mathbf{W}^{(l)}} = \frac{\partial E(W)}{\partial \mathbf{A}^{(l+1)}} (\mathbf{A}^{(l)})^{\top} \tag{3.8}$$

Convolutional neural networks combine three architectural ideas to ensure some degrees of the shift, scale, and distortion invariance: local receptive fields, shared weights (or weight replication), and spatial or temporal sub-sampling [75]. A typical structure for a convolutional network containing a convolutional layer and a sub-sampling layer, is illustrated in figure 3.4.

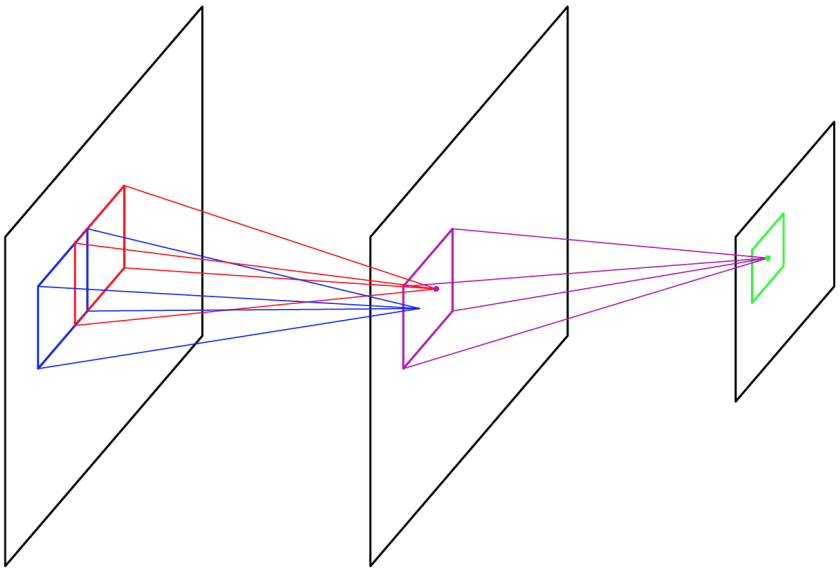


Figure 3.4: Convolutional Neural Network, including a convolutional layer, and a sub-sampling layer. (Source comes from [14])

Local receptive fields can extract elementary visual features such as orientation edges, endpoints, and corners. Information from such features can then be merged in the latter stages of processing to detect higher-order features and ultimately to yield information about the image as the whole. In addition, elementary feature detectors that are useful in one part of the image are likely to be useful across the entire image, even to have identical

weight vectors. Once a feature has been detected, its exact location becomes less important as only its approximate position about other features is relevant.

During feedforward, max-pooling layers pass the maximum values only. Therefore, there is no gradient with respect to non maximum values. Moreover, the maximum value is locally linear with slope 1, with respect to the input that actually achieves the maximum value. Hence, the gradients from the next layer are passed back to only that neurons which achieved the maximum value. All other neurons get zero gradients. In this sense, the max-pooling layers do not do any learning themselves, and they have no parameters to tune for better performance.

The only slight difference among convolutional neural networks lies at convolutional layers. The symbols listed have different meanings from the above. All indexes start from 0 and $m : n$ represents the list $m, m + 1, \cdots, n - 1$.

Because convolution layers have essentially sliding operations, equations of convolutional neural networks (convolutional layers and pooling layers) cannot strictly display regarding matrix operations. The following equations focus more operations between two layers than the kind of patterns across layers. As distinct from layers from fully-connected neural networks, the layers of convolutional neural networks could be regarded as three-dimensional.

For the sliding operation, suppose there are two convolutional layers $L_1[M_1, N_1, K_1]$ and $L_2[M_2, N_2, K_2]$ (normally $K_1 = 3$), and the shape of kernel W is $[F_r, F_l, K_1, K_2]$. In addition, the strides are $[S_r, S_c]$, then for the arbitrary neuron $L_2[m_2, n_2, k_2]$ on layer L_2:

$$L_2[m_2, n_2, k_2] = vec(L_1[S_r m_2 : S_r m_2 + F_r, S_c n_2 : S_c n_2 + F_l, :])^T vec(W[:, :, :, k_2]) \tag{3.9}$$

where $vec(\cdot)$ is the vectorization that flattens a matrix into column vector. Notice there are some constraints between the shapes of two layers $M_2 S_r < M_1$ and $N_2 S_c < N_1$. Likewise equation 3.1, W contains bias.

The back-propagation on the neural networks is still suitable for convolutional neural

networks with a slight modification. The gradients of the kernel in equation 3.9 are:

$$\frac{\partial E(W)}{\partial W[:,:,:,k_2]} = \sum_{m_2,n_2,k_2} \frac{\partial E(W)}{\partial L_2[m_2,n_2,k_2]} L_1[S_r m_2 : S_r m_2 + F_r, S_c n_2 : S_c n_2 + F_l, :] \tag{3.10}$$

and the gradient from the layer L_2 to layer L_1 is:

$$\frac{\partial E(W)}{\partial L_1[S_r m_2 : S_r m_2 + F_r, S_c n_2 : S_c n_2 + F_l, :]} = \frac{\partial E(W)}{\partial L_2[m_2,n_2,k_2]} W[:,:,:,k_2] \tag{3.11}$$

There are different types of pooling operations which are possible, including max pooling, average pooling, L^p pooling, etc. Here, we give the forward propagation format of max pooling only, and the other methods are more or less the same. Given an arbitrary neuron $L_3[m_3, n_3, k_3]$, which performs max pooling on the convolutional layer L_1:

$$L_3[m_3,n_3,k_3] = max(L_1[S'_r m_1 : S'_r m_1 + F'_r, S'_c n_1 : S'_c n_1 + F'_l, k_3]) \tag{3.12}$$

where $max(\cdot)$ selects the maximum element from a matrix or a vector. The size of the window is $[F'_r, F'_l]$; the stride of sliding windows is $[S'_r, S'_c]$. The equation 3.12 tells us that the pooling layer does not contain parameters for tuning itself, i.e., it does not learn any knowledge during the training process.

3.4.2 Structure

Figure 3.5 displays the structure of a deep convolutional neural network employed in the experiment. The rounded blue rectangles (first row) are convolutional units and red rectangles (second row) are fully-connected units. Convolutional units extract features and fully-connected layers transform representation to labels. The structure enables good performance and takes care of efficiency as well, which has been adopted in [1, 71]. During feature extraction, two convolutional layers are embedded, and each follows a max pooling layer. Afterwards, two fully-connected layers transform the representation to a softmax layer, which performs the linear transformation to produce logits.

The size of input, is a $112 \times 112 \times 3$ (color image). The feature maps on the first

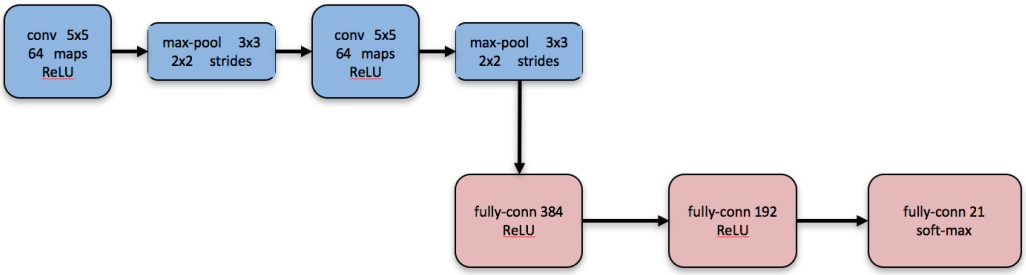


Figure 3.5: The structure of deep convolutional neural network employed in the experiment.

convolutional layer are 5×5 in size and total 64 of them (the first blue rounded rectangle). This is followed by a max pooling layer which has $3 \times 3 \times 1$ windows for each dimension, and the stride of the sliding window for each dimension has size $2 \times 2 \times 1$ (the second blue rounded rectangle). Then, local response normalization [71] is performed. A second convolutional layer has the feature maps at size 5×5 and totally 64 of them (the third blue rounded rectangle). Local response normalization goes before a max pooling layer, which has the same windows and sliding sizes. The red rounded rectangles on the second row of figure 3.5 are fully-connected layers. The first two fully-connected layers employ rectified linear units [88] as activation functions, and the last one uses the soft-max function for final predicted labels. The first fully-connect layer has 576×384 connections of weights while the second one has 384×192.

The structure (e.g., the number of convolutional layers, the number of fully-connected layers) and most hyper-parameters (e.g., the size of layers, the size of kernels) follow the AlexNet [71]. The number of outputs is different because we have 21 objects need to classify.

During the training process, the same layer is deployed across two GPUs (computations are isolated on different mini-batches) and the gradients are averaged for the final update. The experimental settings will be detailed later.

3.4.3 Training

The Section 3.4.1 has discussed the general neural networks, including feed-forward and back-propagation (the state-of-the-art method to train artificial neural networks [14]). This

section will talk about some techniques of the training, e.g., loss function, optimization, over-fitting.

Because the object recognition task deals the classes that are mutually exclusive (each image has only one label), multi-class logistic regression [14] applies a softmax non-linearity to the output of the network and computes the cross-entropy between the normalized predictions and a 1-hot encoding of the ground truth label [1]. Namely, equation 3.4 is replaced by:

$$\mathbf{Y}^{(m,n)} = \frac{\mathbf{Z}^{(m,n)}}{\sum_{m} \mathbf{Z}^{(m,n)}} \tag{3.13}$$

where $\mathbf{Z}^{(m,n)}$ is any entry in matrix $\mathbf{Z}^{(l)}$. Because the entry $t^{(n,k)}$ of the matrix T has to satisfy $t^{(n,k)} \in \{0, 1\}$ (0 indicates the n instance does not belong to the k label) for multi-class logistic regression, the likelihood function of equation 3.13 is:

$$p(\mathbf{T}|W) = \prod_{n} \prod_{k} y_{(n,k)}^{t_{(n,k)}} \tag{3.14}$$

Then, equation 3.5 is replaced by the error (cross-entropy error) function:

$$E(W) = -\ln p(\mathbf{T}|W) = -\sum_{n} \sum_{k} t_{(n,k)} \ln y_{(n,k)} \tag{3.15}$$

The partial derivatives of ∂y_k with respective to ∂a_j is:

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j) \tag{3.16}$$

3.4.3.1 Network training: Early stopping

When training a neural network, we are usually interested in obtaining a network with optimal generalization performance. However, all standard neural network architectures such as the fully connected multi-layer perceptrons are prone to over-fitting [48]. While the network seems to get better and better, i.e., the error on the training set decreases, at some point during training it actually begins to get worse again, i.e., the error on unseen examples increases.

A standard way to solve this problem is to split data into training and validation sets and as over-fitting will increase the error. As we see, the validation error can still go further down after it has begun to increase — plus in a realistic setting we do never know the exact generalization error but estimate it by the validation set error instead.

There are three ways to apply an early stopping criterion [93]:

1. Stop training as soon as the generalization loss on the validation set exceeds a certain threshold;

2. Stop training as soon as the quotient of generalization loss (on the validation set) and progress exceeds a certain threshold;

3. Stop training when the generalization loss (on the validation set) increased in certain successive strips.

Although the systematic differences between the three criteria are only small, the first criterion usually maximizes the probability of finding a good solution [93]. This thesis adopts the first way to implement early stopping. Note that the validation set is never used for weight adjustment.

3.4.3.2 Network training: Dropout

As the neurons of a neural network can be interpreted as parameters, the more neurons a neural network possess, the more likely it over-fits. The key idea of dropout is to randomly drop units (along with their connections) from the neural network during training [111]. There are two steps of dropout. At the training, dropout samples from some different "thinned" networks. At test time, it is infeasible to average the predictions from too many thinned models explicitly. Nevertheless, a very simple approximate averaging method works well in practice. The idea is to utilize a single neural net during test time without dropout. The parameters of the network are essentially scaled-down versions of the trained weights.

3.5 Experiments

Experiments were conducted to validate the aforementioned hypotheses. The first experiment used the model trained from a visual lifelog and perform object recognition on that visual lifelog. The second experiment employed a model trained from a visual non-lifelog and perform object recognition on a visual non-lifelog. If the performance of the first experiment is much worse than the second one, then the hypothesis 1 is confirmed. The third experiment adopted a model trained from a visual non-lifelog and perform object recognition on the visual lifelog. If the performance of the third experiment is extremely bad, then hypothesis 2 is supported, i.e., the visual lifelog and visual non-lifelog are two domains. The second experiment serves as a baseline for performance in the object recognition problem.

3.5.1 Design

Three experiments were performed to support the three research questions correspondingly. The first experiment, in order to explain the poor performance of the object recognition task on a lifelog with insufficient label data, we trained a model on a training set of visual non-lifelog and performed tests on a set of visual non-lifelog as well. The second and third experiments can be taken as a comparison to illustrate that model, trained from visual non-lifelog, will generalize poorly onto a visual lifelog if directly applied. Both of the models were trained on visual non-lifelog data, while the second one was used to predict on the non-lifelog, and the third one was used to predict on the lifelog.

As a consequence, the experiments feed on two domains of data: visual non-lifelog and visual lifelog. The non-lifelog images are split into three sets: the training set, validation set and test set. The visual lifelog has too little data to provide a validation set, and there is no standard solution for that. Considering that a validation set is used to select best hyper-parameters, a training set of visual lifelog data was re-used to make up the loss. In this sense, the visual lifelog data has only two parts: the training set and test set. The training set of visual lifelog data is always used as the validation set, except when used for training

already. All non-lifelog and lifelog images are of size 112×112, as described in detail, in Appendix A.

All experiments in the chapter focus on the object recognition task, essentially multi-class classification. Although other tasks are able to evaluate the hypotheses, object recognition is what we focus on here as it may explain results in other latent factors, because in all data sets used in the experiment, each image or example contains one and only one object and the task of the model is simply to predict which object the example contained, from the test set.

3.5.2 Settings for Experiments

During the training process, an individual example is employed with a series of random distortions to artificially increase the data set size: randomly flipping images from left to right, randomly distorting the image brightness, randomly distorting the image contrast. The flipping operation is performed with a probability of 0.5. Before random distortions, the values of pixels are converted into the range $[0, 1]$. Then $\delta \in [0, 1]$ is randomly generated for each image and added to every pixel of the image as a random brightness value. The sum is rescaled at the end of the process. Random contrast follows the same approach as random brightness. In random contrast, each pixel value is $(x - mean) * factor + mean$ where mean is calculated for each channel and $factor \in [0, 1]$ is randomly chosen for each image.

The training set batch size (the number of training examples in one iteration) is set to be 200. The average moving decay is 0.9999. The number of epochs per decay is 350. The factor of learning rate is 0.1 and the initial learning rate is 0.1.

3.5.3 Predict Lifelog using Lifelog

This experiment trained models on a training set of lifelog data and recognizing objects from images on a test set of lifelog data. Figure 3.6 plots the entropy loss of both training set and test set during training. Note that the test set can not be involved in deciding values of either model nor hyper-parameters. Owing to weight decay in the training process, the

model does not over-fit.

We can observe that the F1-score of the learning set rises rapidly as the number of training examples is small. We can also observe that learning curves oscillate badly because of several reasons. The first reason is the solution surface is not global convex [18], thus learning curve is not smooth. The second reason is the number of training iterations is relatively small, since the curve will look smoother if the number of iterations is larger. The third reason is the results are plotted on the figure at a step of 1000.

There is a notable phenomenon in the figure: there are several obvious decrements (e.g., at an iteration around 15,000, both the training loss and test loss drop). I think it is because the model learns bad parameters. In addition, the small number of iterations and sparse points are plotted to make it more obvious.

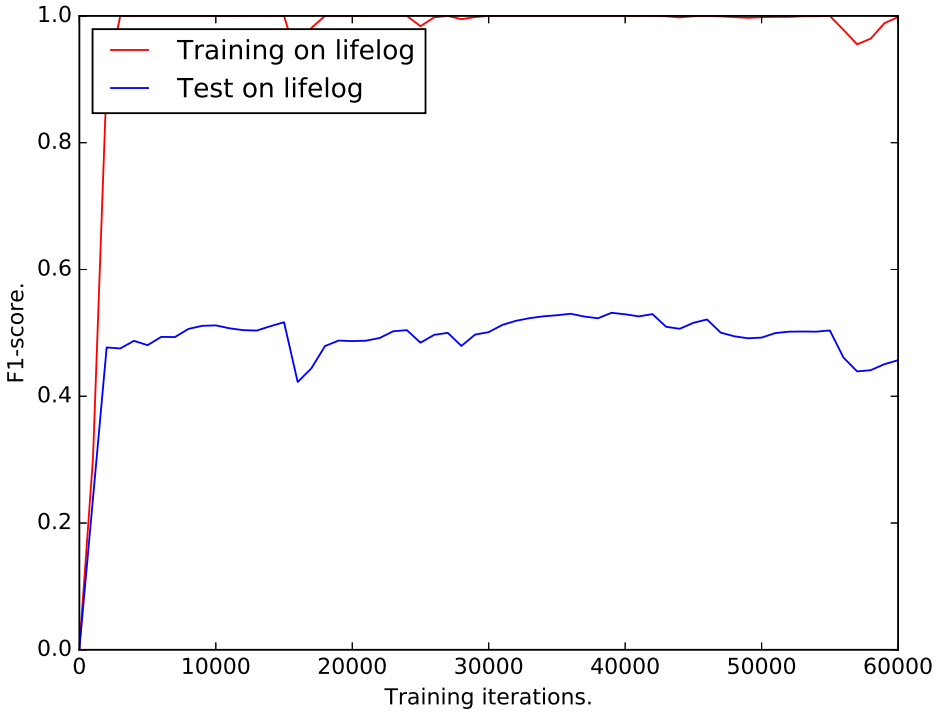


Figure 3.6: The learning curve of training using a convolutional neural network.

Table 3.1 displays the performance, in terms of a confusion matrix, at the epoch of 60,000. Despite the small training set, the weight decay employed prevents over-fitting and

thus overall performance is not too bad. The table suggests the classes with large number of training examples, usually have good performance while those with a small number of training examples have bad performance. The table reveals that the model performs much better on some classes than others. For example, class "bicycle" has no prediction (i.e., it has no true positive) while class "glass" has very high accuracy. Thus it is likely that the prediction will be more accurate on classes with more training examples.

true \ predicted	air	bic	bot	bui	car	cha	cup	dis	doo	fac	gla	han	lam	mob	mot	pap	per	sig	tra	tvm	win
aircon	362	0	1	1	1	0	0	0	0	1	15	16	6	1	0	1	0	12	0	7	0
bicycle	0	0	0	2	1	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0
bottle	1	0	76	6	4	0	0	1	0	0	25	50	9	0	0	2	8	2	0	24	0
building	0	0	2	300	10	3	1	0	2	5	22	132	18	1	0	5	31	15	0	39	0
car	0	0	2	9	122	1	1	0	0	4	8	77	6	0	1	2	9	1	0	9	0
chair	0	0	3	1	2	85	1	0	0	0	2	35	1	0	0	0	4	9	0	0	0
cupboard	1	0	2	4	1	1	256	0	1	0	4	21	3	0	0	6	4	3	0	7	0
dish	0	0	0	0	0	0	0	3	0	2	4	35	3	0	0	3	0	2	0	0	0
door	0	0	0	23	1	0	6	0	27	2	3	50	12	0	0	5	10	11	0	9	0
face	0	0	0	8	2	1	0	0	0	265	14	128	19	0	0	3	4	4	0	3	1
glass	1	0	3	3	1	1	1	0	0	1	524	79	14	0	0	7	9	5	0	16	0
hand	3	0	7	14	6	4	5	2	1	37	34	779	28	0	0	18	15	12	0	21	0
lamp	1	0	1	1	0	0	5	1	0	4	6	33	1753	0	0	14	0	2	0	17	1
mobilephone	1	0	1	3	1	0	0	0	2	1	9	32	5	22	0	7	5	9	0	18	0
motorbike	0	0	0	4	2	0	0	0	0	2	3	25	0	0	1	0	3	1	0	1	0
paper	0	0	2	8	0	5	5	1	1	5	5	59	45	1	0	126	1	0	0	38	0
person	1	0	1	63	14	4	2	0	4	12	131	368	24	3	0	9	246	18	0	39	1
sign	5	0	2	26	3	2	4	0	3	9	37	96	35	2	0	7	13	117	0	44	0
train	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0
tvmonitor	3	0	5	26	1	2	21	0	3	2	43	76	36	0	0	18	13	35	0	734	1
window	0	0	0	5	0	0	1	0	1	0	17	13	19	0	0	6	2	1	0	29	16

Table 3.1: Confusion matrix produced by training on lifelog data, at 60,000 iterations, and test on lifelog data.

Figure 3.7 illustrates that there is a positive correlation between the number of training examples and the accuracy of a class. The evidence approves insufficient training examples of visual lifelog leading to unsatisfactory performance, which agrees on a common phenomenon: a dumb algorithm with lots and lots of data beats a clever one with modest amounts of it [38].

3.5.4 Predict Non-lifelog Data using Non-lifelog Data

A possible concern may arise based on the model used, a convolutional neural network, instead of the data itself. The experiment trained models on the training set of non-lifelog images and recognizes objects from images on the test set of non-lifelog data. Table 3.2 displays the performance, as confusion matrix, at epoch of 60,000. The table supports the fact that although the number of examples for each class of non-lifelog data is also slightly

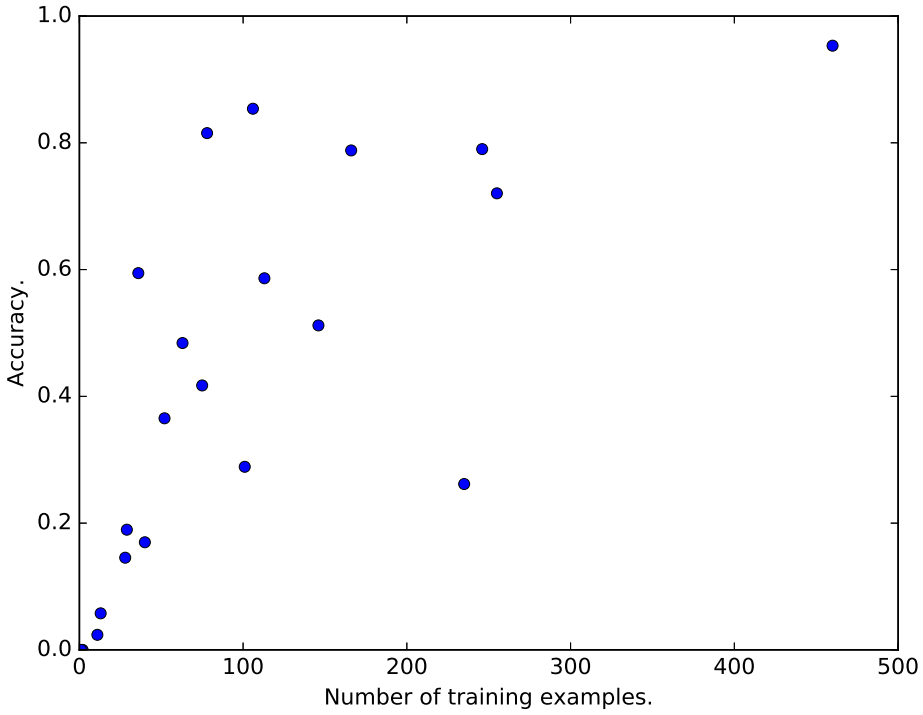


Figure 3.7: The correlation of the number of training examples and corresponding accuracy of each class in the experiment of using lifelog data to predict lifelog data.

imbalanced, all classes are of satisfactory performance. The reason is that no class has a very small number of training examples.

true \ predicted	air	bic	bot	bui	car	cha	cup	dis	doo	fac	gla	han	lam	mob	mot	pap	per	sig	tra	tvm	win
aircon	101	2	0	7	2	2	6	2	8	2	6	1	5	4	0	0	0	9	8	7	5
bicycle	0	116	5	11	3	4	2	4	3	0	4	1	3	2	4	1	1	3	17	3	1
bottle	2	6	113	7	5	4	1	4	6	0	2	1	10	3	0	1	0	2	11	2	5
building	1	0	5	147	1	0	3	1	10	1	5	2	8	1	1	0	1	4	9	2	1
car	0	5	1	6	75	3	0	1	0	1	4	0	2	1	0	0	0	3	5	0	0
chair	2	6	3	9	3	111	2	0	2	3	3	1	3	2	0	1	3	7	16	3	1
cupboard	0	1	2	12	0	8	102	2	10	0	3	0	4	0	3	1	0	6	3	3	1
dish	0	3	4	8	5	4	2	120	8	5	12	3	3	2	3	0	2	6	4	2	2
door	2	3	1	12	0	2	7	2	161	4	5	2	8	3	5	0	3	10	6	3	4
face	0	4	4	8	2	3	0	2	4	129	2	2	7	1	2	0	4	5	5	5	1
glass	0	2	6	9	1	4	1	2	7	4	134	1	5	1	0	0	2	7	2	5	2
hand	0	2	10	4	2	4	0	3	0	9	5	131	8	3	3	0	0	3	5	4	2
lamp	0	2	4	17	1	2	2	4	8	3	6	4	138	5	1	1	2	8	7	3	4
mobilephone	3	2	2	6	1	5	3	1	5	4	7	2	6	118	2	0	0	5	5	4	2
motorbike	0	7	3	8	1	6	0	4	1	5	5	0	5	1	108	0	4	3	9	1	0
paper	1	1	1	2	1	2	0	0	2	1	0	0	4	0	0	28	0	3	0	1	1
person	1	5	5	11	3	3	3	3	8	11	4	4	3	1	2	1	92	2	11	2	1
sign	0	3	4	15	2	5	4	3	9	1	1	0	6	1	1	1	2	127	12	5	0
train	0	3	3	9	3	0	1	0	6	1	2	1	1	1	2	0	0	2	143	2	1
tvmonitor	0	4	3	6	2	2	2	4	3	0	3	2	4	0	3	0	0	5	7	142	0
window	0	2	1	4	1	1	0	2	9	2	2	0	1	1	0	1	0	4	5	2	67

Table 3.2: Confusion matrix produced by training on non-lifelog data, at 60,000 iterations, and test on non-lifelog data. All classes in the table achieve satisfactory performance.

By performing the object recognition task entirely on visual non-lifelog data, the experiment defends the scrutiny of convolutional neural networks on the task, and meanwhile, it suggests inadequate training leads to models that cannot generalize well. Moreover, the performance on non-lifelog data here is expected to be the best that the lifelog experiment can reach.

3.5.5 Predict Lifelog using Non-lifelog

An intuitive solution to supplement the training set of lifelog data is directly using the model, which is trained on non-lifelog data, to perform tasks on non-lifelog data. The model was used to predict labels of non-lifelog shared in the last experiment, as detailed in figure 3.8. The figure plots the entropy loss of the non-lifelog training set, the non-lifelog validation set, and the lifelog training set. As we mentioned in Section 3.5.3, the points are plotted every 1,000 iterations. Because test sets are unseen during the training session, validation sets were used to estimate. Note also, as previously mentioned, in lifelog data, the training set is used as a validation set because of a lack of data. From the figure, we can see that the learning curve of the training set (blue line) approach is rather low because back-

propagation works only on the loss of training examples. Obviously, the loss of validation set of non-lifelog data oscillates below training set of lifelog data.

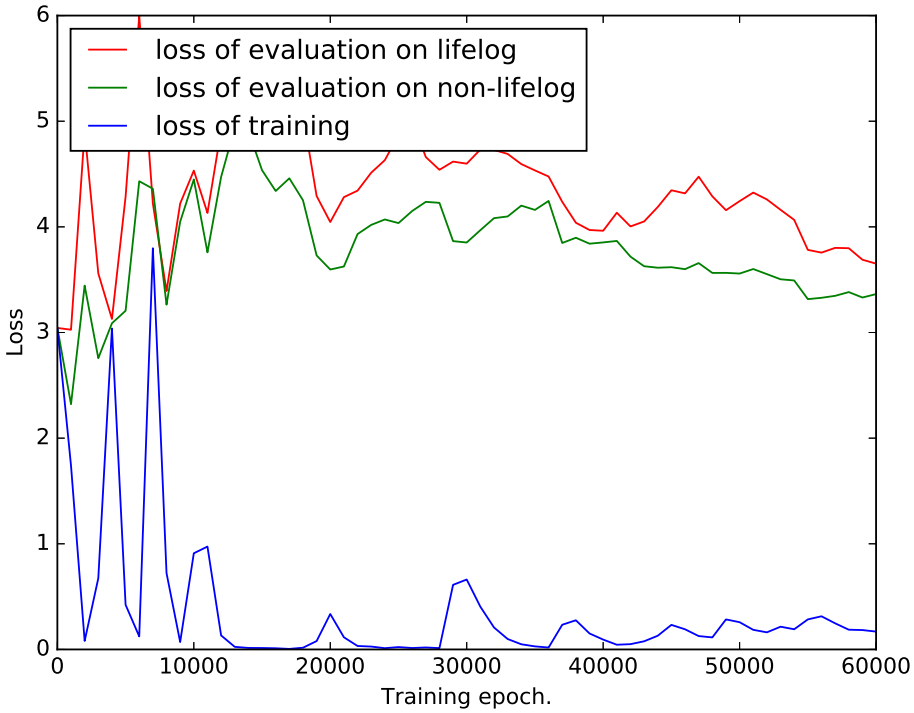


Figure 3.8: The learning curve of training using a convolutional neural network.

The figure tells us that, during the training session, the loss of training set of lifelog is minimal, which is within expectation, because the loss function minimizes the loss of training set. More importantly, the loss of the validation set of non-lifelog data is always smaller than the loss of the training set of lifelog data, which indicates the distribution of lifelog data is different to that of non-lifelog data.

Table 3.1 displays the performance, as a confusion matrix, at epoch of 60,000. The result from the table is much worse than Table 3.2, despite the fact they derived from the same model trained on the same data set. The table and figure confirm the hypothesis 2 that visual non-lifelog and visual lifelog data are of different domains and have different characteristics.

The experiment also points out that it is implausible to perform the task on a visual

true \ predicted	air	bic	bot	bui	car	cha	cup	dis	doo	fac	gla	han	lam	mob	mot	pap	per	sig	tra	tvm	win
aircon	6	0	4	51	0	8	16	93	0	79	49	11	16	7	0	12	14	0	48	10	0
bicycle	0	0	0	0	0	0	0	0	0	4	0	1	1	0	0	0	0	0	2	0	0
bottle	0	0	0	2	0	9	15	21	0	17	33	6	26	2	0	0	8	0	60	9	0
building	2	0	4	20	0	8	54	7	5	77	56	30	12	4	2	0	48	0	235	21	1
car	0	0	0	10	0	6	3	4	0	71	23	60	7	0	0	0	21	0	36	11	0
chair	0	0	1	0	0	6	4	18	0	39	12	8	8	1	0	0	3	0	31	12	0
cupboard	3	0	12	7	0	7	20	1	17	68	60	20	18	1	0	5	12	0	33	27	3
dish	0	0	0	0	0	0	1	13	0	10	7	11	2	0	0	0	0	0	8	0	0
door	1	0	0	5	0	4	45	5	3	14	16	8	9	1	0	0	5	0	14	29	0
face	0	0	0	19	0	1	13	30	0	187	63	50	54	0	0	0	13	2	19	1	0
glass	0	0	4	15	2	46	23	68	2	79	150	9	7	4	3	1	28	0	195	29	0
hand	1	0	1	8	1	13	29	77	1	280	113	235	29	4	0	0	72	0	101	21	0
lamp	3	0	20	282	0	2	14	29	1	482	146	528	122	0	0	53	102	2	38	15	0
mobilephone	0	0	2	0	0	2	0	12	0	34	12	22	3	0	0	0	12	0	8	9	0
motorbike	0	0	0	0	0	2	0	5	0	7	12	0	1	0	0	0	5	0	10	0	0
paper	6	1	5	6	1	9	6	48	3	34	24	81	18	0	0	10	13	3	26	8	0
person	1	0	3	6	0	19	31	98	0	253	127	63	10	3	0	0	93	0	209	24	0
sign	3	0	4	14	0	3	18	67	3	96	47	31	23	1	0	2	18	5	39	30	1
train	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0
tvmonitor	6	0	9	30	10	12	89	47	20	82	68	95	37	11	2	13	25	13	225	223	2
window	0	0	1	1	0	1	4	15	0	23	8	15	4	1	0	0	4	1	18	14	0

Table 3.3: Confusion matrix, which is produced by training on non-lifelog data, at 60,000 iterations, and tested on lifelog data. Prediction results for all classes are poor.

lifelog dataset using a model built from a visual non-lifelog training data.

3.6 Discussion and Conclusions

The chapter pioneers the insight that a visual lifelog can be considered to be in a different domain to a visual non-lifelog and validates the hypothesis using experiments across a visual lifelog and a visual non-lifelog. Experiments are carefully designed to support the hypothesis. The chapter also contributes the format of vectorization computation for fully-connected neural networks and convolutional neural networks which usually appear in the form of summation in most publications.

Chapter 4

Enhancing Visual Lifelog for Object Recognition with Visual Non-lifelog

Chapter 3 introduced the concept of domain adaptation and used several examples from the areas of natural language processing and computer vision to help present the background to how it operates. Moreover, chapter 3 indicated that visual lifelogs and visual non-lifelogs are two separate domains and that visual lifelogs require a much larger dataset than does visual non-lifelogs for learning powerful models, typically with millions of parameters.

This chapter (chapter 4) starts from the perspective of domain adaptation and reveals how cross-domain algorithms could help the object recognition task on visual lifelogs supplemented with non-lifelog images.

4.1 Definitions and Problem Formulation

Following the perspective of machine learning, a rigorous definition of domain adaptation derives from chapter 3, along with the symbolic expressions used in Section 4.2 and Section 4.3. Later in this chapter, the problem of solving the object recognition task for lifelogs using visual non-lifelogs is formulated using domain adaptation concepts.

4.1.1 A Rigorous Definition of Domain Adaptation

Consider the classification task where $\mathcal{X}$ denotes the features space (i.e. the set of all possible observations) and $\mathcal{Y}$ the output space (i.e. the set of all possible labels). Over $\mathcal{X} \times \mathcal{Y}$, the source domain D_s is abundant with the labeled data while the target domain D_t has insufficient labeled data.

The joint distribution of the source domain $P_s(X, Y)$ and the joint distribution of the target domain $P_t(X, Y)$ are different ($P_s(X, Y) \neq P_t(X, Y)$), despite the fact that both joint distributions are unknown [66]. Observations sampled from the same domain are considered to be independent and identical distributed. $P_s(X)$, $P_t(X)$, $P_s(Y)$ and $P_t(Y)$, respectively, standing for the true marginal distributions of X and Y in the source and the target domains. Similarly, $P_s(X|Y)$, $P_t(X|Y)$, $P_s(Y|X)$ and $P_t(Y|X)$ denote the corresponding true conditional distributions in the two domains.

Lowercase x (an observation) and y (a class label) denote a specific value of X and of Y respectively. A pair (x, y) is referred to as a labelled instance. Without any ambiguity, $P(X = x, Y = y)$ or simply $P(x, y)$ should refer to the joint probability of $X = x$ and $Y = y$. Similarly, $P(X = x)$ (or $P(x)$), $P(Y = y)$ (or $P(y)$), $P(X = x|Y = y)$ (or $P(x|y)$) and $P(Y = y|X = x)$ (or $P(y|x)$) also refer to probabilities rather than distributions.

$D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ is used to denote the set of labelled instances in the source domain. In the target domain, we assume that we always have access to a large amount of unlabelled data, and we use $D_{t,u} = \{x_i^{t,u}\}_{i=1}^{N_{t,u}}$ to denote this set of unlabelled instances. Sometimes, we may also have a small amount of labelled data from the target domain, which is denoted as $D_{t,l} = \{(x_i^{t,l}, y_i^{t,l})\}_{i=1}^{N_{t,l}}$. In the case when $D_{t,l}$ is not available, the problem is referred to as unsupervised domain adaptation, while when $D_{t,l}$ is available, the problem is referred to as supervised domain adaptation.

4.1.2 Problem Formulation in Lifelogging Object Recognition

The machine learning task for object recognition in visual lifelogs is the focus of the chapter. The regions of objects from images are cropped and re-sized to the same size. The re-sized

regions are observations. Labels are used to distinguish different objects, and each object has only one label.

4.2 Background to Domain Adaptation Approaches

4.2.1 Related Fields

Domain adaptation has similar settings with a number of prevalent machine learning directions, as depicted in Figure 4.1. In this Figure, the difference between tasks is shown as the vertical column while the difference between observations or availability of training data is shown horizontally. As Chapter 3 introduced, compared with transfer learning, domain adaptation targets that the machine learning tasks between the source and target domains are identical [91].

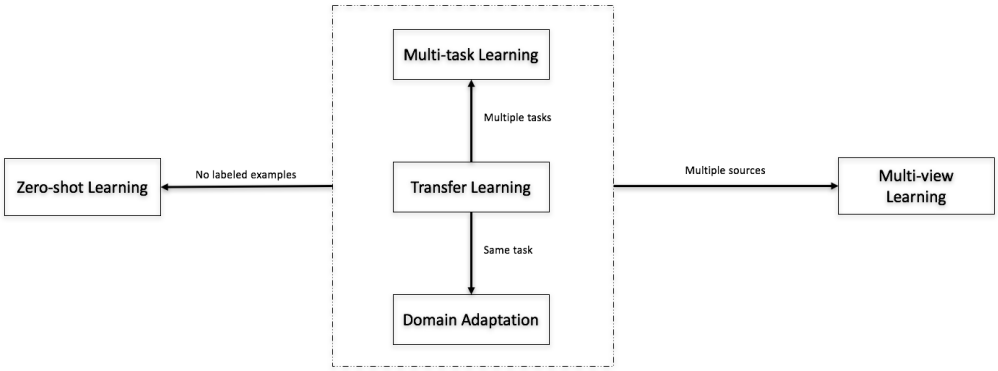


Figure 4.1: The relationship between domain adaptation and other popular relevant machine learning topics

Multi-task learning [6, 131], a.k.a. predicting multivariate responses, considers the generalization of models across multiple related machine learning tasks. Considering a spam filtering application: in a large-scale email system, users create their own email filters (classifiers), which can hardly amount to a global filter that can be used by all users (tasks). Multi-task learning solves the problem by creating a collaborative filter that uses labeled data from individuals.

Zero-shot learning [90, 100, 108] refers to machine learning scenarios in which new classes appear after the learning stage (equivalently, the potential class values are omitted from training examples). A human could easily recognize a new object if s/he obtains the corresponding description without seeing it beforehand.

Multi-view learning techniques are necessary when the data is described by multiple distinct feature sets because single-view learning algorithms tend to over-fit on these high-dimensional datasets [124, 128]. For example, in video classification, the videos can be characterized with respect to vision, audio and even attached comments. For example, most article search engines consider titles, keywords, authors, publishers, dates and content.

4.2.2 Approaches to Domain Adaptation

There are several ways to categorize domain adaptation algorithms [66, 76, 91]. Relevant relation is adopted to categorize domain adaptation approaches[1]: instance-based methods (re-weighting), iterative methods (adjustment), and feature-based methods (representation learning).

Instance-based approaches to domain adaptation assume that distribution shift is caused by sampling bias/shift between marginals and certain parts of the data in the source domain can be re-used for learning in the target domain by re-weighting [123]. Re-weighted or selected instances can reduce the discrepancy between the source and target domains. To correct a sample bias by re-weighting the source labeled data, the source instances close to the target instances are more important.

The parameter-transfer approach transfers knowledge across different domains by sharing some parameters or prior distributions of the hyper-parameters of the models, which is highly dependent on the employed model. The basic assumption of the relational-knowledge-transfer is that some relationships among the data in the source and target domains are similar but overly strong.

Chapter 5 will detail the representation learning and only highly-correlated parts are discussed here. The intuitive idea is to build a common representation between the two

[1]Part of the inspiration for this comes from https://epat2014.sciencesconf.org/conference/epat2014/pages/slides_DA_epat_17.pdf

domains, which makes the two domains appear to have similar distributions, thus enabling effective domain adaptation [10]. Corresponding models [121, 122] directly minimize a trade-off between source-target similarity and source training error (i.e., minimizing the difference between the source and target domains, while at the same time maximizing the margin of the training).

As previously discussed, the cause of the domain adaptation problem is the difference between $P_t(X, Y)$ and $P_s(X, Y)$. Note that while the representation of Y is unchanged, the representation of X can change if we use different features. Such a representation change of X can affect both the marginal distribution $P(X)$ and the conditional distribution $P(Y|X)$. One can assume that under some change of representation of X, $P_t(X, Y)$ and $P_s(X, Y)$ will become the same.

Formally, let $g : \mathcal{X} \rightarrow \mathcal{Z}$ denote a transformation function that transforms an observation x representing the original form into another form $z = g(x) \in \mathcal{Z}$. We define variable Z and an induced distribution of Z that satisfies $P(z) = \sum_{x \in \mathcal{X}, g(x)=z} P(x)$. The joint distribution of Z and Y is then:

$$P(z, y) = \sum_{x \in \mathcal{X}, g(x)=z} P(x, y).$$

If a transformation function g can be found so that under this transformation, we have $P_t(Z, Y) = P_s(Z, Y)$, then we no longer have the domain adaptation problem since the two domains have the same joint distribution of the observation and the class labels. The optimal model $P(Y|Z, \theta^*)$ we learn to approximate $P_s(Y|Z)$ is still optimal for $P_t(Y|Z)$. Note that with a change of representation, the entropy of Y that is conditional on Z is likely to rise from the entropy of Y conditional on X, because Z is usually a simpler representation of the observation than X, and thus less information is encoded. In other words, the Bayes error rate typically increases under a change of representation. Accordingly, the criteria for good transformation functions include not only the distance between the induced distributions $P_t(Z, Y)$ and $P_s(Z, Y)$ but also the incremental amount of the Bayes error rate.

4.3 Domain-Adversarial Training by Back-Propagation

4.3.1 Inspiration from Representation Learning

The idea is that projecting domains of different distributions to same space could be traced back to canonical correlated analysis (CCA) [61], which is a linear algorithm [112]. Linear representations [27, 61, 121] cannot generate sufficiently flexible representations. Thus, more recently, non-linear representations have been studied, including neural network representations [80] and, most notably, the state-of-the-art mSDA [26]. This has mostly focused on exploiting the principle of robust representations, based on the denoising auto-encoder paradigm.

Inspiration was drawn from the theory of domain adaptation, which suggests that for the effective domain transfer to be achieved, predictions must be made based on features that cannot distinguish between the training (source) and test (target) domains [9, 10]. Representation learning for domain adaptation may follow the principle that the learned representations should help the label prediction (discriminativeness) but fight against domain prediction (domain invariance). In other words, the label predictor that predicts class labels is used during training and testing time, and the domain classifier that discriminates between the source and the target domains are used only during training. Whereas the classifier parameters are optimized to minimize their error on the training set, the parameters of the latent deep feature mapping are optimized to minimize the label classifier loss and to maximize the domain classifier loss.

Figure 3.1 shows that the similar but still different pixel distributions on the visual non-lifelog and visual lifelog belong to different domains. The standard convolutional neural networks commit to learning effective representations for only one domain (visual lifelog or visual non-lifelog), and features that are effective to only one domain are discarded. Employed with domain-adversarial concepts, the convolutional neural network can transform features that work well on only one domain to the same space that can be applied to both domains. In this way, those features that are thrown away by the standard convolutional neural network are kept and transformed to fit both the visual lifelog and non-lifelog; i.e.,

features are extracted and transformed at the same time.

4.3.2 Structure and Mathematical Expressions

From the discussion of Section 4.3.1, we can see that feature-based domain adaptation approach is favorable because it separates knowledge transfer from a specific task. In order to keep the consistency of experiments, neural networks are preferred as models for the rest experiments. A paper [47] proposed a representation learning model based on neural networks which perfectly fits.

Domain-adversarial networks depend on the following three components: feature extractor, label classifier, and domain predictor. Based on this principle, the arbitrary architectures could be fabricated depending on specific tasks. The adopted concrete model is displayed in Figure 4.2, which extends the model from chapter 3 to make reasonable comparisons. In the figure, the blue components act as non-linear representation learning; and the green ones perform label classification while the red ones perform domain prediction. Representation learning comprises convolutional and max-pooling layers. Both label classifier and domain predictor consist of perceptions.

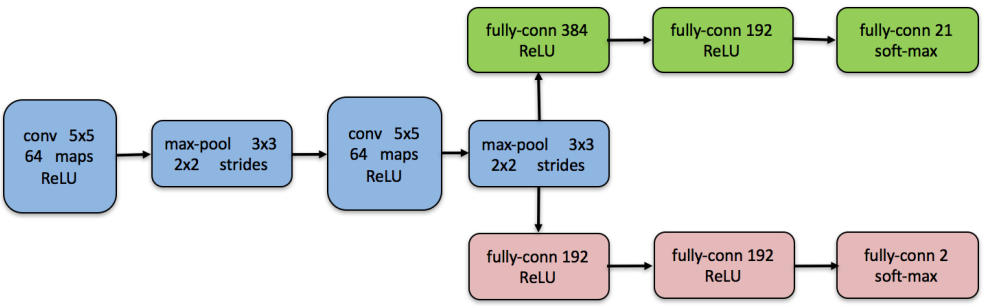


Figure 4.2: The concrete domain-adversarial model

Let $G_f(\cdot;\theta_f)$ be the D-dimensional neural network feature extractor, with parameters θ_f. Additionally, let $G_y(\cdot;\theta_y)$ be the part that computes the label prediction (output layer of networks), with parameters θy, and $G_d(\cdot;\theta_d)$ now corresponds to the computation of the

domain prediction output of the network, with parameters θ_d. The prediction loss is:

$$L_y^i(\theta_f, \theta_y) = L_y(G_y(G_f(x_i; \theta_f); \theta_y), y_i) \tag{4.1}$$

and the domain loss is:

$$L_d^i(\theta_f, \theta_d) = L_d(G_d(G_f(x_i; \theta_f); \theta_d), d_i) \tag{4.2}$$

where the x_i is the input. Combined with Figure 4.2 and the definition in equation 3.9 and equation 3.12, $G_f = L_3^{(4)}(L_2^{(3)}(L3^{(2)}(L_2^{(1)}(x_i))))$.

The training is actually optimized as follows:

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n}\sum_{i=1}^{n} L_y^i(\theta_f, \theta_y) - \lambda(\frac{1}{n}\sum_{i=1}^{n} L_y^i(\theta_f, \theta_d) + \frac{1}{n'}\sum_{i=n+1}^{N} L_y^i(\theta_f, \theta_d)) \tag{4.3}$$

by finding the saddle points $\hat{\theta}_f$, $\hat{\theta}_y$, $\hat{\theta}_d$ such that:

$$(\hat{\theta}_f, \hat{\theta}_y) = \underset{\theta_f, \theta_y}{\operatorname{argmin}} E(\theta_f, \theta_y, \hat{\theta}_d) \tag{4.4}$$

and

$$\hat{\theta}_d = \underset{\theta_d}{\operatorname{argmin}} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d) \tag{4.5}$$

The first term in equation 4.3 is the label classifier loss, and the latter term is the domain predictor loss on both the labelled and unlabelled data of the target domain.

The saddle points defined by equations 4.3 and equation 4.4 can use the following gradient updates to find:

$$\theta_f \leftarrow \theta_f - \mu(\frac{\partial L_y^i}{\partial \theta_f} - \lambda\frac{\partial L_d^i}{\partial \theta_f}) \tag{4.6}$$

$$\theta_y \leftarrow \theta_y - \mu\frac{\partial L_y^i}{\partial \theta_y} \tag{4.7}$$

$$\theta_d \leftarrow \theta_d - \mu\frac{\partial L_d^i}{\partial \theta_d} \tag{4.8}$$

where μ is the learning rate and λ is the adaptation factor. Notably, the trivial equations(equation 4.3, equation 4.4, equation 4.5, equation 4.6, equation 4.7, and equation 4.8) can be found in this paper [47].

It is easy to understand that both the label classifier (equation 4.7) and domain predictor (equation 4.8) try to perform good predictions given the new representations. The second term in equation 4.6 contains two parts: the first part makes it easier for the new representation to perform label classification (discriminativeness), and the second part leads to difficulty in distinguishing the new representation of domains (domain invariance).

Unlike many previous papers on domain adaptation that used fixed feature representations [26, 27], combining domain adaptation and deep feature learning within one training process makes the learning of representations and classifiers as a whole. The goal is to embed domain adaptation into the learning representation process, such that the final classification decisions are made based on features that are both discriminative and invariant to the change of domains, i.e., so they have the same or very similar distributions in the source and the target domains.

As distinct from the introduction of gradient reversal layer [46], this chapter runs experiments on the code implemented exactly as in the gradient updates equations 4.6, 4.7, 4.8.

4.4 Experiments

As in chapter 3, the task recognizes 21 objects from images, and each object contains one object occupying most of the image space. The F1-score is adopted in order to evaluate the prediction performance.

Chapter 3 discussed three experiments: predicting the visual lifelog using a visual lifelog, predicting the visual non-lifelog using a visual non-lifelog, and predicting the visual non-lifelog using a visual lifelog. Each experiment employed training examples from a single domain.

In contrast, in this chapter, the proposed algorithm uses training examples from both the

source and target domains to exploit knowledge from both of them. A baseline approach shares the same training set but performs a prediction directly by using the trained model. To provide a justified comparison, the setting of the hyper-parameters is the same as that used in chapter 3, unless explicitly mentioned.

4.4.1 Experimental Design

The proposed domain adaptation algorithm uses training examples from both domains and tests on a test set from the visual lifelog. The baseline approach uses the same training set and directly performs predictions using the trained model to show the effectiveness of the proposed domain adaptation algorithm. The three experiments from chapter 3 also provide a comparison.

The first experiment, which directly trains models on the training examples from both the visual non-lifelog and the visual lifelog, makes predictions using the neural network structure from chapter 3 and provides a baseline. Two domain adaptation-based model experiments, one with a fixed learning rate and the other with an adaptive learning rate, display the efficacy of the domain adaptation approach.

The lifelog data and non-lifelog data used in the experiments are described in Appendix A.

4.4.2 Experimental Settings

Training examples from the visual non-lifelog and the visual lifelog are randomly mixed, and the number of training examples from the visual non-lifelog is greater than that from the visual lifelog. Following the convention of chapter 3, the inputs are objects that have been cropped from raw images and re-scaled to a size of 112×112. The visual lifelog set cannot provide a validation set, and the final result is evaluated with the model that performs the best.

The domain-adversarial training model, which follows the default setting [47], trains with 0.9 momentum of the stochastic gradient descent and an adaptive learning rate as

follows:

$$r_p = \frac{r_0}{1 + \alpha \cdot p^{\beta}} \tag{4.9}$$

where p is the training progress linearly changing from 0 to 1, $r_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$ (the schedule was optimized to promote convergence and low error on the source domain). The adaptation factor λ is assigned using the following schedule:

$$\lambda_p = \frac{2}{1 + e^{-\gamma \cdot p}} - 1 \tag{4.10}$$

where γ is set to 10 [47].

The fixed learning rate for the domain-adversarial training model is set to 0.1.

The implementation is easy to run on the two-GPU nodes, where a GPU updates the gradient of the label classifier and the other updates that of the domain predictor. After all of the gradients are calculated, their values are constrained between -1 and 1. The implementation has been tested on two types of GPUs: one is 32 NVIDIA K20X GPU cards and the other is NVIDIA GTX 660 Ti GPU cards.

4.4.3 The Baseline Training from Training Examples of Both Domains

Figure 4.3 displays the model performance that used the training examples from both the visual non-lifelog and the visual lifelog on different datasets in terms of their F1-score. The performance of the training set in both Figure 3.8 and Figure 3.6 (most values above 0.8) is obviously much higher than that in Figure 4.3 (all values lower than 0.8). The model in this Figure is trained on a mixture of training examples from both non-lifelog and lifelog data. The F1-score is adopted to evaluate the performances of the model on different datasets. The order of performance on a data set is (from best to worst): training examples from both non-lifelog and lifelog data, test from non-lifelog data, test from lifelog data, and validation from non-lifelog data. Using the training examples sampled from two distinct distributions, the neural network structure that works well for data from a single distribution cannot learn the mixed training examples equally as well as either of the other models.

Figure 4.3 indicates the following long-term tendencies: the F1-scores of the visual

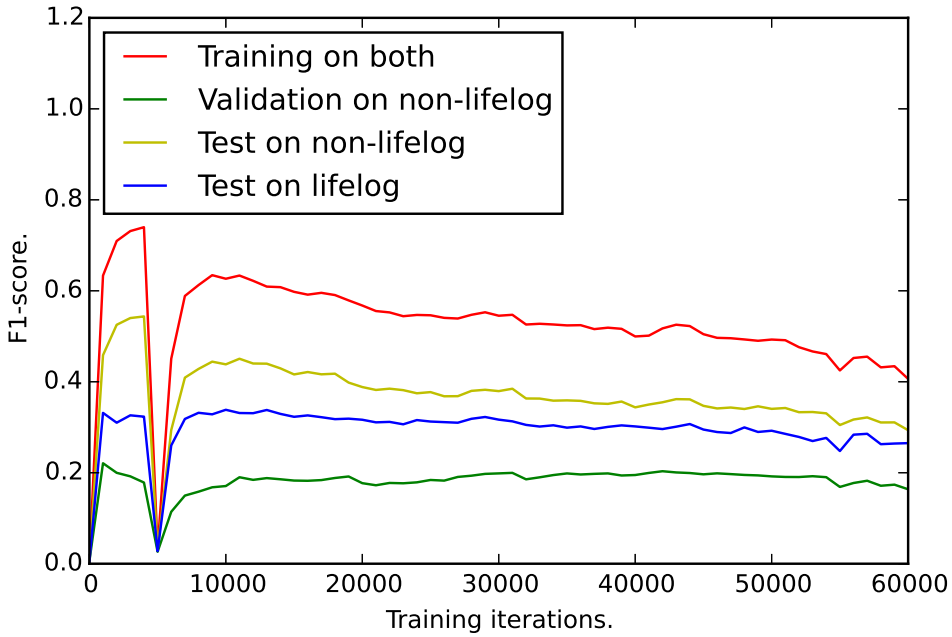


Figure 4.3: The learning curve for training.

lifelog and the visual non-lifelog become closer, and the F1-scores of both domains gradually decrease. The phenomenon occurs because the robust representations that the convolutional neural network is seeking are feasible for both domains. Those features that work well on only one domain are removed. In addition, as the training employs batch gradients, the temporary performance decrease steeply in the middle. Because of the complexity of convolutional neural networks, it is hard to discuss the reasons precisely. I have to point out that it is rare when training convolutional neural networks on a single dataset. Several reasons could raise the phenomenon. Examples in the training set with wrong labels will destroy well-trained models and increase the loss. Considering the numbers of training examples of the classes are imbalanced, insufficient sampling may cause the steep decrease, too.

The training set achieves the best performances, as expected, among the different datasets, as shown in Figure 4.3. The number of training examples of the non-lifelog overwhelms that of the lifelog, and the mixture training set tends to act closer to the visual non-lifelog. The different-domain attribute decides that the test set from the non-lifelog beats that from

the lifelog. The validation set from the non-lifelog is randomly chosen, and the number of examples is relatively small. The nature of test set from the non-lifelog cannot be exactly determined; thus, it could have better or worse performance than the test set from the same domain. Figure 4.3 illustrates the worst results among all datasets.

Notice that the F1-score of the training set is for the entire selection instead of a batch selection. In addition, the performance of the visual lifelog test set will not help in validation, but it does display the overall performance and effectiveness of the result. The model produced from the last iteration of the training process provides the results.

true \ predicted	air	bic	bot	bui	car	cha	cup	dis	doo	fac	gla	han	lam	mob	mot	pap	per	sig	tra	tvm	win
aircon	354	0	1	1	0	0	0	0	0	1	3	1	12	0	0	0	0	51	0	0	0
bicycle	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	7	0	0	0
bottle	0	0	45	1	0	0	1	0	0	1	11	0	11	0	0	0	0	136	0	2	0
building	0	0	1	55	2	4	0	0	1	0	3	0	14	0	0	0	6	487	0	13	0
car	0	0	2	8	70	14	0	0	0	2	12	1	7	0	0	0	8	127	0	1	0
chair	0	0	4	0	0	2	0	0	0	0	12	2	12	0	0	0	0	111	0	0	0
cupboard	0	0	3	0	0	0	196	0	0	0	4	0	16	0	0	0	1	93	0	1	0
dish	0	0	1	0	0	0	0	0	0	0	11	3	16	0	0	0	3	17	0	1	0
door	1	0	1	0	0	0	1	0	2	0	2	1	6	0	0	0	0	139	0	6	0
face	0	0	1	0	0	0	0	0	0	225	68	5	57	0	0	0	22	74	0	0	0
glass	0	0	42	1	0	0	0	0	0	1	252	3	33	0	0	0	2	329	0	2	0
hand	0	0	6	2	0	0	0	0	0	67	124	187	68	0	0	0	310	222	0	0	0
lamp	2	0	1	0	0	0	0	0	1	2	1	0	1799	0	0	0	0	33	0	0	0
mobilephone	1	0	4	0	0	0	0	0	0	5	5	1	4	0	0	0	12	79	0	5	0
motorbike	0	0	0	2	2	4	0	0	0	0	3	1	1	0	0	0	2	27	0	0	0
paper	5	0	0	0	0	0	0	0	3	0	59	12	112	0	0	1	5	103	0	2	0
person	0	0	19	3	0	3	0	0	0	16	94	3	43	0	0	0	109	646	0	4	0
sign	0	0	4	6	0	0	0	0	0	3	21	9	72	0	0	0	14	265	0	11	0
train	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0
tvmonitor	1	0	1	14	1	1	5	0	0	0	8	6	51	0	0	0	4	466	1	460	0
window	0	0	0	0	0	0	0	1	4	0	4	0	9	0	0	0	1	86	0	5	0

Table 4.1: Confusion matrix that is produced by convolutional neural network training on mixed samples, at 60,000 iterations, and tested on non-lifelog data.

Table 4.1 displays the confusion matrix derived from the model at 60,000 iterations. For every number in the cells of the table, the label on the same row indicates their actual identities, and the label on the same column is the predicted result. Although the model trained from the training examples from both domains (performance showed in Table 4.1) is much better than the model using only training examples from the visual non-lifelog (performance shown in Table 3.3), it has slightly worse performance than that of the visual lifelog (performance is shown in Table 3.1). All classes in the table achieve satisfactory performance. The left text column is the ground truth label while the top text row is the prediction label. It can be concluded that when using samples from the visual non-lifelog, the learning of convolutional neural network is affected. As for the visual lifelog, the visual

non-lifelog plays a role as noise. Furthermore, the explanation from Figure 4.3 fits the table. The features that only work on the visual lifelog but the visual non-lifelog are abandoned during training, and the performance of the systems presented in Table 4.1 is slightly worse than those presented in Table 3.1.

4.4.4 Domain Adaptation by Back-propagation

Figure 4.4 displays the performance of the domain adaptation model with an adaptive learning rate that was trained on the training examples from both the visual non-lifelog and the visual lifelog on different datasets in terms of F1-score. The performance order phenomenon on the different datasets was analyzed in Figure 4.3. Although the learning curve is unstable, the domain adaptation model performance is apparently better than that of the standard convolutional neural network structure for every data set, as shown in Figure 4.3. Here the model is trained on a mixture of training examples from both non-lifelog and lifelog data. The F1-score is adopted to evaluate the performances of the model on a different data set. The order of performance on the data set is (from best to worst): training examples from both non-lifelog and lifelog, test from lifelog, validation from non-lifelog. It benefits from projecting the data from both domains to a feature space where both domains follow similar if not identical distribution.

Table 4.2 displays the confusion matrix from the domain adaptation model at 60,000 iterations. For each number in the cells of the table, the label on the same row indicates their actual identities however, the label on the same column is the predicted result. The model is trained with an adaptive learning rate.

The prediction is improved from the model trained on only training examples of the visual non-lifelog (Table 3.3), and it has significantly better performance than the prediction of the standard convolutional neural network trained from the same training set (Table 4.1). The latter observation supports the statement that the domain-adversarial training model can learn a good representation of both the visual non-lifelog and lifelog data to transfer knowledge from the non-lifelog to the lifelog.

Figure 4.5 displays the domain adaptation model performance with a fixed rate that

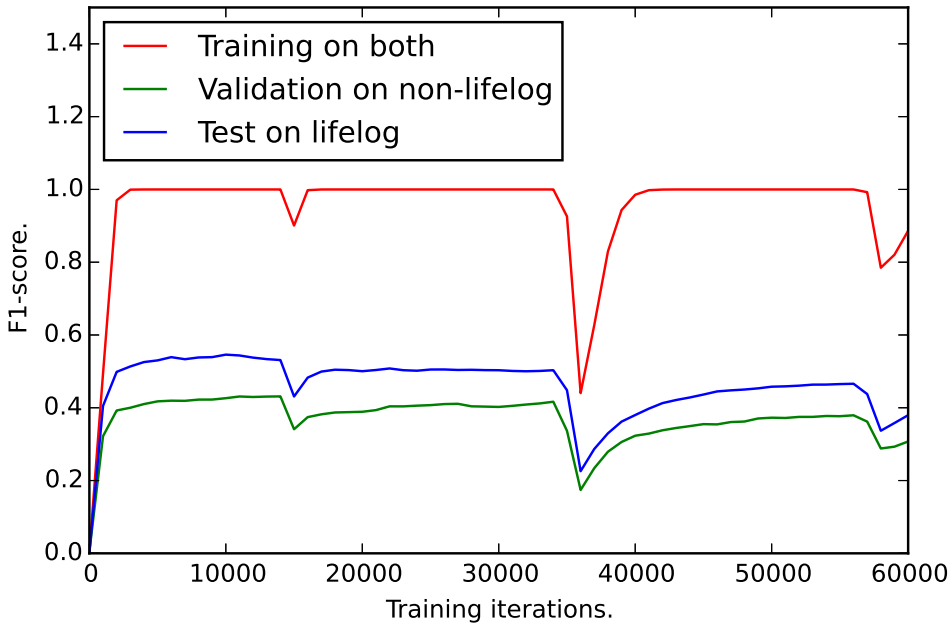


Figure 4.4: The learning curve of domain adaptation model with adaptive learning rate.

true \ predicted	air	bic	bot	bui	car	cha	cup	dis	doo	fac	gla	han	lam	mob	mot	pap	per	sig	tra	tvm	win
aircon	353	0	0	0	3	1	0	11	0	12	0	28	3	1	0	0	4	4	4	0	0
bicycle	0	1	0	0	1	1	0	0	0	1	0	1	0	1	0	0	2	0	0	0	0
bottle	0	1	27	2	28	24	2	13	2	15	46	7	9	2	0	1	23	0	3	3	0
building	5	25	8	170	15	41	7	20	10	77	12	34	29	6	4	0	77	23	22	1	0
car	1	0	0	2	116	10	2	5	1	41	8	31	2	2	0	0	27	3	1	0	0
chair	0	0	0	0	1	74	1	6	3	16	6	22	1	0	0	0	13	0	0	0	0
cupboard	0	0	0	2	0	6	200	1	4	38	4	23	11	1	1	4	10	8	0	1	0
dish	0	0	0	0	0	2	0	1	0	12	2	34	0	0	0	1	0	0	0	0	0
door	5	1	2	1	0	15	2	2	42	14	4	12	13	1	0	7	24	10	1	3	0
face	0	0	0	5	0	0	0	7	0	343	3	80	2	0	0	0	8	4	0	0	0
glass	2	8	3	1	5	18	3	30	3	34	485	34	2	1	0	3	25	7	0	1	0
hand	0	0	0	2	2	19	3	26	1	110	47	678	9	1	0	3	74	6	4	0	1
lamp	1	0	0	1	0	0	1	7	2	8	4	20	1788	0	0	2	1	3	1	0	0
mobilephone	0	0	0	0	2	9	0	1	3	5	18	25	5	18	0	5	9	12	0	4	0
motorbike	0	0	0	0	8	3	1	3	0	1	1	11	1	2	0	0	10	1	0	0	0
paper	6	4	1	1	0	7	1	4	4	38	7	65	20	1	0	123	7	11	2	0	0
person	1	1	4	3	18	45	7	39	1	151	81	138	9	9	1	0	412	11	7	1	1
sign	3	1	5	6	3	14	7	21	4	51	17	80	33	4	0	3	30	113	3	7	0
train	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
tvmonitor	23	6	1	9	25	69	8	4	52	95	24	38	73	23	1	16	34	124	45	349	0
window	4	0	1	1	3	6	4	1	6	23	11	14	6	0	0	5	7	9	1	4	4

Table 4.2: Confusion matrix produced by training on non-lifelog data, at 60,000 iterations, and tested on non-lifelog data. All classes in the table achieve satisfactory performance. The left text column is the ground truth label while the top text row is the prediction label.

trained on the training examples from both the visual non-lifelog and visual lifelog on different datasets in terms of F1-score. The model is trained on a mixture of training examples from both non-lifelog and lifelog data. The F1-score is adopted to evaluate the performances of the model on different datasets. The order of performance on datasets is (from best to worst): training examples from both non-lifelog and lifelog, test from lifelog, validation from non-lifelog. Compared with Figure 4.4, it has overall poorer performance while the performance steadily increases. The final prediction is similar to model learning with momentum, which suggests that the model structure works but that the learning strategy is not important.

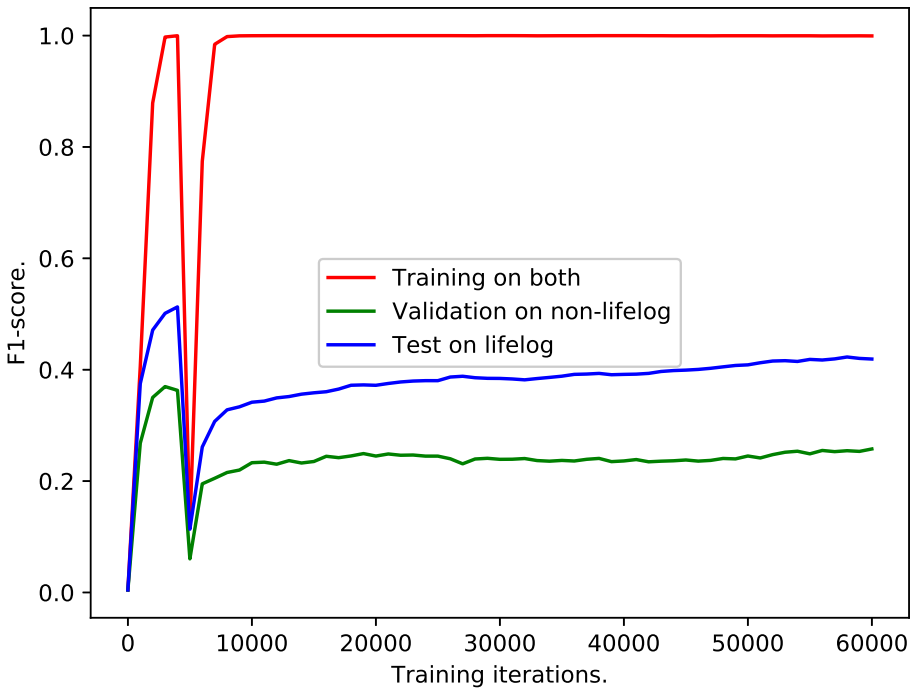


Figure 4.5: The learning curves of domain adaptation model with a fixed learning rate. The legend is at the center of the figure in case it obstructs the curves.

4.5 Conclusion and Contributions

Following the conclusion from chapter 3 that visual non-lifelog and visual lifelog are two domains, this chapter analyzed the domain adaptation problem, related machine learning fields and adopted a representation-based domain adaptation model for deep learning. The experiments successfully validate the usefulness of the domain-adversarial learning algorithm. This chapter answered the second research question in the way that domain-adversarial models are able to transfer the knowledge from the visual non-lifelogs to visual lifelogs.

Chapter 5

Object Detection in Visual Lifelog

The thesis so far has focused on the task of object recognition, including distinguishing the difference between the visual lifelogs and the visual non-lifelogs individually (in Chapter 3) and transferring knowledge from the visual non-lifelogs to the visual lifelogs (in Chapter 4). We turn now to the task of object detection on the visual lifelogs.

In the task of object recognition, any image has one label to indicate which object it contains. To make the classification easier, such images usually display a dominant foreground object (the remaining pixels are usually referred as background). In comparison, object detection deals with images with fewer constraints. An image (for object detection) may contain multiple such objects or none. Object recognition is a component of object detection algorithms adopted in this chapter. The relationships of object detection and object recognition are detailed in Section 5.3.

Lifelog images may have zero or multiple objects, which is the most common scenario for the task of object detection. This chapter proposes to detect predefined objects (with labels and training data provided) from lifelog images and also addresses the challenge that the distributions of objects in lifelog images and non-lifelog images are different.

Object detection on the visual lifelogs has many applications on other tasks of lifelogging, including content-based information retrieval, scene understanding, etc. More details will be discussed in Section 5.7.

This chapter will initially introduce the task of object detection, including the similarity

and difference between the task of object recognition and the history of the development of object detection techniques. Following that, the approach proposed in this chapter will be illustrated in two-fold, region proposal and object recognition. In the end, the applications of object detection on lifelog images will be discussed.

5.1 Object Detection

Humans can easily tell if an object is in an image or not and return the location and extent of the objects if present. Object detection is an active research area of computer vision [51, 52, 117]. Its goal is to tell if an object from a given class exists in an image. The task has many applications in many fields of computer vision, such as content-based image retrieval [79], facial expression analysis [126] and self-driving techniques [40]. Note object detection algorithms cannot detect objects which are not predefined.

In this section, we will compare the task of object recognition and the task of object detection in their definitions, corresponding machine learning problems, and data. Object recognition algorithms apply a label from a set of classes to an image while object detection algorithms return location and extent for each given objects found in an image. For object recognition, an image will be assigned one and only one label. This is a standard multi-class classification problem [14]. Because the locations and extents are continuous variables, object detection can be approximately regarded as a regression problem [14]. More methods to be discussed later use some tricks to convert object detection to object recognition. Because object recognition allows only one object in the images, objects are cropped from original images and rescaled to be the same size. The object detection task has much looser restrictions on the data: The images only have to be of equal size. The output of object recognition is a label indicating the class of the object. The output of object detection is the locations and extents of all given objects. They are usually marked with rectangles (as red rectangles in Figure 5.4).

5.2 A Simple Overview of Object Detection Approaches

Object detection research is best known for its application in face detection [117] and pedestrian detection [30]. Both of face detection and pedestrian detection deal with a predefined class, although an image usually contains multiple faces and pedestrians. Those single class detection results are intuitively easy to achieve with a high performance because useful features could be specially designed for the object and binary classification techniques (e.g., SVMs [17]) are mature enough. Nowadays, the research of object detection has turned to multiple objects detection, i.e., a universal approach that could handle arbitrary objects [51, 97, 116]. Multi-class detection is necessary for many scenarios, such as content-based image retrieval [79] and image caption generation [125], although multi-class detection is more difficult to address.

Existing object detection algorithms can be classified into two groups: algorithms based on regression and algorithms based on classification. Regression desires output consisting of one or more continuous variables. Due to the fact that location and scale (bounding box) are continuous variables, regression-based algorithms are feasible. For example, research [114] demonstrated DNN-based regression by replacing the last layer with a regression layer. Classification-based object detection algorithms are more widely used [51, 116]. As aforementioned, one of the most popular face detection algorithms [117] is based on this approach as well. Combination of both approaches can improve the final performance [51].

The second approach intuitively comprises of two components: the region proposal, proposes potential regions that may contain one and only one object; and object recognition, distinguishes what object the potential regions contain and the regions that do not contain objects of interest are categorized as background.

Many single object detection tasks employ exhaustive sliding-window approaches (shifting various scales and locations over the image) and designed efficient classifiers to discard the most of the false positives [30, 40, 50, 117]. Regarding object detection as a regression problem is a novel modern approach [114], but this does not work well in practice. Branch and bound schemes reduce the number of locations to visit, thus reducing the computa-

tion cost greatly [73]. Methods based on class independent object hypotheses for segmentation [77], i.e., generating multiple foreground/background segmentation, are helpful to detection methods to some degree [23].

5.3 Region Proposal

Region proposal approaches generate rectangular regions from an image that may contain a single object for use in object recognition. The goal of the region proposal is to generate a class-independent, data-driven, rectangular bounding boxes for each object in the image. The challenge of region proposal is that the objects can be located at arbitrary locations with various sizes.

Based on the above reason, exhaustive search approaches usually search every sliding window, i.e., at anywhere with any size, within the images. However, considering the visual search space is huge, most exhaustive search algorithms [30, 117] are equipped with predefined constraints, e.g., setting the maximum size of windows, setting the minimum size of windows, the length of sliding step. In our experiments, we set the minimum size to be 3×3. We don't set the maximum size because we think the potential object could be as big as possible.

5.3.1 Selective Search

The exhaustive search algorithms could be further improved based on an insight that images are intrinsically hierarchical. An object contains several different components; a component contains several different sections; and a section has corners and edges. In the case of the pedestrian detection, a pedestrian has heads, body, limbs; the head of the face has eyes, a mouth, and a nose. Some segmentation algorithms perform well thanks to this principle [23].

The selective search [116] has three aspects: capture all scales (objects can occur at any scale within the image), diversification (produce locations with as many image conditions as possible), and fast to compute.

The first principle is based on the same consideration as the exhaustive search. Namely, every sliding window is taken as a candidate. The selective search algorithm takes a hierarchical bottom-up grouping procedure to form the basis, and it uses region-based features whenever possible. The algorithm firstly uses a fast graph-based segmentation [44] to create initial regions. Then a greedy algorithm is used to iteratively group regions together: at the beginning, the similarities between all neighboring regions are calculated. The two most similar regions are grouped together, and new similarities are computed between the resulting region and its neighbors. The method of grouping the most similar regions is repeated until the whole image becomes a single region. The procedure is similar to hierarchical clustering [14].

The second design principle for selective search [116] is to diversify the sampling and create a set of complementary strategies whose locations are combined later. The algorithm is diversified using three facilities. Firstly, a variety of color spaces is adopted with different invariance properties accounting for different scenes and lighting conditions. Secondly, four complementary, fast-to-compute similarity measures are defined. Thirdly, our starting regions vary.

The last principle ranks the locations, that are most likely to be an object, highest. Selective search algorithm [116] order the combined object hypotheses set based on the order in which the hypotheses were created in each individual grouping strategy.

5.3.2 Visual Lifelog Region Proposal

The selective search algorithm [116] is initially used for effectively generating sliding windows on lifelog images. Recap that chapter 3 points out visual non-lifelogs and visual lifelogs belong to different domains. In other words, it can not be guaranteed an algorithm performs well on visual non-lifelogs performs equally well on visual lifelogs. So the question here becomes; *could we apply selective search algorithm to the visual lifelogs?*

This section intends to offer reasonable argument instead of rigorous mathematical proof for the sake of simplicity. The expected outputs of region proposal of the visual lifelogs are rectangular regions, the same as the output from the visual non-lifelogs. Namely,

despite the fact that lifelog images are usually distorted and different from non-lifelog images, the regions (of lifelog images) proposed for object recognition should still be rectangular. Since the region proposal cannot change the distribution of the regions from the image, the only possible interference is the size of the proposed regions.

Slightly different sizes of the regions have little impact on the performance of object recognition considering that the object recognition task for allows backgrounds in the images. If the proposed region contains too much background (even includes other objects), object recognition algorithms are designed to determine such regions as the background image. As we can see in figure 5.1, an object will usually be proposed in more than one regions. There will be one to be chosen as the best for this object. We can imagine the same result for the small size of the proposed regions.

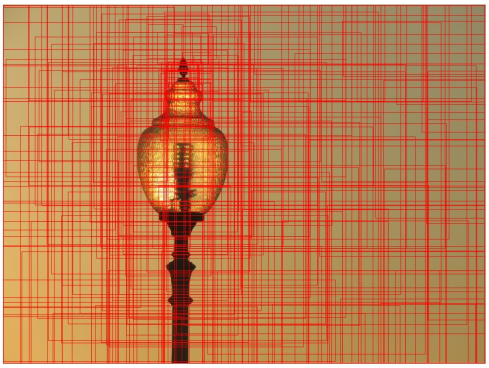
The lifelog images satisfy the first two basic principles of selective search. For the first principle, as stated above, the expected outputs of region proposal on visual lifelogs remains rectangular, so the exhaustive search algorithm applies to lifelog images as well. Lifelog images do not contradict with the second principle either because the regions on those images are hierarchical as well.

The following analysis answers the Research Question 3. The selective search could convert object detection on the lifelog images to object recognition on the lifelog images. Considering the visual lifelogs is a special domain (different from the visual non-lifelogs), the several paragraphs above analysis ensure the selective search is still useful.

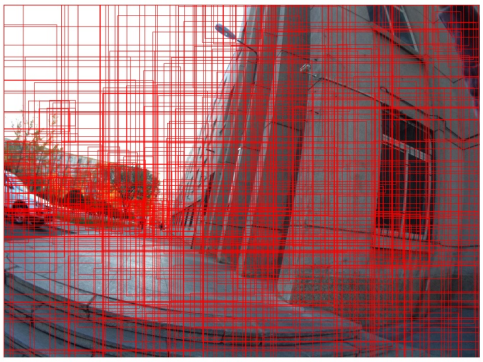
Figure 5.1 displays the proposed regions generated from the selective search on visual non-lifelog and visual lifelog data individually (non-lifelog images are one the left column while the lifelog images are on the right). From the figure, the phenomenon that the regions containing an object attract more proposed regions, can be observed on both non-lifelog images and lifelog images. It shows the selective search works equally well on the visual lifelogs.

(a) A non-lifelog example of a person.

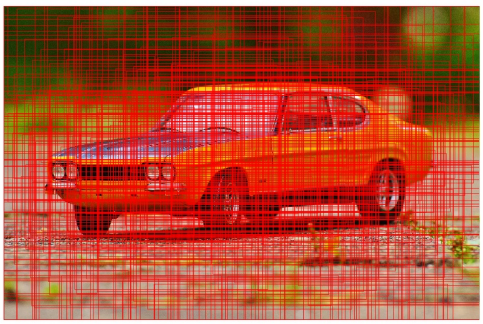
(b) A lifelog image contains person and lamp.



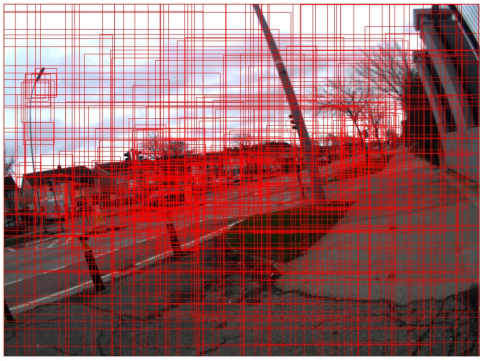
(c) A non-lifelog example of a lamp



(d) A lifelog image contains car, person and lamp.



(e) A non-lifelog example of a car



(f) A lifelog image contains car and lamp.

Figure 5.1: The regions proposal using selective search for every image corresponding to figure 3.1 on Chapter 3. For exhaustive search, we can expect to observe the proposed regions evenly distributed among the whole image. While here, obviously the regions are uneven. Selective search tends to propose regions which are more likely to contain an object.

5.4 Visual Object Recognition of Lifelog using Selective Search

The selective search algorithm proposes a number of rectangular regions of interest from lifelog images. From the Figure 5.1, we can conclude two phenomena: one is that some candidates contain no desired objects; the other is that an object attracts multiple windows of varying sizes. The efforts following selective search are removing candidates that contain only backgrounds, and selecting one of the regions from all the candidates that overlap with each other. Due to the fact that it is impossible to tell the best region of interest (it will be detailed in Section 5.5) among a group of similar region proposals, anyone that contains the object may be the one we are looking for.

In this chapter, we design two approaches for object recognition. The first classifier (recognition with pre-trained model) directly employs the classifier (object recognition system) trained in Chapter 4; the second classifier (recognition with re-trained model) builds a new classifier from the model in Chapter 4 and uses the hyper-parameters from the pre-trained model as initialization. The first classifier is more intuitive than the second one.

5.4.1 Recognition with Pre-trained Model

The domain-adversarial convolutional neural network from Chapter 4 (depicted in figure 4.2) has two output ends: one predicts the label of objects and the other indicates whether the input image belongs to the visual lifelog or the visual non-lifelog. The end that indicates the domain is not the focus of this chapter because we don't care if the input image comes from the lifelog or not. The end that predicts the label of objects has the same number of output neurons as the number of objects (the number is 21 in our case). The output values of neurons are regarded as prediction value (or confidence) of objects and the object with the highest prediction value is taken as the label for the input image.

Note that the input of the model in Chapter 4 does not contain backgrounds, but the input regions contain backgrounds in this chapter. Namely, compared with Chapter 4, the classifier has an extra challenge: it is possible that the input region does not belong to any predefined objects. Considering the output of the label recognition end could be inter-

preted as a likelihood, we could assume that the background will have a low likelihood of all classes. If we are right, the pre-trained model from chapter 4 still could apply to the recognition requirement here with an intuitive modification. If the highest prediction value is below a pre-set threshold, the input region is considered as background.

The threshold is the key for the pre-trained model. A large value for the threshold increases the rate of false negatives, because a region of a real object is rejected by a large threshold and tagged as background. On the contrary, a small value for the threshold increases the rate of false positives, in the way that a region of background is counted by a small threshold and acknowledged as an object. We regard the threshold as a hyper-parameter to tune instead of a parameter to learn, because the threshold is independent of the recognition model.

The recognition with a pre-trained model does not learn any new knowledge at all, i.e., it is not going to change its attached parameters. Except the threshold mentioned just above, another consequence is that the model is not allowed to take training examples. Namely, the selection of value for threshold is independent of the data. A simple solution is to utilize the training examples from chapter 4: feed the domain-adversarial convolutional neural network with those training examples and for each image, the highest prediction value is recorded and among them, while the threshold is set to the lowest one.

For an arbitrary input image X_i, we assume the output is $y_i = f(X_i)$, where $f(\cdot)$ is the end predicts label of objects and y_i is a vector here. The label is predicted as:

$$label = argmax(y_i) \tag{5.1}$$

while the threshold is selected as:

$$threshold = min([max(y_1), \cdots, max(y_i), max(y_n)]) \tag{5.2}$$

5.4.2 Recognition with Re-trained Model

Inspired by [51], the pre-trained domain-adversarial convolutional neural network is employed for object recognition on proposed regions with fine-tuning [130] and the parameters of the pre-trained model are partly used as initialization for the re-training.

Figure 5.2 depicts the procedure for re-training. Chapter 4 trains the classifier using regions from the ground truth only (including visual non-lifelog and visual lifelog). In this strategy, the prediction is only able to produce one and only one object category. Although the strategy perfectly satisfies the requirement of object recognition, it fails on object detection considering regions proposed for object detection could be background. Enlightened by this reason, training examples are supplemented with negative training examples (regions of background) from proposed regions. Intuitively, the negative examples with more overlapping with positive examples are more challenging for the classifiers [14, 17]; therefore, the negative examples are produced from regions that have overlap (in terms of IOU) 0.2-0.4 with the positive examples. Algorithm 1 displays how to calculate IOU. The enlarged training examples are used to re-train the domain-adversarial convolutional neural network.

As we discussed above, an object can relate to several regions in the region proposal and these candidate regions can be of different sizes. Therefore, a necessary step is to change these regions to a fixed size. Another challenge is we hope to output only one region for one object as the result of object detection. The solution is to pick a region that has the highest confidence and then remove those regions that have more than 0.5 overlaps (IOU) with it. For remaining regions, repeat the action until the end.

The classifier, domain-adversarial convolutional neural network, needs to change as well. The label predictor of the domain-adversarial convolutional neural network for visual object recognition of lifelog has 21 output neurons, each of which points to an object category. However, in the case of visual object detection of an object, an extra one is necessary to represent the background. Figure 5.3 details the structure employed for re-training in this chapter.

The only difference between Figure 4.2 and Figure 5.3 is that the label predictor of the

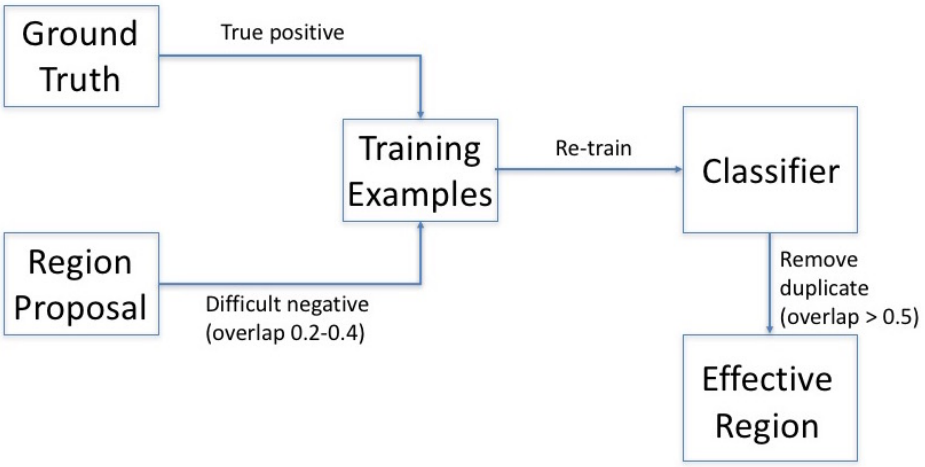


Figure 5.2: The procedure of re-training the domain-adversarial convolutional neural network. The ground truth regions supply the true positive training examples. The region proposal provides the difficult negative which has 0.2-0.4 overlap with the ground truth positives. Those training examples are combined to re-train the classifier. In the end, remove the duplicate regions of interests by getting rid of those have overlap above 0.5 with the one with the highest confidence.

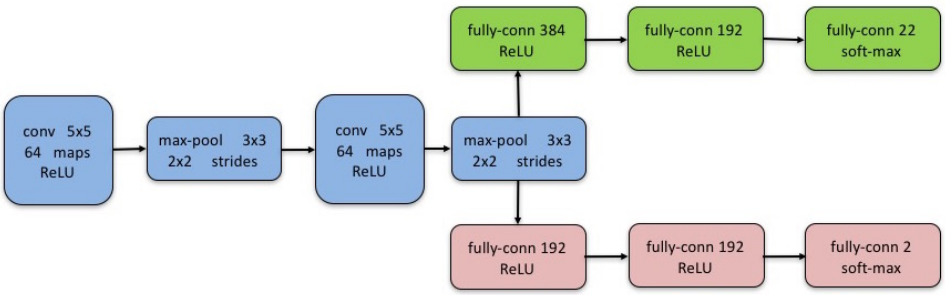


Figure 5.3: The updated domain-adversarial model. The blue components act as non-linear representation learning; and the green ones perform label classifier while the red ones perform domain predictor. Representation learning comprises convolutional and max-pooling layers. Both label classifier and domain predictor consist of perceptions. Notice the output of label predict is 22 now.

latter one has 22 output neurons instead of 21. In this sense, all parameters of the model that obtained from Chapter 4 could be used as initialization for re-training except the output layer of label predictor. Those parameters are highly likely to change during re-training but the initialization will speed the re-training by providing a good initialization.

It is clear that the re-training approach is much more complicated than pre-training approach. The pre-training approach wants the calculation of threshold only while the re-training not only expands the training examples but modifies the classifier and performs training as well.

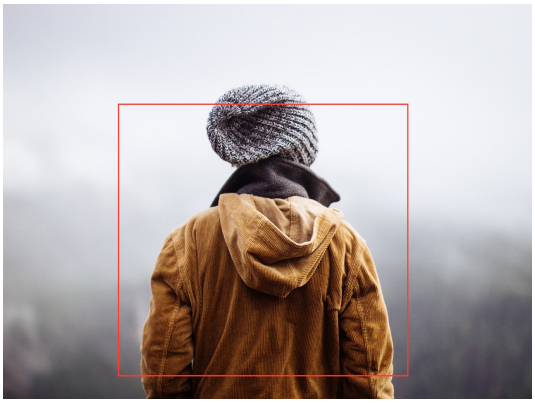
Figure 5.4 displays the output of object detection. A potential benefit of the approach is it not only can handle visual lifelog but visual non-lifelog as well. Note in Figure 5.4d, only the left pedestrian is detected because the right one is too small.

5.5 Evaluation

Objects present in an image may vary in location, size, and aspect ratio. Evaluation for object detection is therefore difficult [25, 60]. Firstly, different from object recognition which is a classification problem, the evaluation value for object detection is continuous. Secondly, it is hard to tell a better candidate window from a group of similar ones. Thirdly, as the regions of interest are rectangular, it is natural that parts of objects are out of the region and some area of the region contains the background. However, it is rather hard to say to what extent the region should not be considered as the region of interest for that object.

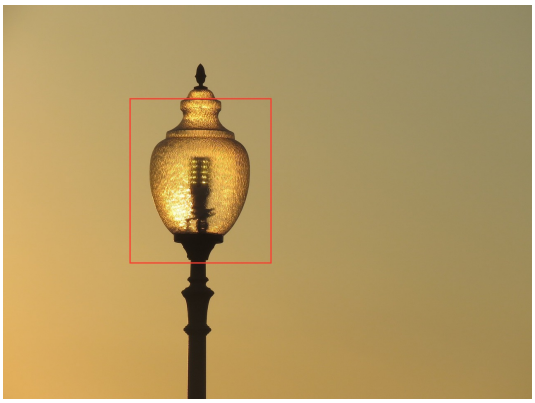
As to the question of the purpose of object detection algorithms, early work [39] emphasizes the category independence, while more recent work [116] focuses on a chosen set of object classes. It is pointed out that the current evaluation rule for object proposal methods is suitable for object detection but is a gameable and misleading protocol for category independent tasks [25]. Though inspiring, the work [25] is still under development.

The Intersection over Union (IOU) algorithm is the consequence of the instinctive idea mentioned above: the larger the area of overlap and the less area not overlapped between

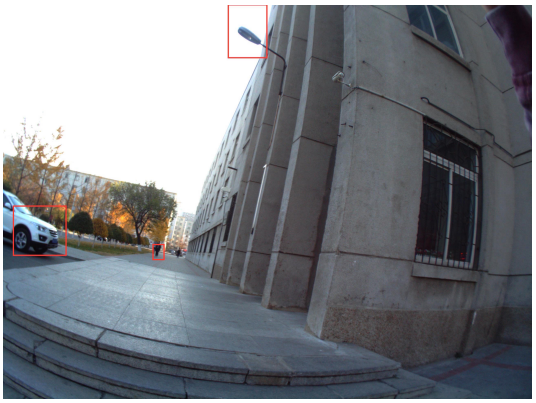


(a) A non-lifelog example of a person.

(b) A lifelog image contains person and lamp.



(c) A non-lifelog example of a lamp



(d) A lifelog image contains car, person and lamp.

(e) A non-lifelog example of a car

(f) A lifelog image contains car and lamp.

Figure 5.4: The red rectangles are results of object detections. For each image, every rectangle comes from one of regions proposed in figure 5.1.

two regions, the more likely they are regions of the same object. For two boxes/regions b_i

and b_j , IOU is defined as:

$$IOU(b_i, b_j) = \frac{area(b_i \cap b_j)}{area(b_i \cup b_j)} \tag{5.3}$$

Because b_i and b_j are rectangular boxes, the calculation of intersection over union is surprisingly easy to implement (depicted in Algorithm 1). The input of Algorithm 1 are two rectangles to be calculated. Each rectangle contains four elements in the following order: the biggest x-coordinate value (x-coordinate of the bottom right corner), the smallest x-coordinate value (x-coordinate of the top left corner), the biggest y-coordinate value (y-coordinate of the bottom right corner), and the smallest y-coordinate value (y-coordinate of top left).

Algorithm 1: Calculation of intersection over union (Python)

```
def IOU(Rct1, Rct2):
    """ Calculate the Intersection of Union.

    Args:
      Rct1: a tuple containing four elements (max_of_x, min_of_x,
          max_of_y, min_of_y).
      Rct2: a tuple containing four elements (max_of_x, min_of_x,
          max_of_y, min_of_y).

    """
    x_overlap = max(0, min(Rct1[0], Rct2[0]) - max(Rct1[1],
        Rct2[1]))
    y_overlap = max(0, min(Rct1[2], Rct2[2]) - max(Rct1[3],
        Rct2[3]))

    # overlap area
    area_overlap = x_overlap * y_overlap

    # total area
```

```
    area_union = (Rct1[0]-Rct1[1])*(Rct1[2]-Rct1[3]) +
        (Rct2[0]-Rct2[1])*(Rct2[2]-Rct2[3]) - area_overlap

    return area_overlap/area_union
```

Intersection over union is suitable for any two regions and it was initially used in set theory. Based on a necessary assumption that the ground truth annotations are perfect, this chapter uses IOU to evaluate the quality of effective regions.

Notwithstanding that the higher IOU value does not necessarily ensure the better region of interest [25], the IOU is adopted for its simplicity and straightforwardness compared to others (e.g., area under the recall curve, volume under the surface, average best overlap).

5.6 Experiments

The experiments have two components: region proposal and object recognition. Two approaches to object recognition of the lifelog images are designed: pre-trained model and re-trained model.

As discussed in Section 5.4.1, the values of the output of the neural network adopted could be interpreted as the likelihood of the label. It inspired a motivation that the neural network may output very low probability on all outputs if the input is nothing related to the given (21) objects. The re-retained model approach regards anything not included in the 21 objects as the 22nd one.

The recognition with pre-trained model does not need re-training, and it does not require extra training data. The only extra effort is predicting regions proposed from selective search based on the model detailed in Figure 4.2. Comparatively, the recognition with re-trained model requires extra training data (the whole procedure is depicted in Figure 5.2), which is based on the model detailed in Figure 5.3.

The data used in the experiments is illustrated in Appendix A.

5.6.1 Settings

The experiments conducted two object recognition approaches to lifelog images. In Chapter 4, we have tried to recognize 21 objects from lifelog images. Since each of those images has one and only one object, we merely need 21 labels. However, for the images obtained from region proposal, it also may have nothing (only background), object not included in the 21 classes, or multiple objects. We don't care that exactly which one happens, so they are assigned the same label ("No object", as the 22nd) in this chapter.

The experiments of object detection on the lifelog images consist of region proposal and object recognition. The performance of region proposal can not be evaluated independently because the algorithm of the region proposal is unsupervised, so we will just simply mention the details of the region proposal. The performance of object recognition can be displayed via learning curve of the training phrase and confusion matrix (as used in previous chapters). The performance of object detection on the lifelog images will be evaluated using IOU (which can be calculated using Algorithm 1).

5.6.2 Recognition with Pre-trained Model

Table 5.1 depicts the performance of recognition with a pre-trained domain-adversarial convolutional neural network for every object category. The first row of the table lists the total proposed region (TPR) using selective search. Rest rows appear in pairs, describing the number of effective regions (ER(x)) and corresponding percentage out of the proposed region (ERP(x)=ER(x)/TPR) for every object category. Namely, a proposed region is effective if it has an IOU value more than 0.5 with a given ground truth region. The threshold of a pair of ER(x) and ERP(x) is calculated as $56.71/x$, where x is selected from $\{1, 2, 4, 8, 16, 32, 64, 128\}$, so the thresholds correspondingly are $\{56.71, 28.36, 14.18, 7.09, 3.54, 1.77, 0.89, 0.44\}$.

The table supports the statement from Section 5.3.2 that the number of effective regions increases as the threshold decreases. Apart from that, several interesting phenomena could also be observed. The direct recognition approach performs poorer in some categories (e.g., train, bicycle, dish) than others (e.g., lamp, tv monitor), because there are relatively

fewer training examples provided by the former categories. The number of effective regions between values of 0.89 and 0.44 (ERP(64) and ERP(128)) is similar, which indicates the values of most confidences are between those intervals. At the same time, the number of effective regions where the threshold above or equal to 7.09 is zero.

It can be concluded that the values of most confidences are much lower than expected (56.71). Apparently, recognition with pre-trained domain-adversarial convolutional neural network could not accomplish the recognition part for visual object detection of lifelog images.

5.6.3 Recognition with Re-trained Model

The procedure has been discussed in Section 5.4.2. The addition of difficult negative training examples will also be reshaped to fit the size of input of model depicted in Figure 5.3. We don't care the outputs of domain predictor, so they are ignored from the results. The training phrase adopts same tuning tricks from Chapter 3 and Chapter 4. The initial values for each layer of the model depicted in Figure 5.3 is loaded from pre-trained model inherited from Chapter 4 except the output layer on the label predictor because the number of neurons has changed from 21 to 22.

The learning curve of the re-training model is plotted in Figure 5.5. From the figure, we could see that the after F1-score reaches very high value fast, it is much more robust compared to learning curves in Chapter 3 and Chapter 4.

Table 5.2 depicts the confusion matrix (the images does not contain any from the 21 objects are labelled "No object") at 60,000 iterations. It shows similar performance on each label and displays average good performance.

3136 regions can not be resized because of bugs from the package employed for this research, so 3219 images from test set remained. Table 5.3 displays the result of recognition with the re-trained model which has much better improvement than that from Table 5.1. The object categories that do not perform well (e.g., train, bicycle, dish) still repeat from the re-trained model.

	aircon	bicycle	bottle	building	car	chair	cupboard	dish	door	face	glass
TPR	19160	2957	6184	46858	10534	109571	14366	4291	7357	228746	124236
ER(1)	0	0	0	0	0	0	0	0	0	0	0
ERP(1)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ER(2)	0	0	0	0	0	0	0	0	0	0	0
ERP(2)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ER(4)	0	0	0	0	0	0	0	0	0	0	0
ERP(4)	0	0	0	0	0	0	0	0	0	0	0
ER(8)	0	0	0	14	0	0	0	0	0	0	0
ERP(8)	0.0	0.0	0.0	0.0004	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ER(16)	5	0	14	4174	5	120	64	0	0	3466	545
ERP(16)	0.0003	0.0	0.0023	0.0891	0.0005	0.0011	0.0045	0.0	0.0	0.0152	0.0042
ER(32)	217	1	317	12254	270	1627	378	8	47	14271	12445
ERP(32)	0.0113	0.0003	0.0513	0.2615	0.0256	0.0148	0.0263	0.0019	0.0064	0.0624	0.1002
ER(64)	443	2	481	13756	941	2175	481	28	104	16985	23195
ERP(64)	0.0231	0.0007	0.0778	0.2936	0.0893	0.0199	0.0335	0.0065	0.0141	0.0743	0.1867
ER(128)	456	2	496	13868	1030	2210	484	30	110	17119	23948
ERP(128)	0.0238	0.0007	0.0802	0.2960	0.0978	0.0202	0.0337	0.0070	0.0150	0.0748	0.1928

	hand	lamp	mobilephone	motorbike	paper	person	sign	train	tvmonitor	window
TPR	527042	1267507	71303	600	208	53778	262990	8523	27603	592
ER(1)	0	0	0	0	0	0	0	0	0	0
ERP(1)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ER(2)	0	0	0	0	0	0	0	0	0	0
ERP(2)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ER(4)	0	0	0	0	0	0	0	0	0	0
ERP(4)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ER(8)	53	16997	0	0	0	0	2	0	15	0
ERP(8)	0.0001	0.0134	0.0	0.0	0.0	0.0	0.0	0.0	0.0005	0.0
ER(16)	20000	197911	35	0	0	61	3946	0	3083	0
ERP(16)	0.0379	0.1561	0.0005	0.0	0.0	0.0011	0.0150	0.0	0.1117	0.0
ER(32)	69726	341877	850	1	3	4596	13049	0	6838	8
ERP(32)	0.1323	0.2697	0.0119	0.0017	0.0144	0.0855	0.0496	0.0	0.2477	0.0135
ER(64)	79528	351055	1234	1	3	8176	17328	0	7851	12
ERP(64)	0.1509	0.2770	0.0173	0.0017	0.0144	0.1520	0.0659	0.0	0.2844	0.0203
ER(128)	79821	351528	1265	2	5	8482	18220	0	7974	12
ERP(128)	0.1515	0.2773	0.0177	0.0033	0.0240	0.1577	0.0693	0.0	0.2889	0.0203

Table 5.1: The first row displays the number of total proposed regions (TPR) for each category. From second row, the row ER(x) displays the number of effective regions (ER) with confidence above the threshold $56.71/x$ while the row ERP(x) displays the percentage (ERP) of effective regions from total proposed regions with confidence above the threshold $56.71/x$. The parameter x is selected from $[1, 2, 4, 8, 16, 32, 64, 128]$. The threshold, calculated from equation 5.2, is 56.71.

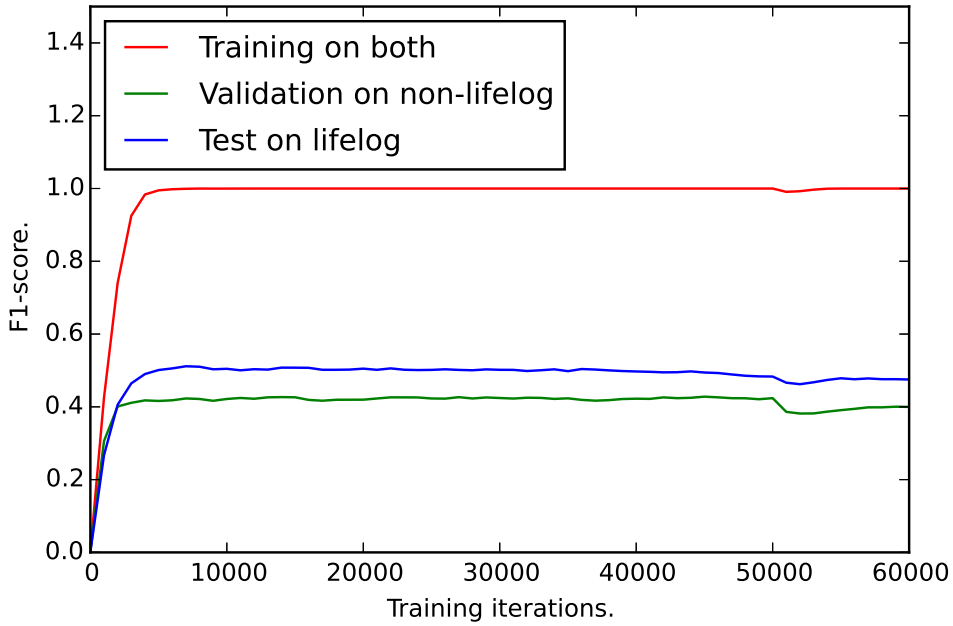


Figure 5.5: The learning curve of re-trained model.

true \ predicted	air	bic	bot	bui	car	cha	cup	dis	doo	fac	gla	han	lam	mob	mot	pap	per	sig	tra	tvm	win	non
aircon	372	0	0	0	1	0	0	2	0	0	0	0	3	0	1	2	0	7	0	3	0	33
bicycle	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	6
bottle	2	0	73	0	0	3	0	2	0	0	3	0	3	0	0	0	4	1	0	1	0	116
building	2	4	1	134	1	0	6	0	3	1	1	2	2	1	3	2	4	6	5	1	1	406
car	0	0	0	0	104	0	0	0	0	3	1	5	0	1	1	0	1	0	1	2	0	133
chair	0	0	1	0	0	61	1	1	6	0	0	7	0	0	0	0	2	1	0	2	0	61
cupboard	0	0	0	0	0	2	237	2	5	3	1	1	1	1	0	5	3	0	0	7	1	45
dish	0	0	0	0	0	0	0	9	0	0	2	8	1	0	0	1	0	2	0	0	0	29
door	1	0	0	0	0	0	8	1	61	0	0	0	3	0	0	3	2	2	0	5	1	72
face	0	0	0	0	1	1	0	1	1	323	0	14	0	0	0	0	0	1	0	0	0	110
glass	1	2	8	0	2	1	3	10	2	1	420	6	7	1	0	5	2	3	0	6	0	185
hand	0	0	2	0	0	0	2	7	1	16	2	424	1	0	0	7	10	1	0	0	0	513
lamp	1	0	0	0	0	0	1	0	0	0	0	4	1767	0	0	5	0	1	0	0	0	60
mobilephone	0	0	0	0	2	0	0	0	0	1	0	3	0	26	0	1	1	2	0	12	0	68
motorbike	0	0	1	0	3	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	34
paper	1	0	0	0	0	0	1	1	0	1	1	5	4	0	0	91	0	0	0	0	1	196
person	0	0	2	2	1	2	3	6	3	14	0	11	3	2	1	0	212	1	1	1	0	675
sign	3	2	7	3	1	3	6	9	1	7	3	11	7	5	0	9	7	114	0	18	1	188
train	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2
tvmonitor	2	2	2	1	4	2	2	1	2	1	7	6	5	3	1	10	0	2	3	676	0	287
window	0	0	0	0	0	1	1	1	4	0	1	3	3	0	0	5	1	1	0	13	20	56
No object	26	27	19	48	18	16	30	22	35	35	45	74	54	43	26	54	56	21	34	76	9	18007

Table 5.2: Confusion matrix of the recognition performance is produced by the re-trained model, at 60000 iterations, and test on lifelog data.

	aircon	bicycle	bottle	building	car	chair	cupboard	dish	door	face	glass
GT	245	5	91	413	111	68	125	30	79	196	400
ER	0	0	0	377	0	3	0	15	12	1	0

	hand	lamp	mobilephone	motorbike	paper	person	sign	train	tvmonitor	window
GT	575	959	53	20	222	416	212	1	715	60
ER	0	2	0	0	17	0	15	1	6	28

Table 5.3: The prediction performance of recognition with re-trained model

5.7 The Relation to Other Tasks in Visual Lifelog

Object detection can be extended to many research tasks in the computer vision. For example, the face detection enables future research efforts in face tracking, pose estimation, and expression recognition [126]. Similarly, object detection of the lifelog images could also be extended into many research tasks of lifelog images as well.

The content-based image retrieval [79] of the lifelog images is an important use case. One of the motivations of the lifelogging is to enable lifelogger easily access and recall their information, including images and documents. The content-based image retrieval is expected to return the images which contain the desired objects. It could be easily implemented using object detection. Object detection algorithm marks the objects in each image, which are used as vocabulary. The rest is then similar to the common document retrieval [83]. In this case, the content-based image retrieval does not need the information of the location of each object.

Scene understanding requires a semantic understanding of the elements that surround the lifelogger, such as objects, people and environments, which are an important factor to decide the scene [15]. The objects drawn from the lifelog images can form a vocabulary of concepts which characterizes the surrounding scene. The first work [20] on object recognition in the domain of lifelogging was able to successfully analyze the temporal consistency, co-occurrence and relationships within the detected objects. The work also concluded that the relationships could help the scene or concepts understanding.

The scene understanding could also be extended to aid the memory of the aging population [22]. For example, detection and recognizing a face from a lifelog album tells its owner when and where the face was met.

5.8 Conclusion and Contributions

In this chapter, we turn the research focus to object detection on lifelog images from object recognition (of Chapter 3 and Chapter 4). We compared object recognition and object detection. This chapter proved the selective search algorithm also applies to the lifelog

images. Moreover, two experiments were conducted to find the best way to perform object recognition within the detection framework. In summary, this chapter decomposed the object detection into region proposals and the object recognition. That is the answer to the last research question.

This chapter has two contributions. On the one hand, the thesis is not a complete work without this chapter because lifelog images are intrinsically images in the wild [28], i.e., a lifelog image naturally contains arbitrary objects. On the other hand, visual object detection of lifelog could be easily extended to other tasks of visual lifelogs.

Chapter 6

Conclusions and Future Directions

Facing the challenge that visual lifelog is suffering from insufficient training data, the thesis explores object detection in lifelog images, introducing information from the abundant source of visual non-lifelog data. At the start of this work (in Chapter 1), we pointed out the limitations of this thesis: the thesis neither addresses all the tasks for visual lifelogs, nor guarantees the approach is useful in other non-visual fields.

From the analysis of Chapter 1, we could see that lifelog images have larger feature space with many different frequencies of the objects contained therein. Chapter 2 points out that because of the privacy issues and the difficulties of the collection, the number of the available lifelog images for training is rather limited. Chapter 1 proposes to use the non-lifelog images to help the object detection task for the lifelog images.

By answering the three related research questions that were presented in Chapter 1, the thesis proves the above approach works. The answers to the three research questions complete a unified framework that could tell which pre-defined objects exist in the lifelog images. The second research question is raised from the answer to the first research question. The third one is based on the second one.

6.1 Answers to Research Questions

Research Question 1

Research Question 1 asked: What is the relation between visual non-lifelog and visual lifelog in terms of similarity?

Chapter 1 raises three answers to the question: the same, similar or absolutely different. Chapter 3 points out the objects in lifelog images have a different distribution in representation than in non-lifelog images, even if the lifelog images and non-lifelog images have the same size. In other words, lifelog images are similar to non-lifelog images. This conclusion hints that it is possible to transfer knowledge from the visual non-lifelog to visual lifelog, which leads to following two research questions.

The analysis and conclusion above are also proven by experiments. If the visual lifelog and the visual non-lifelog were the same, then model trained on non-lifelog images is expected to have similar performance on the test sets between the visual lifelog and the visual non-lifelog. But the results are distinctly different.

Moreover, an experiment that trains the model on the lifelog images and tests on the lifelog images indicates that some classes of objects are not well recognized because the number of training examples is insufficient. As a comparison, another experiment trains a model on the visual non-lifelog and predicts on the visual non-lifelog. These two experiments support the conjecture that object recognition needs more training data in the visual lifelogs than that of the visual non-lifelogs because the objects have larger feature space in the visual lifelogs.

Research Question 2

Research Question 2 asked: How could we transfer useful knowledge from visual non-lifelogs to help object recognition on visual lifelogs?

The answer to Research Question 1 reveals that it is possible to exploit the knowledge from the visual non-lifelogs and apply it to the visual lifelogs since they are similar. But it is still unknown how to transfer.

Chapter 4 starts by introducing transfer learning and domain adaptation. It then analyzes the visual lifelogs and visual non-lifelogs from the perspective of domain adaptation. Following Chapter 3, we have the conclusion that visual lifelogs and visual non-lifelogs are two different domains. Since the visual non-lifelogs have many more training examples than visual lifelogs, in our experimental work, the non-lifelog images are taken as the source domain and the lifelog images is taken as target domain.

Chapter 4 designs an experiment which trains a model from the combination of training examples from both the visual lifelogs and the visual non-lifelogs. The experiment displays the performance of the model evaluated on the visual lifelogs and the visual non-lifelogs individually. Domain adversarial convolutional neural networks are then used to transfer the knowledge from the visual non-lifelogs to the visual lifelogs.

In theory, a method that works well for the domain adaptation may address our problem. The domain adversarial convolutional neural networks were state-of-the-art.

Research Question 3

Research Question 3 asked: How could we transfer the object recognition task to the object detection for the lifelog images?

The goal of the thesis is to handle the object detection task for the lifelog images. However, the above research questions are discussed within the task of object recognition, because we can't perform object detection on the non-lifelog images. Therefore, we know that we can only transfer the knowledge from objects of the visual non-lifelog to objects of the visual lifelog.

The last step is how to transform the object recognition task to the object detection task. Chapter 5 uses selective search which finds potential regions (containing objects). It also points out we don't need to worry about the "correct regions" during the selective search since the transformation remains in the source domain.

To explore which will be best to reduce object detection to object recognition for lifelog images, Chapter 5 conducts two approaches: pre-trained convolutional neural networks (interprets the probability produced by the model as distance to a class) and re-trained

convolutional neural networks (regards the background as another class). The experiments show that the latter one is much better than the former one.

6.2 Contributions

We identify and summarize what we consider to be four primary contributions from this research, namely:

1. In Chapter 3, we find out that the non-lifelog images and lifelog images are different in terms of visual content and content framing, but similar in distributions of pixels. We propose that visual lifelogs and visual non-lifelogs belong to two separate domains. The relationship of lifelogs and non-lifelogs have never been studied before, as to the best knowledge of the author, the dissertation presents the first study to explore and discuss the connection and relation between visual lifelogs and visual non-lifelogs. It lays the foundation not only for the rest of the thesis but new perspectives and potential solutions to the challenge of retrieval from lifelog archives.

2. Chapter 4 focuses on the object recognition task for visual lifelogs. The chapter successfully transfers the knowledge from non-lifelog images to lifelog images. The chapter is the first work to discuss the visual lifelogs in the perspective of transfer learning. The thesis addresses the object recognition problem on visual lifelog images using domain-adversarial convolutional neural networks, and alternative models can be developed following this idea.

3. The biggest contribution of chapter 5 is that it converts object recognition to object detection on the lifelog images. Both object recognition and object detection are common tasks in computer vision, but object detection has more direct applications, such as multimedia retrieval. It points out the selective search does not affect the transfer procedure. The chapter also discusses the application of visual object detection of lifelog for other tasks of lifelogging.

4. The last contribution from this work is a dataset of non-lifelog images which are

necessary to lifelog images adopted in the thesis. Most data of the dataset comes from the ImageNet. The users are able to generate this dataset following the steps in Appendix A.

6.3 Future Research Directions

Since this is very much exploratory work in the area of lifelogging, there are a number of future research directions that we can identify from this work. We now present the most interesting of these challenges.

Label Set Expansion

The direct application of the work is multimedia retrieval. Lifeloggers could utilize enhanced search functionality for their visual archives that contain specific objects. Moreover, the work could be used to indicate which objects may be contained in an image. However, this work only focuses on 21 objects. As for the common objects encountered in everyday life, 21 is far from enough. The number 21 is picked because of the limits of the datasets that we employed in this work. In other words, we have only 21 labels to train with. It is straightforward to expand the number of objects we can handle by expanding the annotated objects of the dataset. It is necessary for the desired objects have to be annotated in the lifelog dataset and corresponding non-lifelog training examples have to be collected. After that, methods and approaches similar to those presented in this dissertation can be applied.

Model Exploration

The thesis focuses on object detection for lifelog images. However, the methods and algorithms adopted in this thesis are preliminary. Therefore, more cutting edges algorithms or ad-hoc algorithms could be developed and tested. In other words, with better models, we may be less likely to miss objects or propose wrong objects.

For example, we used a CNN which is similar to AlexNet [71] for the experiments of object recognition. But there are more advanced CNNs could take its place, such as

GoogleNet [113] and ResNet [58]. For the object detection, candidates could be Faster RCNN [97] and Mask RCNN [57].

Lifelog Search Engine

In the thesis, our goal is detecting objects on lifelog images which is essentially indexing the lifelog images. Although as mentioned in Chapter 3, indexing is the most important and most difficult to build a multimedia search engine, more efforts are required to build a lifelog search engine.

Lifelog search engine is useful in a lot of scenarios in the real world, such as lifelog digital diary [21], concept detection [118], and digital memory maintenance as an assistive technology.

Additional Tasks

Although the thesis works on two tasks, object recognition and object detection, the idea that enhancing lifelogs using non-lifelogs, could be easily employed on any other task as well. The domain adversarial convolutional neural networks are task-dependent, but there are different transfer learning algorithms could handle different tasks. Moreover, the transfer representation learning provides a general adapter for further specific algorithms.

Modalities Exploration

This thesis focuses on images only as the source of lifelog data. However, lifelogs are typically multimodal and as such could also include other sensors' data, such as GPS, acceleration, and even sound. Those modalities can provide additional information which images can't. The idea and methodology proposed in this thesis could be easily adopted in other modalities.

This work is similar to the above one because when the data changes, the task naturally must change also. For example, we can perform object detection on the images, but could not do the same on the GPS data.

Appendices

Appendix A

Data Explanation

In this appendix, a brief introduction is given to the data sets used in this thesis. This includes non-lifelog data used for recognition, lifelog data for recognition, a mixture of non-lifelog and lifelog data used for recognition (mentioned earlier as a mixture of non-lifelog and lifelog data) and lifelog data used for detection. The approaches we take to processing the data and the data sets are independent: the discoveries and models in the thesis can work on other data sets and the data sets illustrated here could also be used for other approaches.

Chapter 3 uses both non-lifelog data for recognition and lifelog data for recognition. Chapter 4 uses lifelog data for recognition, non-lifelog data for recognition and a mixture of the two. Chapter 5 uses original lifelog images and a mixture of lifelog and non-lifelog data for recognition.

A.1 Non-lifelog Data for Recognition

The source of non-lifelog data for recognition tasks in this thesis comes from the large-scale image data sets which contain images that each has one and only one object. Reliable as they are, great additional efforts were taken due to their limited categories of objects, copyright issues, inconsistent format, etc.

ImageNet [35], a largescale ontology of images built upon the backbone of the Word-

Net[1] structure, is used as main source of non-lifelog data. Up to this point, it has 14,197,122 images and 21,841 synsets[2]. ImageNet has benefited from adopting the hierarchical structure of WordNet. ImageNet considers the rare cases in a class (called "synonym set" in ImageNet), e.g., "building" synonym set contains the building of the house, office, hall, and restaurant. Despite the fact that ImageNet covers thousands of classes, the object "hand" or similar concepts are still missing. Thus in our work images of the object "hand" are supplemented with images from Flickr[3].

ImageNet uses the hierarchical structure of WordNet [86], a lexical database of English. Each meaningful concept in WordNet, possibly described by multiple words or word phrases, is called a synonym set or synset. There are approximately 80,000 noun synsets in WordNet. Figure A.1 provides two rows of image examples and their corresponding synsets from ImageNet. For each synset, 6 randomly sampled images are presented. In each row, the direction from left to right follows concepts from root to leaf, i.e., the concepts of the right synset is a subset of the left synset.

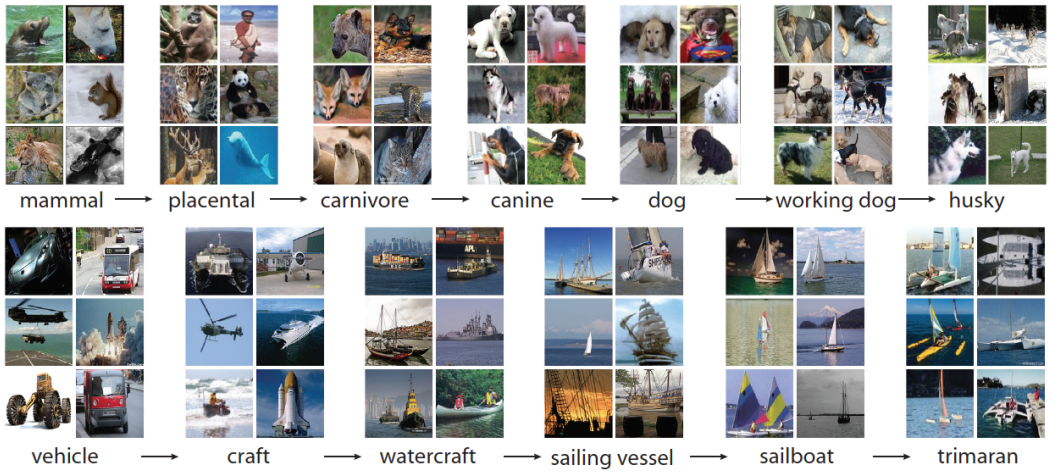


Figure A.1: A snapshot of two root-to-leaf branches of ImageNet: the top row is from the mammal sub-tree; the bottom row is from the vehicle sub-tree. For each synset, 6 randomly sampled images are presented in the Figure. The source is from [34].

Figure A.2 displays the statistical information associated with common sub-trees in the Fall 2011 release of ImageNet [34]. It tells that every synset has different number of images.

[1] https://wordnet.princeton.edu/
[2] http://www.image-net.org/
[3] https://www.flickr.com/

subtree	# synsets	avg # of images per synset	total # of images
amphibian	94	590	55,510
animal	3,822	732	2,798,930
appliance	51	1,163	59,343
artifact	7,450	749	5,582,339
bird	856	948	812,069
covering	946	818	774,362
device	2,385	674	1,609,552
fabric	262	689	180,777
fish	566	494	279,775
flower	462	734	339,383
food	1,495	669	1,001,193
fruit	309	607	187,583
fungus	303	452	137,187
furniture	187	1,042	194,948
geological formation	151	838	126,567
invertebrate	728	572	416,832
mammal	1,138	821	934,450
musical instrument	157	891	139,899
person	21	1,152	24,208
plant	1,666	599	999,163
reptile	268	707	189,521
sport	166	1,207	200,402
structure	1,239	763	945,590
tool	316	551	174,271
tree	993	568	564,040
utensil	86	912	78,442
vegetable	176	764	134,518
vehicle	481	777	374,135

Figure A.2: Statistics of common sub-trees in the Fall 2011 release of ImageNet. The sub-trees listed are not mutually exclusive to each other. Source is from [34].

One of the main assets of WordNet [86] lies in its semantic structure, i.e., its ontology of concepts. Similar to WordNet, synsets of images in ImageNet are interlinked by several types of relations, the "is-a" relation being the most comprehensive and useful. The "is-a" relation forms a hierarchy of synsets, or more specifically a directed acyclic graph (DAG) [34]. Using ImageNet as a dataset for training concept detection thus has three obvious advantages: it is large, it has a hierarchical structure and it is relatively easy to download having been used by many other researchers which means that our work can be

compared against work of others because we share this training dataset.

To fairly compare non-lifelog images and lifelog images, the same varieties of objects from non-lifelog images and lifelog images are chosen. Because the ImageNet has a sufficiently wide range of objects, 21 objects are selected because they are the maximum number of objects that our lifelog dataset can provide. The lifelog dataset will be detailed in Section A.3.

As ImageNet does not possess the copyright or ownership of the images that have been added to WordNet synsets, it merely provides the URL links to images so researchers can download and use them directly. This means that on occasion, images may be missing and as a consequence, nothing will be returned when the researcher tries to retrieve the image or a warning will be returned, as shown in Figure. A.3.

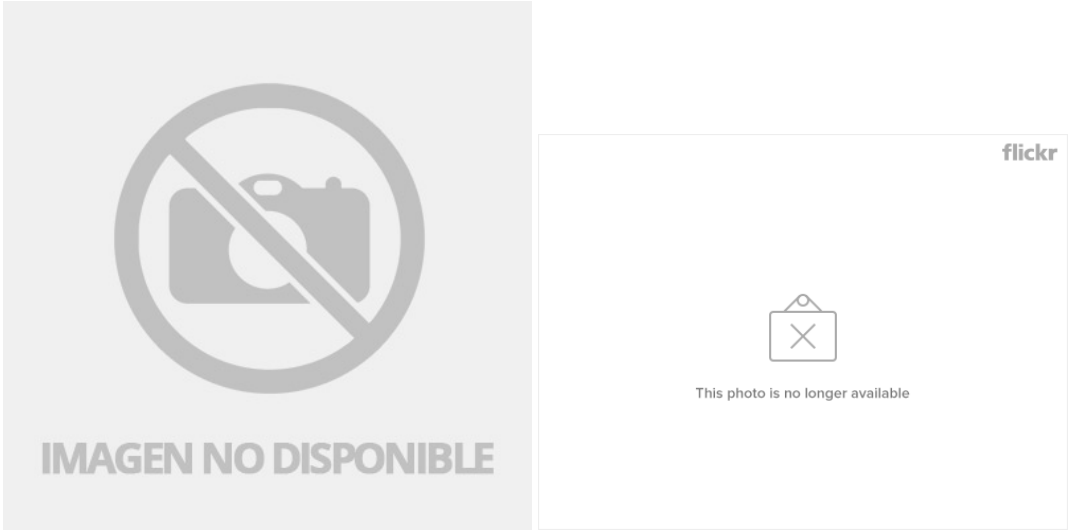


Figure A.3: Examples of downloaded images when original links are missing from ImageNet.

There are other issues as well. A downloaded file may have the wrong file extension, for example, a file with ".jpg" can be an HTML file. Starting from the file with URL links to images, four operations are performed to download, filter and pre-process images. Details of this processing are given for better reproducibility:

Step.1 Requested. The "wget" command downloads images according to the provided URL

links with a default setting[4]; however we tried to download images at most twice;

Step.2 Corrected. We check for correct extensions of the downloaded files and only retain files of format ".jpg", ".png" and ".gif";

Step.3 Actual. Manually find a sample of invalid images (as in Figure. A.3) and computationally remove common duplicates;

Step.4 Final. Discard all 2-dimensional (grey scale) images; while for 4-dimensional images, only keep the first one as they are similar; discard corrupted images; re-size images to 112×112.

The four columns in table A.1 show the number of images obtained after the corresponding four steps above for each object.

For the work done in this thesis, we focused on 21 specific objects which are described in Table A.1. This table shows in column 1, the changes of the number of each class after every operation. Note as Step. 4 does not change the number of images, the final number of examples is the same after Step. 3. The instances are further split into training, validation and test sets according to the proportions 6:2:2. Note that in our work we do not intentionally balance the number of images for each class ("window" has 524 vs "door" has 1,214), because when we train a model for each object/concept then the model itself should adapt to this common situation of variable amounts of training data.

In some cases on ImageNet, there is an ambiguity in terms of general concepts such as signs. In our subset of ImageNet, some images have both signs and trains which are labeled as "sign". In such cases, we remove these images to avoid ambiguity.

A.2 Lifelog Data for Detection

Table A.2 lists the publicly accessible egocentric datasets which contain the ground-truth annotation of objects and they are taken from the summary of [15] (see reference [15] for more datasets). The table provides the overall descriptions, annotation types, modality and

[4] https://www.gnu.org/software/wget/

	Requested	Corrected	Actual	Final
	Step. 1	Step. 2	Step. 3	Step. 4
aircon	1,727	1,008	883	883
bicycle	1,344	1,149	939	938
bottle	1,228	1,034	923	923
building	1,421	1,117	1,017	1,016
car	910	595	534	534
chair	1,460	1,059	909	907
cupboard	1,290	959	806	806
dish	1,186	1,114	992	992
door	1,462	1,428	1,214	1,214
face	1,570	1,459	949	948
glass	1,337	1,143	975	973
hand (flickr)	995	995	994	988
lamp	1,847	1,259	1,114	1,112
mobilephone	1,422	1,181	915	915
motorbike	1,380	1,014	857	856
paper	407	277	241	240
person	1,242	1,179	881	881
sign	1,199	1,177	1,012	1,012
train	1,312	1,167	929	906
tvmonitor	1,399	1,129	960	958
window	1,230	568	525	524
Average	1,303	1,048	884	882

Table A.1: Number of ImageNet Images after each step of the download process

addresses for download. The datasets consisting of images is more usable for our work than datasets of videos because images are easier to handle and the sequential characteristics of videos have no benefit for the task of object detection. Although AIHS [68] has much more data than EDUB [16], the reason EDUB is adopted instead of AIHS is that AIHS has many more categories of objects: the EDUB has 21 object classes while AIHS has only two objects among 45 concepts including places, objects, and activities. The NTCIR-Lifelog (100-day) dataset is also not adopted for this work because as it uses the output of the Caffe [65] as the source of the visual annotation, instead of manual annotations (ground-truth) and these automatic annotations will have a significant error rate.

The egocentric dataset of the University of Barcelona (EDUB) is a visual lifelog dataset composed of 4,912 images acquired by 4 people over 2 days using the Narrative[5] wearable

[5] www.getnarrative.com

Abbreviation	Description	Annotation Type	Modality	Full Name (download address)
EDUB [16]	4,912 images are collected by 4 people, 2 days per person.	Objects	Images	Egocentric Dataset of the University of Barcelona
AIHS [68]	45,612 images from 2 weeks.	Objects, activities, scenes	images	All I Have Seen
IEOD [99]	10 videos from 2 subjects manipulating 42 objects. Objects are labelled and foreground plus background are segmented.	Objects in hands	Videos	Intel Egocentric Object Dataset
GTEA [42]	10 videos from 2 subjects manipulating 42 objects. Objects are labelled and foreground plus background are segmented	Objects in activities	Videos	GeorgiaTech Egocentric Activities
ADLD [42]	10 video clips are collected for each activity and each clip spans 30 seconds.	activities, object tracks, hand positions, and interaction events	Videos	Activities of Daily Living Dataset
NTCIR-Lifelog [67]	100-day dateset collected from 3 lifeloggers and 1,000–1,500 images per day.	Images, time, automatic annotation of visual concepts, locations and motions (the automatic annotation is defined in chapter 3).	Images and other data	Lifelog for NTCIR

Table A.2: Summary of important available public egocentric datasets for object detection, recognition, or segmentation.

camera. The dataset is divided into 8 different days which capture daily life activities like shopping, eating, riding a bike, working, attending meetings, commuting to work, etc. The actual number of images collected may be lower than the device is able to collect because some images would be deleted considering privacy and lifeloggers may forget to turn on the device sometimes. The objects appearing in the images were annotated using the online tool LabelMe [102] and their annotation files include the bounding contour of objects and the name of objects.

The original lifelog images allocated for training in the task of object detection turns out to be 1,693 (accounting for 34%), while the number of lifelog images for the test is 3,219 (accounting for 66%). The number of blank (no object contained) lifelog images is 875.

A.3 Lifelog Data for Object Recognition

The expected input into the object recognition task is different from that of the object detection task. Object detection allows multiple objects or no object at all on the image, but object recognition requires its input have one and only one object. Section A.2 makes the decision to take EDUB [16] as the lifelog dataset. The original lifelog image may have multiple objects or no object, so it is necessary to crop regions of objects from lifelog images. The EDUB dataset offers the coordinates for each object.

In order for better recognition and evaluation, the original contour annotations are converted to rectangular annotations. The stratagem here is to find the minimum square area that could cover all coordinates from the annotation. The advantage of the stratagem is that it contains all the necessary pixels, while the disadvantage is that the cropped area may carry pixels not belonged to the object. The processing python code is given below:

Algorithm 1: Counter annotations to rectangular annotations (Python)

```
# The annotated contour (obj) consists of a series of points,
    each has x and y to indicate the coordinates.
x = [points.text for points in obj.iter('x')]
```

```
y = [points.text for points in obj.iter('y')]
x = map(int, x)
y = map(int, y)
# The biggest rectangular annotation of contour annotation is
    decided by its top left and bottom right points.
max_x = max(x)
min_x = min(x)
max_y = max(y)
min_y = min(y)
```

The cropped bounding boxes are re-scaled to the consistent size of 112×112, because the cropped areas have different sizes while classifiers will work better when the dimensions of inputs are consistently the same.

	Total	Training set	Test set
aircon	530	106	424
bicycle	10	2	8
bottle	260	52	208
building	732	146	586
car	315	63	252
chair	179	36	143
cupboard	392	78	314
dish	65	13	52
door	199	40	159
face	565	113	452
glass	831	166	665
hand	1,232	246	986
lamp	2,299	460	1,839
mobilephone	145	29	116
motorbike	53	11	42
paper	377	75	302
person	1,175	235	940
sign	506	101	405
train	4	1	3
tvmonitor	1,274	255	1,019
window	138	28	110
average	537.2	107.4	429.8

Table A.3: Total number of EDUB data and the numbers of training and test sets for each class.

Because lifelog data is insufficient, especially for some classes, it is only divided into training and test sets, in proportion 1:4, and the validation set is left vacant. Table A.3 lists the numbers for the EDUB data when splitting the results. The first column records the total amount, the second and third columns are the numbers of images from training and test sets respectively. From the table, there is an obvious fact: the numbers of images in each class are obviously imbalanced (e.g., the object "train" has only 4 instances while the object "lamp" has thousands), which is caused by the occurrences of objects encountered in everyday life. By default, they are all used as part of the test set unless explicitly mentioned.

Appendix B

Author's Publications

Up to 2017, I have published six papers.

1. Dancheng Li, Weipeng Jin, Guoqi Liu, Xiang Shen, Tengqi Ye, and Zhiliang Zhu (2010). The research on automatic generation of testing data for Web service. In: *Information Science and Engineering (ICISE) 2010 2nd International Conference on* (pp. 1629-1632). IEEE.

2. Ying Liu, Tengqi Ye, Guoqi Liu, Cathal Gurrin, and Bin Zhang. Demographic attributes prediction using extreme learning machine. In *Extreme Learning Machines 2013: Algorithms and Applications*, pages 145-165. Springer, 2014.

3. TengQi Ye, Brian Moynagh, Rami Albatal, and Cathal Gurrin. Negative faceblurring: A privacy-by-design approach to visual lifelogging with google glass. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 2036-2038. ACM, 2014.

4. Lijuan Marissa Zhou, Brian Moynagh, Liting Zhou, TengQi Ye, and Cathal Gurrin. Memlog, an enhanced lifelog annotation and search tool. In *MultiMedia Modeling*, pages 303-306. Springer, 2015.

5. TengQi Ye, Tianchun Wang, Kevin McGuinness, Yu Guo, and Cathal Gurrin. Learning multiple views with orthogonal denoising autoencoders. In *International Confer-*

ence on Multimedia Modeling, pages 313-324. Springer, 2016.

6. Tianchun Wang, TengQi Ye, and Cathal Gurrin. Transfer nonnegative matrix factorization for image representation. In *MultiMedia Modeling*, pages 3-14. Springer, 2016.

The first two works (paper 1 and paper 2) took place when I was undergraduate at Northeastern University, China. The settings of paper 2 is improper. Some formulations of paper 5 are incorrect. I found those error long time after they published. Paper 4 and paper 3 are demo papers. I developed software for paper 1 and paper 4. I did limited work on paper 6.

Bibliography

[1] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Omid Aghazadeh, Josephine Sullivan, and Stefan Carlsson. Novelty detection from an ego-centric perspective. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3297–3304. IEEE, 2011.

[3] Kiyoharu Aizawa. Digitizing personal experiences: Capture and retrieval of life log. In *11th International Multimedia Modelling Conference*, pages 10–15. IEEE, 2005.

[4] Kiyoharu Aizawa, Ken-Ichiro Ishijima, and Makoto Shiina. Summarizing wearable video. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 398–401. IEEE, 2001.

[5] S Annadurai. *Fundamentals of digital image processing*. Pearson Education India, 2007.

[6] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[7] Nuria Bel, Cornelis HA Koster, and Marta Villegas. Cross-lingual text categorization. In *International Conference on Theory and Practice of Digital Libraries*, pages 126–139. Springer, 2003.

[8] C Gordon Bell and Jim Gemmell. *Total recall: How the e-memory revolution will change everything*. Dutton, 2009.

[9] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[10] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 137–144, 2007.

[11] Yoshua Bengio, Ian J. Goodfellow, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2015.

[12] Michael W Berry and Murray Browne. *Understanding search engines: mathematical modeling and text retrieval*, volume 17. Siam, 2005.

[13] Alejandro Betancourt, Pietro Morerio, Carlo S Regazzoni, and Matthias Rauterberg. The evolution of first person vision methods: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(5):744–760, 2015.

[14] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[15] Marc Bolaños, Mariella Dimiccoli, and Petia Radeva. Towards storytelling from visual lifelogging: An overview. *IEEE Transactions on Human-Machine Systems*, PP(99):1–14, 2016.

[16] Marc Bolaños and Petia Radeva. Ego-object discovery. *arXiv preprint arXiv:1504.01639*, 2015.

[17] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[18] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[19] Vannevar Bush. As we may think. *ACM SIGPC Notes*, 1(4):36–44, 1979.

[20] Daragh Byrne, Aiden R Doherty, Cees GM Snoek, Gareth JF Jones, and Alan F Smeaton. Everyday concept detection in visual lifelogs: validation, relationships and trends. *Multimedia Tools and Applications*, 49(1):119–144, 2010.

[21] Niamh Caprani. *The design of an intergenerational lifelog browser to support sharing within family groups*. PhD thesis, Dublin City University, 2013.

[22] Niamh Caprani, John Greaney, and Nicola Porter. A review of memory aid devices for an ageing population. *PsychNology Journal*, 4(3):205–243, 2006.

[23] Joao Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010.

[24] Vijay Chandrasekhar, Cheston Tan, Wu Min, Liyuan Li, Xiaoli Li, and Joo-Hwee Lim. Incremental graph clustering for efficient retrieval from streaming egocentric video data. In *ICPR*, pages 2631–2636. Citeseer, 2014.

[25] Neelima Chavali, Harsh Agrawal, Aroma Mahendru, and Dhruv Batra. Object-proposal evaluation protocol is' gameable'. *arXiv preprint arXiv:1505.05836*, 2015.

[26] Minmin Chen, Kilian Q Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-encoders for nonlinear representations. In *ICML*, pages 1476–1484, 2014.

[27] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 767–774, New York, NY, USA, July 2012. Omnipress.

[28] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region

proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1201–1210, 2015.

[29] Sarah Clinch, Paul Metzger, and Nigel Davies. Lifelogging for'observer'view memories: an infrastructure approach. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1397–1404. ACM, 2014.

[30] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[31] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[32] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[33] Ana Garcia del Molino, Cheston Tan, Joo-Hwee Lim, and Ah-Hwee Tan. Summarization of egocentric videos: A comprehensive survey. *IEEE Transactions on Human-Machine Systems*, 2016.

[34] Jia Deng. *Large scale visual recognition*. PhD thesis, PRINCETON UNIVERSITY, 2012.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[36] Aiden R Doherty. *Providing effective memory retrieval cues through automatic structuring and augmentation of a lifelog of images*. PhD thesis, Dublin City University, 2009.

[37] Pedro Domingos. A unifeid bias-variance decomposition and its applications. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 231–238. Morgan Kaufmann Publishers Inc., 2000.

[38] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[39] Ian Endres and Derek Hoiem. Category-independent object proposals with diverse ranking. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):222–234, 2014.

[40] Markus Enzweiler and Dariu M Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2179–2195, 2009.

[41] Alircza Fathi, Jessica K Hodgins, and James M Rehg. Social interactions: A first-person perspective. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1226–1233. IEEE, 2012.

[42] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On*, pages 3281–3288. IEEE, 2011.

[43] Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1(1):1–47, 1991.

[44] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[45] Sven Fleck and Wolfgang Straßer. *Privacy Sensitive Surveillance for Assisted Living – A Smart Camera Approach*, pages 985–1014. Springer US, Boston, MA, 2010.

[46] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1180–1189, 2015.

[47] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[48] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/-variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[49] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits: fulfilling the memex vision. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 235–238. ACM, 2002.

[50] David Geronimo, Antonio M Lopez, Angel D Sappa, and Thorsten Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1239–1258, 2010.

[51] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[52] Rafael C Gonzalez and Richard E Woods. Digital image processing. *Nueva Jersey*, 2008.

[53] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Ntcir lifelog: The first test collection for lifelog research. In *Proceedings of the 39th Annual ACM SIGIR Conference*, 17-21 July 2016.

[54] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Overview of ntcir-12 lifelog task. 2016.

[55] Cathal Gurrin, Alan F Smeaton, and Aiden R Doherty. Lifelogging: Personal big data. *Foundations and Trends in Information Retrieval*, 8(1):1–125, 2014.

[56] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2010.

[57] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[59] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. Sensecam: A retrospective memory aid. In *UbiComp 2006: Ubiquitous Computing*, pages 177–193. Springer, 2006.

[60] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2016.

[61] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[62] Feiyan Hu. *Periodicity Detection and its Application in Lifelog Data*. PhD thesis, Dublin City University, 2016.

[63] Feiyan Hu and Alan Smeaton. Periodicity intensity for indicating behaviour shifts from lifelog data. In *IEEE International Conference on Bioinformatics and Biomedicine — International Workshop on Biomedical and Health Informatics*, Dec 2016.

[64] Tim Jacquemard. *Ethics of lifelog technology*. PhD thesis, Dublin City University, 2014.

[65] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[66] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 2008.

[67] Hideo Joho, Frank Hopfgartner, and Cathal Gurrin. Information retrieval and learning with lifelogging devices: a session for interaction and engagement at iconference 2016. 2016.

[68] Nebojsa Jojic, Alessandro Perina, and Vittorio Murino. Structural epitome: a way to summarize ones visual experience. In *Advances in neural information processing systems*, pages 1027–1035, 2010.

[69] Vaiva Kalnikaite, Abigail Sellen, Steve Whittaker, and David Kirk. Now let me see where i was: understanding how lifelogs mediate memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2045–2054. ACM, 2010.

[70] Kris Makoto Kitani. *Modeling and Recognizing Human Activities from Video*. PhD thesis, 2008.

[71] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[72] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

[73] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2129–2142, 2009.

[74] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[75] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[76] Qi Li. Literature survey: domain adaptation algorithms for natural language processing. *Department of Computer Science The Graduate Center, The City University of New York*, pages 8–10, 2012.

[77] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[78] Ying Liu, Tengqi Ye, Guoqi Liu, Cathal Gurrin, and Bin Zhang. Demographic attributes prediction using extreme learning machine. In *Extreme Learning Machines 2013: Algorithms and Applications*, pages 145–165. Springer, 2014.

[79] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[80] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 97–105, 2015.

[81] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[82] Steve Mann. Wearable computing: A first step toward personal imaging. *Computer*, 30(2):25–32, 1997.

[83] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[84] Michael Massimi, Khai Truong, David Dearman, and Gillian Hayes. Understanding recording technologies in everyday life. *IEEE Pervasive Computing*, 9(3):64–71, 2010.

[85] Jochen Meyer, Steven Simske, Katie Siek, Cathal Gurrin, and Hermie J Hermens. Beyond quantified self: data for wellbeing. In *Extended Abstracts on Human Factors in Computing Systems*, 2014.

[86] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[87] Ion Muslea, Steven Minton, and Craig A Knoblock. Active learning with multiple views. *Journal of Artificial Intelligence Research*, pages 203–233, 2006.

[88] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[89] Paul Over, Jon Fiscus, Greg Sanders, David Joy, Martial Michel, George Awad, Alan Smeaton, Wessel Kraaij, and Georges Quénot. Trecvid 2014–an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, page 52, 2014.

[90] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418, 2009.

[91] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[92] Paulina Piasek. *Case studies in therapeutic SenseCam use aimed at identity maintenance in early stage dementia*. PhD thesis, Dublin City University, 2015.

[93] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[94] Peter Prettenhofer and Benno Stein. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127. Association for Computational Linguistics, 2010.

[95] Zhengwei Qiu. *A lifelogging system supporting multimodal access*. PhD thesis, Dublin City University, 2013.

[96] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity recognition from accelerometer data. In *AAAI*, volume 5, pages 1541–1546, 2005.

[97] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[98] Xiaofeng Ren and Chunhui Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3137–3144. IEEE, 2010.

[99] Xiaofeng Ren and Matthai Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *Computer Vision and Pattern Recognition Workshops,*

2009. CVPR Workshops 2009. IEEE Computer Society Conference on, pages 1–8. IEEE, 2009.

[100] Bernardino Romera-Paredes and PHS Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2152–2161, 2015.

[101] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[102] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[103] Michael S Ryoo and Larry Matthies. First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2737, 2013.

[104] Bahjat Safadi, Philippe Mulhem, Georges Quénot, and Jean-Pierre Chevallet. Lig-mrim at ntcir-12 lifelog semantic access task. *Proceedings of NTCIR-12, Tokyo, Japan*, 2016.

[105] Harrisen Scells, Guido Zuccon, and Kirsty Kitto. Qut at the ntcir lifelog semantic access task. *Proceedings of NTCIR-12, Tokyo, Japan*, 2016.

[106] Abigail J Sellen and Steve Whittaker. Beyond total capture: a constructive critique of lifelogging. *Communications of the ACM*, 53(5):70–77, 2010.

[107] Alan F Smeaton, Kevin McGuinness, Cathal Gurrin, Jiang Zhou, Noel E O'Connor, Peng Wang, Brian Davis, Lucas Azevedo, Andre Freitas, Louise Signal, et al. Semantic Indexing of Wearable Camera Images: Kids'Cam Concepts. In *Proceedings of the ACM Multimedia 2016 Workshop on Vision and Language Integration Meets Multimedia Fusion Amsterdam, The Netherlands October 16, 2016*. ACM Press, 2016.

[108] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[109] Sibo Song, Vijay Chandrasekhar, Ngai-Man Cheung, Sanath Narayan, Liyuan Li, and Joo-Hwee Lim. Activity recognition in egocentric life-logging videos. In *Computer Vision-ACCV 2014 Workshops*, pages 445–458. Springer, 2014.

[110] Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[111] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[112] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

[113] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[114] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013.

[115] Estefania Talavera, Mariella Dimiccoli, Marc Bolanos, Maedeh Aghaei, and Petia Radeva. R-clustering for egocentric video segmentation. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 327–336. Springer, 2015.

[116] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[117] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[118] Peng Wang. *Semantic interpretation of events in lifelogging*. PhD thesis, Dublin City University, 2012.

[119] Peng Wang, Lifeng Sun, Shiqiang Yang, and Alan F Smeaton. Improving the classification of quantified self activities and behaviour using a fisher kernel. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 979–984. ACM, 2015.

[120] Peng Wang, Lifeng Sun, Shiqiang Yang, and Alan F Smeaton. Training-free indexing refinement for visual media via multi-semantics. *Neurocomputing*, 2016.

[121] Tianchun Wang, TengQi Ye, and Cathal Gurrin. Transfer nonnegative matrix factorization for image representation. In *MultiMedia Modeling*, pages 3–14. Springer, 2016.

[122] Wei Wang, Hao Wang, Chen Zhang, and Fanjiang Xu. Transfer feature representation via multiple kernel learning. In *AAAI*, pages 3073–3079, 2015.

[123] Xuezhi Wang and Jeff Schneider. Flexible transfer learning under support and model shift. In *Advances in Neural Information Processing Systems*, pages 1898–1906, 2014.

[124] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[125] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[126] Ming-Hsuan Yang, David J Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1):34–58, 2002.

[127] TengQi Ye, Brian Moynagh, Rami Albatal, and Cathal Gurrin. Negative faceblurring: A privacy-by-design approach to visual lifelogging with google glass. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 2036–2038. ACM, 2014.

[128] TengQi Ye, Tianchun Wang, Kevin McGuinness, Yu Guo, and Cathal Gurrin. Learning multiple views with orthogonal denoising autoencoders. In *International Conference on Multimedia Modeling*, pages 313–324. Springer, 2016.

[129] Ryo Yonetani, Kris M Kitani, and Yoichi Sato. Recognizing micro-actions and reactions from paired egocentric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2629–2638, 2016.

[130] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[131] Jian Zhang, Zoubin Ghahramani, and Yiming Yang. Learning multiple related tasks using latent independent component analysis. In *Advances in neural information processing systems*, pages 1585–1592, 2005.

Glossary

Accuracy Accuracy is a measurement to describe the percentage of the true values. 63

Activation function Activation function is the last operation before the data fed into next layer. It usually does not change the shape of inputs. 52

Bias The bias is an offset to neurons. 52

Bystander People who get involved while lifelogger perform lifelogging. 17

Convolutional neural network Convolutional neural network are a subset of neural networks. They are a family of machine learning models which are widely adopted in computer vision. Convolutional neural networks usually have convolutional layers as first several layers. 51

Domain adaptation Domain adaptation is a subcategory of transfer learning. It focuses on the problem when feature spaces of source domain and target domain are different. 42

Exhaustive search It is a method to propose regions in object detection. Its idea is to proposes regions everywhere given some guidance. 88, 141

Fully-connected neural network Fully-connected neural network are a subset of neural networks. Every neuron has a connection to every enuron on adjacent layers. 51

Lifelog The data produced during lifelogging. 15

Lifelogger People who perform lifelogging. 15, 140

Lifelogging An activity which is related to capturing, recording, and sharing everyday life. 15, 140, 141

Max pooling Max pooling is a way of pooling. Max pooling chooses the maximum value from several values of previous layer. 56

Neural network Neural networks in the thesis are artifical neural networks by default. They are a family of machine learning models. 51, 140, 141

Selective search Just as exhaustive search, it is a method to propose regions in object detection, too. It is a more advanced techniques which could produce more effective regions. 88

Total capture The complete record of everyday life. It is an ideal mode of lifelogging. 17

Transfer learning Transfer learning is a research topic in machine learning that focuses on utilizing knowledge gained from solving one problem and applying it to a different but related problem. 42, 140

Weight A weight is the value of a connection between two neurons of neural networks. 52