CARDEA: A CONTEXT-AWARE AND INTERACTIVE VISUAL PRIVACY CONTROL FRAMEWORK

by

RUI ZHENG

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Computer Science and Engineering

October 2016, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

RUI ZHENG

CARDEA: A CONTEXT-AWARE AND INTERACTIVE VISUAL PRIVACY CONTROL FRAMEWORK

by

RUI ZHENG

This is to certify that I have examined the above M.Phil. thesis

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by

the thesis examination committee have been made.

ASSISTANT PROF. PAN. HUI, THESIS SUPERVISOR

PROF. QIANG YANG, HEAD OF DEPARTMENT

Department of Computer Science and Engineering

21 October 2016

ACKNOWLEDGMENTS

thank god

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CARDEA: A CONTEXT-AWARE AND INTERACTIVE VISUAL PRIVACY CONTROL FRAMEWORK

by

RUI ZHENG

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

The growing popularity of mobile and wearable devices with builtin cameras, the bright prospect of camera related applications such as augmented reality and lifelogging system, the increased ease of taking and sharing photos, along with advances in computer vision techniques, have greatly facilitated peoples lives in many aspects, but inevitably raised peoples concerns about visual privacy at the same time.

Motivated by the finding that peoples privacy concerns are influenced by the context, in this thesis, we propose Cardea, a contextaware and interactive visual privacy control framework that enforces privacy policies according to peoples privacy preferences. The framework provides people with finegrained visual privacy control using: *i)* personal privacy profiles, with which people can define their contextdependent privacy preferences; *ii)* different visual indicators: face features and tags, for devices to automatically locates individuals who request privacy protection; *iii)* hand gestures, for people to temporarily update and flexibly inform cameras of their privacy preferences.

Benefited from recent progresses in face and object recognition, Cardea offers a way for context-dependent privacy control in a natural and flexible manner, which differs from tag and marker based systems. We design and implement the framework consisting of Android client app and cloud control server, with convolutional neural networks as core of the image processing module. Our evaluation results confirm such framework is practical and effective, showing promising future for contextaware visual privacy control on mobile and wearable devices.

CHAPTER 1

INTRODUCTION

The concern about visual privacy has been growing in last decade with increasing adoption of video surveillance systems for security reasons. The statistics shows there are 125 video surveillance cameras per thousand people in U.S. by 2014 [1]. Momentum of new technologies such as the Internet of Things (IOT) will keep driving global video surveillance market in following years, which will raise more privacy concerns.

Other than closed-circuit television (CCTV) surveillance systems for security reasons, handheld devices such as camera phones are also used extensively for the recording of meaningful life moments. Recently, coming with the explosion of products in augmented reality (e.g., Google Glass), robotics (e.g., iRobot Create platform), and gaming (e.g., Kinect), is more and more cameras being embedded in these platforms for the enhancement of life experiences. The trend of embedding cameras, especially in wearables, will keep growing, an example of which is smart contact lens [2]. However, the ubiquitous presence of cameras, the ease of taking photos and recording videos, along with "always on" and "non overt act" features threaten individuals to have private or anonymous social lives, raising people's concerns of visual privacy.

More specifically, photos and videos captured without getting permissions from bystanders, and then uploaded to social networking sites, can be accessed by everyone online, potentially leading to invasion of privacy. Malicious applications on the device may also inadvertently leak captured media data [3].

Benefited from research breakthroughs from deep learning community [4], current vision perception systems are advancing fast in their capabilities of understanding image and video contents [5]. Nowadays, recognition technologies can link images to specific people [6, 7, 8], places [9], and general objects [10], making what previously unsearchable now searchable [11], thus reveal far more private information than expected.

Both legal and technical measures have been proposed to resolve visual privacy concerns. For instance, Google Glass is banned at places such as banks, hospitals, and bars [12]. However, prohibition of cameras usage does not resolve the issue fundamentally, instead it may intrude people's rights to capture happy moments. As a result, there are growing needs to design technical solutions to protect individuals' visual privacy in a world with pervasive cameras. Technical solutions that have been proposed so far are still limited, in the way that they are mostly based on static policies, thus users can not flexibly express their individualized privacy preferences based on surrounding contexts when they are captured. Moreover, previous works require users to wear visual markers such as hats [13] for the detection of interested persons, or clip tags such as QR codes [14, 15] for the fetching of privacy polices. Despite technical feasibilities of these approaches, the extra need of setting up markers/tags and the resulting aesthetically unpleasant appearance will hinder users' willingness to adopt these solutions.

Therefore, the motivation of this thesis is to seek a more natural, userfriendly, flexible, and fine-grained mechanism for people to express, modify, and control their individualized privacy preferences. Under this guideline, we propose Cardea, a context-aware and interactive visual privacy control framework, which lets individuals control their visual privacies through: *i)* personal privacy profiles, with which people can define their contextdependent privacy preferences; *ii)* different visual indicators: face features and tags, for devices to automatically locates individuals who request privacy protection; *iii)* hand gestures, for people to temporarily update and flexibly inform cameras of their privacy preferences. When using Cardea, the device will automatically compute context factors, compare them with peoples privacy profiles, and finally enforce privacy policies conforming to peoples privacy preferences. To our knowledge, this is one of the pioneering works that leverages deep learning models, more specifically convolutional neural networks (CNN) [16], to enable visual privacy control in a context-specific and interactive manner.

The rest of the thesis is organized as follows: We first review and discuss related works on visual privacy control in Chapter 2. Following that we introduce convolutional neural networks, the core of Cardea's image processing module, and their applications on related computer vision problems. We then give details about the design, implementation and evaluation of Cardea in Chapter 4. Finally, we share our thoughts on possible future work and conclude the thesis in Chapter 5.

CHAPTER 2

RELATED WORKS

2.1 User Studies

Glass-style AR devices and lifelogging devices come into people's lives in recent years, bringing more and more privacy concerns among public, from the perspectives of both recorders and bystanders. Some user studies investigate these concerns by conducting privacy surveys and *in situ* interviews [17, 18, 19, 20]. Here we summarize some findings from these works:

- All user studies find that most participants care about their appearance in recored contents, and welcome a consent mechanism so that they can have controls such as: Stop the recording when they don't want to be recorded; Obfuscation of their identities in recording-time and sharing-time.
- Study in [18] says that lifeloggers care about the privacy of bystanders, and they prefer automated *in situ* controls of privacy contents to relieve the burden of physical control.
- Privacy is vague, and depends on many factors, which we name it as contexts, defined by primitives such as *Who, What, When, Where, Why and How*. All studies confirm that there are many reasons that make contents private. A breakdown of potential design axes for privacy-mediating technologies is proposed in [17].

Inspired by the design axes proposed in [17], in Table 2.1 we reviewed related works by comparing some major factors such as their problem settings, technical solutions, etc.

Table 2.1: Mobile cameras include cameras in smartphones, AR, VR, lifelogging and other wearable devices. Computer vision methods may be assisted with extra sensors such as RFID tags [21], FIR imagers [22]. There are other factors like implementation layer (app level or os level) that are not listed due to space limitation.

year	privacy survey	problem setting		technical solution		enforcement time		privacy object	
		video surveillance	mobile cameras	computer vision	cryptography	in-situ / run time	access / distribution	user	bystander
I-Pic: A Platform for Privacy-Compliant Image Capture [20]									
2016	✓		✓	✓	✓	✓			✓
Recent work which allows people to broadcast their privacy preferences and appearance information to nearby devices through BLE. These preferences are based on social context.									
What You Mark is What Apps See [23]									
2016			✓	✓		✓	✓	✓	
Propose a system that give users control to mark secure regions for third-party applications. It is implemented within Android camera subsystem.									
Sensitive Lifelogs: A Privacy Analysis of Photos from Wearable Cameras [19]									
2015	✓		✓						
Analyze the photos collected in [18], seeking to understand what makes a photo private and what participants said about their images.									
Screenavoider: Protecting Computer Screens from Ubiquitous Cameras [24]									
2014			✓	✓		✓	✓	✓	
Present a framework that controls the collection and disclosure of lifelogging datasets which contain computer screens and possible sensitive contents.									
PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces [25]									
2014			✓	✓		✓	✓	✓	
Introduce a prototype for owners of first-person cameras to 'blacklist' sensitive places (like bathrooms and bedrooms).									
Privacy.Tag: Privacy Concern Expressed and Respected [14]									
2014			✓	✓	✓		✓		✓
Propose using QR code as privacy tag to link an individual with his photo sharing preferences. These preferences are based on web domains.									
Privacy Behaviors of Lifeloggers using Wearable Cameras [18]									
2014	✓		✓			✓	✓	✓	✓
Conducted an *in situ* user study on privacy behaviors of 36 participants who wore lifelogging devices for a week.									
Courteous Glass [22]									
2014			✓	✓		✓			✓
Wearable camera integrated with a FIR (far-infrared) imagers that turns off recording when new persons or specific gestures are detected.									
World-Driven Access Control for Continuous Sensing [15]									
2014			✓	✓		✓	✓	✓	
Propose a general framework that allows objects to explicitly specify its access policies. Policy triggers can be visual indicators or anything that can be detected in other research works.									

Table 2.1: Continued from previous page

year	privacy survey	problem setting		technical solution		enforcement time		privacy object	
		video surveillance	mobile cameras	computer vision	cryptography	in-situ / run time	access / distribution	user	bystander
In Situ with Bystanders of Augmented Reality Glasses: Perspectives on Recording and Privacy-Mediating Technologies [17]									
2014	✓		✓			✓	✓	✓	✓
Investigate the privacy perspectives of individuals when they are bystanders around AR devices. Conducted 12 field sessions in cafés and interviewed 31 bystanders regarding their reactions to a co-located AR device.									
A Scanner Darkly: Protecting User Privacy From Perceptual Applications [26]									
2013		✓	✓	✓		✓	✓	✓	
Perceptual applications can only access transformed objects such as sketches, faces, etc.									
Enabling Fine-Grained Permissions for Augmented Reality Applications With Recognizers [27]									
2013			✓	✓		✓		✓	
Third party AR applications can only access high-level objects such as Skeleton, Hand Position, etc.									
PriSurv: Privacy Protected Video Surveillance System Using Adaptive Visual Abstaction [21]									
2008		✓		✓			✓	✓	✓
Propose a privacy control mechanism for surveillance videos based on closeness between content objects and content viewers. RFID tags are used to improve the detection of people in videos.									
Respectful Cameras: Detecting Visual Markers in Real-Time to Address Privacy Concerns [13]									
2007		✓		✓			✓	✓	✓
A video surveillance system that allows people who wish to remain anonymous wear colored markers such as hats or vests, and their faces will be blurred.									
Privacy Management for Portable Recording Devices [28]									
2004			✓		✓		✓	✓	
Propose an approach that closed closed devices can encrypt data together during recording utilizing short range wireless communication to exchange public keys and negotiate encryption key. Only by obtaining all of permissions from people who encrypt the recording can one decrypts it.									

2.2 Solutions and Guidelines

As listed in Table 2.1, we conclude previous research works from such aspects:

- **Problem Setting:** Previous researches focused on privacy issues in CCTV, after the spread of smartphones, research focus has been shifted to potential user privacy leakage caused by installed third party applications. A recent trend is the adoption of wearable devices, which make it harder for people to notice that they are being captured. Thus bystander privacy in such settings is gaining more attention these days.

- **Technical Solution:** Encryption and decryption are mostly used when fetching privacy policies, while recognition technologies are mainly used for detection of people and perception of context. Computer vision techniques are also used in policy enforcement stage. Extra economical sensors can be integrated to ease the hardness of computer vision tasks. Wireless technologies are used in some works for the broadcasting of bystanders' privacy policies.

- **Enforcement Time:** *In-situ* control mechanisms pose high requirements of devices' computational power, however fast developments of hardwares and breakthroughs in computer vision encourage researches to try more *in-situ* solutions. Bystander privacy control ideally requires an *in-situ* solution, while user privacy concern usually surfaces at sharing time and is generically easier to be handled.

- **Privacy Object:** Just as other factors, user privacy and bystander privacy don't exclude each other. Essentially user privacy is just a special case of bystander privacy.

- **Other factors:** It is preferable for control mechanisms to be integrated in operation system or even hardware level. Most solutions are based on the assumption that users are trusted. Other design considerations include opt-in vs opt-out, policy push vs policy pull, etc. Each option has advantages and limitations, thus should be considered case by case.

Due to limitation in hardwares as well as algorithms, previous designs are mostly limited on specific settings and simple techniques, though many works mention that policies should be situational, individualized and dynamic. This motivates us to move further in the direction of providing a more general as well as practical control service. From the discussion of related works, we can see that privacy control system design is a complicate task. In current stage it is not possible to solve all problems in one shot, therefore Cardea's design is also limit in such ways:

- Though not limited to, it starts from and aims at protecting bystander privacy.

- Currently only computer vision methods are used, and it tries to provide an *in situ* solution, but relies on a stable connection with a center server. Part of computations are offloaded to center server is because mobile phones can not afford to load all deep learning models and run all computer vision algorithms.

- It is implemented in application layer, thus can not prevent users from using other applications to get raw camera data.
- We choose opt-out policies for aesthetic reason, trying to let photographers enjoy the real world image while still take care of privacy concerns for bystanders who mind their appearances in that image.

2.3 Possibilities and Challenges

The processing power of mobiles and computers are increasing extremely fast in recent years. Nvidia's Tegra Processors [29] are used in many auto-pilot systems. Nvidia's Tegra X1 already achieves 1 teraflops 16bit floating points performance [30]. Many high performance smartphones are now equipped with high end processors such as Qualcomm's Snapdragon 820s [31], which makes them able to run certain heavy computer vision algorithms that is impossible before. A first thought would be deploying state-of-the-art deep learning models on these platforms to extract more accurate context informations. However, context is an abstract concept and thus hard to be defined, thus Cardea is limited in how well context is represented in the design. The way users interact with a system is also important for acceptance of the system, how to ease the burden for bystanders to express their dynamic privacy preferences pose another challenge to our design. From implementation point of view, we have to take care of many things such as resource limitations, issues when scaling up, uncontrolled environments, etc. After an introduction of deep learning and convolutional neural networks in next chapter, we will give details of how we tackle these challenges in Chapter 4.

CHAPTER 3

CONVOLUTIONAL NEURAL NETWORKS

3.1 Deep Learning

Deep learning [32, 4] is part of a broader family of machine learning methods that focus on representation learning. Unlike rule based methods used in early days of artificial intelligence, deep learning targets at tasks that are easy for people to perform but hard for people to describe formally—problems that we solve intuitively. Former methods have proved success in problems that can be completely described by a very brief list of completely formal rules, thus easily provided ahead of time by the programmer. That led to the defeat of world chess champion Garry Kasparov by IBM's Deep Blue chess-playing system in 1997 [33]. However, rule based or knowledge based methods fail at intuitive tasks such as recognizing objects or speech. The reason behind this inability is because these tasks require immense amount of knowledge about the world, and much of this knowledge is subjective and intuitive, thus difficult to articulate in a formal way. The difficulties faced by AI systems relying on hard-coded knowledge suggest that AI systems need the ability to acquire their own knowledge, by extracting patterns from raw data, just like how human learns from experiences. This capability is known as machine learning. It does not take long for people to realize that performance of machine learning algorithms depends heavily on the representation of data fed into these algorithms. Under a good representation, factors of variation can be disentangled and non important factors will be discarded. People used to put lots of efforts on hand designing features to get a good representation for every domain specific problem. Unfortunately seeking a good representation for a problem can be as difficult as solving the problem itself. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts, thus achieves great power and flexibility by representing the world as a nested hierarchy of concepts. Fig 3.1 shows a hierarchy relation from AI to deep learning.

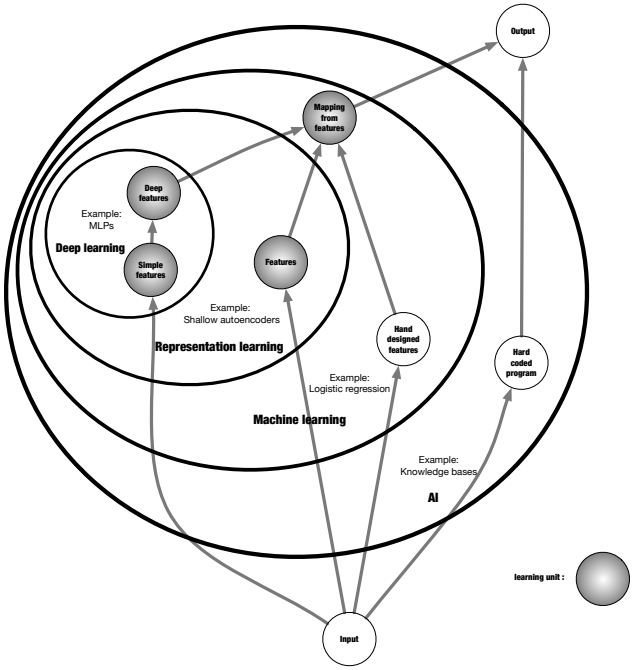

Figure 3.1: Venn diagram showing relations between deep learning, representation learning, machine learning, and AI. An example and work flow are also included for each section.

3.1.1 Three Waves

Deep learning has been rebranded many times under different names, only recently become well known as ”deep learning”. Broadly speaking, there have been three waves of development of deep learning: in 1940s-1960s known as *cybernetics* [34, 35], in 1980s-1990s known as *connectionism* [36], and the current resurgence starts from 2006 [37, 38, 39]. These three waves witness the evolving from simple perceptron, to distributed representation and stochastic gradient descent algorithm, and finally to today's various deep structures. The resurgence of deep learning benefits from the facts: computers are faster, datasets are bigger and a good initialization of model parameters. Faster computers make it possible to train deeper models, bigger datasets relieve deep models from overfitting, and enable them to learn more meaningful mid-level features that are more generalizable, and a good initialization through layer wise pretraining provides a proper prior and makes supervised training on later stages much easier.

Since the early stage of deep learning, neuroscience is regarded as an important source of inspiration. However, it is no longer a predominant guide for the field because we simply do not have

enough information about the brain to use it as a guide. It is seemed as a more general principle of learning multiple levels of composition, which can be applied in machine learning frameworks that are not necessarily neurally inspired.

3.1.2 Breakthroughs

We highlight some of the breakthroughs brought by deep learning in recent years:

- In the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012, a convolutional neuron network won this challenge for the first time and by a wide margin, bringing down the state-of-the-art top-5 error rate from 26.1% to 15.3% [40], since then the top-5 error rate keeps dropping down each year [41, 42] and in last year dropped to 3.6% using deep residual network (ResNet) [43]. Deep networks also had spectacular successes for pedestrian detection and image segmentation [44, 45, 46] and yield superhuman performance in traffic sign classification [47].

- In March, Google DeepMind's AlphaGo defeated world champion Lee Sedol using deep reinforcement learning [48, 49]. Prior to that, they also showed that a deep reinforcement system is capable of learning to play Atari video games and reaching human-level performance on many tasks [50].

- The application of deep learning on other fields such as speech recognition [51, 52, 53] and machine translation [54, 55] all lead to great successes. The past years surfaced many new trends such as generative adversarial networks, neural Turing machines etc [56, 57, 58, 59, 60]. The years ahead are full of challenges and opportunities to improve deep learning even further and bring it to new frontiers.

3.1.3 Artificial Neural Networks

A Single Neuron

Fig 3.2 shows how to coarsely model a biological neuron: Each neuron receives input signals from its dendrites and produces output signals along its (single) axon. The axon eventually branches out

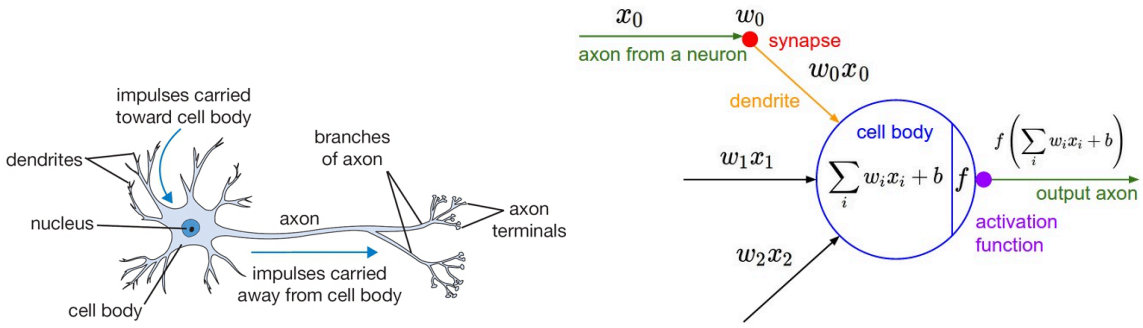

Figure 3.2: A cartoon drawing of a biological neuron (left) and its mathematical model (right). Photo taken from *Module1: Neural Networks* in [61]

and connects via synapses to dendrites of other neurons. In the basic model, the dendrites carry the signal to the cell body where they all get summed. If the final sum is above a certain threshold, the neuron can fire, sending a spike along its axon. In the computational model, we assume that the precise timings of the spikes do not matter, and that only the frequency of the firing communicates information. Based on this rate code interpretation, we model the firing rate of the neuron with an activation function f, which represents the frequency of the spikes along the axon. Historically, a common choice of activation function is the sigmoid function σ, since it takes a real-valued input (the signal strength after the sum) and squashes it to range between 0 and 1. Hereafter in this thesis, we will take standard naming convention and refer a hidden unit as a computational neuron.

Activations

Depends on the activation, neurons can have different properties and impacts on the whole network. Figure 3.3 shows most commonly used activation functions:

Sigmoid Sigmoid is used historically to simulate the firing rate of a neuron but recent years falls out of favor and is rarely used now. It saturates at either tail of 0 or 1, and at these regions the gradient is almost zero, thus it kills gradients from propagating to previous layers, making deep networks not trainable. Another drawback of sigmoid activation is that neurons in later layers will not receive zero-centered inputs, this will cause all positive or negative gradients on the weights during backpropagation, which may lead to undesirable zig-zagging dynamics in the gradient updates of the weights. However, once gradients are added up across a batch

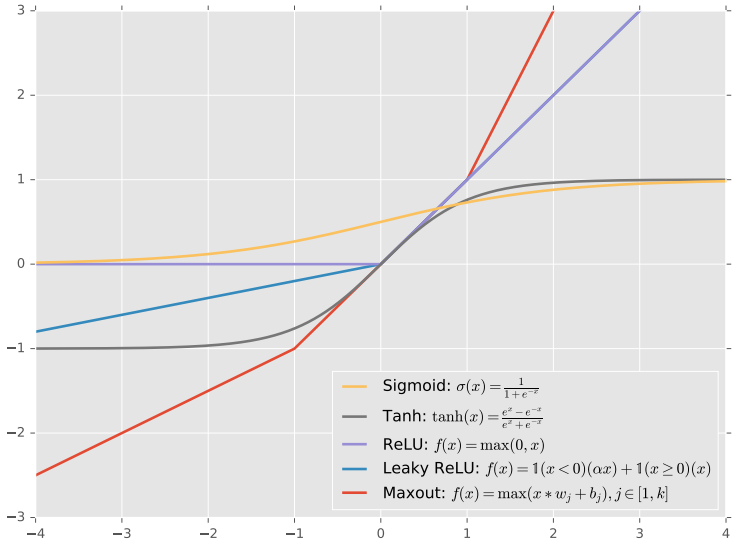

Figure 3.3: Commonly used activation functions.

of data, the final update for the weights can have variable signs, somewhat mitigating this issue.

Tanh Though similar to sigmoid non-linerality in that it also suffers from saturization and thus vanishing gradients problems, tanh is always preferred to sigmoid because its output is zero centered.

ReLU Rectified linear unit becomes popular in recent years. [40] used it in their winning 2012 ImageNet competetion and found that it greatly accelerated the convergence of stochastic gradient descent optimization process. Also comparing to other activations, it is a much more light weight operation since it is a simple thresholding operation. However, ReLU unit can ”die” if a large enough gradient changes the weights such that the neuron never activates on new data.

Leaky ReLU Similar to ReLU, but can help fix the ”dying ReLU” problem by modifying the flat side of ReLU to have a small gradient [62].

Maxout Introduced in [63], Maxout unit enjoys all the benefits of a ReLU unit, and does not have its drawbacks. Maxout activation can implement ReLU activtions and approximate any convex activation function.

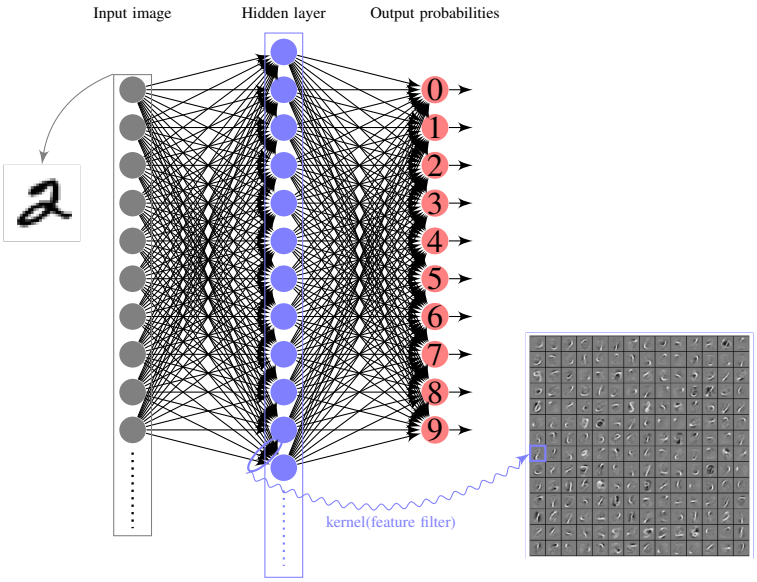

Figure 3.4: A regular feed forward neural network with 1 hidden layer. A 28x28 mnist digit image is flattened and fed into the network, it outputs probabilities of being a certain digit from 0 to 9. The learned kernel for each hidden unit is also shown.

Basic Network Structure

With appropriate loss function, a single hidden unit can function as a linear classifier. However, due to simplicity, its representation power is limited. For example, XOR operation can not be implemented with a single unit, but can be implemented with 3 units. For the network to have enough learning capacity to solve complicate tasks, hidden units are usually aggregated and stacked to form a layer by layer structure. With appropriate regularization, the network can learn powerful features and perform well for many tasks. A regular neural network is shown in Fig 3.4, a deep neural network can have many hidden layers. [64] provides an interactive play ground for tinkering with neural networks.

Backpropagation Algorithm

During the optimization process of loss function, we need to calculate the gradients with respect to all the weights and update them. A naive way is keep all the weights fixed and only tweak one weight to see how it changes loss, thus get its gradient. However, this method is very inefficient since it has to tweak order of the number of weights times to get gradients for all the weights.

The efficient way to do this is backpropagation algorithm. Detailed derivations on backpropagation algorithm can be found on tutorials [65, 61]. We highlight some points here:

- It helps to think of the mapping from input to output that the neural network represents as a computational graph. Each node in this computational graph is simple operation like $+$, $*$, $\max$, or any operation like σ that have a simple derivable gradients. In such way, any complicate mapping function is decomposed into simple unit operations, and backpropagation refers to the node by node backward propagation of gradients from loss to input using chain rule.

- There can be many paths from loss to any variable in the network, gradients flowed through all these paths backwards to this variable get added up when calculating gradient on this variable.

- In neural networks, the preactivations of a layer z and activations of previous layer o have relation $z = w^T o$ where w are the weights between these two layers, it can be seen that gradients for w are proportional to activations o of previous layer, this implies that the scale of the data has an effect on the magnitude of gradients for the weights. When input data is not scaled well, the gradient can be huge, so data preprocessing is a good practice.

Training

Below we list most of the practical concerns when training neural networks:

Data Preprocessing Common data preprocessing steps include mean subtraction, normalization, PCA whitening [66]. Mean subtraction is to zero-center the data. Normalization refers to normalizing the data dimensions so that they are of approximately the same scale. PCA whitening is to first apply PCA on the dataset, followed by a normalization step on the principle components. For most computer vision tasks with big image datasets as input, mean subtraction is enough.

Weight Initialization Mostly used weight initialization strategy is random sampling from a damped gaussian distribution. Improvements on randomly initialization are focused on

calibrating the variances with respect to number of inputs for each neuron such that its output's variable does not blow up. For ReLU units, it is recommended to draw weights from $\sqrt{\frac{2.0}{n}}\mathcal{N}(0,1)$ as suggested in [62]. A recently developed technique called batch normalization explicitly forces each neuron's activations for a minibatch to take on a unit gaussian distribution through whitening among this minibatch [67]. Batch normalization can be interpreted as doing preprocessing at every layer of the network, but integrated into the network itself in a differentiable manner. In practice networks that use batch normalization are significantly more robust to bad initialization.

Regularization $\mathcal{L}2$ regularization $\frac{1}{2}\lambda w^2$ is still the most commonly used regularization term. Other than that, $\mathcal{L}1$ regularization $\lambda|w|$ leads to sparsity in weights, it is useful when the problem is concerned with explicit feature selection. Dropout [68] is a simple but effective way of regularization, in practice it smears activations with a certain probability during training, which can be interpreted as sampling a neural network within the full neural network, and only updating the parameters of the sampled network based on the input data, therefore effectively there are many more neural nets working as an ensemble to eventually perform the classification.

Loss Functions In classification problems, most commonly used data loss functions are hinge loss and cross-entropy loss. Hinge loss is defined as $L_i = \sum_{j \neq y_i} \max(0, f_j - f_{y_i} + 1)$ where L_i and y_i refer to loss and ground truth label of ith sample, f_j is the output score for class j. Hinge loss is used with SVM classifier. Cross-entropy loss is defined as $L_i = -\log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}})$, and used with Softmax classifier. For regression problems, $\mathcal{L}2$ distance $L_i = ||f - y_i||_2^2$ are normally used. Whenever possible, it is recommended to see if a regression problem can be morphed to a classification problem by quantization of outputs into bins, because classification is an easier task and gives confidences of each class.

Optimization There are many optimization options when training deep neural networks. Starting from vanilla SGD (stochastic gradient descent), integration of momentum improves convergence rate and adaptive algorithms ease the pain of tuning the learning rates. [69] gives a comprehensive overview of related algorithms.

Miscellous To find good hyperparameters such as initial learning rate, regularization strength, it is suggested to use random search over grid search [70] and stage the search from coarse to fine. Bagging of neural networks trained independently is a reliable approach to improve the performance.

3.2 Convolutional Neural Networks

3.2.1 Architecture Overview

3.2.2 Layers

3.3 Applications

3.3.1 Classification

3.3.2 Detection

CHAPTER 4

CARDEA

4.1 System Design

Recalling related works in Chapter 2, what motivates the design of Cardea are the follow:

- People's privacy concerns are dependent on context. Although in certain circumstances locations are strong hints of possible privacy intrusion, generally what individuals are doing and with whom are more essential and crucial factors that directly relate to privacy.
- People's privacy preferences vary from each other, thus they should be able to express their personal privacy preferences.
- People's privacy preferences may change from time to time, therefore they need a way to change such preferences easily.

To achieve these objectives, we propose following solution:

- explain what composes context in Cardea
- registration of privacy preferences
- hand gesture for flexibility

Table 4.1: Scene categories.

Scene category	Scenes
Eating	banquet hall, beer garden, bistro, cafeteria, coffee shop, diner restaurant, food court, sushi bar
Entertainment	ballroom, bar, discotheque, pub
Shopping	bazzar, clothing store, department store, flea market, florist shop, general store, gift shop, jewelry shop, market, shoe shop, shopping mall, supermarket
Working	classroom, conference, cubicle, lecture room, library, office, reading room
Public places	crosswalk, downtown, field road, forest, freeway park, picnic area, street
Transportation	airplane cabin, airport, bus station, bus interior, subway station, train station, railroad track
Exhibition	art gallery, museum
Religion	cathedral, chapel, church, mosque, pulpit, temple
Illness	hospital, nursing home
Nudity	bathroom, beach, coast, lagoon, lavatory, swimming pool, shower, jacuzzi

4.2 Implementation

4.2.1 Scene Classification

Data Preparing and Preprocessing

For scene classification, we use pre-trained model of Places2 dataset provided by [71]. In the time Cardea project was conducted, Places2 dataset provided by the authors contained 401 categories with more than 8 million training images, and the pre-trained model was based on AlexNet structure [40]. By the time this thesis is writing, the dataset is deprecated and the new Places2 dataset contains 365 categories. And the authors provide more pre-trained models based on different network structures [72].

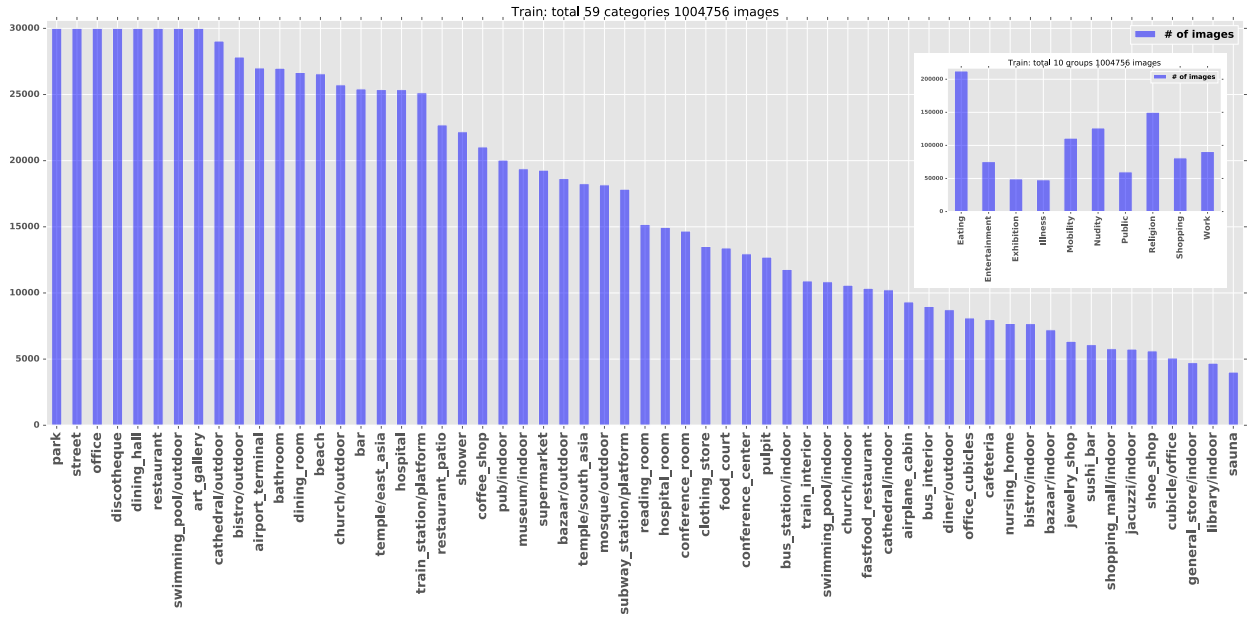

Figure 4.1: Number of images for each category and each group (inset).

Note that in the dataset we used, there is a non-uniform distribution of images per category for training, ranging from 4,000 to 30,000, mimicking a more natural frequency of occurrence of the scene. Among the 401 categories, we choose 59 scene categories that are close to daily life and in such scenes people may have privacy concern. In total this subset composed of 1 million training images and 2950 validation images (50 validation images for each category). We also group these 59 scene categories into 10 groups based on contextual similarity, such that people have similar reasons for privacy in scenes that are in the same group (e.g. people don't want to be captured in bathroom and beach is both because of nudity concerns). The distribution of training images among categories and groups is shown in Fig 4.1.

Training Procedures

Our training step is just a standard fine-tuning process, which is extensively used in transfer learning [73, 74]:

❶ Using pre-trained model as feature extractor, we extract the features at *fc7* layer for images belonging to the 59 picked categories. Other than shuffling the features, we also augment

the features such that all categories have same amount of features. Though the natural frequencies of occurrence are obviously different among different scenes, we argue that for the purpose of privacy protection, all the scene categories should be equally important, thus categories imbalance is not what we favored. The augmentation step can be implemented using weighted loss layer, but we take simple way of bootstrapping features for categories with less images. After this step, all features are cached and stored in lmdb format.

❷ Train a softmax classifier of the 59 categories using the extracted features. We choose to train a classifier for categories and then add up the output probabilities to predict the group, rather than directly train a group classifier, is because category classifier tells more about the image, and our desired property is equal weights among scene categories rather than groups.

❸ Both feature extraction and classifier training are implemented using Caffe library [75, 76]. In this step we merge the feature extraction part of pre-trained model and the softmax classifier into a single model by copying weights. Now Caffe has the option of specifying layers with fixed weights, thus simplifying the fine-tuning and deployment process.

Other than improving the validation accuracy from 0.56 to 0.57, shuffling also makes training converges faster. With augmentation to relieve category imbalance issue, the classifier can finally achieve 0.600 validation accuracy on the 59 categories. There is no other benchmarking result specifically on the subset we choose, but recent benchmark gives 53%-56% validation accuracy on the new Places2 dataset with 365 categories [72], suggesting our model is competitive. The higher validation accuracy of our model is due to the smaller scale of classification problem we are dealing with.

Prediction

For prediction, we get probability of a group by summing up the probabilities of all categories belonging to this group, and output the most probable group as prediction of an image. Our model's group prediction accuracy for the validation set is 82.8%. Fig [] shows some prediction examples. As seen from the examples, given an image, the predicted category probabilities are usually stratified to few categories within same group, thus group prediction is resilient to perturbation of

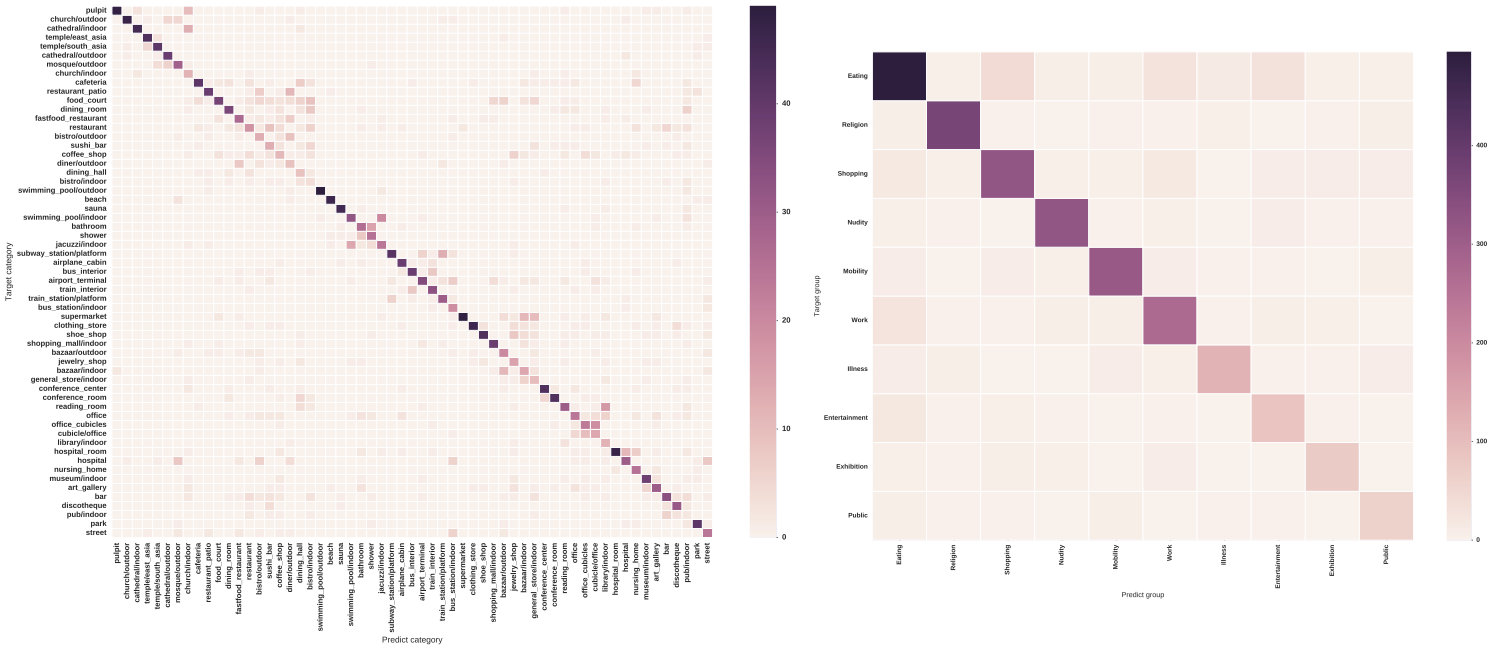

Figure 4.2: Confusion matrices for category prediction and group prediction.

category prediction. The way we group categories can be seemed as a hard-coded clustering step, which makes prediction more robust to noise. The failure cases are mostly due to natural context ambiguity from a image (e.g. image with object in focus, therefore not enough hints for scene inference). Labeling the 342 non selected categories as extra group will amplify the ambiguity issue, even for images with less ambiguity, doing so will stratify probabilities to the extra group and lead to wrong prediction. In other words, a 59 way classifier leads to higher recall for selected scenes and grouping leads to higher accuracy. This is also reflected in confusion matrices shown in Fig 4.2, category confusion matrix shows some clustering structure which is in accordance with the groups we manually assigned. However, only using top 1 category for prediction sacrifices prediction accuracy, which can be avoided by grouping as shown in group confusion matrix.

4.2.2 Face Recognition

4.2.3 Gesture Recognition

4.2.4 Deployment on Android

4.2.5 System Integration

data flow

screen shots

decision tree

4.2.6 Overview

technical focus , final object is to prove feasibility of proposed solution, shed lights on future explorations

4.3 Evaluation

CHAPTER 5

CONCLUSION AND FUTURE WORK

Bibliography

[1] *Number of surveillance cameras per thousand people by country*. URL: http://www.statista.com/statistics/484956/number-of-surveillance-cameras-per-thousand-people-by-country/.

[2] Heather Kelly. *iPhone photography is cool, eyeball photography is cooler*. URL: http://money.cnn.com/2016/05/12/technology/eyeball-camera-contact-sony.

[3] Tara Seals. *Popular Android Camera App Leaks Sensitive Data*. URL: http://www.infosecurity-magazine.com/news/popular-android-camera-app-leaks/.

[4] Ian Goodfellow Yoshua Bengio and Aaron Courville. "Deep Learning". Book in preparation for MIT Press. 2016. URL: http://www.deeplearningbook.org.

[5] Jiwon Kim. *Awesome Deep Vision*. URL: http://github.com/kjw0612/awesome-deep-vision.

[6] Yaniv Taigman et al. "Deepface: Closing the gap to human-level performance in face verification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1701–1708.

[7] Yi Sun et al. "Deepid3: Face recognition with very deep neural networks". In: *arXiv preprint arXiv:1502.00873* (2015).

[8] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 815–823.

[9] Tobias Weyand, Ilya Kostrikov, and James Philbin. "Planet-photo geolocation with convolutional neural networks". In: *arXiv preprint arXiv:1602.05314* (2016).

[10] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[11] Alessandro Acquisti, Ralph Gross, and Fred Stutzman. "Face recognition and privacy in the age of augmented reality". In: *Journal of Privacy and Confidentiality* 6.2 (2014), p. 1.

[12] Alison Spiegel. *Lost Lake Cafe, Seattle Restaurant, Kicks Out Patron For Wearing Google Glass*. URL: http://www.huffingtonpost.com/2013/11/27/lost-lake-cafe-google-glass_n_4350039.html.

[13] Jeremy Schiff et al. "Respectful cameras: Detecting visual markers in real-time to address privacy concerns". In: *Protecting Privacy in Video Surveillance*. Springer, 2009, pp. 65–89.

[14] Cheng Bo et al. "Privacy. tag: Privacy concern expressed and respected". In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM. 2014, pp. 163–176.

[15] Franziska Roesner et al. "World-driven access control for continuous sensing". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2014, pp. 1169–1181.

[16] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[17] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. "In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies". In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM. 2014, pp. 2377–2386.

[18] Roberto Hoyle et al. "Privacy behaviors of lifeloggers using wearable cameras". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 571–582.

[19] Roberto Hoyle et al. "Sensitive lifelogs: A privacy analysis of photos from wearable cameras". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 1645–1648.

[20] Paarijaat Aditya et al. "I-pic: A platform for privacy-compliant image capture". In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (New York, NY, USA, 2016), MobiSys*. Vol. 16. 2016.

[21] Kenta Chinomi et al. "PriSurv: privacy protected video surveillance system using adaptive visual abstraction". In: *International Conference on Multimedia Modeling*. Springer. 2008, pp. 144–154.

[22] Jaeyeon Jung and Matthai Philipose. "Courteous glass". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM. 2014, pp. 1307–1312.

[23] Nisarg Raval et al. “What You Mark is What Apps See”. In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (New York, NY, USA, 2016), MobiSys*. Vol. 16. 2016.

[24] Mohammed Korayem et al. “Screenavoider: Protecting computer screens from ubiquitous cameras”. In: *arXiv preprint arXiv:1412.0008* (2014).

[25] Robert Templeman et al. “PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces.” In: *NDSS*. 2014.

[26] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. “A Scanner Darkly: Protecting user privacy from perceptual applications”. In: *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE. 2013, pp. 349–363.

[27] Suman Jana et al. “Enabling fine-grained permissions for augmented reality applications with recognizers”. In: *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. 2013, pp. 415–430.

[28] J Alex Halderman, Brent Waters, and Edward W Felten. “Privacy management for portable recording devices”. In: *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. ACM. 2004, pp. 16–24.

[29] *Tegra Processors*. URL: `http://www.nvidia.com/object/tegra.html`.

[30] *Tegra X1 Processor*. URL: `http://en.wikipedia.org/wiki/Tegra#Tegra_X1`.

[31] *Snapdragon 820 Processor*. URL: `http://en.wikipedia.org/wiki/List_of_Qualcomm_Snapdragon_devices#Snapdragon_820_and_821`.

[32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[33] Feng-Hsiung Hsu. *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton University Press, 2002.

[34] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[35] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. DTIC Document, 1985.

[37] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. “A Fast Learning Algorithm for Deep Belief Nets”. In: *Neural Comput.* 18.7 (July 2006), pp. 1527–1554.

[38] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *Science* 313.5786 (2006), pp. 504–507.

[39] Yoshua Bengio et al. “Greedy layer-wise training of deep networks”. In: *NIPS*. 2007.

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

[41] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014).

[42] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

[43] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015).

[44] Pierre Sermanet et al. “Pedestrian detection with unsupervised multi-stage feature learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 3626–3633.

[45] Clement Farabet et al. “Learning hierarchical features for scene labeling”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1915–1929.

[46] Camille Couprie et al. “Indoor semantic segmentation using depth information”. In: *arXiv preprint arXiv:1301.3572* (2013).

[47] Dan CireAn et al. “Multi-column deep neural network for traffic sign classification”. In: *Neural Networks* 32 (2012), pp. 333–338.

[48] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (2016), pp. 484–489.

[49] *AlphaGo versus Lee Sedol*. URL: `http://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol`.

[50] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[51] George Dahl, Abdel-rahman Mohamed, Geoffrey E Hinton, et al. “Phone recognition with the mean-covariance restricted Boltzmann machine”. In: *Advances in neural information processing systems*. 2010, pp. 469–477.

[52] Li Deng et al. “Binary coding of speech spectrograms using a deep auto-encoder.” In: Citeseer. 2010.

[53] Geoffrey Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[54] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[55] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[56] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680.

[57] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[58] Alex Graves, Greg Wayne, and Ivo Danihelka. “Neural turing machines”. In: *arXiv preprint arXiv:1410.5401* (2014).

[59] Kelvin Xu et al. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *Proceedings of The 32nd International Conference on Machine Learning*. 2015, pp. 2048–2057.

[60] Chris Olah and Shan Carter. *Attention and Augmented Recurrent Neural Networks*. 2016. URL: http://distill.pub/2016/augmented-rnns/.

[61] *CS231n: Convolutional Neural Networks for Visual Recognition.* 2015. URL: http://cs231n.github.io/.

[62] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1026–1034.

[63] Ian J Goodfellow et al. “Maxout networks.” In: *ICML (3)* 28 (2013), pp. 1319–1327.

[64] *Neural Network Playground*. URL: http://playground.tensorflow.org/.

[65] Michael Nielson. *Neural Networks and Deep Learning*. 2016. URL: http://neuralnetworksanddee
com/index.html.

[66] *PCA whitening*. URL: http://ufldl.stanford.edu/tutorial/unsupervised/
PCAWhitening/.

[67] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[68] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[69] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2016. URL: http:
//ufldl.stanford.edu/tutorial/unsupervised/PCAWhitening/.

[70] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* 13.Feb (2012), pp. 281–305.

[71] *Places2 Dataset Project*. URL: http://places2.csail.mit.edu/index.html.

[72] *Release of Places365-CNN*. URL: http://github.com/metalbubble/places365.

[73] Ali Sharif Razavian et al. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 806–813.

[74] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[75] *Caffe*. URL: http://caffe.berkeleyvision.org/.

[76] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).