Fine-grained and Continual Visual Recognition for Assisting Visually Impaired People

MARCUS KLASSON

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av Teknologie doktorexamen i elektroteknik fredagen den 18 januari 2022 klockan 14.00 i Sal F3, Lindstedtsvägen 26, Kungliga Tekniska Högskolan, Stockholm.

Abstract

In recent years, computer vision-based assistive systems have enabled visually impaired people to use automatic object recognition on their mobile phones. These systems should be capable of recognizing objects that are important for the user on a fine-grained level. To this end, we have focused on the particular application of classifying food items which can be challenging for blind/low-vision people since visual information is often required for distinguishing between similar items. In Paper A, we present a challenging image dataset of groceries taken in grocery stores where each item is hierarchically labeled to capture the fine-grained structure of the various items. Furthermore, we demonstrate in Paper B how more easily accessible information about the items, such as web-scraped images and text descriptions, can be utilized for enhancing the classification performance of groceries compared to only using the real-world images for training.

A valuable feature of assistive vision systems is the capability of adapting to new object classes. The main challenge here is to avoid catastrophically forgetting previously learned knowledge when the classifier is updated with new classes. In Paper C, we propose a new continual learning setting for replay-based methods that aligns well with real-world needs where constraints are placed on processing time rather than the storage capacity of old samples. We then study the timing of replaying certain tasks and show that learning replay schedules over which tasks to replay can be critical for the final classification performance in our proposed setting. Finally, in Paper D, we present a method based on reinforcement learning for learning a policy for selecting which tasks to replay at different times. The benefit of our learned replay scheduling policy is that it can be applied to any new continual learning scenario for mitigating catastrophic forgetting in a classifier without additional computational cost.

To conclude, I will discuss some potential future directions for the development of the next generation of computer vision-based assistive technologies.

Sammanfattning

hej

List of Papers

A **A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels**
Marcus Klasson, Cheng Zhang, Hedvig Kjellström
In *IEEE Winter Conference on Applications of Computer Vision (2019)*

B **Using Variational Multi-view Learning for Classification of Grocery Items**
Marcus Klasson, Cheng Zhang, Hedvig Kjellström
In *Patterns, Volume 1(8) (2020)*

C **Learn the Time to Learn: Replay Scheduling for Continual Learning**
Marcus Klasson, Hedvig Kjellström, Cheng Zhang
Under submission

D **Meta Policy Learning for Replay Scheduling in Continual Learning**
Marcus Klasson, Hedvig Kjellström, Cheng Zhang
Under preparation

Acknowledgements

hej

Acronyms

List of commonly used acronyms:

CL	Continual Learning
CNN	Convolutional Neural Network
FGIR	Fine-Grained Image Recognition
LSTM	Long Short-Term Memory **MLP** Multilayer Perceptron **PCA** Pr
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
VAE	Variational Autoencoder
VCCA	Variational Canonical Correlation Analysis
VI	Vision Impairment

Contents

Part I

Overview

Chapter 1

Introduction

Vision is probably the most important of all senses that humans possess. Our society is built on having this ability. For example, if we would like to cross a street, there are thick colored stripes on the road or signs above head height that indicate where the cross walk is located such that we can cross the street in an appropriate way. Another example is how we use text to communicate with each other, where words and sentences are composed by structured sequences of symbols that constitute a specific language. Furthermore, it has been shown that learning from both images and text can improve comprehension over learning from text only [1,2]. Possessing normal vision capabilities basically make everyday tasks easier when it comes to reaching destinations in the world, communicating with other people, and learning new concepts.

In 2020, it was estimated to be 43.3 million people who are blind and 295 million people with moderate to severe visual impairment in the world [3]. To enhance the mobility of visually impaired (VI) people, there exist various kinds of assistive devices and tools, such as screen readers and Braille typewriter machines, for supporting them with receiving information and communicating through text. More recently, several computer vision-based assistive vision tools have emerged in the form of wearable devices and mobile applications for helping VIs with tasks where visual information is a must, for example, object recognition [4–6] and wayfinding in natural environments [7–9] and .

Despite the recent successes in computer vision [10–12], these methods can face several challenges when deployed in the real-world which makes their recognition performance suffer. For example, it can be difficult for the methods to distinguish between similar items on a fine-grained level, such as different brands of apples and pears, as well as performing robustly in environments with noisy backgrounds and poor lighting. Part of the reason for such challenges is that specifying a model of the visual world that has been injected with knowledge about the rich complexity that can exist in images is very difficult [13]. Therefore, there is a necessity for developing computer vision methods that can recognize different

appearances of objects, adapt to changes of known objects, and learn what new objects look like. At the same time, these tasks should be possible to execute in a time-efficient and robust manner.

In this thesis, we address the challenges on robustness in fine-grained classification as well as how the method can learn to recognize new object classes. We will begin this introduction by briefly describing vision impairments in Section 1.1, followed by a summary of assistive vision technologires in Section 1.2. Then we describe the scope of the thesis in Section 1.3 and summarize the contributions of the included papers in Section 1.4. Finally, in Section 1.5, we give the outline to the rest of the contents in this thesis.

1.1   Vision Impairments

Vision impairment (VI) is defined as the decrease of one's ability to see from various distances [14]. There are different types of VIs ranging from various degrees of blindness to having issues with seeing from far or near distances. The visual capabilities are in general assessed by measuring the *visual acuity* (sharpness) of seeing, for example, a letter or symbol, from some fixed distance. The visual acuity measured differently based on whether near- or far-sighted VI is assessed. For far-sighted VI, the visual acuity is calculated by the ratio between the distance that the subject can see the item and the distance a normal-sighted person could recognize the item. When assessing near-sighted VI, one checks the font size of letters that the subject can see using a standardized point system for measuring the symbol size [15].

In 2020, it was estimated that 338 million people possess moderate to severe VI globally, including 43 million people that are blind [3]. Furthermore, the World Health Organization (WHO) have estimated that at least 2.2 billion people live with a near or distance VI, where at least 1 billion cases could have been prevented or yet has to be addressed [15]. The untreated cases are projected to grow to 1.7 billion people by 2050 mainly due to population growth in the world as well as increased aging among the populations [3]. The leading causes for vision loss are uncorrected refractive errors, untreated cataracts, age-related macular degenerationm, glaucoma, diabetic retinopathy, where 90% of such cases are preventable and treatable [16]. The causes for vision loss also differs between countries and areas with different incomes.

There exists several tools for assisting VI people with everyday tasks. The *white cane* is probably the most common tool among VI people which is used for wayfinding to help the user anticipate what is present in their near surroundings. Also, guiding dogs are used for enhancing mobility by helping VI people to maintain a direct route, avoid obstacles, and prepares owner by stopping at curbs and stairways until they are told to proceed [17]. There also exist several tools for recognition tasks. For example, currency markers are used for keeping track of different bills in wallets, color indicators can be used to tell the user of the color of

clothes, and labeling apparatus are used for distinguishing between similar items. Means for communication also exists in the form of Braille keybords and screen readers that are used in both computers and mobile phones to provide nearly equal opportunities for VI people when it comes to office-related tasks. There has been a recent emergence of various devices that are aimed to assist VI people with object recognition tasks which we will discuss next.

1.2 Assistive Vision Technologies

Cameras are used by people with VIs, including blindness, to record events and memories similarly as normal-sighted people [18]. This has opened up for opportunities where VI people can use their cameras for more than recording events, for example, object recognition, document text recognition, and color identification. Object recognition has been shown to be considered an everyday challenge, where VI people would like to ask questions about objects where visual information is necessary for identification [19]. For example, it can be very difficult to distinguish between different containers and packages that have similar shapes but different content without being able to see. These findings have encouraged development of technical aids that use computer vision for assisting VI people.

In the last decade, we have seen several variants of assistive vision technologies emerging on the market. There exist many applications for mobile phones where various visual tasks have been cramped in into the app, such as object and face recognition, barcode scanning, color and currency identification, and text recognition [20–23]. Moreover, there exists wearable devices with similar capabilities as the mobile phone apps [24,25] that also use computer vision for assistance. An alternative to the computer vision-based apps there are other mobile applications where VI users can have a video call with sighted volunteers that help them with any kind of task requiring visual capabilities [26,27]. Despite that these assistive vision technologies has opened up for VI people being more independent, there remains several challenges to tackle regarding system requirements [28–30] and privacy concerns [31–33].

Current assistive vision technologies face several challenges that needs to be tackled to enhance their utility for VI people. In the past decade, machine learning techniques have been applied successfully to various computer vision tasks such as object recognition [10,34,35], generating scene descriptions [12,36,37], and visual question answering [38–40]. In addition to better computer hardware, the main reason for these successes is the immense data collection and annotation that is required for obtaining large-scale computer vision datasets. However, the annotation becomes even more costly if the object classes should be separated based on fine-grained details about the objects, which makes it challenging for assistive vision systems to provide users with further information about objects than the general object class. Another challenge is how to update the assistive vision devices with information about new objects to recognize and ensuring that

the system is still able to recognize the previous known items correctly. Furthermore, assistive vision devices should have the ability to answer questions about the surroundings of the user, should perform in real-time and be robust when applied in different environments, as well as ensuring privacy for the user.

1.3   Scope of Thesis

This thesis is focused on two applications for machine learning and computer vision-based assistive technologies, namely *fine-grained classification* [41] and *continual learning* [42, 43]. Fine-grained classification involves identifying subcategories and details of general object classes, which can be important when distinguishing between visually similar items. An example is when one has to distinguish between two juice packages from the same brand where the main ingredients are apples and oranges in the packages. The general setting in fine-grained classification is that all data and classes to learn are given all at once to the classifier to learn, but can be extended to the continual learning setting where the classes to learn are divided into tasks that are learned at different points in time. Continual learning methods are used for updating the classifier's current knowledge with information about the new classes and making sure that the classifier remembers the previously learned classes. The common denominator of these fields is classification, but both have challenges of their own that has to be addressed before adding such features into assistive vision devices. Next, we describe the challenges that we have focused on in this thesis.

Fine-grained Classification

One main challenge for fine-grained classification is the data collection procedure and there are several reasons for this. Firstly, the annotation of the collected data becomes more time-consuming as the annotators must know specific details about the objects to label the data as accurately as possible. Secondly, as fine-grained classes might be rare, there might be few examples per class that the classifier can learn from to discriminate between the objects. An application where an assistive vision device would need to learn fine-grained classes from sparse datasets is grocery shopping for helping VI people [5, 44]. Grocery items usually require visual information to distinguish between them, for example, when one needs to know how the ingredients differ in two juice packages. This also goes for raw grocery items where it might be difficult for a VI customer to tell the difference between two different brands of green apples unless the customer knows how the apples smell or how the texture of their peel differs when touching them. Furthermore, situations in the grocery store environment can disturb the recognition performance of the assistive vision device, for example, when multiple and misplaced items appear in the camera view and also when there are poor lighting settings in some areas of the store. Collecting training data that covers all

possible scenarios that can occur in the store would be a cumbersome procedure. Our goal is to reduce the need for training data in the grocery stores by collecting web-scraped information about the items and using this for easing the learning of the classifier.

Continual Learning

The main challenge in continual learning is called *catastrophic forgetting* [45] which means that the classifier will overwrite previously learned knowledge with information about the new objects of interest during learning. Therefore, we must use additional training techniques that prevents this forgetting effect to maintain the recognition performance on all classes during the lifespan of the classifier. A simple yet efficient approach in continual learning for mitigating catastrophic forgetting is replay-based methods [46, 47]. The main assumption is that we are allowed to keep a low number of examples from every seen class in a small memory buffer. The idea is then to mix the old examples with the training data from new classes, such that we learn the new classes and aim to retain the performance on the old classes by replaying the memory examples for the classifier.

Most previous works on replay-based continual learning ignores the time to replay certain tasks. However, the timing of rehearsal has been shown to be very important for humans to retain memory on various tasks [48–52]. Moreover, in contrast to the constraint on the small memory size, machine learning systems used in real-world applications may be limited by processing times rather than data storage capacity [Add REFs]. In such settings, there is a need for methods that select what data from the huge storage to replay as the problem of catastrophic forgetting still remains. Our goal is to demonstrate that scheduling over which tasks to replay can be crucial for continual learning performance in this setting. Hence, we will need to develop efficient methods that can automatically propose replay schedules that mitigate catastophic forgetting in classifiers to enable this strategy in real-world settings.

1.4 Thesis Contributions

In this section, we provide summaries of the included papers as well as stating the contributions of each author to the manuscripts.

Paper A: A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels

Marcus Klasson, Cheng Zhang, Hedvig Kjellström. In *IEEE Winter Conference on Applications of Computer Vision (WACV) 2019*.

Summary We collect a dataset with natural images of raw and refrigerated grocery items taken in grocery stores in Stockholm, Sweden, for evaluating image

Table 1.1:   Examples of grocery item classes in the Grocery Store dataset. We display four different items (coarse-grained class in parenthesis), followed by two natural images taken with a mobile phone inside grocery stores. Next comes the web-scraped information of the items consisting of an iconic image and a text description. We have highlighted ingredients and flavors in the text description that are characteristic for the specific item.

Class Labels	Natural Images	Iconic Images	Text Descriptions
Granny Smith (Apple)			"...**green** apple with **white, firm** pulp and a **clear acidity** in the flavor."
Royal Gala (Apple)			"...**crispy** and **very juicy** apple, with **yellow-white** pulp. The peel is **thin** with a **red yellow** speckled color."
Tropicana Mandarin (Juice)			"...is a **ready to drink** juice **without pulp** pressed on **orange**, **mandarin** and **grapes**. Not from concentrate. Mildly **pasteurized**."
Yoggi Vanilla (Yoghurt)			"...**creamy vanilla** yoghurt original... added **sugar** than regular flavored yoghurt. Great for both **breakfast and snacks**."

classification models on a challenging real-world scenario. The data collection was performed by taking photos of groceries with a mobile phone to simulate a scenario of grocery shopping using an assistive vision app. Furthermore, we downloaded iconic images and text descriptions of each grocery item by web-scraping a grocery store website to enhance the dataset with information describing the semantics of each individual item. The items are grouped based on their type, e.g., apple, juice, etc., to provide the dataset with a hierarchical labeling structure. We show some examples of grocery item classes and their corresponding web-scraped information in Table 1.1.

We provide benchmark results evaluated using pre-trained and fine-tuned CNNs for image classification. Moreover, we take an initial step towards utilizing the rich product information in the dataset by training the classifiers with representations where both natural and iconic images have been combined through a multi-view VAE.

Author Contributions   CZ and HK presented the idea and the data collection procedure for the natural images and web-scraped information. MK performed the data collection including visiting the grocery stores for taking the natural

images and the web-scraping of the grocery store website for iconic images and text descriptions. MK performed all the experiments. All authors contributed to discussing the results and contributed to writing the manuscript.

Paper B: Using Variational Multi-View Learning for Classification of Grocery Items

Marcus Klasson, Cheng Zhang, Hedvig Kjellström. In *Patterns, Volume 1(8) (2020)*.

Summary   We investigate whether training image classifiers can benefit from learning joint representations of grocery items using multi-view learning over the natural images and web-scraped information of the grocery items in the Grocery Store dataset (see Paper 1.4). We employ a deep multi-view model based on VAEs called Variational Canonical Correlation Analysis (VCCA) [53] for learning joint representations of the different data types, i.e., natural images, iconic images, and text descriptions. We performed a thorough ablation study over all data types to demonstrate how they contribute individually to enhancing the classification performance. Furthermore, we apply two classification approaches where we (i) train the classifier on the joint latent representations, and (ii) using a generative classifier by incorporating a class decoder to the VCCA model that can be used for classifying images.

We performed a thorough ablation study over all data types to demonstrate how they contribute individually to enhancing the classification performance. To gain further insights into our results, we visualized the learned representations of the grocery items from VCCA and discussed how the iconic images and text descriptions help the model to better distinguish between the groceries. Our results show that the iconic images help to group the items based on their color and shape while text descriptions separate the items based on differences in ingredients and flavor. Figure 1.1 shows visualizations of the latent representations projected in 2-dimensional space using Principal Component Analysis (PCA), where we illustrate how the latents change when adding either the iconic image or text description into the VCCA model. Finally, we concluded that utilizing the iconic images and text descriptions yielded better classification results than only using natural images.

Author Contributions   CZ and HK presented the idea and all authors contributed to formalizing the methodology. MK performed all the experiments and created the visualizations. All authors took part in discussing the results. All authors contributed to writing the manuscript.

Paper C: Learn the Time to Learn: Replay Scheduling for Continual Learning

Marcus Klasson, Hedvig Kjellström, Cheng Zhang. Submitted to *International Conference on Machine Learning (ICML) 2022*.

Summary   In this paper, we show that learning the time to replay different tasks can be critical for continual learning (CL) performance in replay-based methods. As the main assumption in replay-based CL is that only a small set of historical data can be re-visited for mitigating catastrophic forgetting, most works have focused on improving the sample quality of the replay memory. However, in many real-world applications, historical data is accessible at all times, e.g., by storing it on the cloud. But although all historical data could be stored, retraining machine learning systems on a daily basis is prohibitive due to processing times and operational costs. Therefore, small replay memories are still needed in CL to mitigate catastrophic forgetting when learning new tasks. To this end, we propose to learn the time to learn for a CL system, in which we learn schedules over which tasks to replay at different times. Inspired by human learning, we demonstrate that scheduling over the time to replay is critical to the final CL performance with finite memory resources. We then illustrate our idea with scheduling over which tasks to replay by learning such policy with Monte Carlo tree search. We perform extensive evaluation showing that learning replay schedules can significantly improve the performance compared to baselines without learned scheduling. We also show that our method can be combined with any replay-based method and memory selection technique. Finally, our results indicate that the learned schedules are also consistent with human learning insights.

Author Contributions   CZ presented the idea and MK and CZ contributed to formalizing the methodology. MK performed all the experiments. All authors took part in discussing the results and contributed to writing the manuscript.

Paper D: Meta Policy Learning for Replay Scheduling in Continual Learning

Marcus Klasson, Hedvig Kjellström, Cheng Zhang. Under preparation for conference submission.

Summary   In this paper, we propose a reinforcement learning-based method for learning policies for replay scheduling that can be applied in new continual learning scenarios. We demonstrated in Paper C that learning the time to replay different tasks is important in continual learning. However, a replay scheduling policy that can be applied in any continual learning scenario is currently absent,

which makes replay scheduling infeasible in real-world scenarios. To this end, we propose using reinforcement learning to enable learning general policies that can generalize across different data domains. The learned policy can then propose replay schedule that efficiently mitigate catastrophic forgetting to improve the continual learning performance without any additional computational cost in the new domain. We compare the learned policies to several replay scheduling baselines and show that the learned policies can improve the continual learning performance given task orders and datasets unseen during training.

Author Contributions CZ presented the idea and MK and CZ contributed to formalizing the methodology. MK performed all the experiments. All authors took part in discussing the results and contributed to writing the manuscript.

1.5 Thesis Outline

The rest of the thesis is organized as follows. We provide some background and preliminaries on the machine learning methodology used in this thesis in Chapter 2. Chapter 3 is focused on our contributions in fine-grained classification where we first describe the related work in this field followed by explaining the frameworks used in our work. Similarly, in Chapter 4, we focus on our contributions in continual learning by first describing the related work to place our contributions in context and then describing the framework we used for enabling replay scheduling in continual learning. Finally, in Chapter 5, we provide our conclusions of the presented works and present some research directions that we believe would be interesting to look deeper into in the future.

MK: todo: fix references to chapters!

(a) VAE$_x$

(b) VAE$_x$ Apples

(c) VAE$_x$ Juice/Yoghurt

(d) VCCA$_{xi}$

(e) VCCA$_{xi}$ Apples

(f)   VCCA$_{xi}$
Juice/Yoghurt

(g) VCCA$_{xw}$

(h) VCCA$_{xw}$ Apples

(i)   VCCA$_{xw}$
Juice/Yoghurt
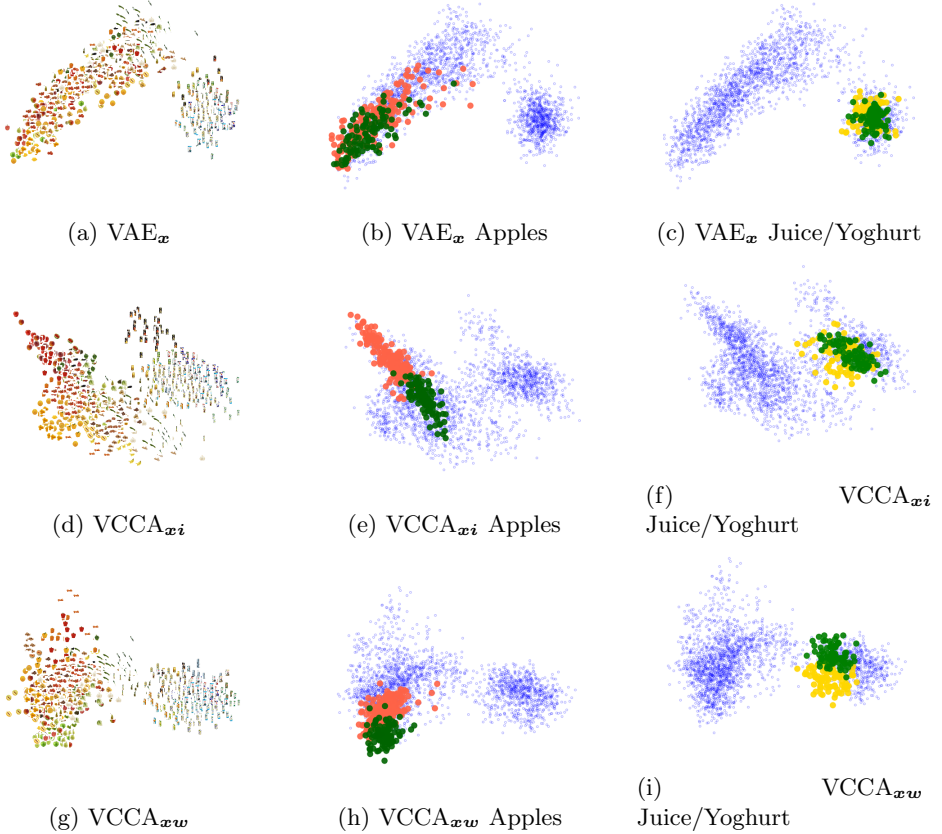
Figure 1.1: Visualizations of the latent representations projected in 2D space with PCA to illustrate how the latent space changes when learning the latent representations from either the iconic images i or the text descriptions w in addition to the natural images x. In the first column, we show the latent representations plotted using the iconic images of the corresponding object class for VAE$_x$, VCCA$_{xi}$, and VCCA$_{xw}$, where the subscript on the models indicates which data views that were utilized. In the second column, we focus on the red and green apple classes in the dataset to demonstrate how the iconic images separates these items based on their different colors. Similarly, in the third column, we focus on the visually similar juice (yellow dots) and yoghurt (green dots) packages to demonstrate how the text descriptions separates these items based on their different ingredients and flavors. The blue dots corresponds to all other grocery item classes.
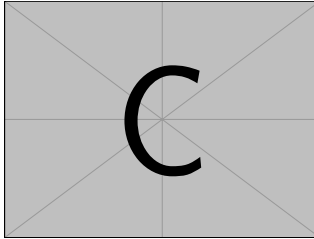
Figure 1.2: **MK: Paper C could illustrate the new CL setup with a figure that shows a network and a scheduler that needs to fetch a small replay memory from a huge pool of historical data.**

Figure 1.3: **MK: Paper D could show illustration of the RL agent scheduler that gets performance measures as input and outputs an action proportion of how to select the replay memory. RL agent could also have a replay buffer where data is collected from several environments. Maybe it can also show the test case, so that it would be two separated "at training/test phase".**

Chapter 2

Background

The goal with this chapter is to provide the reader with preliminaries that are useful for comprehending the included papers. We assume that the reader has some knowledge in calculus, linear algebra, and probability theory, but we intend to keep it on a basic level. First, we give the notation that will be used throughout the thesis. Then, we will introduce a selection of related works to place the thesis into context.

2.1 Notation and Terminology

We will begin by providing some algebraic notation that will be used for representing various types of data in the thesis. Scalars (both integer and real) are denoted by italic letters such as a. Vectors are denoted by lowercase bold italic letters such as $\boldsymbol{x}$, where all vectors are assumed to be column vectors. A superscript T denotes the transpose of a vector or matrix, such that $\boldsymbol{x}^T$ becomes a row vector. Matrices are denoted as uppercase bold italic letters such as $\boldsymbol{W}$. The notation $(w_1, \ldots, w_m)$ denotes a row vector with m elements, where the corresponding column vector is denoted as $(w_1, \ldots, w_m)^T$.

A dataset is denoted by the set $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$, where $\boldsymbol{x}^{(i)}$ is the i-th example among the N data points. Each data point is assumed to belong in a space of vectors denoted by $\mathcal{X}$, such that $\boldsymbol{x} \in \mathcal{X}$. The data generating distribution is denoted by $p_{data}(\mathcal{X})$ which is usually unknown. To provide an example, we let the vector $\boldsymbol{x} = (x_1, \ldots, x_m)$ represent a flattened image of m pixels. In this case, all possible images that can exist belong to the space $\mathcal{X}$ and the data generating distribution $p_{data}(\mathcal{X})$ gives the probability of how likely each image is to occur in the world. In supervised learning, there is also a target, either denoted as $y^{(i)}$ or $\boldsymbol{y}^{(i)}$, associated with $\boldsymbol{x}^{(i)}$. The target belongs to the target space $\mathcal{Y}$, where the space is discrete $\mathcal{Y} = \{1, \ldots, C\}$ for classification tasks over C number of classes, or continuous $\mathcal{Y} = (-\infty, \infty)$ over an interval of real values for regression tasks.

Throughout this thesis, we take a machine learning approach to solve tasks by

tuning an adaptive model using a dataset called the *training set*. In our case, we will represent the model with a function $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ that allows us to predict outcomes of events/data $\boldsymbol{x}$ from the task of interest. The parameters $\boldsymbol{\theta}$ expresses the function and we use machine learning algorithms for tuning parameters with the given dataset during the training phase. Once the model is trained, we often enter the *test phase* where we want to evaluate the model by predicting outcomes on an unseen dataset called the *test set*. The ability to predict outcomes of new data that is different from the examples seen during training is called *generalization*, which is a central goal for most applications in machine learning and pattern recognition.

2.2   Problem Settings in Machine Learning

Machine learning problems can be divided into three main fields, namely, *supervised learning*, *unsupervised learning*, and *reinforcement learning* (RL). Since this thesis includes work from each of these problem settings, we will briefly introduce these topics to provide the reader with context on the tasks that we are trying to solve.

MK: TO-DO: Add references to papers and sections!

Supervised Learning.   In this setting, we are given a dataset $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{N}$ where each data example $\boldsymbol{x}^{(i)}$ is accompanied with a target $\boldsymbol{y}^{(i)}$. The goal is to estimate a function $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ that assigns the correct target to each example in the training set as accurately as possible. In classification problems, each target belongs to one of K discrete categories, such that $\boldsymbol{y} = \{1, \ldots, K\}$, and we want to predict which of the categories that new data belongs to. Classification tasks will be involved in all included papers of this thesis wherein Paper A and B we focus on assigning the correct product category to images of grocery items. Another problem type in supervised learning is regression where the targets are continuous and real-valued. An example of a regression task is to predict the outdoors temperature tomorrow given the observed temperature today.

Unsupervised Learning.   Here, we are given a dataset $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ without access to any corresponding targets. The goal in this unsupervised setting may then be to find hidden structures in the given dataset. For example, we might be interested in discovering groups of similar examples with *clustering* techniques, or we may want to use *density estimation* where we approximate the true data distribution p_{data} with a parametric distribution $p_{\boldsymbol{\theta}}$ using the collected dataset, or we may want to project high-dimensional data into two or three dimensions for *visualization* purposes. We will get back to these goals when we introduce representation learning in Section X.

Reinforcement Learning. For these problems, we have a *learning agent* that wants reach a goal in an environment by performing a given set of actions. After performing an action, the agent observes the state of the environment and receives a reward from the environment saying how good or bad the taken action was to reach the goal. The objective for the agent is to maximize the reward signal within the time the agent reaches the goal. The agent then has to learn a policy for deciding which actions to perform in certain situations in the environment. The policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})$ is a mapping from perceived states $\boldsymbol{s}$ in the environment to actions $\boldsymbol{a}$ that should maximize the reward signal. An example of a task that can be framed as a RL problem is the so called Mountain Car problem, where the agent is a car that is trying to drive up to the top of a hill. The state represents the position and velocity of the car and the agent must take actions that will move the car forward or backwards. The objective is to reach the goal with as little time as possible, and the agent is encouraged to do so by the environment by sending the agent a negative reward for every time step that passes without reaching the top of the hill. We will return to the RL framework later when we describe the prerequisites for Paper D.

There exist many different methods for solving problems within these three fields. In this thesis, we employ deep learning methods which has been successfully applied in each field by representing the models with deep neural networks [10, 54, 55].

2.3 Deep Learning

In this section, we give a brief overview of deep learning [56] which is the main building block for the models we use in this thesis. Deep learning contains a family of machine learning models based on neural networks that are parametric function approximators used for representing some function of interest. Much of the successes of deep learning methods have been in supervised learning settings, especially in applications where there are large amounts of labelled data and sufficiently large model in terms of number of parameters in the network. In the following sections, we will introduce the deep learning frameworks and models that we have used in this thesis.

Deep Neural Networks

Neural networks is a class of machine learning models which popularity have grown immensely due to their ability to learn from large and high-dimensional datasets. Moreover, neural networks have been successfully applied in various number of fields in computer vision [10, 34], natural language processing [57], and reinforcement learning [55, 58]. These models are constructed by stacking layers of parameters that extract intermediate representations of the input data. The last layer outputs the target answer from the queried input and is specific for

the task. For example, in image classification, the last layer outputs class scores representing which class the image is most likely to belong to.

Next, we will describe three popular types of neural networks, namely, multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). **MK: TO-DO: It would be nice to have some simple illustrations of how the networks process the data, I can take inspiration from how other people have done it.**

Multilayer Perceptrons.   The simplest form of feedforward neural networks is the MLP. Let the vector $\boldsymbol{x} = (x_1, \ldots, x_d)^T$ be some form of data where x_i is the i-th feature for $i = 1, \ldots, d$. Every layer in the MLP constitutes of weights $\boldsymbol{W}$ that are used for transforming the input such that the output reveals some hidden structure useful for solving the task of interest. The transformation is performed with a matrix multiplication, i.e., $\boldsymbol{h} = \boldsymbol{W}\boldsymbol{x}$, to receive the interme-diate representation $\boldsymbol{h}$. An essential part for enabling neural networks to learn non-linear functions is to add an activation function right after the matrix mul-tiplication of each layer. Otherwise, the neural network would only be capable of learning linear functions since the matrix multiplication is a linear mapping. A common activation function is the $a(\boldsymbol{x}) = \max(0, \boldsymbol{x})$, or the so called Rectified Linear Unit (ReLU) activation, which outputs $\boldsymbol{x}$ when $\boldsymbol{x} > 0$ or otherwise zero. By stacking two layers together in a neural network with a ReLU activation, we then obtain the representation $\boldsymbol{h} = \boldsymbol{W}_2 \max(0, \boldsymbol{W}_1\boldsymbol{x})$. Note here that this rep-resentation could be predicted class scores, such that $\boldsymbol{h} = \hat{\boldsymbol{y}}$, if we would use two-layer MLP for a classication task.

Convolutional Neural Networks.   For data where the spatial order of each feature can be salient for prediction tasks such as image classification, we need a network that can capture relationships between features. CNNs are special kinds of neural networks that can process data with grid-like structures. Convolutional layers constitutes of a set of filters with adaptable weight parameters. To produce the output, we slide each filter across the input across the width and height of the input and compute dot products between the filter weights and the input at any position. Each filter will then produce a 2-dimensional feature map that gives the responses of that filter at every spatial position. The 2-dimensional feature maps from all filters are then stacked depth-wise to obtain the output volume. The obtained feature maps are often downsampled along their spatial dimensions using a pooling operation after the activation function. The parameter sharing in convolutional layers where each weight in a filter is applied to every position of the input comes from the idea that if some visual features are important in one part of the image, it should intuitively be useful at some other location as well. Furthermore, this design choice also makes the model require fewer parameters and a lower number of operations to compute the outputs.

Recurrent Neural Networks. RNNs are a family of neural network models specialized for processing sequences of data. Similar to CNNs, these models use parameter sharing by applying the same weights across several time steps. The parameter sharing is important in RNNs as it enables the model to handle different sequence lengths as well as being capable of recognizing relevant information that can appear at different locations in the sequence. Many RNNs follow the same procedure through the equation $\boldsymbol{h}^{(t)} = f_{\boldsymbol{\theta}}(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)})$, where the RNN produces the current state $\boldsymbol{h}^{(t)}$ by incorporating the input data $\boldsymbol{x}^{(t)}$ at time t into the previous hidden state $\boldsymbol{h}^{(t-1)}$. Hence, the hidden state $\boldsymbol{h}^{(t)}$ will now contain information about the whole past sequence up to time t. In most applications, there will be an extra output layer that reads the information from state $\boldsymbol{h}^{(t)}$ to make predictions. An example application for RNNs is predicting the next word in a sentence given previous words, where the RNN should store the necessary information about previous words to predict the rest of the sentence. A common choice of RNN model is the Long Short-Term Memory [59] (LSTM) which mitigates problems with vanishing gradients during the training phase.

For training deep networks, *loss* functions are used for measuring how well the network performs to solve the task of interest. In classification tasks, the cross-entropy loss is commonly used where the predicted class scores $\hat{\boldsymbol{y}}$ are compared against the true target class $\boldsymbol{y}$,

$$\mathcal{L}_{CE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{k=1}^{K} \boldsymbol{y}[k] \log \hat{\boldsymbol{y}}[k], \qquad (2.1)$$

where the true target vector $\boldsymbol{y}$ uses a one-hot representation where the true class i is denoted in the vector by setting the i-th element in $\boldsymbol{y}$ to one, as in $\boldsymbol{y}[i] = 1$, and zero elsewhere.

Probably the most common optimization algorithm for deep learning is stochastic gradient descent (SGD). The model parameters are updated by first computing the gradient of the loss function with respect to the weights of the network, as in $\nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})$, for a single input-output pair with backpropagation [60]. We can then update the parameters with the equation

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}), \qquad (2.2)$$

where η is the learning rate which is an important parameter for SGD determining how much the weights should be updated with the computed gradient.

Autoencoders

A common architecture type for deep learning in unsupervised learning are autoencoders for learning hidden representations of unlabeled data. Autoencoders are commonly used for dimensionality reduction of high-dimensional data, where the lower-dimensional representation can be used for classification tasks, or to

visualize hidden structures in the data that are hard to reveal from the original input data. The objective of the model is to reconstruct the original input data. The network architecture is built using two networks called *encoder* and *decoder* with a bottleneck layer between the networks for extracting the hidden representation $\boldsymbol{h}$. The encoder and decoder architectures can be of any neural network type, such as MLPs, CNNs, or RNNs, that fits the given dataset. The encoder is used for obtaining the hidden representation of the input data, while the decoder tries to reconstruct the original input from the obtained representation. Therefore, the idea is that the learned representation should contain the relevant information for reconstructing the data.

Mathematically, we denote the decoder as $f_{\boldsymbol{\theta}}$ and the encoder as $g_{\boldsymbol{\phi}}$. The encoder extracts the hidden representation $\boldsymbol{h} = g_{\boldsymbol{\phi}}(\boldsymbol{x})$ from the input $\boldsymbol{x}$, then the decoder produces a reconstruction $\hat{\boldsymbol{x}} = f_{\boldsymbol{\theta}}(\boldsymbol{h})$ from $\boldsymbol{h}$. We train the encoder and decoder simultaneously by minimizing a reconstruction loss $\mathcal{L}(f_{\boldsymbol{\theta}}(g_{\boldsymbol{\phi}}(\boldsymbol{x})), \boldsymbol{x})$, for instance mean-squared error loss, using SGD similarly as for the feedforward networks described above. There exist various kinds of methods for improving the quality of the learned representations in autoencoders. For example, we can adjust target task by adding noise to the inputs and let the decoder reconstruct the original input from noise variants, or we can induce different constraints in the bottleneck layer to, for example, obtain a sparse lower-dimensional representation of the data. Next, we will introduce the variational autoencoder which originates from latent variable models.

Variational Autoencoders

The variational autoencoder [61] (VAE) is a variant of autoencoders where learning is viewed from the perspective of probabilistic modeling. These models come from the family of deep generative models, where the goal is to approximate some underlying data distribution p_{data} with a parametric distribution $p_{\boldsymbol{\theta}}$ learned from a dataset $\mathcal{D} \sim p_{data}$. A common approach for estimating $p_{\boldsymbol{\theta}}$ is to use a latent variable model that infers hidden structures in the data to facilitate learning the distribution. VAEs is a deep latent variable model that uses neural networks for learning $p_{\boldsymbol{\theta}}$ making the training scalable to large high-dimensional datasets.

The main idea with introducing latent variables is that they should encode some semantically meaningful information about the observed data. Latent variable models are usually expressed by the joint distribution

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}), \tag{2.3}$$

where $\boldsymbol{z}$ denotes the latent variables and $\boldsymbol{x}$ the observed variables that represents the observed data points. The distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the likelihood of the data and $p(\boldsymbol{z})$ is the prior distribution for the latents. This model describes the generative process of the data $\boldsymbol{x}$ by following the steps 1) sample the latent vector $\boldsymbol{z} \sim p(\boldsymbol{z})$ from the prior, and 2) generate data point $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{z})$ from the sampled latent $\boldsymbol{z}$. We are now interested in learning the model $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ that best fits a

given dataset $\mathcal{D}$, as well as inferring the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ over the latent variables $\boldsymbol{z}$ given the data $\boldsymbol{x}$.

The overall goal with latent variable models is to maximize the marginal log-likelihood $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ given a dataset $\mathcal{D} \sim p_{data}$. However, computing $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ by with marginalizing out the $\boldsymbol{z}$ from the model $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \, d\boldsymbol{z}$ is in general intractable due to the many settings of $\boldsymbol{z}$ we would need to evaluate. Consequently, the posterior distribution also becomes intractable since $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})/p_{\boldsymbol{\theta}}(\boldsymbol{x})$ from Bayes' rule. Variational inference [62, 63] is a technique for enabling learning of latent variable models. The idea of variational inference is to provide means for calculating the marginal log-likelihood $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ by selecting a parameterized distribution $q_{\boldsymbol{\phi}}$ for approximating the true posterior distribution. In VAEs, the approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is represented as a neural network with parameters $\boldsymbol{\phi}$ that outputs the latents $\boldsymbol{z}$ given data points $\boldsymbol{x}$. With this approach, we can now form a lower bound on the marginal log-likelihood given by

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \mathbb{E}_{z \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - D_{\mathrm{KL}}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p(\boldsymbol{z})]. \qquad (2.4)$$

The right-hand side is called the evidence lower bound (ELBO) and comprises of two quantities that we can evaluate to train the model. The expectation over the log-likelihood $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ can be estimated with Monte Carlo sampling. The KL divergence between $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ can be computed analytically depending on how we choose these distributions. The standard choice for the prior is to use a zero-mean unit-variance Gaussian distribution $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$ where $\mathbf{I}$ is the identity matrix. The approximate posterior is also selected to be a Gaussian distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})))$, where the encoder network parameterized by $\boldsymbol{\phi}$ outputs the the means $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x})$ and standard deviations $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})$ for the latent dimensions. The latent vector is sampled using the "reparametrization trick" [61, 64] by computing $\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}) + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ denotes element-wise multiplication, which enables backpropagating gradients through the sampling operation. The likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the decoder network that tries to reconstruct the original input to the encoder. The likelihood distribution depends on the type of data $\boldsymbol{x}$ we wish to generate. If $\boldsymbol{x}$ is a continuous variable, then we can let the decoder output Gaussian parameters for the likelihood similar as for the encoder.

Multimodal Learning using Autoencoders

Deep Reinforcement Learning

Chapter 3

Fine-grained Recognition

This chapter presents an approach for enhancing fine-grained classification performance of grocery items by using web-scraped information. We focus on classification of grocery items due to applicability in assistive vision and its potential to enhance the independence of visually impaired (VI) people [ADD groceries/shopping/object recognition for VI REFs]. Initially, we were interested in learning classifiers with natural images taken in the grocery stores combined with web-scraped information about the grocery items, such as iconic images and text descriptions from supermarket websites. Using iconic images have been used in grocery image classification earlier [ADD grocery paper REFs], however, utilizing text descriptions was as far we know absent for this application even if it has been successfully applied in other image classification problems [65–67]. Thus, we collected our own dataset of grocery items images using a mobile phone camera as well as web-scraped images and text descriptions to study whether this multi-view approaches would benefit training the classifiers (Section 2). We then select a multi-view learning framework based on the Variational Autoencoder (VAE) for investigating how the different data views affect the fine-grained classification performance (Section 3).

3.1 Related Work

In this section, we will briefly discuss the related work on fine-grained image recognition [41], particularly when learning from external information, and multi-view learning.

Fine-grained Image Recognition

The goal with fine-grained image recognition (FGIR) is to distinguish between images with multiple visually similar sub-categories that belong to a super-category. For example, various attempts have been made to discriminate between sub-

categories of different animals [68], cars [69], fruits [70], retail products [71], etc. The challenge is to recognize differences that are sufficient for discriminating between objects that are generally similar but differ in fine-grained visual details. In recent years, the successes with deep learning in computer vision have encouraged researchers to explore various approaches for FGIR that can broadly be divided into three directions for recognition by utilizing (i) localization-classification sub-networks, (ii) end-to-end feature encoding, and (iii) external information. In (i), the goal is to find object parts that are shared across the sub-categories for discovering details that make the part representations different. This can be achieved by utilizing feature maps from the activations of convolutional layers as local descriptors [72–74], employing detection and segmentation techniques for localizing object parts[REFs], or leveraging attention mechanisms when common object parts are difficult to represent or even define [REFs]. With (ii), the goal has been to learn features that are better at capturing subtle and local differences by, for instance, performing high-order features interactions [75] as well as designing novel loss functions [Add REFs]. In the third approach (iii), the goal is to leverage external information, for example, web data and multimodal data, in FGIR as additional supervision to the images. We will put more focus on the approach on FGIR with external information next, as we use this approach in Paper A and B.

Recognition with External Information

Learning fine-grained details about objects often requires large amounts of labeled data. To ease the need for large amounts of accurately labeled images, there have been several attempts to let either web-scraped or multimodal data influence learning the fine-grained features of the sub-categories to boost the FGIR performance. Web-scraped images may be noisy in the sense that retrieved images may have high-variations of the objects. For example, the objects of interest can look different in appearance, and there could also be other irrelevant objects in the images that potentially occlude the category to recognize. Hence, incorporating web-scraped data into the training set may establish a domain gap between the easily acquired web data and the original training set which we need to overcome by reducing the domain gap or reducing the negative effects of the noisy web data that can disturb the learning. Another direction than using web-scraped data is to utilize multimodal data, for example, images, text and knowledge bases, for boosting the classification performance. In FGIR, the goal is to establish a joint representation between the images and additional data sources, where the additional data should act like extra guidance for learning useful representations that capture the fine-grained details of objects. Text descriptions have been a popular data type to combine with images, which can be both easy and cheap to collect as they can be accurately generated by non-experts. High-level knowledge graphs of objects have also been used and can contain rich knowledge useful for fine-grained recognition. In addition to FGIR, both web-scraped and multimodal

external information has been used for zero-shot learning to transfer knowledge from annotated categories to new fine-grained categories. In Paper A, we collect web-scraped images and text descriptions of grocery items to accompany real-world images of groceries for FGIR. Then, in Paper B, we perform a study using multi-view learning to investigate how the external information can enhance the classification performance. Next, we will cover the related work for the multi-view learning approach that we used.

Multi-view Learning

Learning from several data sources and modalities

3.2 Dataset Collection

In this section, we describe our procedure for collecting the image dataset of grocery items. As the target use case is grocery shopping with an assistive vision device, we visited several supermarkets and collected natural images of the groceries with a mobile phone camera to imitate such scenarios. Hence, the collected images will capture situations that can be challenging for the assistive device, such as, various lighting conditions, multiple instances and classes present, hand occlusions, and misplaced items. All images were taken with a single targeted item in mind, such that each image is paired with a single label. For items which belong to a clear super-class, for example, various kinds of apples and milk packages, we also provided the general class of the items to establish a hierarchical labeling structure of the data. Collecting natural images of the grocery items is unfortunately a time-consuming process. Furthermore, as the surroundings in every grocery store varies, it may be difficult to build accurate classifiers that can recognize fine-grained details solely from natural images. Hence, we need some cheaper procedure that can complement the collection of real-world images for boosting the classification performance of the groceries.

We have complemented the image dataset with external information from the web of each grocery item that can be used for training classifiers. In the past years, most supermarket chains have the option for consumers to purchase groceries online from their websites. The website usually provides each grocery item with an iconic image of the item on a white background, a text description that describes the flavor and ingredients of the item, as well as nutrition values if applicable. We downloaded these information types of all grocery item classes by web-scraping the online shopping website of a supermarket chain. We show four examples of grocery items and their web-scraped information in Table 3.1. Since these data types are on a class-based level, we can use the web-scraped information as weak supervision to guide the classifier to learn fine-grained details that helps discriminating between visually similar items.

Table 3.1: Examples of grocery item classes in the Grocery Store dataset. We display four different items (coarse-grained class in parenthesis), followed by two natural images taken with a mobile phone inside grocery stores. Next comes the web-scraped information of the items consisting of an iconic image and a text description. We have highlighted ingredients and flavors in the text description that are characteristic for the specific item.

Class Labels	Natural Images	Iconic Images	Text Descriptions
Granny Smith (Apple)			*"...**green** apple with **white, firm** pulp and a **clear acidity** in the flavor."*
Royal Gala (Apple)			*"...**crispy** and **very juicy** apple, with **yellow-white** pulp. The peel is **thin** with a **red yellow** speckled color."*
Tropicana Mandarin (Juice)			*"...is a **ready to drink** juice **without pulp** pressed on **orange**, **mandarin** and **grapes**. Not from concentrate. Mildly **pasteurized**."*
Yoggi Vanilla (Yoghurt)			*"...**creamy vanilla** yoghurt original... added **sugar** than regular flavored yoghurt. Great for both **breakfast and snacks**."*

3.3 Multi-view Representation Learning of Grocery Items

This section describes the approach we took for learning representations of grocery items that are shared across the available data types. We employ a deep latent variable model called Variational Canonical Correlation Analysis [53] (VCCA) for learning the shared representation. The main assumption in VCCA is that each data view have been generated from the same latent space. The goal then is to learn this latent space that captures the correspondences between all views into representations shared across the views for the grocery items. This representation can then be utilized for enhance the learning more accurate classifiers as well as for performing tasks such as synthesis and prediction of novel images. Next, we describe how to enable learning the shared latent space.

Capturing variations from each view in the learned representation is performed by predicting the original views from the latent space. To obtain the latent representation, we extract the representation by encoding the natural images with neural network. The extracted representation is then used for predicting each view individually by inputting the representation through separate neural networks. Note that we only use the natural images for extracting the latent

representation here since it is the only view that is available at test time when we want to use the learned classifier in the grocery store. We have two options for exploiting the new representation to train classifiers. The first option is to train the classifier with the latent representations after we have learned the latent space as described above. The second option is to train the classifier and learning the latent space simultaneously by adding an additional classifier network predicting the class label with the latent representation as input.

MK: TO-DO: Need to introduce the ELBO. I should also add a figure of the architecture for extra clarity on the method.

3.4   Experiments

3.5   Discussion

Chapter 4

Continual Learning

This chapter introduces the idea of replay scheduling for mitigating catastrophic forgetting in continual learning (CL). The problem setting of CL is on learning tasks of recognizing a new set of classes with a dataset given at the current time step. In the standard setting, one main assumption is that the data from past tasks can never be fully revisited by the model. However, in the real-world, many organizations record data from incoming streams for storage rather than deleting it [76, 77] [Add at least 1 more REF]. In contrast to the assumption on data storage in standard CL, we suggest a new setting where we assume that all seen data is accessible at any time for the model to revisit. The challenge then becomes how to select which tasks that needs to be remembered via replay as the data is still incoming from a stream. We propose to learn the time when replaying a certain task is necessary when the model is updating its knowledge with new incoming tasks. In Paper C, we propose the new CL setting where historical data is accessible and introduce the idea of replay scheduling and how it can be used in CL. In Paper D, we propose a framework based on reinforcement learning [78] (RL) for learning replay scheduling policies that can be applied in new CL scenarios.

4.1   Related Work

4.2   Replay Scheduling in Continual Learning

4.3   Meta-Policy Learning for Replay Scheduling

4.4   Experiments

4.5   Discussion

Chapter 5

Conclusions and Future Directions

5.1   Conclusions

5.2   Future Directions

- Video data for object recognition instead of images for making systems easier to use. And use a disability-first approach when collecting the data

- Federated Learning for decentralizing model updates

- Uncertainty Quantification - How to make the classifiers trustworthy?

Disability-first Approaches

Federated Learning

Bibliography

[1] Alexander Eitel, Katharina Scheiter, Anne Schüler, Marcus Nyström, and Kenneth Holmqvist. How a picture facilitates the process of learning from text: Evidence for scaffolding. *Learning and Instruction*, 28:48–63, 2013.

[2] Anne Nielsen Hibbing and Joan L. Rankin-Erickson. A picture is worth a thousand words: Using visual images to improve comprehension for middle school struggling readers. *The Reading Teacher*, 56, 2003.

[3] Rupert Bourne, Jaimie D Steinmetz, Seth Flaxman, Paul Svitil Briant, Hugh R Taylor, Serge Resnikoff, Robert James Casson, Amir Abdoli, Eman Abu-Gharbieh, Ashkan Afshin, et al. Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the global burden of disease study. *The Lancet global health*, 9(2):e130–e143, 2021.

[4] Dragan Ahmetovic, Daisuke Sato, Uran Oh, Tatsuya Ishihara, Kris Kitani, and Chieko Asakawa. Recog: Supporting blind people in recognizing personal objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[5] Rabia Jafri, Syed Abid Ali, Hamid R Arabnia, and Shameem Fatima. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *The Visual Computer*, 30(11):1197–1222, 2014.

[6] Hernisa Kacorri. Teachable machines for accessibility. *ACM SIGACCESS Accessibility and Computing*, (119):10–18, 2017.

[7] James Coughlan and Roberto Manduchi. Functional assessment of a camera phone-based wayfinding system operated by blind and visually impaired users. *International Journal on Artificial Intelligence Tools*, 18(03):379–397, 2009.

[8] Hernisa Kacorri, Eshed Ohn-Bar, Kris M Kitani, and Chieko Asakawa. Environmental factors in indoor navigation based on real-world trajectories of blind users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[9] Jack M Loomis, Reginald G Golledge, Roberta L Klatzky, and James R Marston. Assisting wayfinding in visually impaired travelers. In *Applied Spatial Cognition*, pages 179–202. Psychology Press, 2020.

[10] 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.

[11] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[12] 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. PMLR, 2015.

[13] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[14] World Health Organization. International Classification of Diseases 11th Revision (ICD-11), 2022. URL https://icd.who.int/en. Accessed 2022-03-22.

[15] World Health Organization. *World report on vision*. World Health Organization, 2019.

[16] Jaimie D Steinmetz, Rupert RA Bourne, Paul Svitil Briant, Seth R Flaxman, Hugh RB Taylor, Jost B Jonas, Amir Aberhe Abdoli, Woldu Aberhe Abrha, Ahmed Abualhasan, Eman Girum Abu-Gharbieh, et al. Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study. *The Lancet Global Health*, 9(2):e144–e160, 2021.

[17] Roberto Manduchi and James Coughlan. (computer) vision without sight. *Communications of the ACM*, 55(1):96–104, 2012.

[18] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P Bigham. Supporting blind photography. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 203–210, 2011.

[19] Erin Brady, Meredith Ringel Morris, Yu Zhong, Samuel White, and Jeffrey P Bigham. Visual challenges in the everyday lives of blind people. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2117–2126, 2013.

[20] Microsoft Corporation. Seeing AI, 2017. URL https://www.microsoft.com/en-us/ai/seeing-ai. Accessed 2022-03-14.

[21] Patrick Clary. Lookout: an app to help blind and visually impaired people learn about their surroundings, 2018. URL https://blog.google/outreach-initiatives/accessibility/lookout-app-help-blind-and-visually-impaired-people-learn-about-their-surro. Accessed 2022-04-05.

[22] Inc Cloudsight. TapTapSee, 2013. URL https://taptapseeapp.com/. Accessed 2022-03-22.

[23] Envision. Envision App, 2018. URL https://www.letsenvision.com/envision-app. Accessed 2022-03-22.

[24] OrCam. OrCam MyEye 2, 2019. URL https://www.orcam.com/sv/myeye2/. Accessed 2022-03-22.

[25] Envision. Envision Glasses, 2020. URL https://www.letsenvision.com/envision-glasses. Accessed 2022-03-22.

[26] Be My Eyes. Be My Eyes, 2017. URL https://www.bemyeyes.com/. Accessed 2022-03-22.

[27] Aira Tech Corp. Aira, 2017. URL https://aira.io/. Accessed 2022-03-14.

[28] Tai-Yin Chiu, Yinan Zhao, and Danna Gurari. Assessing image quality issues for real-world problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3646–3656, 2020.

[29] Hernisa Kacorri, Kris M Kitani, Jeffrey P Bigham, and Chieko Asakawa. People with visual impairment training personal object recognizers: Feasibility and challenges. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5839–5849, 2017.

[30] Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. *arXiv preprint arXiv:1912.01100*, 2019.

[31] Tousif Ahmed, Roberto Hoyle, Kay Connelly, David Crandall, and Apu Kapadia. Privacy concerns and behaviors of people with visual impairments. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3523–3532, 2015.

[32] Danna Gurari, Qing Li, Chi Lin, Yinan Zhao, Anhong Guo, Abigale Stangl, and Jeffrey P Bigham. Vizwiz-priv: A dataset for recognizing the presence and purpose of private visual information in images taken by blind people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 939–948, 2019.

[33] Roberto Hoyle, Robert Templeman, Steven Armes, Denise Anthony, David Crandall, and Apu Kapadia. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 571–582, 2014.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[35] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[36] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[37] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[38] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[39] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[40] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10294–10303, 2019.

[41] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[42] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[43] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[44] Patrick E Lanigan, Aaron M Paulos, Andrew W Williams, Dan Rossi, and Priya Narasimhan. Trinetra: Assistive technologies for grocery shopping for the blind. In *2006 10th IEEE International Symposium on Wearable Computers*, pages 147–148. IEEE, 2006.

[45] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[46] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[47] Tyler L Hayes, Giri P Krishnan, Maxim Bazhenov, Hava T Siegelmann, Terrence J Sejnowski, and Christopher Kanan. Replay in deep learning: Current approaches and missing biological elements. *Neural Computation*, 33(11):2908–2950, 2021.

[48] Frank N Dempster. Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330, 1989.

[49] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

[50] Karri S Hawley, Katie E Cherry, Emily O Boudreaux, and Erin M Jackson. A comparison of adjusted spaced retrieval versus a uniform expanded retrieval schedule for learning a name–face association in older adults with probable alzheimer's disease. *Journal of Clinical and Experimental Neuropsychology*, 30(6):639–649, 2008.

[51] T. Landauer and Robert Bjork. Optimum rehearsal patterns and name learning. *Practical aspects of memory*, 1, 11 1977.

[52] Paul Smolen, Yili Zhang, and John H Byrne. The right time to learn: mechanisms and optimization of spaced learning. *Nature Reviews Neuroscience*, 17(2):77, 2016.

[53] Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2016.

[54] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[55] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[56] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[57] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[58] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[59] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[60] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[61] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[62] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

[63] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[64] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[65] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011.

[66] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[67] Sebastian Bujwid and Josephine Sullivan. Large-scale zero-shot image classification from rich and diverse textual descriptions. *arXiv preprint arXiv:2103.09669*, 2021.

[68] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[69] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[70] Yushan Feng Saihui Hou and Zilei Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *IEEE International Conference on Computer Vision*, 2017.

[71] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. Rpc: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*, 2019.

[72] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1134–1142, 2016.

[73] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4148–4157, 2018.

[74] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6599–6608, 2019.

[75] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[76] Peter Bailis, Edward Gan, Samuel Madden, Deepak Narayanan, Kexin Rong, and Sahaana Suri. Macrobase: Prioritizing attention in fast data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 541–556, 2017.

[77] Tom M. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42:30–36, 1999.

[78] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Part II

Included Papers

Paper A

Name for Paper A

Marcus Klasson, Cheng Zhang, Hedvig Kjellström

Abstract

Abstract aby stract

A.1 Introduction

hej hej här är en artikel

what do you think this is?

hello hello city

Paper B

Name for Paper B

Marcus Klasson, Cheng Zhang, Hedvig Kjellström

Abstract

Abstract aby stract

B.1   Introduction

hej hej här är en artikel

what do you think this is?