Context Aware Face Recognition

by

Nina Taherimakhsousi
B.Sc., Computer Engineering, IAU, Iran, 2006
M.Sc., University of Ferdowsi and
Sharif University of Technology, Iran, 2008

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

Context Aware Face Recognition

Supervisory Committee

Dr. Hausi A. Müller, Supervisor
(Department of Computer Science. University of Victoria)

Dr. Alex Thomo, Departmental Member
(Department of Computer Science. University of Victoria)

Dr. Panajotis Agathoklis, Outside Member
(Department of Electrical and Computer Engineering. University of Victoria)

Abstract

In common face recognition systems the recognition rate is not sufficient for today's applications, and systems only work in conditional databases and fail in unconstrained conditions.

The problem addressed in this dissertation is how to exploit context information to enhance face recognition. Therefore, this dissertation focuses on the investigation of dynamic context management and adaptivity to: (i) improve context awareness and the exploit of the value of contextual information to enhance the recognition rate in face recognition systems, and (ii) improve the dynamic capabilities of adaptivity in face recognition systems by controlling the relevance of contextual information collecting, analyzing and searching context.

Context awareness and adaptivity pose significant challenges for face recognition systems. Regarding context awareness, the first challenge addressed in this dissertation is data collection that can automatically analyze images in order to categorize and summarize contextual information. The second challenge arises from data extraction due to the big size of database of faces. Concerning adaptivity, the third challenge is to improve adaptive learning and classifier method with respect to variations. The fourth challenge, related also to adaptivity, concerns the high rate of videos generated by users from a dense urban area to decentralized cloud infrastructure. The fifth and sixth challenges concern the human's visual system in terms of contextual information in face recognition.

Given these challenges, to improve context awareness and adaptivity in face recognition systems we made four contributions. First, we proposed our framework for location-based face recognition. The framework comprises location-centric image databases to recognize faces in images that have been taken at nearby locations frequently visited by individuals. Second, we defined contextual information and architectural designed for context aware face recognition systems. Third, we designed contextual information extraction algorithm with an architecture for context aware video-based face recognition, which decentralizes cloud computing on the SAVI network infrastructure. Fourth, we designed an experimental study of face recognition by humans. The experimental study provided insights into the nature of cues that the human visual system relies upon for achieving its impressive performance serving as the building blocks for the developed context aware face recognition system.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

During the years of my graduate studies, I have been blessed to have so many supportive people around me, to each of whom I am deeply grateful.

Firstly, I would like to express my sincere gratitude and respect to my research supervisor, Professor Hausi A. Müller, for his unbelievably kind support, patience, and the never-ending encouragements, without which the completion of this research work would not have been possible. He not only taught me how to think critically and independently as a researcher but also how to be a good academia member. I am grateful to him for giving me the chance to explore other aspects of academia, such as teaching, developing and updating courses, and lab supervising. All of the valuable opportunities he provided me with, as well as always being available and enthusiastic about holding scientific discussions, helped me to build up my self-confidence and taught me how to perform a research independently. A big thanks from the bottom of my heart, and I owe you a lot.

I don't want to let this opportunity pass without acknowledging all the people that played an important role during my time at UVic, encouraged me academically and also gave me their friendship. I would specially like to thank to Wendy and Nancy in the Computer Science department; they were always ready to provide help and advice whenever I needed it. Also, I would like to thank to my classmates, my colleagues in the lab, and my other friends at UVic. Thanks to all of you for your help and for making this stage of my life more enjoyable. Dr.Ulrike Stege, Dr. Sudhakar Ganti, Dr. Alex Thomo, Dr. Pan Agathoklis, Dr.Jim Tanaka, Dr. Maia Hoeberechts, Dr. Alexandra Branzan Albu, Dr. Amirali Baniasadi and of course, special thanks to my colleagues in Rigi Research Dr. Lorena Castaneda, Dr. Andreas Bergen, Stephan Heinemann, Charlie Magnuson, Pratik Jain, Ernest Aaron, Dr. Ron Desmarais, Prashanti Priya Angara, Miguel Jimenez.

Of course, I cannot forget to thank those that I left back in Iran and that are my support network in the distance, my family and friends. I would like to express my deepest respect and appreciation to my parents and my sister, for their unconditional love, strong support, and continuous encouragements without which the completion of my graduate studies would not have been possible. Special thanks to all my friends who were always there for me and helped me to stay positive during tough circumstances. Some goes to my friends Samira Motalebi, Dr. Amineh Amini, Naghmeh Banisadr, Didar Barghlame, Dr. Alireza Tari, Alireza Hajiany, Dr. Azadeh Fattahi,

Dr. Majid Soleimani nia, Laurie Barnas, Dr. Maryam Ahmadi, Dr. Maryam S. Mirian, and Dr. Sara Rouhani.

Finally, I would like to acknowledge the "Natural Sciences and Engineering Research Council of Canada" (NSERC) for their financial support during the course of this research.

DEDICATION

This thesis is dedicated to
My love who believes in the richness of learning
My son who made me keen on learning

Chapter 1

Introduction

1.1 Motivation

Face recognition in humans is subconsciously associated with contextual information from the environment and social parameters [AT13]. Contextual information helps us to identify faces in daily social interactions and humans may fail to recognize the observed face without this information [MB13]. Hence, taking the contextual information into account in real-world face recognition applications is of vital importance to enhance the performance and reliability of the automatic face recognition systems [AAC16]. Contextual information includes information related to the image of the scene surrounding the person, camera context such as location and image capture time, and the social context that describes the interactions between people. Further to cognitive approach, the statistical approach can also be used to tackle the face recognition problem. There is also a significant statistical correlation between contextual information and image information, which enables statistical operators to achieve a higher recognition rate based on contextual information [Riv14]. In a more general manner, the face recognition problem can be approached with information theory, because real-world face recognition is an open problem and contextual information is not redundant with respect to database and image information [SAW94]. Overall, contextual information can be used to perform face recognition faster and more confidently but why does the performance of automatic face recognition systems need to be improved?

One answer to the above question pertains to the availability of cameras in multiple sensory devices that allows capturing numerous images and videos and thereby

creating vast archives of data. These huge amounts of data are required to be analyzed in order to categorize, summarize and make them searchable to retrieve the information that the user may need [LXT+18]. Thus, there is a need for high performance automatic face recognition systems. This need motivated us to conduct the current research study which includes two key aspects.

The first aspect is to engage contextual information in face recognition systems and exploit its value to improve recognition rate and systems performance. By making the system capable of gathering and processing contextual information, consciously and continuously, from internal and external entities that can affect the accomplishment of the system, we offer context awareness. In other words, the system must be able to model, acquire, process, provide, and dispose contextual information. In context aware face recognition systems, relevant context observations can be gathered from images and other resources. For instance, location and time can be obtained from first user cameras.

Traditional face recognition systems only support conditional database and rigid templates and do not result in quality output in unconstrained conditions. This limits the application of such systems in real-world settings. However, new algorithms may be developed so that the systems can perform effectively. This led to the second key aspect of this research, making systems self adaptive and responsive to variation of contextual information (e.g., sensor noise, viewing distance, and illumination [YBR06]) that may affect the expected system behavior. This is accomplished by training at runtime using adaptive learning algorithms and results in a system which modifies itself at runtime and according to changes of contextual information.

1.2 Problem Statement, Research Challenges, Goals, and Questions

In common face recognition systems: (1) the recognition rate is not sufficient for today's applications and, (2) systems only work in conditional databases and fail in unconstrained conditions. To advance the state of the art of face recognition systems we identified two main research avenues: (1) context awareness and (2) adaptivity. Thus, this research has been driven by the following two main motivations:

M1. The need for improving context awareness and the exploitation of the value of contextual information to enhance the recognition rate in face recognition systems.

M2. The need for improving the dynamic capabilities of adaptivity in face recognition systems by controlling the relevance of contextual information collecting, analyzing and searching context.

1.2.1 Problem Statement

This dissertation addresses the research problem of how to exploit context information to enhance face recognition:

> Context aware face recognition in which contextual information helps solve the face recognition problem effectively requires automatic data collection, contextual information extraction, and adaptive learning. For face recognition systems to become smarter: (1) Contextual information must be added to exhibit an explicit relationship with the face recognition system; (2) The resulting face recognition system must adapt to the relevant image database and contextual information entities at runtime.

1.2.2 Research Challenges

The section outlines the research challenges we addressed in this research. These challenges are classified according to two main parts of face recognition systems and this study on human visual system for context aware face recognition. Challenges RCH1 and RCH2 concern the research related to data collection and context extraction. Challenges RCH3 and RCH4 concern the improvement of the adaptive learning and classification algorithms. Challenges RCH5 and RCH6 concern the human visual system in a manner of contextual information.

Data collection and context extraction

RCH1. Developing a system that can automatically analyze images in order to categorize, summarize and recognize faces needs more information than just raw images. Therefore, an automatic data collection method is required to make an image database equipped with contextual information such as location, time, and image content.

RCH2. A big database of faces reduces both accuracy and speed of the face recognition system. Therefore, a mechanism is required to project face images onto a feature space to make the classifiers faster. Hence, significant features that are principal components of the faces are needed.

Adaptive learning and classification

RCH3. Face recognition methods have worked on databases under well controlled conditions such as frontal full-screen faces which is not the case here. Therefore, a robust adaptive learning and classifier method is required with respect to variations such as size, view, expression, and light of faces.

RCH4. Given the nature of video arrival, a high arrival rate of videos generated by users can easily overwhelm the paths into the centralized cloud infrastructure. Therefore, a decentralized cloud infrastructure is required to scale well beyond this to millions of concurrent uploads from a dense urban area.

Human visual system

RCH5. The use of natural images in a human experiment design makes the design difficult because of the substantial variance in the images. Thus, the underlying mechanisms responsible for this seemingly complicated task need be isolated.

RCH6. The context base face recognition task itself is an open human experiment task, which means that participants can employ a range of different strategies to solve it. Therefore, creating and updating an abstract representation of each individual's facial identity is required.

1.2.3 Research Goals

From the findings of our exploratory study and taking into account our research questions, we stated the goals of this dissertation as follows.

The long-term goal of this research, beyond this dissertation, is to investigate innovative techniques to optimize the design, implementation, maintenance and evolution of a context aware face recognition system.

The short-term goal of this dissertation is to investigate the application of context aware techniques to improve context awareness throughout the face recognition process.

1.2.4 Research Questions

Based on our research goals and challenges, we defined the following four research questions:

RQ1. How to form the location categories effectively? How to take location information into account in the feature extraction processes? How to search efficiently to recognize the faces? How to advance the recognition steps, and minimize response times by taking advantage of Future Internet nodes such as the SAVI network?

RQ2. How does the use of contextual information impact face recognition performance? How do selected types of contextual information affect the face recognition performance for different scenarios? Is a certain type of context more effective than others for certain scenarios?

RQ3. How can a web scale face recognition system exploit contextual information accurately, efficiently and adaptively with millions of web users and billions of photos?

RQ4. What is the effect of contextual information on human face recognition? How does contextual information affect human face recognition accuracy and response time?

1.3 Methodological Aspects

1.3.1 Research Methodology

In this research we use parallel sequential mixed methods; combining human vision and computer vision approaches [Bra17]. Figure 1.1 depicts our research methodology in two visions: First a collection and analysis of databases with contextual information using our computer vision approach. Second, the experimental studies and analysis of data with our human vision approach to support our previous computer vision approach. In both approaches results provide feedback to the system.

Limitations

We recognize that the research area of face recognition is broad and complex. This dissertation focuses on the two main areas of research mentioned earlier in this chap-

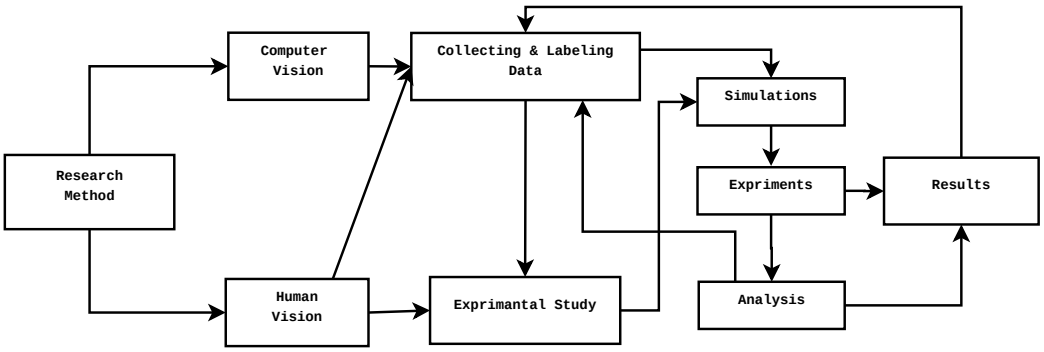

Figure 1.1: Research methodology

ter: context awareness and adaptivity on face recognition. Thus, the scope of this dissertation is related to those research areas. Our contributions, implementations and experiments are focused on our main research interest.

We identify two potential major limitations: *availability of databases with contextual information* and the *database size*. To overcome the limitation related to the availability of databases and required acquisition sensors, we design our collection method, which provides an adaptive mechanism to gather images with contextual information dynamically. However, in our evaluation we have a limited number of images from a real-world database generated by third party applications (e.g., social networks and websites) and multi-sensory devices (e.g., Google glasses and Microsoft HoloLens). Also, we use simulation techniques for those multi-sensory devices that are not readily available for our use.

1.3.2 Research Approach

The first step in this research is to explore the current state of *face recognition*. We found simplified systems by classic databases, such as CMU-PIE [GMC+10], FERET [PMRR00] and CMU-Pittsburgh AU-Coded [KCT00], and research that concentrated on face representations that are invariant to changes in pose [DT16a], illumination [YLQ+17], and facial expression [RPP13]. The findings of the aforementioned exploratory studies contributed to measuring the similarity between face images. Identification is performed based only on face pixels which need heavy training and work just for its own database. Consequently, current face recognition systems face with some challenges with real-world databases and face changes. This constitutes research gap directs us toward building a smart face recognition system capable of identifying face changes. Additionally, our findings revealed that even when there are

approaches to recognize changes in pose, illumination, and facial expression, there is a lack of runtime support to represent execution time challenges, such as new faces in databases [BTJ+13, TM15b] and automatically adding new data features [TM15a].

The second step is to designing our system for context aware face recognition. To do so, we studied different types of methods and focused on those suitable to represent the contextual information to create databases. More importantly, we focused on methods that are capable of being adapted. As a result, we defined an adaptive database creation method based on certain types of contextual information such as location [TM15b].

Third, we focused on different components of our approach that can support the context awareness idea. We defined and categorized what we call contextual information and designed a method to extract contextual information. Then, we analyzed our developed method by different contextual information to demonstrate the usefulness of contextual information in face recognition systems. In addition, we designed and implemented our system architecture for video based face recognition dealing with runtime requirements and providing adaptive capabilities [TM16]. Our architecture takes advantage of the SAVI network [KBLG13], and a mixture of expert learning methods [ME14] to support self-learning and adaptability. We performed a qualitative assessment in which we compared our approach with related approaches. Also, we evaluated our approach by measuring the accuracy and the capability of the architecture.

Fourth, we focused on the rapid and efficient human visual system based on contextual information which performs in a manner that surpasses any existing computational model. The recognition speed is a central aspect of our perception as the fast recognition of faces is crucial to many of our activities. As an experimental study, we created a context aware database and ran simulations in which contextual information changes, thus affecting users' recognition.

1.4 Contributions

This section summarizes the four main contributions of this dissertation.

C1: Location-based Face Recognition Approach

Our first contribution is a framework for location-based face recognition. We provide a detailed description of the method for using location information within the proposed algorithm. The framework comprises location-centric image databases to recognize faces in images that have been taken at nearby locations frequently visited by individuals. The approach is defined as follows: (1) given a set of known images of faces for training and another set of faces of the same people as a testing set, (2) recognize each face in the testing set, (3) each face image associates with the location information, and (4) creates many clusters of locations from the training set where each location cluster contains a set of individuals, who have images in that location, and images of their interacted people. Finally, (5) the user can take an image and attach the location information, then send it to the system and query for recognizing the face in the image. The system will answer the recognition question and return the identification to the faces in the image.

C2: Context Definition and Smart Applications for Face Recognition

Our second contribution is a definition and an architectural design for context aware face recognition systems and their smart applications. Context is broadly defined as information relevant to something under consideration which can include information from non-face regions of the image, information related to the capture of the image, or the social network context of the interactions between people. The useful contextual information for face recognition is defined here in four categories: (1) face context, which provides information about a face. Such as anthropometric measurements, skin color, and distance between face parts, (2) pixel context, which is the non-face regions of the image information such as distinctive clothing, classes, and other faces, (3) sensor context, which is knowing the capture conditions of an image such as location, time, and brightness, (4) interacted context, which is the information about social relationship, weak labels, age, and gender. This architectural design is based on contextual information that assists face recognition systems act smarter. This

study demonstrates how face recognition applications can become smarter using the contextual information.

C3: Automatic Context Extraction and Decentralized Cloud Computing on SAVI Network

The third contribution is the design of a contextual information extraction algorithm as follows: (1) context aware filters are initially applied with low computational complexity on a subset of the selected video frames, then (2) more complex context aware filters are applied which extract features and relevant contexts, in order to increase face recognition accuracy, (3) detected faces are normalized to the same size and finally, (4) the detected faces are automatically added to the cloud base database along with contextual information which is the main element of adaptability.

Also, this contribution includes an architecture for context aware video based face recognition, which decentralizes cloud computing on the SAVI network infrastructure: (1) a video from an individual mobile device travels as far as its currently associated SAVI node, (2) computer vision analytics run on a SAVI node VM in near real time, (3) the Data Manager runs in an individual VM on the SAVI network to manage the storage of the videos and database with the associated contextual information, (4) the data is logically organized as a collection of videos, (5) results of the processing along with contextual information (such as the VM details, location, start time and video duration) are sent to the SAVI core, and (6) the labels and contextual information in the SAVI core can guide and facilitate deeper and more customized searches of the contents of a video during its retention period on a SAVI node's VM.

C4: Recognizing Faces in Different Contexts by Humans

This contribution demonstrates a design of an experimental study of face recognition by humans. The experimental study provides insights into the nature of cues that the human visual system relies upon for achieving its impressive performance serve as the building blocks for the developed context aware face recognition system. It also showed that the benefit of reinstatement is diminished when encoding contextual information is associated with many study episodes. This experimental study includes the following steps: (1) a database of individual images is created with and without contextual information (e.g., workplace, working clothes, and generally neu-

tral emotional expressions), (2) design study phase, participants viewed a series of faces from the database with contextual information. Participants were instructed to remember these faces. (3) design the test phase, faces were presented with and without contextual information. The response indicated whether the face was old or new (i.e., one from the study phase or not, respectively), and (4) the results of the contextual information effect the response time and accuracy for both old and new faces.

1.5 Dissertation Outline

The remaining chapters of this dissertation are organized as follows:

Chapter 2: Research Background—presents four background topics relevant to the research of this dissertation: (1) the *feature extraction* which is used for scene understanding and face recognition, or pose estimation that can be used for video representation, (2) *Internet video clustering*, which is the technological domain of this research, (3) the foundational concepts of *context awareness*, which concerns a recent thrust in computer vision, and (4) the core conceptual element of this dissertation: *adaptive learning*, which is one of the most interesting methods that has great potential improving performance in machine learning.

Chapters 3-6 present our four contributions, respectively, as outlined in Section 1.4 above and the proof of concept, which includes case studies, implementations, test scenarios, simulations, results and findings.

Chapter 7: Summary, Conclusions and Future work—summarizes the research and the contributions of this dissertation, presents the conclusions and discusses potential future work.

1.5.1 Publications

- Andreas Bergen, **Nina Taherimakhsousi**, Pratik Jain, Lorena Castañeda,and Hausi A. Müller: Dynamic context extraction in personal communication applications. In Proceedings 2013 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2013), pages 261–273. IBM Corporation. [BTJ+13]

- **Nina Taherimakhsousi**, Hausi A. Müller: Context-based face recognition for smart web tasking applications. In Proceedings 2nd Workshop on Personalized Web-Tasking (PWT 2014) at Tenth IEEE World Congress on Services (SERVICES 2014), IEEE, pages 21–23. [TM14]

- Andreas Bergen, **Nina Taherimakhsousi**, Hausi A. Müller: Adaptive management of energy consumption using adaptive runtime models. In Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015), ACM, pages 120–126. [BTM15]

- **Nina Taherimakhsousi**, Hausi A. Müller: Location-based Face Recognition Using Smart Mobile Device Sensors. In Proceedings International Conference on Computer and Information Science and Technology (CIST 2015), IEEE, pages 111-116. [TM15b]

- **Nina Taherimakhsousi**, Hausi A. Müller: Context-aware real-time video analytics. In Proceedings Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2015), pages 223–226. IBM Corporation. [TM15a]

- **Nina Taherimakhsousi**, Hausi A. Müller: Context Aware Video Analytics with Decentralized Cloud on the SAVI Network. 4th International IBM Cloud Academy Conference (ICACON 2016). IBM Corporation. [TM16]

- Andreas Bergen, **Nina Taherimakhsousi**: Software Energy Optimization in the Cloud. In Proceedings Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2016), ACM, pages 243–249. IBM Corporation. [BT16]

- Juan C. Muñoz-Fernández, Alessia Knauss, Lorena Castañeda, Mahdi Derakhshanmanesh, Robert Heinrich, Matthias Becker, **Nina Taherimakhsousi**: Capturing Ambiguity in Artifacts to Support Requirements Engineering for

Self-Adaptive Systems. 23rd intl. Working conference on Requirements Engineering: Foundation for Software Quality (REFSQ) 2017. [MFKC+17]

1.6 Chapter Summary

This chapter presented the motivation, research challenges, questions, goals and methodology, as well as an overview of the contributions and publications resulting to this dissertation. We introduced the two main topics of this research, context awareness and adaptivity and explained the research methods and approaches to improve context awareness and adaptivity in face recognition systems as part of the four contributions. Figure 1.2 summarizes this dissertation including the relationships among research challenges, questions, goals, contributions, publications.

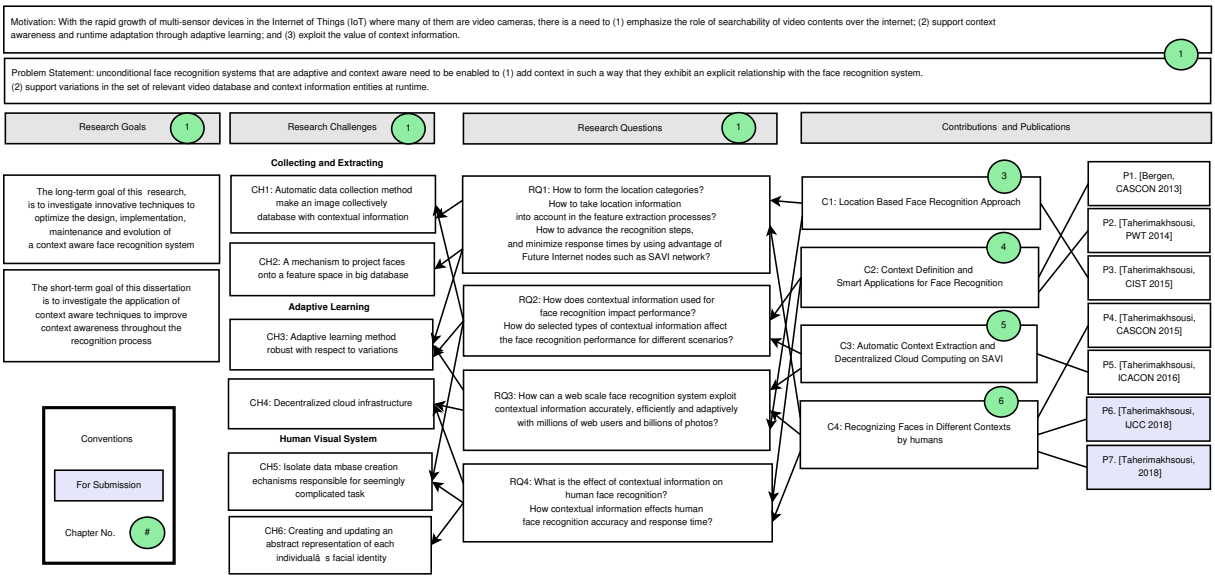

Figure 1.2: Dissertation roadmap

Chapters 2 and 7 were excluded from this map since they belong to the Background and Summary, Conclusions, and Future work, respectively.

Chapter 2

Research Background

There is extensive literature on different aspects of face recognition. In this chapter, we focus on literature that directly relates to our research. First, we review the main research foundations of automatic face recognition systems as well as their recognition process. Second, we provide an overview of different approaches of context based object recognition. Third, we provide an introduction to research about context in face recognition systems. Finally, we present a summary of the main research foundations that are relevant to the research presented in adaptive face recognition systems needed for our research.

2.1 Face Recognition State-of-the-art

Face recognition is routinely and subconsciously performed by humans during social interactions. The availability of personal computers and fast embedded computing systems has attracted increasing attention in automatic image and video processing in diverse applications, such as biometric authentication, recommendation systems, surveillance, as well as human and computer interaction (e.g., using Google Glass) [HG14].

In recent years, a variety of approaches have been proposed to tackle the problem of face recognition. However, there is plenty of room for investigating new approaches to achieve a variety solutions for the unconstrained version of these problems and in different contexts. Many of the proposed algorithms in the literature perform extremely well on unconstrained datasets, such as the Labeled Faces in the Wild

(LFW) [HLM14, TYRW14]. However, their underlying objectives are often unclear in the context of unconstrained face matching [MCC+19].

Although there has been a lot of encouraging achievements in the unconditional face recognition, the trajectory is shifting towards the role of such large unconstrained databases [PJXS16]. One general approach is to divide the problem of face recognition into subtasks of achieving invariance towards transformations of a face [LLP14]. Furthermore, recent significant progress on the supervised protocols of the LFW dataset approaching human accuracy has concealed the necessity for understanding the fundamental problems in vision tasks such as recognition [VDDP18].

Given the current trend, it is essential to develop methods that are based on fundamental principles, rather than just beating the current state-of-the-art, but to increase our understanding of the problem itself. So far, significant effort has been expended and reported in the literature about generating implicit invariance to specific individual or a small subset of these transformations at once. However, there is no study on an approach which generates explicitly invariant features to any unitary modeled transformation while being explicitly discriminative [ZCPR03, ANRS07, JA09, SGC15, DT16b, RB17].

Amongst recent works, the majority has focused on unconstrained face verification, and relies greatly on locating accurate and dense facial landmarks and descriptors to extract over-complete information from the image. They may also use 3D modeling in the algorithm [GMSR18]. Many of these systems are also closed sets [AWR+16].

There is also a different class of algorithms that are based on deep learning which have gained popularity recently [PVZ+15]. These algorithms utilize a large amount of data (high sample complexity) and increase model complexity significantly [DT18]. Although these methods have been widely successful, they failed to provide a better understanding of the problem because of their complex models and over-complete feature extraction combined with unconstrained testing protocols.

Due to the current trend in unconstrained face recognition, large-scale databases comprise an immense amount of certain unspecified types of transformations in each image. However, other modes of transformations such as translation, rotation and scaling are excluded by providing aligned faces. Having no control over the type and amount of other transformations tends to bias the development of face recognition systems in which it is unclear why some algorithms work well and others don't. Most of the current face verification methods use hand-crafted features. In addition, these

features are often combined to improve performance. The systems that currently have the highest performance employ tens of thousands of image descriptors [PWLG19].

2.1.1 Traditional Face Recognition Process

Figure 2.1 depicts a face recognition system which recognizes faces in images captured from a camera. It includes four modules: (1) segmentation, (2) feature extraction, (3) classification, and (4) decision. In addition, facial models of the N enrolled individuals are stored in the system, to be used by the classification module to produce matching scores for each individual. During the process, the segmentation module is used to isolate faces in the image, which produces the regions of interest. Then, discriminant features are extracted from each ROI (e.g. eigenfaces [TP91a] of local binary patterns [AHP06, LL12]) to produce the corresponding pattern $\mathbf{d} = (d[1], ..., d[F])$ (where F is the dimension of the feature space). In the next step, the classifier compares the obtained pattern to the facial model of each enrolled individual i , which produces the corresponding matching scores $s_i(\mathbf{d}), (i = 1, ..., N)$.

The facial models are typically designed in advance employing one or several reference patterns, from which the same features have been extracted and their nature depends on the type of classifier used in the system. For example, using a template matcher, a facial model of an individual i can be a collection of one or several reference patterns $\mathbf{r}_{i,j}(j = 1, ..., J)$. In such a case, matching scores for each operational pattern $\mathbf{d}$ would be computed from distance measures to these patterns. Neural networks (e.g., multi-layer perceptrons [TDH16] and neural networks [Sch15]) or statistical classifiers (e.g., naive Bayesian classification [CSG+03]) may also be used to perform classification, in which the facial models would consist of parameters that were estimated during their training using the reference patterns $\mathbf{r}_{i,j}$ (e.g., neural networks weights, statistical distribution parameters).

Finally, the final response gets produced by the decision module according to the application. For instance, an identification system for surveillance may predict the identity of the observed individual with a maximum rule, using the highest matching score to select the enrolled individual, while a verification system for access control generally confirms the claimed identity by comparing the corresponding matching score to a decision threshold.

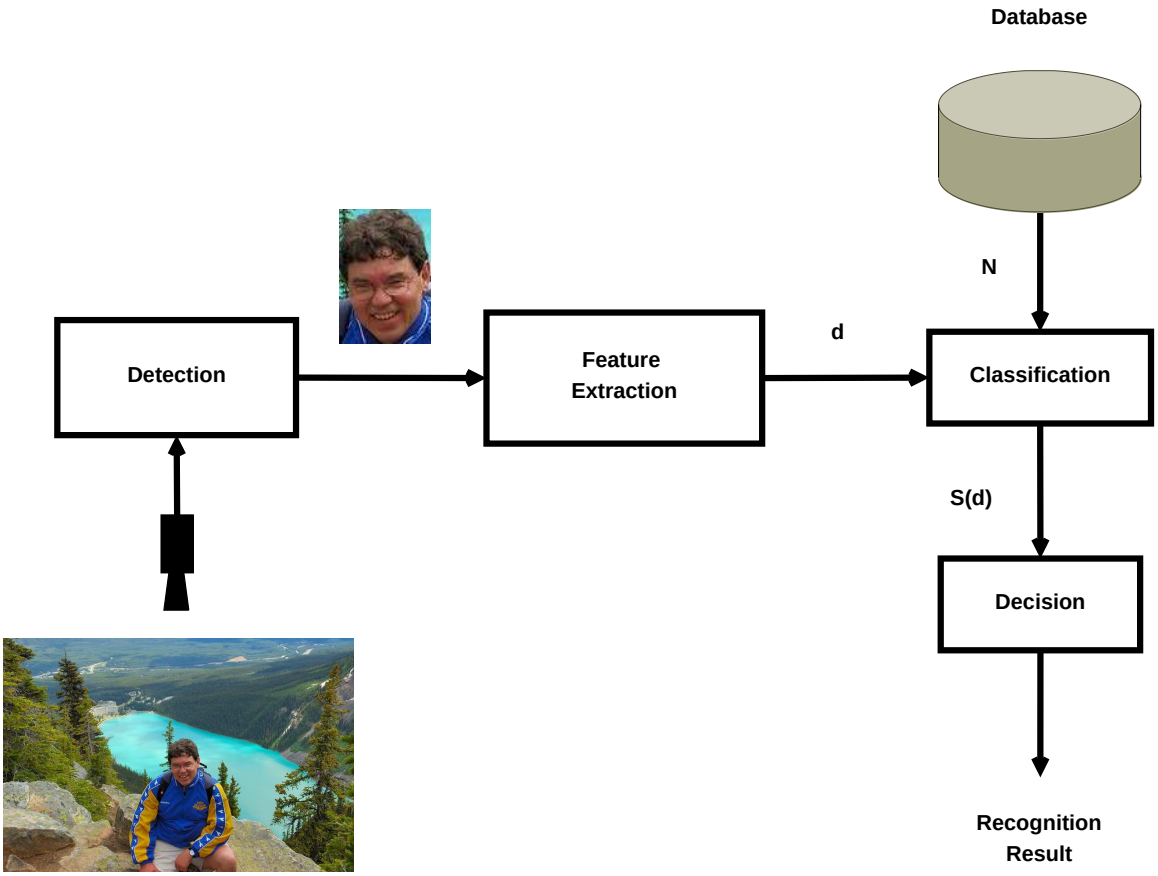

Figure 2.1: Scheme of a generic face recognition system. Each database image is captured with similar pose, illumination, distance, and expression

2.2 Context Aware Object Recognition

One of the research avenues in computer vision is using context in object detection and recognition. In the article entitled "Pictures Are Not Taken in a Vacuum" the authors use image metadata (i.e., "data about data") and image capture time to differentiate indoor and outdoor images [LBB16].

Martin et al. [GRSG+18] and Nikita et al. [DMS18b] demonstrated the context of a scene as well as the connection between context and object detection in 3D and 2D, respectively. In particular, they reduced misclassification by developing a context aware object recognition system that constrains the belief to comply with the probabilistic spatial context models. Zhang et al. [ZDZ18] describe learning the co-occurrence and relative co-locations of objects improves object recognition. In other works, authors integrated location into models to exploit the concept of considering relative location for object categorization [TLZR15, ZLSR17, MSG17, DMS18a].

2.2.1 Feature Extraction

In an important survey on context based object recognition, Galleguillos and Belongie categorized context information into three groups: (1) semantic context which refers to the probability of an object being observed in some scenes but not in others, and from the modeling perspective, can be expressed in terms of the corresponding object's probability of co-occurrence with other objects and the probability of occurrence in certain scenes, (2) spatial context which corresponds to the possibility of finding an object in certain positions with respect to other objects in the scene, and (3) scale context which make use of the fact that objects in the scene have a limited set of size relations with other objects [GB10]. They theorized that contextual knowledge can be any information that is not directly produced by the appearance of an object. In fact, it can be obtained from the nearby image data, image tags, or annotations as well as the presence and location of other objects.

The context of an object can be represented in terms of its relationship with other objects in the scene, e.g., co-occurrence based context model [DMS18a]. Heitz and Koller [HK08] proposed a terminology for this concept and introduced a "stuff and things" context model. In their model, the terms "stuff" and "things" are used to distinguish between "materials" that have uniform or repetitive patterns of fine scale properties, but have no distinctive spatial extent or shape (stuff) from "other objects with clearly defined size and shape". They claimed that classifiers for either things or

stuff can benefit from the proper use of contextual information. In another report a classification of contextual models was proposed for object recognition which is called Scene Based Context (SBC) models. SBC models consist of contextual inference based on the statistical summary of the scene and contextual information representation in terms of relationships among objects in the image [RB09].

There are also other proposed methods to model the contextual information in a comprehensive manner, e.g., [GGDH18]; however, these methods are very specified and designed for one certain computer vision task. Thus, they cannot be generalized for our target in context aware face recognition. We also notice that our work follows the research trend of stacking which uses the output of the classifiers as the input for the next layer of classifier [WFHP16]. We also more specifically looked at the auto-contextual information extraction models for weakly supervised image labeling, e.g., AutoContext [GJMG18] and Texton [RBF+16].

Object hierarchy contextual information has recently drawn much attention [WJ15, ZK15]. The object hierarchy approach extends the research of object co-occurrence contextual information under the assumption that objects are related with a semantic hierarchy. Object relationship, with the increased number of object categories, is naturally exhibited as a hierarchical structure. Contextual information modeling with numerous object categories seeks to model this relationship with high level semantic structure or learn from contextual information [AAL+15].

2.2.2 Modeling and Classification

Although there are many reports on contextual information representation and modeling, few of them focus on context awareness between object detection and classification, namely, high level task context.

In object classification, the task is aimed more for finding whether the image contains a certain kind of object rather than where it is. The task is solvable due to the facts that: (1) many data sets only concern the objects which occupy most of the images [ZKSE16], (2) the same category objects often share similar scene level information, and (3) the current prevalent object classification pipeline uses the sophistic feature encoding and learning method to extract image specific information which often reveals the object-specific contents, e.g., Fisher Vector Coding [PSM10] and SVM classifier [Sut16].

The methods used in object classification are often built with a top-down approach in which global information is used to infer the existence of a local object. For object detection, the task aims to localize the object within the image.The object detector mostly models the object appearance [CSF+13] or object shape [FGMR10, For14] through the annotated object samples while disregarding the contextual information defined by the surrounding object. The localized feature of the object detector restricts the model to differentiate the false alarm effectively which occurs at obviously different contextual information. Harzallah et al. [HJS09] introduced the pioneering work for object detection and classification with engaging contextual information after classification process with probability combination.

Additionally, the contextual information is commonly considered as extra features. Most of the existing strategies [SPMV13, PCMY15, GTM+16, SSGC17] utilize the contextual information via feature concatenation, feature fusion or combination, and take the contextual information as an independent feature. However, contextual information may have unstable distribution, and its reliability and noise level may not be controllable. Therefore, it requires adaptive context awareness with proper constraints to avoid the inappropriate usage of contextual information. In this dissertation, we follow this line to design the learning scheme for utilizing contextual information for our face recognition system.

2.3 Context in Face Recognition

As illustrated in Figure 2.1, a face of unknown identity is compared against a database of face images with known identities, where each database image is captured with similar pose, illumination and expression. There are significant differences between the technical challenge of face recognition in general and the problem we are addressing. For our face recognition system users, developing a dataset of their face images is inconvenient at best and impossible at worst.

Recently researchers have attempted to recognize people from contextual information that extends beyond face pixel data. Generally, contextual information is used to imply acceptable co-occurrence of various parts or features of an object or face [BVS14]. In particular, this is supportive in scenarios where the identities of people in an image have to be deduced [ADB+99]. Contextual information were used as secondary information along with face features to aid identification performance

Figure 2.2: Examples of contextual information that can be incorporated in order to enhance the face recognition performance. Images are taken from the Images of Groups [GC08] and Gallagher Collection Person [GC09] datasets.

[LKK+19]. As shown in Figure 2.2, different kinds of contextual information were used for improving recognition performance.

However, the idea of contextual information has been developed gradually. One of the initially proposed models incorporating contextual information to aid face recognition utilizes clothing information as secondary information for face recognition [ZCLZ03, SD12].

In another work, a semi-automated model for face and contextual information features was introduced which is based on a probabilistic Bayesian framework and performs face recognition in family photos. The model presents a candidate list of potential faces from which the user should choose the correct face. In another semi-automated model for face recognition temporal information, spatial information, as well as social contextual information are incorporated to aid face recognition [DSC+05]. Here, temporal information refers to the exact time the image was captured per smart device; spatial information refers to the smart device ID from the image sharing network and location of the smart device; and social contextual information only refers to the identity of the smart device user. A specific logger was designed and implemented on smart devices to track the aforementioned contextual information. In this work, face recognition was performed using Sparse-Factor Analysis (SFA) by combining face features and contextual information. The results of

their experiment demonstrated that utilizing contextual information improves the performance of face recognition compared to using either information independently.

A fusion model which engaged clothing information with face recognition results was proposed later to help face recognition performance. This model introduced a clothes recognition algorithm and its outcome was integrated into a spectral clustering algorithm to perform face recognition. Logic constraints were applied in a clustering algorithm to corresponding different faces. The results demonstrated that the performance of face recognition can be improved with clothing information and logic constraints [SL06].

A Markov Random Field (MRF) based model has been proposed for face recognition, by combining clothing features and facial features [ALGS07]. Time temporal contextual information was created for each event based on the clothing information of corresponding detected faces. Multiple levels of features, e.g., cloth's color and texture feature, were applied to encode clothing features, then the Loopy Belief Propagation (LBP) algorithm was engaged for detecting MRF inference [LL12, LLZZ15]. In another work, a clothing segmentation algorithm was introduced based on graph cuts [GC08]. To train the face recognition system a probabilistic model and extracted features from the face and clothing pixels were used. The model claimed to be efficient for image collections, where the number of faces in an image are known and some faces have already been recognized by the user. Hence, the model allows detection and recognition the remaining faces.

A decade ago, researchers started proposing models which were motivated by the large scale availability of contextual information on virtual social networks. One of the proposed models utilizes contextual information for complementing face recognition algorithms and automatically labeling face images in Facebook [SZD08]. The images and contextual information were collected from a set of Facebook users. Then, to train a Conditional Random Field (CRF) algorithm, the labeled images were used to link faces which were detected in the network images. The results of the experiment demonstrated an improved face recognition performance when contextual information was incorporated in the proposed model.

Logical contextual constraints was also attempted to get incorporated into the model of adaptive learning to recognize the faces in the group images [KHAB09]. This was done based on the previous contextual information and labeling the photos using match and non-match constraints. In a work relevant to group images an algorithm was introduced for incorporating a family's contextual information into

a face recognition model [WGLF10]. In the model, a weakly supervised labeling was used in group images and tried to label each face. Then, to train the system face features and family relationship contextual information were implemented in a graphical model based on the face appearance position in the image. The experimental results demonstrated an improved efficacy of the face recognition rate in group images.

Soft biometric traits, descriptive features, and contextual information was used for face recognition based on a Bayesian weighting algorithm [SKR+11]. In this approach, all the weights for faces in the dataset get updated based on the descriptive features and context aware extracted features of the images. A graph-based algorithm was proposed for labeling two faces in a single image based on the relationship between the group image [CHL12]. To understand the network between faces, the algorithm creates graphs and subgraphs from a dataset of group images which is called Bag of Face sub Graph (BoFG) and is based on the co-occurrence of faces in different images. To train the BoFG a Naive Bayes classifier is implemented. The results demonstrated an improved performance, compared to other models that utilize image pixel features for performing the face recognition.

In another work on social network images, a re-ranking algorithm was proposed. This algorithm uses context based rules to enhance the classification performance of any classifier [BVS14]. Rule mining is used for deriving associations between faces in group images. Multiple rules are produced and utilized to obtain context based weights. To train the classifier these weights were combined with the normalized weights obtained from the classifier to re-rank the weights. In another work, a model was proposed to incorporate album based costs in a recognition framework [HHM+14]. In order to include contextual information obtained from image albums, personal and social costs are considered in the optimization of a structural Support Vector Machine (SVM).

Another model to update the rankings obtained from an existing face recognition system was proposed in which a social graph (created from training images) is utilized [BGSV15]. Each node represents a subject in order to learn the contextual information between the subjects. For a given group image, in order to perform context-aided face recognition the face recognition scores obtained from a traditional face recognition system are combined with those obtained from the social graph. Recently, a model was introduced for utilizing multi-level contextual information at the face, image, and group image levels [LBL+16]. At the face level, the algorithm employs clothes information and body contextual information, while in images of

groups, a joint distribution of identities as well as contextual information is used to guide the face recognition. The proposed model presents a framework consisting of SVMs and Conditional Random Fields (CRF) to combine the aforementioned levels of contextual information in the recognition system.

Kinship verification scores was incorporated as contextual information in the face recognition system [KVS+17]. In the first step of the proposed model, a deep learning algorithm was employed for kinship verification. In the next step, to train the system an SVM classifier was used based on a score-level fusion with face verification and the result of likelihood ratio. A multi-model system was proposed based on face and ear recognition which was incorporated with social behavioral information extracted from virtual social networks [SPG17]. The scores of face and ear contextual features were fused at the score-level to recognize faces. This method uses no non-face pixel features, although it is stated that the combination of contextual information and content-based models is favorable.

In one of the most recent work, context information was merged into a classifier ensemble for face recognition or continuous authentication [SRSZ18, NBNF17, LLZ+17, SRSZ18]. In another recent work, a Siamese Convolutional Neural Network (SCNN) was proposed which used contextual information of face images such as yaw, pitch, and face size to improve the face recognition rate [STSG18]. As it is apparent in previous works, the definition and utilization approaches of contextual information varies across the spectrum of research. For instance, the early stages of this research focused on incorporating non-face pixel information in the face recognition, while the later stages focused on utilizing virtual social network graphs.

In Chapter 4 of this dissertation we describe contextual information definition. Temporal information, such as the time the image is taken, remains important for classifying the events images. With the development of virtual social network, and proliferation of related contextual information through smart devices, most of the recently proposed models focused primarily on virtual social networks to enhance the recognition rate. A combination of contextual information derived from virtual social networks and traditional approaches could further improve recognition performance and response time.

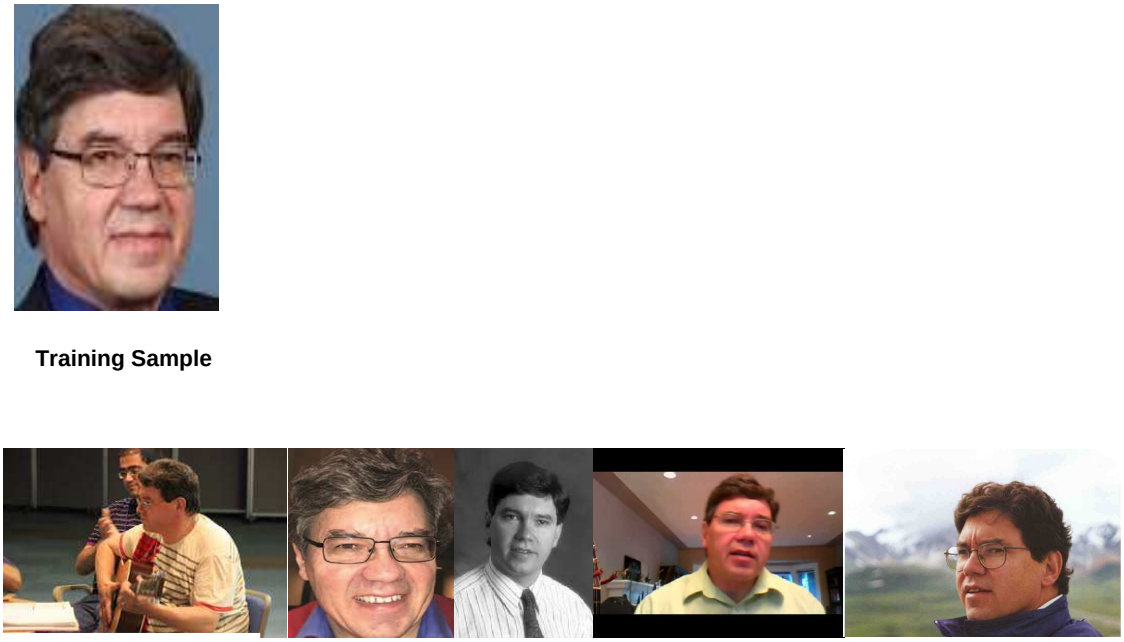

Figure 2.3: Need for adaptive face recognition systems due to changes in age, makeup, face view and facial expression

2.4 Adaptive Face Recognition System

Face recognition is still an open and challenging problem in computer vision, as there are insufficient training face samples for individuals and vast variability in the number of face testing samples. Furthermore, as Figure 2.3 shows aging and change of face makeup can affect face features. Additionally, inconsistency in capturing testing face images cannot effectively be considered in the training process. [HA15]. To address all these variabilities in the training and testing processes, self-adaptive system can be engaged effectively in promising solutions [AAEF15]. Self-adaptive systems adapt themselves with changes from training to testing processes. Self-adaptive systems can be treated independently as traditional face recognition systems are not capable of addressing changes in an individual's face samples. Here, we describe the needs for self-adaptability in face recognition systems in terms of level of adaptation.

In spite of continued progress in face recognition in recent decades, 100% face recognition is not achievable due to the nature of the technology [MCRA18b, DD18]. The major barrier to achieve perfect in face recognition's accuracy rate is pertinent to

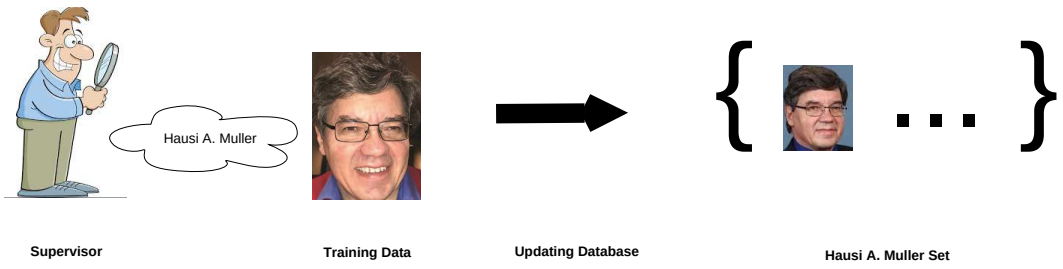

Figure 2.4: Supervised adaptive face recognition scheme in which the training face images are labeled by the supervisor

the limited number of training face images as well as the existence of significant face image variations in the testing process [HBRZ19, MCRA18a, OMR18]. Moreover, the face recognition system is expected to test face images captured by different devices that could result in substantial differences in image quality [PLdC18]. Considering all the aforementioned variations, a traditional face recognition system cannot remain consistent in performance over the time without retraining itself. Several solutions have been proposed so far to compensate the face variation's impact such as using computer graphic algorithms to simulate age relevant changes and most recently, engaging adaptability in face recognition systems. In face recognition, multiple methods have been developed to handle variability in lighting [DHWG17], illumination [YBR06, SS15], head pose [VSLC+17], face expressions [WRKN16] and the presence of glasses and sunglasses [KR16]. However, trying to compensate all these variations concurrently results in more issues in false positive and true negative face recognition rate. Computer graphic methods have also been used to generate a unique look of an individual face image. For example, simulating the effect of ageing in individual faces can be used to compensate variance in age related changes [PGDCL16, PPdCL17].

In addition, the graphic algorithm of face image generation often depends on the pre-processing that may be susceptible to estimation error if it does not get corrected artificially. Adaptive face recognition has recently been proposed as a solution to track the changes and to train the system with individuals face images variation. This can be done by supervising the individual face template [AAR+16] or model [DGM+16] to get updated, using operational data. The adaptive face recognition system has an extra module than a traditional system which is called adaptation module or updating module [Rat15]. The individual face reference will be generated every time that the update process is invoked. Therefore, a reference management

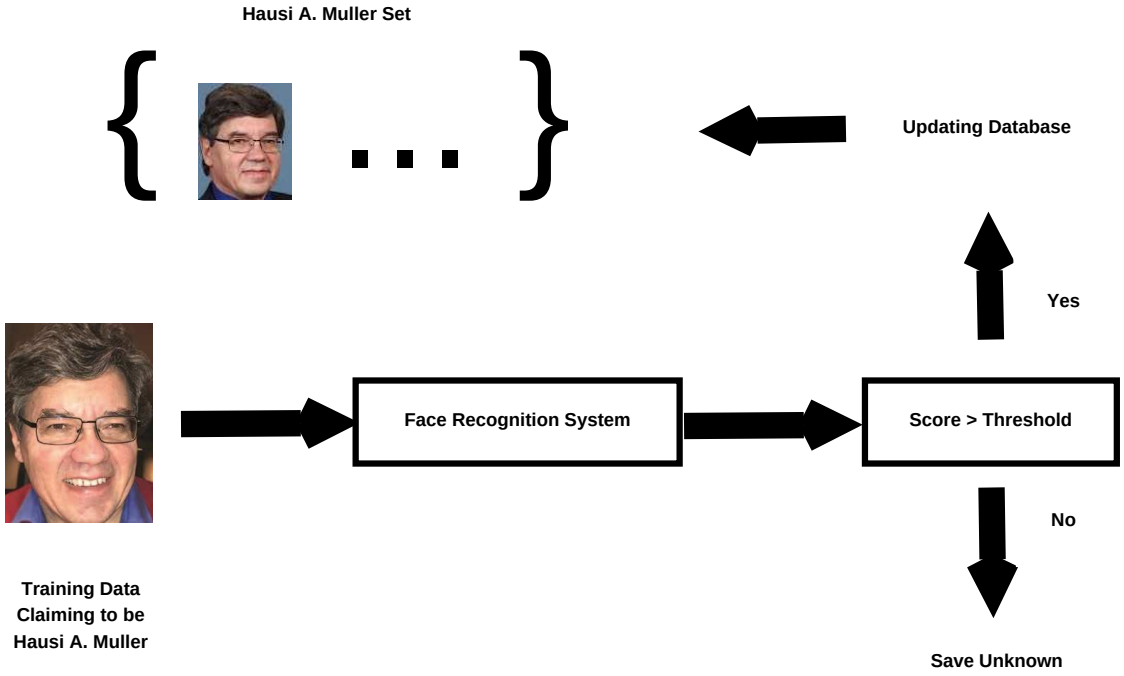

Figure 2.5: Self-adaptive face recognition scheme in which the face recognition system adapts itself

approach is essential. The supper template approach can be used to maintain only a single large common reference that embeds all the information about face [DPBB14]. Model based approach is an alternative to update the existing reference by replacing or appending the recently acquired face image to the template face image set.

2.4.1 Supervised vs. Unsupervised Adaptive Face Recognition Systems

In the supervised adaptation method, a supervisor can name the individual faces [GMY17]. Conversely, in an unsupervised method, the system names the individual faces (whether being a match or not). The system attempts to infer the name and only those faces whose names are inferred confidently are used to adapt the reference [SLZ+17]. Supervised adaptation can obviously achieve better performance than that of unsupervised adaptation. Figures 2.4 and 2.5 demonstrate an example of supervised and unsupervised adaptation.

2.4.2 Self-training vs. Co-training in Adaptive Face Recognition Systems

In the semi-supervised adaptation method, the system infers the face name from the operational data with engaging self-training and co-training [DGH+18, PKR15] as two main methods. In self-training, the algorithm applies highly confidently labeled input face images to update the reference [LCPM+19, TvSA17]. A label propagation scheme may be used in an offline self-training method to determine whether or not the operational data should be used for adaptation [OOF+16]. In co-training, the mutual and complementary help of two or more biometrics individual data is used to adapt the reference. The input sample can better be captured with a substantial variation in co-training method, thus, leading to better performance than self-training [MMX+17].

2.4.3 Image-based vs. Video-based Adaptive Face Recognition Systems

The type of input individual face data used in adaptive systems plays an important role in adaptation. Specifically, in a video-based face recognition, the face name can be applied for the entire video as the face is being sampled remains unchanged for the entire video frames. This is due to the fact that each consecutive pair of images in the frames are apart from each other by a fraction of the second. This is the identity constancy property [TMMR15]. In contrast, the face name cannot be applied from one face to another in case of the images that are acquired from a camera as there is only one image captured in a single acquisition session. Thus, this is a more challenging problem as the identity constancy property cannot be applied.

2.4.4 Level of Adaptivity

The adaptation process can take place at the score or decision level simultaneously with the adaptation at the template or model level. In one proposed adaptive face recognition approach, the final accept/reject decision is rendered independently by exploiting the face image quality to adapt the matching score [RMR11]. The reason of using adaptation at score-level is related to the fact that the distributions of match (genuine) and non-match scores are dependent on the condition of acquisition. Con-

versely, adaptation at the decision level adapts the decision module to the changing conditions.

2.4.5 Online-adaptivity vs. Offline-adaptivity in Face Recognition Systems

Choosing the appropriate adaptation method is influenced by hardware limitations. For instance, an online adaptation method is more appropriate when there is insufficient memory size available. In this case the system updates its parameter when an input face has been successfully authenticated and deemed suitable for adaptation (online) [AAR+16, ZBL17]. In contrast, if sufficient computational memory is available, the offline adaptation can be used in which case the adaptation process can wait until a certain time or when the buffer memory is fully accompanied [DlTGSG15].

2.5 Chapter Summary

In this chapter we introduced related works and background materials essential to understanding our research. We covered recent advances in face recognition in Sections 2.1 and 2.3, and context aware object recognition in Section 2.2. This background information is helpful in understanding and solving the problems we are addressing in our research. Section 2.4 showed different levels of adaptivity and also highlights some of the main features of adaptive face recognition systems as used in face recognition systems. The next chapter presents our approach to characterize and design of a location based face recognition system that will be used in our context aware face recognition system.

Chapter 3

Location-based Face Recognition Approach

This chapter presents the first contribution of this dissertation, the characterization and the architectural design of a location-based face recognition system. There are two research questions associated with this contribution:

RQ1. *How to form the location categories? How to take location information into account in the feature extraction processes? How to search efficiently to recognize faces? How to advance the recognition steps, and minimize response times using the features of Future Internet nodes such as the SAVI network?*

RQ3. *How can a web based face recognition system exploit contextual information accurately, efficiently and adaptively?*

To answer these research questions, we concentrate on the location information coming from smart mobile devices and designed as an attribute for a face recognition system.

We focus on two neglected information components of face recognition systems:

(1) *location information* as an attribute, where facial and location attributes are important in face recognition, thus defining a hybrid learning algorithm for these attributes; as well as searching and matching methods for maintaining the system under changing conditions; and

(2) *social network information* for when the system recognizes a face appearing in a particular location, social friends' images are used to train the classifier for the location.

This chapter is organized as follows. Section 3.1 presents the location-based face recognition system as well as the location-based recognition method and algorithm. Section 3.3 explains the experimental setup and highlights selected accuracy critical parameters in the classification stage. Section 3.4 reports on the evaluation and results. Finally, Section 3.5 summarizes this chapter.

3.1 Improving Face Recognition with Location Information

Humans recognize faces in social interactions based on environment and social parameters subconsciously. The relevant context includes information related to the image of the scene surrounding a person. This includes camera context such as location information and image capture time as well as the social context that describes the interactions between people and helps us identify faces in daily social interactions. Without context, even humans may fail to recognize the observed face. With the proliferation of smart mobile devices, such as phones and tablets, people are increasingly instrumented. These devices can track locations and moving directions of their users. Furthermore, many of us use them as replacement for more sophisticated cameras because smart mobiles are easy to carry and convenient to use.

A study[1] showed that from the photos taken in 2017 smartphones took 85%, tablets 4.7%, and regular cameras 10.3%. In Yahoo's Flickr, a place where people store and share their digital photographs, iPhone was responsible for more photos posted to this site than any other device in 2017.[2] Also, growing online mobile photo-sharing services, such as Instagram,[3] enables its users to take pictures and videos, and share them on a variety of social networking platforms using mobile devices. Photos taken with these smart devices include sensor-based context information, such as location, time, humidity, temperature, and acceleration pertinent to the moment when the photo was captured. The annotated measures of, smart mobile device can be exploited to sensors aid face recognition tasks.

[1]Smartphones Cause Photography Boom, https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/, retrieved December 2018.

[2]Top camera used by the flicker community in 2017 https://expandedramblings.com/index.php/flickr-stats/ (May 2018), retrieved December 2018.

[3]https://en.wikipedia.org/wiki/Instagram

This chapter concentrates on location information amongst all the available context information, and investigates how to exploit location information in face recognition problems. Our approach here is to reduce the search space of the recognition problem by discernment of the user location. If a user takes photos at a certain location, there is a high chance that the user will be in the image which is taken at this location. Consequently, when we want to recognize a face, we can save some effort and gain accuracy by first comparing faces that are in the photos taken at locations where that face normally appears.

In our approach for location-based face recognition, we present the first treatment of location information as a facial attribute and design a hybrid face recognition algorithm. This approach includes a search and match method. If a search fails using the current location, then it will search over an extended database. Also we show that this approach is surprisingly accurate when taking the social network information into account. When we recognize that a user appears in a particular location, her social friends will likely show up in that location too. Social friends' images are used to train the face classifier for each location.

3.1.1 Location-based Face Recognition

Here we introduce the location-based face recognition system and provide a detailed description of the method for using location information within our algorithm. Our approach assumes location-centric image databases to recognize faces in images that have been taken at or nearby locations frequently visited by these individuals.

This face recognition problem is defined as follows: Given a set of known images of faces for training and a set of the same set of people as a testing set, and recognizing each face in the testing set. In this system, each face image associates with the location information. The system creates many clusters of locations from the training set. Each location cluster contains a set of users with associated images in that location and their friends' images. The user can take an image and attach the location information, then send it to the system and query for the face in the photo. The system will answer the recognition question and return the identification to the faces in the image.

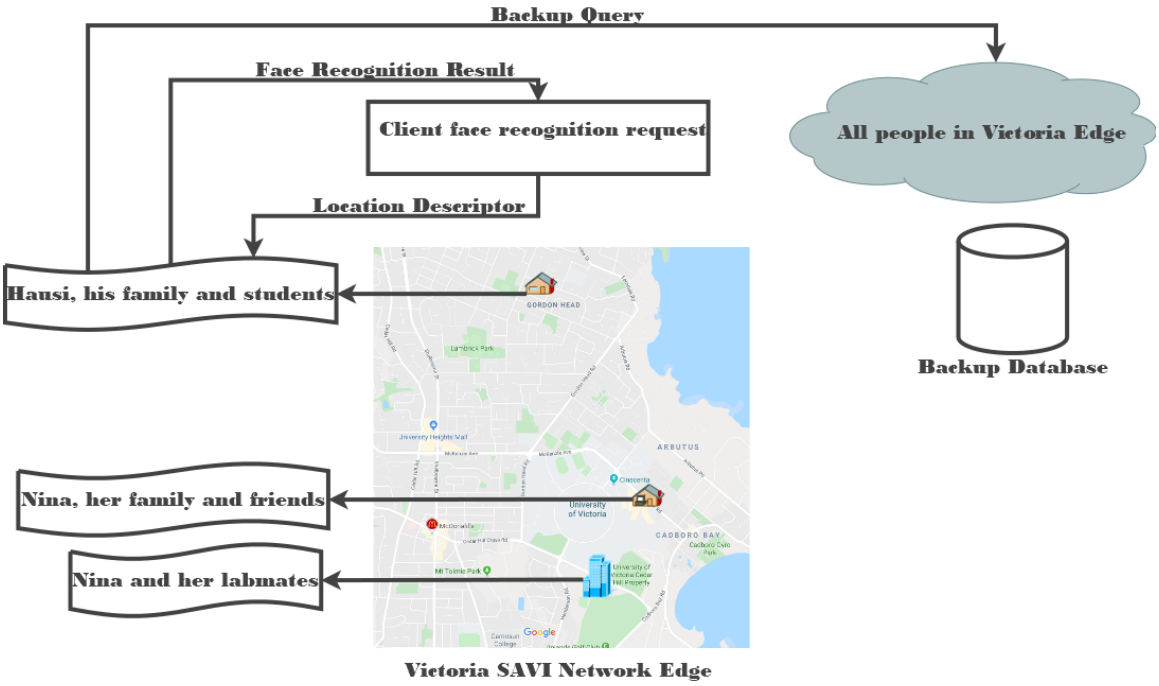

Figure 3.1: Schematic representation of our location-based face recognition approach

3.1.2 SAVI Network Smart Edges

Figure 3.1 illustrates our location-based face recognition approach. Using a smart mobile device the user sends images and associated recognition queries to SAVI network edges for processing.[4] The feature extraction stage is performed on the user smart mobile device. Extracted features and location information are sent to the SAVI server for face recognition. On the SAVI server, we categorize the location-based database.

The database at each location includes images of faces that have previously appeared at that location as well as social network friends' images. Because social friends are more likely to visit an individual in the database, we can insert images of those friends and associate them with the individual's location. We also maintain a backup database which represents a collection of all images in different databases on the SAVI network's edges as depicted in Figure 3.2. Should the system fail at any location database, we extend the search to the backup database to recognize the face image.

3.1.3 Categorizing Location Information

We have a database of training images based on location information which represent each location point with its longitude and latitude. Initially, the database contains n points corresponding to n different categories. At each step of database categorization, we merge the two categories if the distance between two categories is the minimum among all pairs of categories. The distance between two categories X and Y is defined in Equation 3.1:

$$d(X,Y) = \frac{1}{|X||Y|} = \sum_{x \in X} \sum_{y \in Y} d(x,y) \tag{3.1}$$

where $d(x,y)$ is the Euclidean distance between points x and y. We keep merging categories until the minimum distance in each iteration is above a threshold or the number of categories we want to obtain is reached. For this we used the real-time cluster moving object algorithm [LHY04].

[4] https://www.savinetwork.ca/

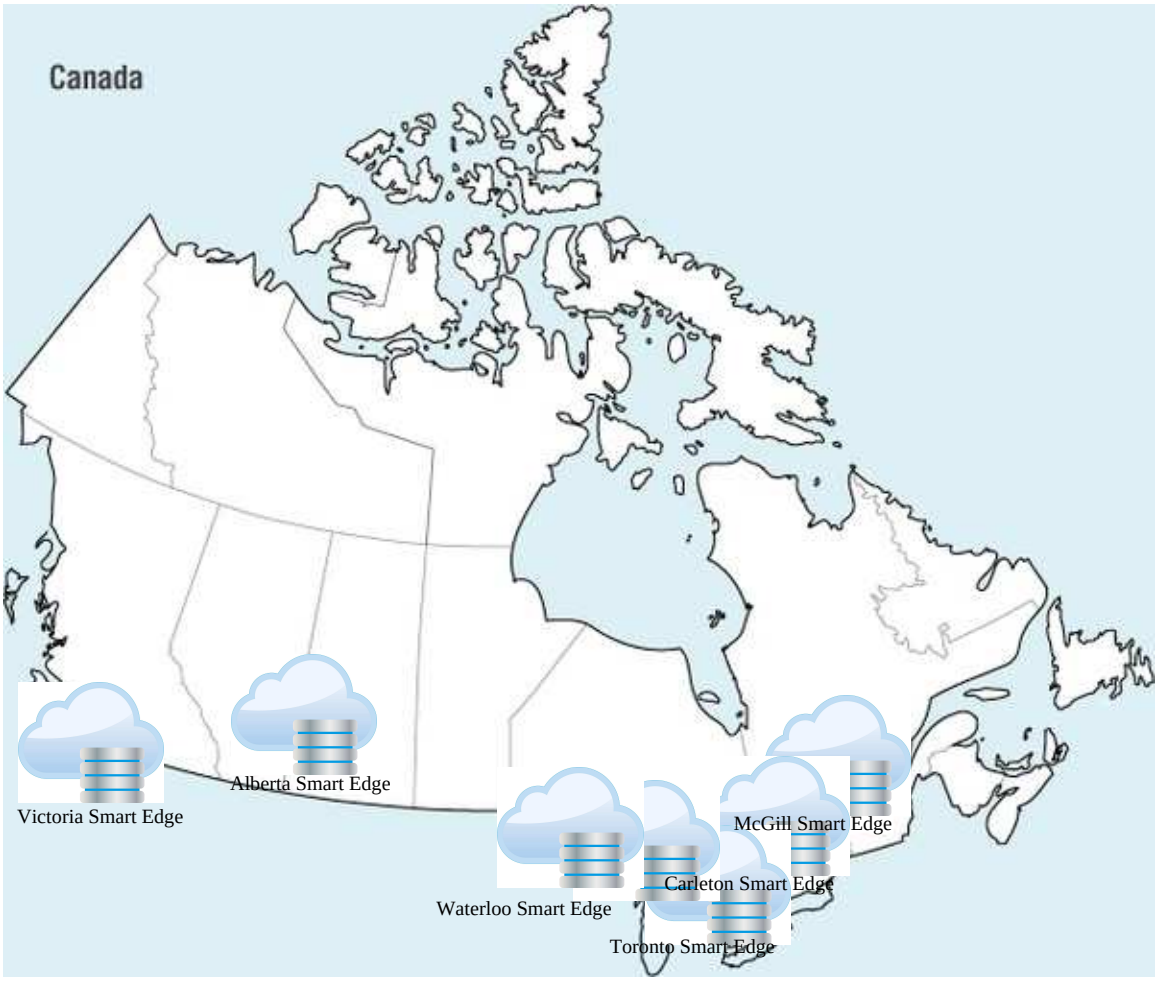

Figure 3.2: Backup databases on smart edges in the SAVI networks [LG]

3.1.4 Location-based Features Preparation

Facial features are associated with the location information for training the classifier at each SAVI network edge. We used the Viola-Jones face detection method which is an efficient face detection method and suitable for real-time applications to detect faces from user images [VJ04]. Detected faces are normalized to the same size. We employed the algorithm presented by Taherimakhsousi et al., to extract face features from each face image which is efficient and practical [TEH09]. Moreover, utilizing the method shown by Tan et al., we reduce the lighting effects on face features [TLLJ10].

3.2 Location-based Teacher-directed Learning

Figure 3.3 illustrates how we deployed the location-based face recognition system on the SAVI network which includes two parts, the smart mobile device user and the SAVI network edge. Face features are extracted from images and trained with the Mixture of Experts (MoE) classifiers as described in Appendix A using the multi-layer perceptron, MLP, and expert classifiers with teacher-directed learning on the SAVI network edge. Each location has its own MLP expert classifier. In the teacher-directed learning (TDL) method, teacher information is included in the training process [EKEY08]. We use TDL to specialize the experts in their corresponding locations, in a way that, according to the location of the input training image, only the weights of the corresponding expert(s) are updated. In the training phase, the weights of MLPs are learned using the error back-propagation (BP) algorithm. To apply TDL to the BP algorithm, updating of the weights in the learning process is controlled, such that only the weights of the expert having the input training features in its location are updated and the others are kept unchanged. For instance, for a training sample of $Location_1$, only the weights of experts responsible for $Location_1$ are updated (cf. Figure 3.4).

When face features are received on the SAVI network edge, it rates the location of the received face features and searches the closest expert classifier in the database on the SAVI network. The accuracy of each location-based expert shows how well the face image is classified by MoE. If the accuracy is below a certain threshold, we send that face feature to the backup database to classify over the backup database on the other SAVI network edges.

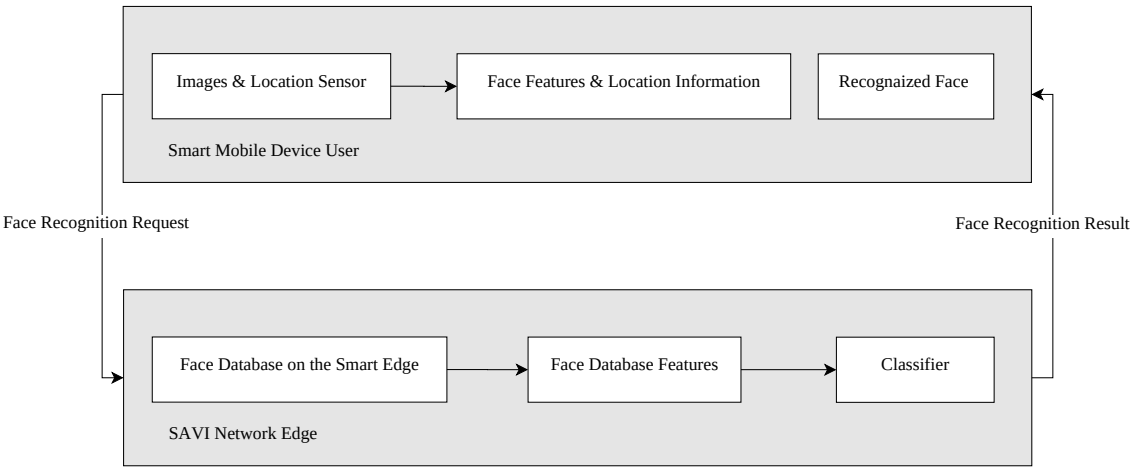

Figure 3.3: Our location-based face recognition system the smart mobile device user and the SAVI network edge

3.3 Experimental Setup for Location-based Face Recognition

This experiment investigates the performance of our model in recognizing faces with location information. In this experiment, the performance of our model is compared against those of MoEs without location attribute in which the experts and gating network receive a common input. We conducted a five-fold cross validation to test our location-based face recognition method. In each test, 80% of the images were used as the training set and 20% of the unseen images were used as the testing set.

The database that we used for this experiment contains 1000 images from 100 labeled faces. The dataset has nine locations. Our location-based face recognition system includes the labeled faces and the social network relations as the training set. Further, each location is associated with face images of friends from a user's social network. We also created a backup database containing images of those face images on the SAVI network servers, which is used when the local SAVI network edge fails to recognize the face based on recognition thresholds.

As shown in Figure 3.4, we formed one expert for each location. Therefore, the MoE used in this model had nine experts. To form eigenfaces, for instance for $Location_1$, we used 400 training images of these location and mapped them to the Principal Component Analysis (PCA) space [TP91b] of dimension 40. For the gating network, which mediates between these experts, we used a global eigenface of dimension 40.

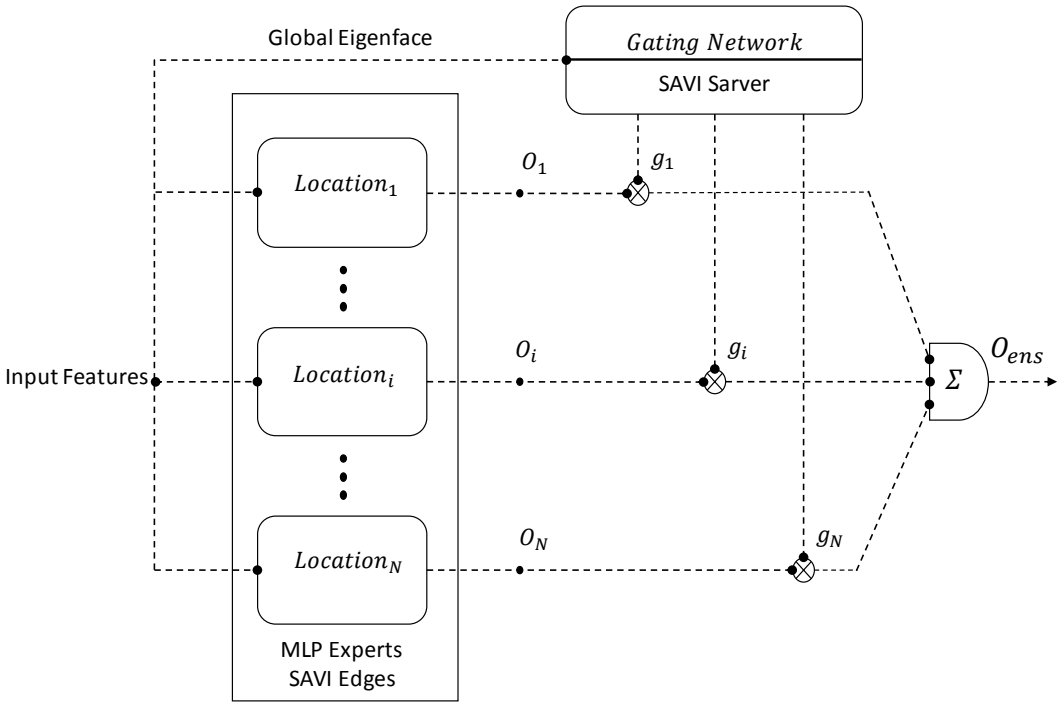

Figure 3.4: The sketch of our teacher-directed location-based learning method. Different from the conventional MoE, the experts receive input features from their corresponding location and the gating network, which is to mediate between the experts, and has global features in its input layer; as a result, each expert is specialized on a specific location.

Table 3.1: Location-based MoE classifier accuracy rates obtained through 5-fold cross validation at nine different locations

Locations	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**
L 1	0.889	0.901	0.891	0.75	0.846
L 2	0.891	0.923	0.933	0.941	0.877
L 3	0.965	0.938	0.942	0.968	0.989
L 4	0.687	0.736	0.745	0.684	0.692
L 5	1.000	1.000	0.975	1.000	1.000
L 6	0.766	0.741	0.795	0.689	0.690
L 7	0.625	0.632	0.687	0.692	0.627
L 8	0.535	0.516	0.576	0.594	0.581
L 9	0.715	0.735	0.749	0.754	0.752

3.4 Results and Discussion for Location-based Face Recognition

To evaluate the performance of our location-based model, we trained and tested two MoEs with global eigenfaces and location-based eigenfaces with and without TDL. In all experiments, the gating network had a global eigenface in its input layer. The validation results are presented in Table 3.1. The results in each row of Table 3.1 are the averages of 10 times testing the location-based model, each time trained with different initial random weights. The MLPs of all experts had 40 input nodes for PCA components and 10 output nodes for location attributes.

These models were implemented and tested on the dataset under the same condition as the experiment with the intermediate unseen images. The results are tabulated in Table 3.2. We can clearly observe that the recognition rate of the location-based models are higher than the MoE method without location information. The recognition rate of the location-based with teacher-directed learning method is 25% higher than that of the MoE method without location information.

The time complexity of the recognition algorithm is $O(\log k)$, where k is the number of locations in the our location-based face recognition system. The time complexity of recognizing the MoE is $O(n)$, where n is the number of known faces at each location-based database. The search space requirement of MoE only depends on the number of known faces in each database, but not on the number of training set face images, which reduces the search space when the training set is large.

In addition to face recognition based on social networks, we could recognize faces in photo albums. We believe that many of the observations also hold when using different recognition algorithms. It would be useful to investigate whether the insights gained from these experiments can be generalized to other face recognition approaches. In particular, it would be interesting to see how well these algorithms perform on an entire social network. We expect that the results of such a comparison will add to a scalable and accurate approach to recognize faces from on-line social networks.

Table 3.2: Recognition rate comparison between our location-based method and two of the most closely methods implemented and tested on our dataset

Method	*Recognition Rate (%)*
MoE location-based eigenface with TDL	88.5
MoE location-based eigenface without TDL	79.3
MoE global eigenface	63.5

3.5 Chapter Summary

This chapter presented the first main contribution of this dissertation: our location-based face recognition system for smart mobile devices answering research questions RQ1 and RQ3 (cf. Figure 1.2).

In Section 3.1, we introduced a location-based face recognition system on SAVI network smart edges. moreover, we presented facial and location feature extraction and preparation. In Section 3.2 we disused a teacher-directed learning method based on location attributes with MoE classifiers using MLP experts with TDL to specialize classifiers in their corresponding locations on SAVI network edges.

In Section 3.3, we presented the experimental setup for investigating the performance of our location-based face recognition method. We discussed the results in Section 3.4. We showed how our method reduces search space when the training set is large.

The next chapter defines the context, categorizes contextual information, and describes an architectural design for context aware face recognition systems and their smart applications. This architectural design is based on contextual information that assists face recognition systems to act smarter.

Chapter 4

Context Definition and Smart Applications for Face Recognition

This chapter presents our definition and characterization design for the context aware face recognition systems and their smart applications, our second contribution of this dissertation.

In Chapter 2, we presented the state-of-the-art-face recognition systems in which researchers focused on measuring the similarity between face images and identification based on only face pixels. However, we discussed that there is a need for smart face recognition applications that can exploit and understand the context of images.

There are two research questions associated with this contribution:

RQ2. *How does the use of contextual information impact face recognition performance? How does selected types of contextual information affect the face recognition performance for different scenarios? Is a certain type of context more effective than others for certain scenarios?*

RQ4. *What is the effect of contextual information on human face recognition? How does contextual information affect human face recognition accuracy and response time?*

To answer these research questions, we concentrate on the contextual information that a human uses to recognize faces and categorize them, specifically contextual information that assists smart applications in the face recognition process.

This chapter is divided into four parts: First, we describe the need of contextual information to improve face recognition process in Section 4.1 which is the conceptual

foundation of this chapter and the starting point towards the context aware face recognition system. Second, we introduce our definition of context for the use in face recognition systems in four categories: face context 4.2.1, pixel context 4.2.2, sensor context 4.2.3, and social context 4.2.4 in Section 4.2. Third, we demonstrate how context aware face recognition helps applications become smarter in Section 4.3. Finally, we summarize this chapter in Section 4.4.

4.1 Improving Face Recognition Process with Contextual Information

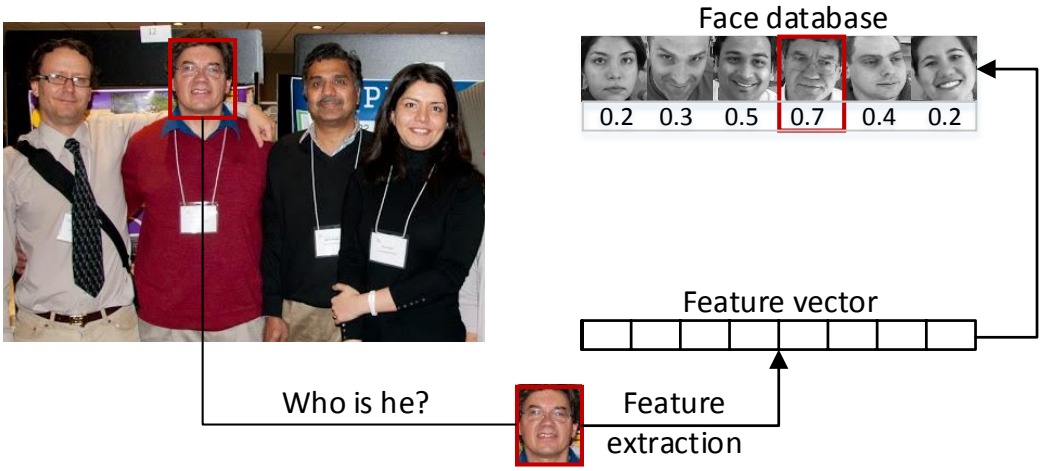

Figure 4.1: The face of interest is cropped and features are extracted from the face pixels and compared to each face in the database for identification

As illustrated in Figure 4.1, a face of unknown identity is compared against a database of face images with known identity, where each database image is captured with similar pose, illumination and expression. There are significant differences between the technical challenge of face recognition in general and the problem we are addressing in this thesis. Intuitively, recognition of the man wearing red would be aided by knowing the identity of his colleagues, the location of image capture, the image capture date, the fact that he wears a conference badge, and any other available contextual information. In contrast to a face only approach, the human visual

system brings with it a wealth of rich contextual information to recognize faces in images; faces are just one component that humans use for recognizing faces.

Figure 4.2: A)faces are embedded in the context of location, clothing, gender, and height. B) a set of faces without contextual information rather than face pixels

Figure 4.2 shows the limitations of using only facial features for recognizing people. When five faces are cropped and scaled from this image in the same fashion as images from the simplified classic databases such as PIE database [SBB02], it is difficult to determine how many different faces are actually present. Even if it was known that there are five different individuals, the problem would not be much easier [SKP15]. In fact, the five from the same family.[1] When the faces are shown in context with their clothing, gender, and height it becomes almost trivial to recognize which images belong to which person.

[1]Family of Same faces! An online photo where the photographer's name is unknown. Web 11 Jan 2017, http://www.funnyjunk.com

4.2 Context Definition in Face Recognition Systems

The contextual information that a human uses to recognize faces can often be modeled by using inference over image data, data about data, and large amounts of statistical data which model human interactions and social network context. Probabilistic models and machine learning are used to integrate contextual information into the interpretation of faces in images. Context is broadly defined as information relevant to something under consideration [AAC16]. Context can include information from non-face regions of the image, information related to the capture of the image, or the social network context of the interactions between people. Applications for these ideas include:

- Recognizing faces from descriptions, names.
- Finding plausible names and demographic data for any person in an image database.
- Understanding the relationships between people.
- Recognizing faces from a small number of labeled examples.
- Plausible naming for faces in large scale applications.

Table 4.1 shows examples of context that are considered in the research.

4.2.1 Face Context

Current methods generally focus on measuring the similarity between face images which represent human faces semantically via facial components such as eyes, mouth and face outline. The face shape models are not based on statistical learning and it still could not deal with faces of large variation in appearance due to pose, rotation, illumination and background changes.

4.2.2 Pixel Context

Pixel context information such as distinctive clothing or glasses can be useful for recognizing faces in images. Further, because people tend to appear in images with

friends and family, the identities of other people in an image aid our recognition of a person of interest. Even the position of a person in the image is important. For example, babies are often held by another person when photographed.

4.2.3 Sensor Context

Simply knowing the capture conditions of an image can help identifying people in the image. The image capture time is particularly relevant, as it allows us to group multiple images in the collection captured at the same event into clusters. Within an event, it is likely that a given person will maintain a constant appearance and wear the same clothing. The geographic location of the image capture is intuitively useful for determining the identities of people in the image.

4.2.4 Social Context

Social context refers to information about people and their society that is useful for understanding images. For example, because specific first names rise and fall in popularity over time and are selected based on the gender, culture, or location of the child, a first name provides prior information about the age, gender and origin of a person.[2]

When multiple people appear in an image, their social relationships are related to their age, gender, and relative position within the image. The distributions of relative ages between spouses [NB05], parents and children [MHS+07], and siblings [CMM+05] are either documented in, or can be estimated from demographic statistics. A standard actuarial table allows us to consider life expectancy as a contextual information.[3]

Of course, contextual clues are interrelated and each one is only known with some degree of certainty. For example, knowing the first name of a face provides some information about the age and gender of the person. Likewise, if the age and gender are known, the uncertainty about the person's name decreases. We will use

[2]U.S. Social Security Administration. Baby name database. http://www.socialsecurity.gov/OACT/babynames.

[3]NCHS Analytic. reporting guidelines: The national health and nutrition examination survey (nhanes). Technical report, National Center for Health Statistics, Centers for Disease Control and Prevention, Hyattsville, Maryland. URL. 2006. http://www.cdc.gov/nchs/data/nhanes/nhanes0304/nhanes analytic guidelines, 2012.

probabilistic graph models to represent this uncertainty and allow all evidence to be considered.

Table 4.1: Different types of contextual information useful for recognizing faces in an image

Face Context	Pixel Context	Sensor Context	Social Context
Distance between eyes	Clothing	Flash illumination	Social relationship
Skin color	Relative pose	Brightness	First name
Anthropometric measurements	Posture	Location	Height
Distance measurement	Glasses	Time	Ethnicity
Midline of eyes	Hats		Age
Eye localization	Other faces		Gender

4.3 Smart Applications for Context Aware Face Recognition

This section presents how applications can become smarter using context aware face recognition.

4.3.1 Personalized Web Tasking

With the scope of personalized web tasking, we automate smart web tasks based on context aware face recognition and introduce innovative applications of context aware face recognition for smart web tasks. In other words, context aware face recognition system provides assistance to the user for automating web tasks [TM14]. For example, personalized web-tasking objectives aim to improve the user's experience by automating repetitive and ordinary tasks to accomplish web tasks [VM13]. With the proliferation of digital cameras, personal websites, and social networking, there is a growing need to analyze images and videos [WL13].

Automation in personalized web-tasking relies on the user's personal context (i.e., interests, preferences, and personal information). By identifying the user in an image or video during web tasking, we can extract personal context. Moreover, using existing context for a relevant image or video (i.e., location, time, non-face pixel information, and social information) will involve further face recognition.

4.3.2 Adaptive Environments

To build adaptive environments, context aware face recognition can be combined with speech and standard interactions like mouse movements and keystrokes by detecting the user's affective states [CSCG16]. Also, context aware face recognition can build and used for social aware systems in a same way [JS15].

4.3.3 Gaming

Recently, it can be observed that there is a shift in focus towards the design of video games for individuals, so as to increase perceived value [CK19]. Indeed, a benefit of such player centred design is that it ideally results in enhanced gameplay experience for players regardless of their gender, age or experience [BWL+14]. Here is the place that gaming experience for individuals can be improved by context aware

face recognition of characters or mission according to the player's emotional responses [MB14, BMO+15].

4.3.4 Commercial Video Chat

Commercial video chat is now commonplace in people's lives for a combination of work and personal needs. However, the functionality and user experience that existing commercial video chat can provide are still rather limited and sometimes unsatisfying. For instance, a user may be interested in gleaning information about the people in the audience of a video conferencing session [BTJ+13] so that the speaker can make her live presentation more engaging if she knows names and information of people in the audience.

4.3.5 Web-based Class Environment

Consider a web-based class environment, such as a Massive Open Online Course (MOOC) [Kop11], with an online community of students, teaching assistants, and professors. The most important thing that helps students succeed in an online course is interpersonal interaction and support [AS16]. Detecting students' frustration by context aware face recognition system can help improve e-learning experience.

During the course of a lecture, a MOOC instructor performs many web tasks that require interactions with students including answering questions. The instructor-student interaction can be greatly eased and facilitated with automated face recognition given a context aware face recognition database for the entire MOOC course.

4.3.6 Personal Media Management

Cloud services for storing mobile users' photos and videos promise unprecedented ease of access and an enhanced overall user experience. Advanced mobile networks facilitate media sharing and today's wide array of mobile devices provides high-quality creation and consumption of media [VR14].

However, as users push increasing amounts of media into the cloud, the problem of organizing and managing the content becomes a burden. For example, a user may wish to find photos of a friend or relative, but without smart organization, she must scroll through pages of thumbnails looking for the desired images.

Existing solutions for personal media management leverage basic context to provide limited search and filter functionality. EXIF image data may include the date and time of the image and in some cases the GPS coordinates indicating the location. Other EXIF information, such as the camera type and exposure setting, is suitable for other applications, but does not provide much value for the search problem [RCR15]. Another source of context that is potentially helpful is user annotations. Users may tag a photo or video, or arrange content into directory structures with a meaningful naming convention. However, this approach is time-consuming and particularly challenging on mobile devices without keyboards or drag-and-drop capabilities. Given the effort required, it is typically only used by small groups of users, such as professional photographers or enthusiasts.

Content processing algorithms that generate fully-automated high-level descriptions of videos and photos promise to be useful, but there are few successful implementations in this area and the current state-of-the-art-face leaves room for improvement. Fortunately, content processing to assist users with a semi-automated process of tagging and organizing content is a good compromise [BZL+13]. Advances in context aware face recognition can generate sufficient information to be a viable part of a user content management solution.

4.4 Chapter Summary

This chapter constitutes the second contribution of this dissertation. It allows us to answer research questions RQ2 and RQ4 (cf. Figure 1.2). In Section 4.1, we described improving face recognition process by contextual information. In Section 4.2 we introduced our definition of context for using in face recognition systems by dividing contextual information to four categories: (1) face context which has contextual information of face components such as: eyes, mouth, and face outline, (2) pixel context which has contextual information of other part of image such as distinctive clothing, glasses or the identities of other people in an image, (3) sensor context which has contextual information of capture conditions of an image such as location and time, and (4) social context which has contextual information about people and their virtual society that is useful for understanding images. In Section 4.3 we introduced some applications of context aware face recognition system and described how context aware face recognition system can help them become smarter.

The next chapter pretenses the design of a contextual information extraction algorithm and context aware database creation method which is the main element of adaptability. As same as our architectural design of decentralizes cloud computing on SAVI network infrastructure to support self-learning and adaptability on real-time context aware face recognition system.

Chapter 5

Automatic Context Extraction and Decentralized Cloud Computing on SAVI Network for Context Aware Real-time Video Analytics

This chapter presents our design of a contextual information extraction algorithm and decentralize cloud computing on SAVI network for Context Aware Real-time Video Analytics (CAVA), which is our third contribution of this dissertation. In Chapter 4, we presented the definition of contextual information for us in face recognition systems, which contextual information is the main key of context aware face recognition systems. In this vision, there is a need for context aware feature extraction and self adaptive learning.

There is three research question associated with this contribution:

RQ1. *How to form the location categories effectively? How to take location information into account in the feature extraction processes? How to search efficiently to recognize the faces? How to advance the recognition steps, and minimize response times by taking advantage of Future Internet nodes such as the SAVI network?*

RQ2. *How do the use of contextual information impact face recognition performance? How does selected types of contextual information affect the face recognition performance for different scenarios? Is a certain type of context more effective than others for certain scenarios?*

RQ3. *How can a web scale face recognition system exploit contextual information accurately, efficiently and adaptively with millions of web users and billions of photos?*

To answer these questions first we concentrate on feature extraction, specifically features that assist our face recognition in the achievement of system goals. We implement two levels of context aware filters: (1) context aware filters [YBR06] are initially applied with low computational complexity on a subset of the selected video frames, then (2) more complex context aware filters are applied which extract features and relevant contexts, in order to increase face recognition accuracy. After we extracted features the detected faces are normalized to the same size, and finally the detected faces are automatically added to the cloud base database along with contextual information which is the main element of adaptability.

Then, we define our architecture for CAVA as follows, which decentralizes cloud computing [Sat17] on SAVI network infrastructure: (1) a video from an individual mobile device travels as far as its currently associated SAVI node, (2) our self-adaptive face recognition algorithms run on SAVI node VM in near real time, (3) the Data Manager runs in an individual VM on the SAVI network to manage the storage of the videos and database with the associated contextual information, (4) the data is logically organized as a collection of videos, (5) results of the processing along with contextual information (such as the VM details, location, start time and video duration) are sent to the SAVI core, and (6) the labels and contextual information in the SAVI core can guide and facilitate deeper and more customized searches of the contents of a video during its retention period on a SAVI node's VM.

This chapter is organized as follows. Section 5.1 describes the SAVI-based architecture of CAVA. Section 5.2 introduces our feature extraction method which is the conceptual foundation of this chapter and the starting point towards the realization of adaptability in our system. Section 5.3 describes our extracted data management process. Section 5.4 refers to our adaptive labeling and learning method and Section 5.5 shows our two steps data searching based on extracted contextual information. Section 5.6 describes our experimental testbed on SAVI network. Sections 5.7 and 5.8 indicate how we evaluate CAVA and how CAVA is extremely computationally intensive. Finally, Section 5.9 summarizes the chapter.

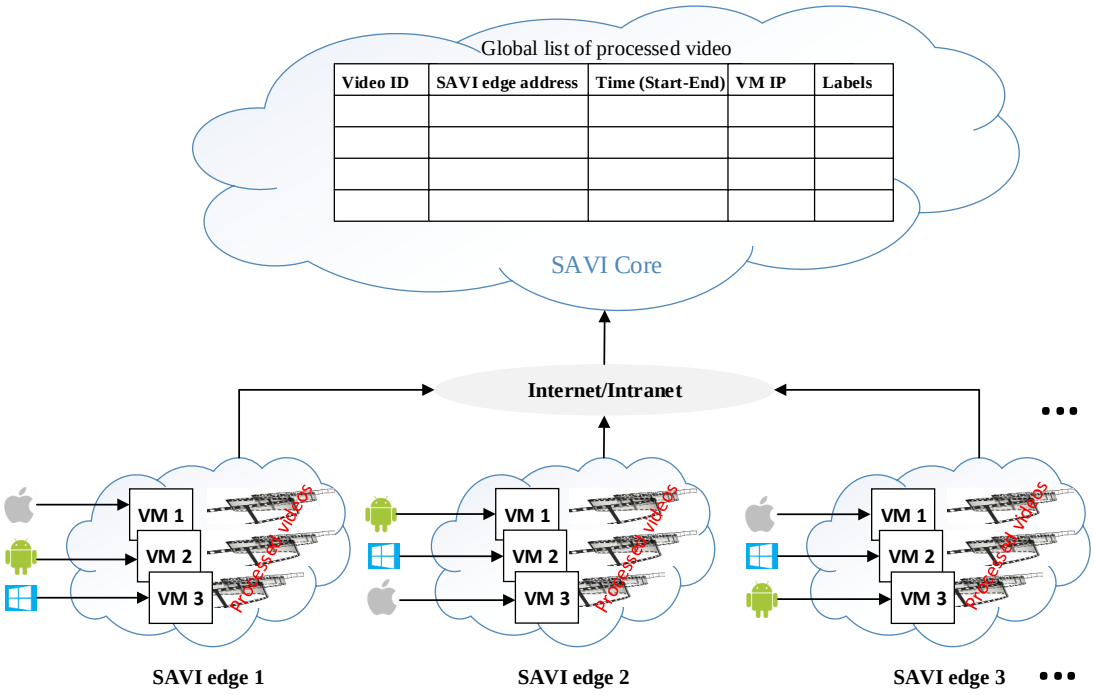

Figure 5.1: Our decentralized cloud computing SAVI network infrastructure

5.1 SAVI-based Architectural Design

A main challenge for the cloud is the large number of incoming videos from many cameras [PGGS17]. Every minute, three hundred hours of video are uploaded to YouTube[1]. Scaling well beyond this to millions of concurrent uploads from a dense urban area will be difficult [BBS+17]. To solve this problem, we design a SAVI network-based architecture that uses a decentralized cloud computing infrastructure in the form of SAVI edges (cf. Figure 5.1). A SAVI edge is an architectural element arising from the convergence of mobile computing and cloud computing. It represents in between the mobile device and the SAVI core. A SAVI edge can be viewed as a data center that brings the cloud closer [BTM15]. SAVI edges were originally created to address end-to-end latency in interactive applications and bandwidth considerations.

In our architecture a video from a mobile device only travels as far as its currently associated SAVI edge. Computer vision analytics run on SAVI edge VMs in near real time, and only the results (recognized faces and objects) along with contextual

[1]YouTube, http://www.kmol.cz/KMOL/www.youtube.com/yt/press/statistics.html, retrieved May 2019.

information (such as the VM details, location, start time and video duration) are sent to the SAVI core. The labels and contextual information in the SAVI core can guide deeper and more customized searches of the contents of a video segment during its retention period on a SAVI edge VM.

5.2 Context Aware Video Processing

The SAVI edge VM processes the video uploaded by the mobile device adhering to user privacy settings. It is the only component entity, apart from the mobile device itself, that accesses the original, non-processed video. As such, it forms the SAVI-based counterpart of the mobile device: an entity that the user trusts to store personal contents, but with much more computational and storage resources. When a user has a video ready for upload, the SAVI VM requests the data manager to allocate storage space for the processed video. The data manager, running as a SAVI edge-wide service in a separate VM, organizes the SAVI edge storage and contextual information database. Figure 5.2 shows how a user video is processed inside the SAVI VM before being stored on the SAVI edge.

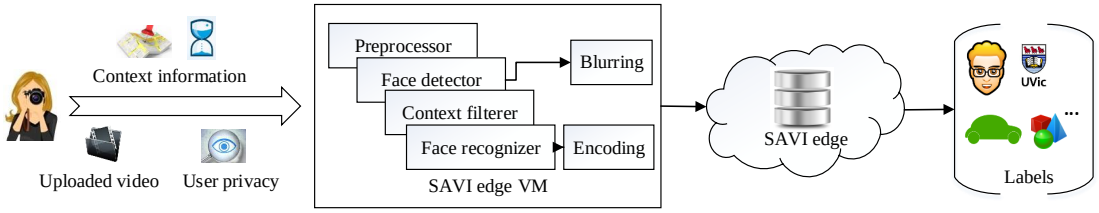

Figure 5.2: Overview of the CAVA

The video processing is implemented using OpenCV, C++ and python in multiple steps. Generally, the extraction of context features includes five following steps:

- Step 1: We select a subset of the video frames for actual processing. Video processing is too computationally complex to perform at the native video frame rate. We apply, context aware filters with low computational complexity [MSG17, XYW+16, MWY+16].
- Step 2: We apply Viola-Jones face detection method which is an efficient face detection method suitable for real-time applications [Wan14]. Detected faces are normalized to the same size in order to increase face recognition accuracy.

- Step 3: Then detected face will be blurred [LVH+17](i.e., the algorithm gives an array with the bounding boxes of the detected faces) or recognized based on the user's privacy settings as depicted in Figure 5.3.

- Step 4: We also employed the algorithm by Taigman et al. to extract face features which allow us to scale the training up to, the previously unexplored range of, hundreds of millions of training samples [TYRW].

- Step 5: The final step is to feed the context aware descriptors (including video pixel extracted context and associated contextual information) into self-adaptive classifiers. The classifiers for these algorithms are Mixture of Experts (MoE) classifiers (For more information about MoE see Chapter 3 and Chapter A).

5.3 Extracted Data Management

The Data Manager runs in an individual VM on the SAVI network. It manages the storage of the videos and the database with the associated contextual information. The data is logically organized as a collection of videos. We define a video as a record of the reality during a continuous period in time and location. Each video contains one or more streams, each representing a single informational dimension of the reality. Streams can be audio, video, GPS coordinates or any other sensor information captured.

All video and contextual information are stored on local disk storage in the SAVI edge. The contextual information describing the different streams is stored in a MySQL database and initially includes the video ID of the stream, capture time, duration, access control rights as well as a location, delineating the area in which the video has been captured. This contextual information will allow the users to restrict the scope of their search (e.g. by specifying in user query to return only videos captured in the downtown at noon). These labels describe the recognized faces and their positions in the video. Our main motivation stems from bandwidth considerations, as explained earlier. However, since VMs are located at the SAVI edge, all users connected to a specific edge will upload videos that are mostly captured in the same area of the SAVI edges. This characteristic may be used as a first-order heuristic to determine which SAVI edges to be searched when a user performs a query on the video search engine.

5.4 Video Context Labeling

The labeling of processed video contents is a background activity performed by the VM on the SAVI edge. In our current implementation, each processed video is analyzed individually by our method to obtain labels for that video. In a next version of our architecture, we will include the analysis of sequences of processed videos so that we can tag human-meaningful actions, such as running, or swimming, playing. For each labeled video, an entry is created in a dedicated label table of the SAVI data center database.

Each entry contains the label, the ID of the video segment and a confidence score. For example, an entry "old man, uni.mp4, 222, 86" indicates that our model labeled the video with 86% confidence an old man in frame 222 of the video uni.mp4. After label and feature extraction, these labels are also propagated to the collection of video segments in the SAVI core. As a proof of concept, we use a Python-based implementation of Taherimakhsousi et al.'s self-adaptive image categorization and classification algorithm [TM15b, HTME09]. This enables us to identify, with acceptable levels of false positives and false negatives, the 40 classes of common objects such as faces, cars, human, trees, inside and outside or logos.

An additional source of information about image (frame of video) contextual information can come from the preprocessing step. Since preprocessing is performed before labeling process, some important features of an image may not be available to downstream image processing algorithms. For example, a face recognition algorithm may not correctly label an image with faces because they are blurred. The obvious solution of labeling before processing is not acceptable because users expect their videos to be filtered first, before they are exposed to the SAVI edge video processing algorithm. We therefore allow the processing step to export labels that are discovered after filtering process. For example, the label "face" would have been exported as an attribute of a frame with faces. Thus, the labels for a video are the union of those obtained during filtering by the user's video processing within the SAVI edge VM, and those obtained during labeling by SAVI edge's video processing.

Rome, Itay
Roman Colosseum
Wearing white
5/9/2015

Victoria, Canada
Hausi A. Müller
University of Victoria
Wearing glasses
Wearing black suit
Wearing tie
5/1/2012

Figure 5.3: Two examples of processed videos

5.5 Context Aware Video Searching

Our designed architecture uses two steps to search a video based on context information. First, the user performs a conventional search on the SAVI core collection. The user's query includes context information such as time and location, as well as labels extracted by labeling. A production version of the our architecture would likely use a distributed cloud database, such as BigTable to ensure a scalable global service. The result of this step is a list of videos and their processed thumbnails. The IP addresses of the SAVI edges on which those videos are located can also be obtained from the collection. We therefore perform the next search step that filters the actual context to reduce the returned results to a more relevant set. This step is computationally intensive but can be run in parallel on SAVI edges.

We provide a suite of filters for common search attributes such as color and texture patches. For more complex image content such as objects and logos, the users can train own classifiers and insert them into the collection. Using a plugin interface, image processing code fragments called filters can be inserted into the result stream. These code fragments allow user-defined classifiers to examine video segments and to discard irrelevant parts of them, thus reducing the volume of data presented to the user.

To illustrate this context aware search, consider a search for "any videos taken on May 2015 between 11am and 4pm in Rome, showing a girl in the Roman Colosseum and wearing a white jacket." The first step of the search would use the time and location information and the face label to narrow the search.

The result is a potentially large set of thumbnails from processed videos that cover the specified location. From a multi-hour period of video capture by all visitors, this may narrow the search to a few hundred or few thousand thumbnails. Using a color filter tuned to white, followed by a composite color/texture, most of these thumbnails

may be discarded. Only the few thumbnails that pass this entire bank of filters are presented to the user. From this small set of thumbnails, it is easy for the user to pick the result (cf. Figure 5.3).

5.6 Experiment on SAVI Network Testbed

In exploring the limits to scaling the CAVA, we identified the bottlenecks in the system through experiments. After the description of our experimental testbed in section 5.7 we separately characterize the load incurred by video collecting, video processing and video labeling in section 5.8.

To evaluate CAVA, we developed a testbed consisting of mobile devices and the University of Victoria SAVI edge. A core-i5 laptop was connected to the SAVI edge over a private access point to avoid of collisions caused by other mobile devices not participating in this experiment. The access point was connected via a private LAN network to the University of Victoria SAVI edge.

To evaluate heads-up display representative of devices such as the Microsoft HoloLens, we recorded 5-minute videos with a smart mobile device while walking indoors/outdoors at the University of Victoria. The videos were recorded which their resolutions set to $1268 \times 720p$ and $634 \times 360p$ with 30 frames per second that are representative formats of maximum and minimum available on the Microsoft HoloLens[2] camera.

To generalize the video dataset, especially for the video processing and video labeling performance evaluation, we added 50 first person videos from YouTube, in both high and low resolutions.

[2]Microsoft HoloLens, https://developer.microsoft.com/en-us/mixed-reality, retrieved March 2019.

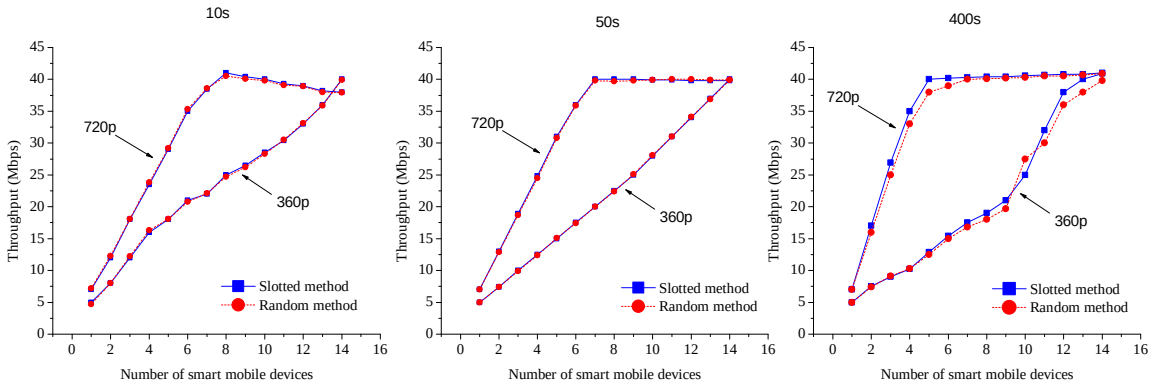

Figure 5.4: Throughput for 720p and 360p resolutions and 10s, 50s, and 400s video frame rates. The cumulative throughput of the SAVI network.

5.7 Evaluation

The number of smart device users that contribute to the SAVI edge's capacity and wireless network bandwidth are the most important challenge that we will explore in the scalability of our architecture. The system load includes video capturing, video processing and video labeling. It is essential to realize which of these processes lead to the scalability bottleneck on the SAVI network testbed.

5.7.1 Video Collecting Cost

In this experiment we quantify the wireless network capacity for the number of video uploads from multiple users to the corresponding SAVI edge with collaboration with my colleague Dr. Andreas Bergen.

First we validated the maximum TCP throughput of our wireless network. When running the experiment with iPerf2[3] on our laptop, the average throughput was around 180 Mbps. However, when running iPerf2 on our Android mobile device, the measured throughput dropped to 40 Mbps. Clearly, the maximum throughput for each device is limited by the processing capacity of the device and the implementation of the TCP stack and WiFi driver [BJKK15].

Second, we experiment with actual video upload, gradually increasing the number of phones uploading to the SAVI edge. To avoid any potential overhead of video processing, we run a Python server on the SAVI edge to calculate the total throughput received and the throughput per mobile device which allows us to set the baseline for how many users can be supported by the SAVI testbed.

To allow for repeatability of the experiments, we divided videos captured by the smart mobile device in fragments of $10s$, $50s$, and $400s$ by FFmpeg[4]. These fragments were uploaded to the devices beforehand and reused during our experiments.

Then, we compare random and slotted upload methods. In the random method, the smart mobile device will start the upload of the first segment after a delay randomly selected in the interval $[0, n]$, with n equal to the length of the video segments used. The subsequent segments become available for upload one by one every n seconds ($10s$, $50s$, and $400s$).

[3] A network traffic tool for measuring TCP and UDP performance, https://sourceforge.net/projects/iperf2/, retrieved May 2019.

[4] A complete, cross-platform solution to record, convert and stream audio and video, https://ffmpeg.org/, retrieved May 2019.

In the slotted method, the interval of n seconds is divided into a number of slots equal to the number of participating smart mobile devices. For example, with $10s$ segments and two smart mobile devices, one device will upload its first segment at $t = 0s$, $10s$, and so on. Whereas the other device will upload at $t = 5s$, $15s$, and so on. This method tries to avoid as much as possible collisions with other devices. If we have more devices, the individual slots will become shorter and it becomes more likely that the upload of a video segment is still in progress when the next device starts transmitting. So the individual throughput of each device will start degrading.

We conducted experiments for the 720p and 360p videos. Figure 5.4 shows averaged results over five iterations and the cumulative throughput received by the SAVI edge and are indicative for the total load on the CAVA.

For both resolutions, the throughput to the SAVI edge increases linearly with the number of mobile devices, until the maximum wireless network throughput is reached. We observed that the throughput per mobile device starts to drop when more than 12 mobile devices uploading 360p videos and 7 mobile devices uploading 720p videos.

The overall throughput does not increase significantly with the use of slotted method. We attribute this to the fact that the transmission speed of a smart mobile device is limited. Even when a mobile device has the full channel capacity at its disposal, it will not reach the maximum bit rate. We expect that the performance of the slotted method will increase with better hardware and software implementations.

5.7.2 Video Processing Cost

The video processing consists of five steps: video preprocessing, filtering of frames based on contextual information, video sampling, context aware blurring, and encoding. To measure the individual throughput of each step, we ran each step individually on a SAVI VM. The results are shown in Figure 5.5.

We omitted the results of the early-discard, since this is a computationally trivial step. Note that the context aware filtering is the only step that can be parallelized, other steps rely on the sequential order of the frames. Accordingly, to obtain the results we ran eight threads for face detection/recognition and face blurring, while we used one thread for each of the other steps.

Clearly, by reducing the resolution the throughput of the personal VM increases, but decreases the video processing accuracy. To study the trade-off between throughput and system accuracy, we selected 720p database videos and created lower resolu-

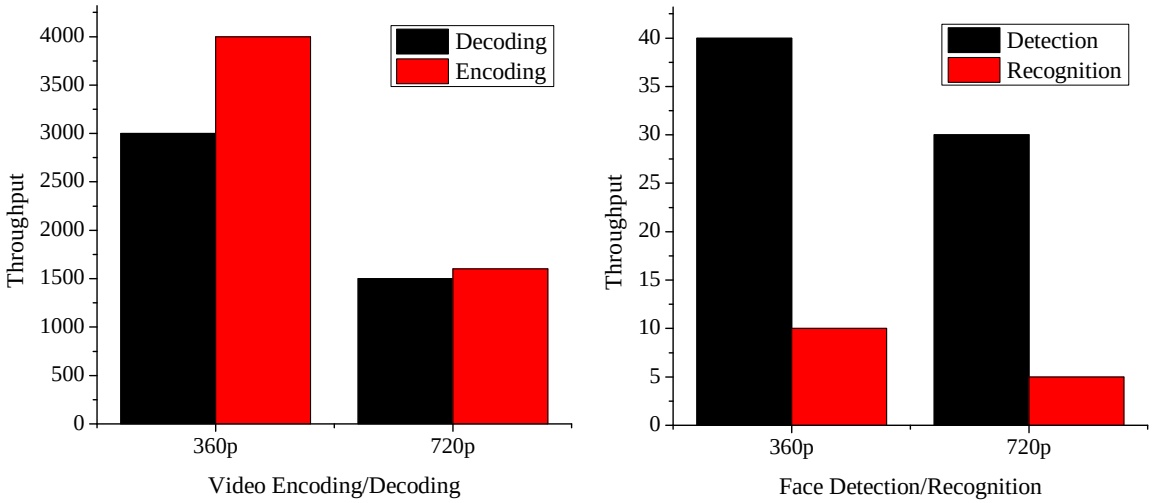

Figure 5.5: Performance of CAVA video processing

tion copies of each video. Experiments show that scaling down the resolution results in a higher throughput but has decreased the detection accuracy from $90 - 100\%$ to $20 - 40\%$ for different experiments. Since the faces in a low resolution video are not detectable, we trained the classifiers with 40×40 pixels to detect small faces. Our experimental results suggest that video resolutions should be at least 480p to trust the system.

5.7.3 Video Labeling Cost

The context aware video labeling involves running computer vision methods inside a VM on the SAVI edge. The key research question is what is the throughput of the video labeler in frames per second? Therefore, to evaluate the impact of the resolution on the system accuracy, we labeled YouTube videos in 720p and 360p resolutions.

We compare the throughput of the video labeling process for the running as a native process, or inside a SAVI VM with eight virtual CPUs. We considered the frames detected by the filters in the original video as the ground truth and compared how many of those frames are found in the same video at 360p resolution. When compared to the processing, the detection accuracy of the filters is more robust to lowering the resolution.

Overall, we can conclude that in our current algorithm we can perform the labeling on the 360p resolution without losing too much accuracy (i.e., less than 10%).

5.8 SAVI Capacity Allocation

Here we indicated how video processing and Video labeling are extremely computationally intensive. In section 5.7, we study how the available SAVI edge hardware should be allocated between the video labeler and the personal VMs. Allocating more virtual CPUs to the personal VMs in favor to the indexer allows to increase the number of processed videos, but the number of non-labeled processed videos will keep increasing if the labeler has not sufficient processing power.

We used four quad-core systems. We dedicated two of them to the labeler. The resources on the other two machines were distributed over a number of personal VMs for the video processing. The Data Manager is running on one of the machines used for video processing. Videos were uploaded every 50 s at 720p resolution and processed through the CAVA steps.

Video processing is performed at the original resolution for maximum accuracy, while the video labeler will first resize the processed video to 360p before running the video context detection methods. We varied the number of virtual CPUs of the VM, the number of users and the number of frames per second of video that is forwarded to the early-discard and context aware filtering steps of Figure 5.2. These two parameters jointly determine the load on the CAVA. If two users uploading at $50fps$ and each SAVI VM selects 1 out of 50 frames for processing, the SAVI VMs are expected to produce 2 processed videos every second.

When the video labeling is indicated as the bottleneck, it means we had to reduce the selection rate of the SAVI VMs and hence not fully utilized the resources allocated to the VM. Conversely, when the video processing is the bottleneck, it means that the labeler has spare resources and that the configured selection rate is the maximum that can be achieved with the given size of the SAVI VMs. Note that the number of users and the number of video segments increases the overhead of virtualization and the overhead of database transactions, respectively.

5.9 Chapter Summary

In this chapter we present a framework integrating the complete chain of processing from the video to the visual content search. Having a context aware searchable collection of processed videos provides a unique view on the world to be leveraged by many existing and future applications.

We process uploaded videos on VMs running on SAVI edges based on contextual information such as time, location. The VMs establish a clear demarcation line where the original videos can be processed. We investigated both frame rate and resolution as parameters to scale up the system. Video processing only a subset of the video frames results in a linear reduction of the system load, and still allows the user to guess the contents of a video from a limited set of frames.

The next chapter presents the experimental study of face recognition by humans, which is our fifth contribution of the dissertation which defines and implements a design to investigate the potential of the contextual information on accuracy rate and response time on human's vision.

Chapter 6

Recognizing Faces in Different Contexts by Humans

Have you ever seen someone who looks familiar, however; you can not remember who that person is and where you met?

This chapter presents an experimental study of face recognition by humans, which is our fourth contribution of the dissertation. There is one research question associated with this contribution:

RQ4. *What is the effect of contextual information on human face recognition? How does contextual information affect human face recognition's accuracy and response time?*

To answer this question, we define a recognition memory study, which implements a design to investigate the potential of the contextual information on accuracy rate and response time.

This chapter is organized as follows: Section 6.1 describes the functionalities of human episodic memory by the way of recognition memory tasks. Section 6.2 describes the implementation of experiment of the face recognition task. Section 6.5 demonstrates results of accuracy (6.5.1) and response time (6.5.2) for context based face recognition tasks. Then, we discuss the finding of the study in Section 6.6. Finally, Section 6.7 summarizes the chapter.

6.1 Memory of Faces

Human episodic memory is commonly studied through recognition memory tasks, in which the prior occurrence of stimuli is assessed [Yon02, YP04]. Recent memory studies have suggested that this ability draws on two distinct processes: familiarity and recollection [LYRR13, SAC16]. Familiarity refers to a general feeling of having encountered a human face before, without conscious access to contextual information details, such as the time, place of the visit or any other thing about the person. Recollection, on the other hand, refers to the conscious retrieval of specific details related to the encoding episode. These specifics are not limited to spatio temporal context, but may as well consist in thoughts one had at the time of first visit, or other items present at encoding [YP04]. Recognition judgments made on the basis of familiarity are fast, automatic, and reflect a general feeling of knowing that a given item in the recognition test was presented previously. On the other hand, recollection requires more contextual information details about studied items and results in slow and more effortful judgments [SAC16].

In a recognition memory study [EZGBM07], they were able to show that familiarity and the associated FN400 effect are sensitive to study-test manipulations of intrinsic item features (e.g. the color of an object), but insensitive to contextual manipulations. That is, the FN400 old-new effect was diminished if the color of objects was changed from study to test, but was not affected by a change of arbitrary background shapes, even though this specific contextual information was available to subjects in a direct source memory test. [TOR01] has reported a contextual influence on the FN400 effect. They had subjects study object images on highly salient landscape scenes, and manipulated the Old/New status of objects, contexts, and their specific combinations, resulting in five test conditions: old objects presented on the same background as the one on the study (i.e., identical repetition), old objects presented on an old context but rearranged with respect to study, old objects presented on new backgrounds (Old/New), and new objects presented on either old (New/Old) or new (New/New) backgrounds. Instructions were to judge the old/new status of objects irrespective of context (inclusion task). The authors reported an FN400 effect only for same and rearranged repetitions but not for Old/New items.

The aim of the present study is to test positive effect of contextual information on face recognition by adopting the [TOR01] design and manipulating the potential

of the contexts to have faster and more accurate face recognition response. This was done by implementing a design that followed the [TOR01] procedure.

6.2 Method

In this experiment, the adopted version of the [TOR01] face recognition task was used to test the contextual information effect on accuracy rate and response time. The design followed the one reported by [TOR01] in many aspects. Throughout the experiment, face images (not object images) of varying sizes have been used. Note that the only reason for different sizing of faces was the application of the cueing technique which comprises face size manipulations. If all faces were virtually the same size, cues would not be very helpful. There has been no study on face size manipulations. Also, faces were not presented super imposed on background images.

6.2.1 Participants

Ten subjects for each test (working clothes congruent test 6.3.1 and scene congruent test 6.3.2)-in total 20 subjects-, five females and five males undergraduate and grad students at the University of Victoria participated in the study. They were not rewarded. The mean age of the participants was 30.33 years (SD = 3.40), and all participants were right-handed. They had normal or corrected-to-normal vision and reported no neurological illness, dyslexia, or symptoms of prosopagnosia during the preliminary interviews at the beginning of the experimental session. All participants reported being healthy. All participants gave informed consent.

6.2.2 Apparatus

Experiment was conducted on a 24-Inch 1920×1200 resolution LCD flat panel monitor. Participants were seated in a room, free of movement, at 70 cm from a computer screen piloted from a PC computer. Subjects were instructed to fixate on the center of the screen, and they were instructed to minimize eye blinks while stimuli were on the screen.

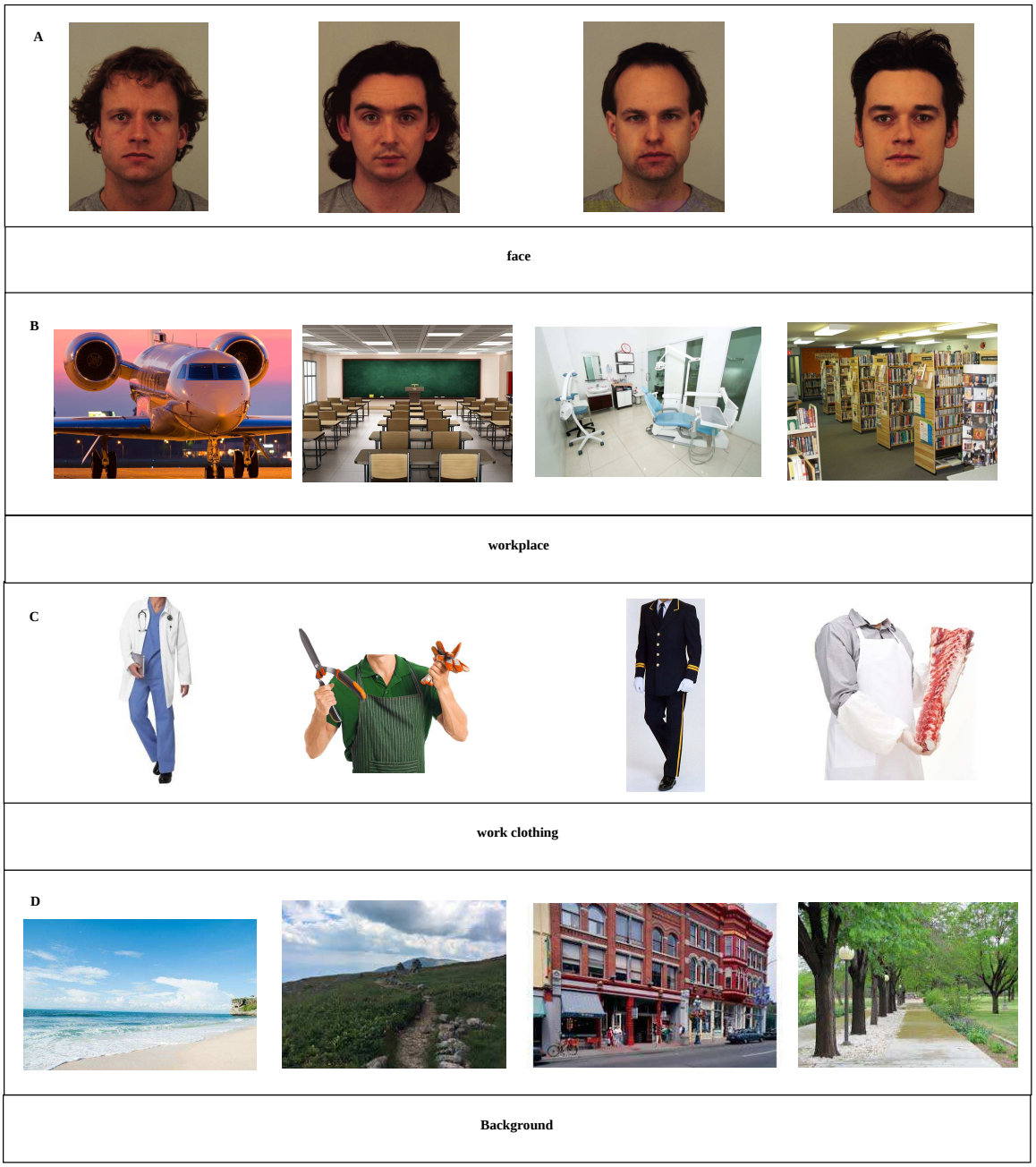

Figure 6.1: Stimuli used in experiments. A shows the four face images [LDB+10]. B shows the four images of a workplace. C shows four images used for working clothes. D shows four images used for the scene. All images used for workplace, working clothes and background were downloaded from the internet

6.2.3 Stimuli

Faces with contextual information stimuli were taken from chromatic photographs and included contextual information. The means of contextual information included workplace, working clothes, and generally neutral emotional expressions. There was no priori information on the size, position, or number of the targets in a single photograph. There was also a very wide range of distractor images, which could be outdoor or indoor scenes, natural landscapes or workplace with working tools with the limitation that no animal or person appeared. We did not use experimentally manipulated images. Rather, and following [JWVMB11], the images encompassed a diverse range of natural photographs taken of the identities at different ages, from different angles, and in different lighting conditions. The individuals in the photographs were unfamiliar to participants. The occupations are an airline pilot, barber, bus driver, butcher, chemist, civil engineer, computer engineer, cook, dancer, dentist, doctor, drummer, gardener, interior designer, librarian, mechanic, nurse, office receptionist, optometrist and teacher. Stimuli used in experiments can be seen in Figure 6.1.

6.3 Design

6.3.1 Working Clothes Congruent

In each test of working clothes congruent, participants studied ten faces with contextual information (from 20 images of different occupations) in the chromatic photographs and approximately 0.75 min later were tested with those ten faces with contextual information in the scene congruent photographs (16.6% of the trials), scene incongruent in the monochromatic photographs (16.6% of the trials) and isolate faces (without contextual information) in the chromatic photographs (16.6% of the trials) plus ten new faces in the scene congruent chromatic photographs (16.6% of the trials), ten new faces in the scene incongruent in the monochromatic photographs (the other 16.6% of the trials) and ten new isolate faces (without contextual information) in the monochromatic photographs (the other 16.6% of the trials) for a total of 60 trails. Figure 6.2 shows the memory paradigm for working clothes congruent schematically.

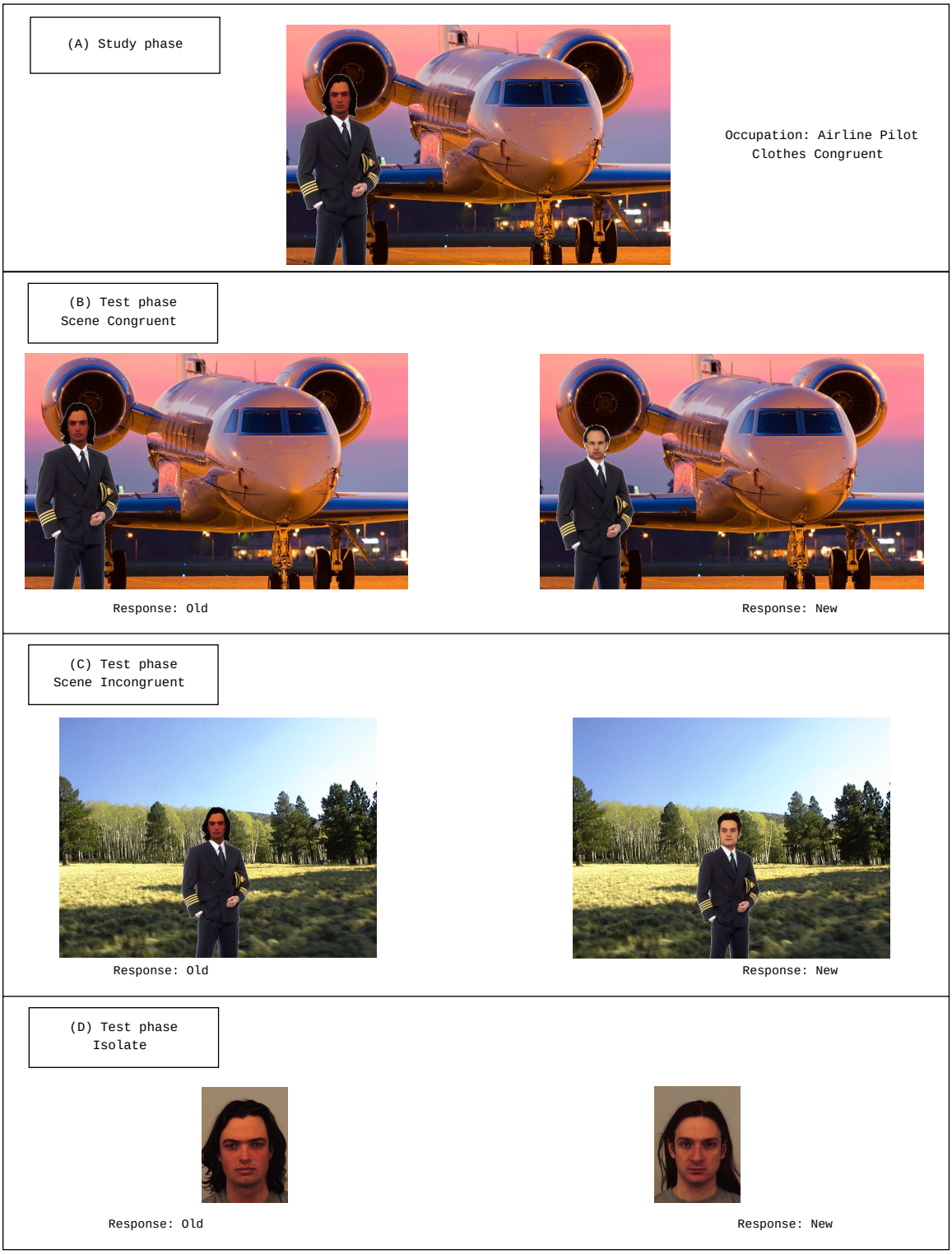

Figure 6.2: Schematic representation of the Test 1 memory paradigm. (A) In the study phase, participants viewed a series of faces with contextual information. Participants were instructed to remember these face. (B), (C) and (D) In the test phase, faces were presented in scene congruent, scene in congruent and isolate respectively. The response indicated whether the face was old or new (i.e., one from the study phase or not, respectively)

6.3.2 Scene Congruent

In each test of workplace congruent, participants studied ten faces with contextual information (from 20 images of different occupations) in the chromatic photographs and approximately 0.75 min later were tested with those ten faces with contextual information with the working clothes congruent photographs (16.6% of the trials), working clothes incongruent in the monochromatic photographs (16.6% of the trials) and isolate faces (without contextual information) in the chromatic photographs (16.6% of the trials) plus ten new faces in the working clothes congruent chromatic photographs (16.6% of the trials), ten new faces in the working clothes incongruent in the monochromatic photographs (the other 16.6% of the trials) and ten new isolate faces (without contextual information) in the monochromatic photographs (the other 16.6% of the trials) for a total of 60 trails. Schematic representation of the memory paradigm in workplace congruent can be seen in Figure 6.3.

6.4 Procedure

In the working clothes congruent test 6.3.1 and the scene congruent test 6.3.2 each study trial included a 1-s gray fixation cross at eye level on a black background followed by a 1.5-s face presentation, with contextual information related to the face.

Participants for each test (working clothes congruent or scene congruent) were told to try to remember the face for a subsequent memory test. In the test phase of the scene congruent and working clothes congruent tests, faces appeared with or without contextual information and isolate faces, and participants pressed one of the two buttons on each trial according to whether they thought the face had appeared in the study phase or not (i.e., old or new). Each test trial included 1.5-s fixation, 0.5-s face, fixation, and then response.

Figure 6.3: Schematic representation of the Test 2 memory paradigm. (A) In the study phase, participants viewed a series of faces with contextual information. Participants were instructed to remember these face. (B), (C) and (D) In the test phase, faces were presented in working clothes congruent, working clothes incongruent and isolate respectively. The response indicated whether the face was old or new (i.e., one from the study phase or not, respectively)

6.5 Results

6.5.1 Accuracy

In working clothes congruent experiment Section 6.3.1 and scene congruent experiment Section 6.3.2, we found a huge difference in accuracy rates for recognizing the old faces with and without contextual information (cf. Figure 6.4 (A) and Figure 6.5 (A)). Also, you can see from Figure 6.4 and Figure 6.5 a scene as a context has more positive effect on accuracy than a working clothes context. Also, it is clear that contextual information does not have a negative effect on the accuracy rate of the recognition of new faces (cf. Table 6.1).

Assuming easier encoding elaboration for the old faces in the context rather than the old faces without any contextual information, because more is known about the former, we wanted to make sure that any greater reinstatement advantage for the faces with context was the result of greater ease of associating the person to the encoding context than elaboration. We predicted that memory for faces in the context would be more affected than memory for faces without context because it is more likely in the former case that the face was successfully associated to the context in the first place.

6.5.2 Response Time

The main effect of contextual information was due to the faster responses recognizing old faces than new faces. The main effect of face recognition showed that it took longer to respond to the without context condition than the with context condition. The main effect of contextual information showed that response time in with context condition was significantly faster than the response time to the without context condition for old faces (cf. Figure 6.4 (B) and Figure 6.5 (B)). In addition, Figure 6.4 (B) and Figure 6.5 (B) shows the effect of scene contextual information which generally makes face recognition faster than face recognition with working clothes contextual information (cf. Table 6.1).

Figure 6.4 (B) depicts the mean response time of all participants in the working clothes congruent experiment for old and new faces in with and without contextual information about scene condition. Figure 6.4 (B) shows that the mean response time is faster to the old faces with scene contextual information than old faces without scene contextual information. This is in agreement with the accuracy rate of recognition

of the old faces with and without scene contextual information (cf. Figure 6.4 (A)). The faster response occurs while the participants recognizing faces with higher rate of accuracy.

Figure 6.5 (B) depicts the mean response time of all participants in the scene congruent experiment for old and new faces in with and without working clothes contextual information condition. Figure 6.5 (B) shows that the mean response time is faster to the old faces with working clothes contextual information than old faces without working clothes contextual information. This is in agreement with the accuracy rate of recognition of the old faces with and without working clothes contextual information (cf. Figure 6.5 (A)). The faster response happens while the participants recognizing faces with higher rate of accuracy.

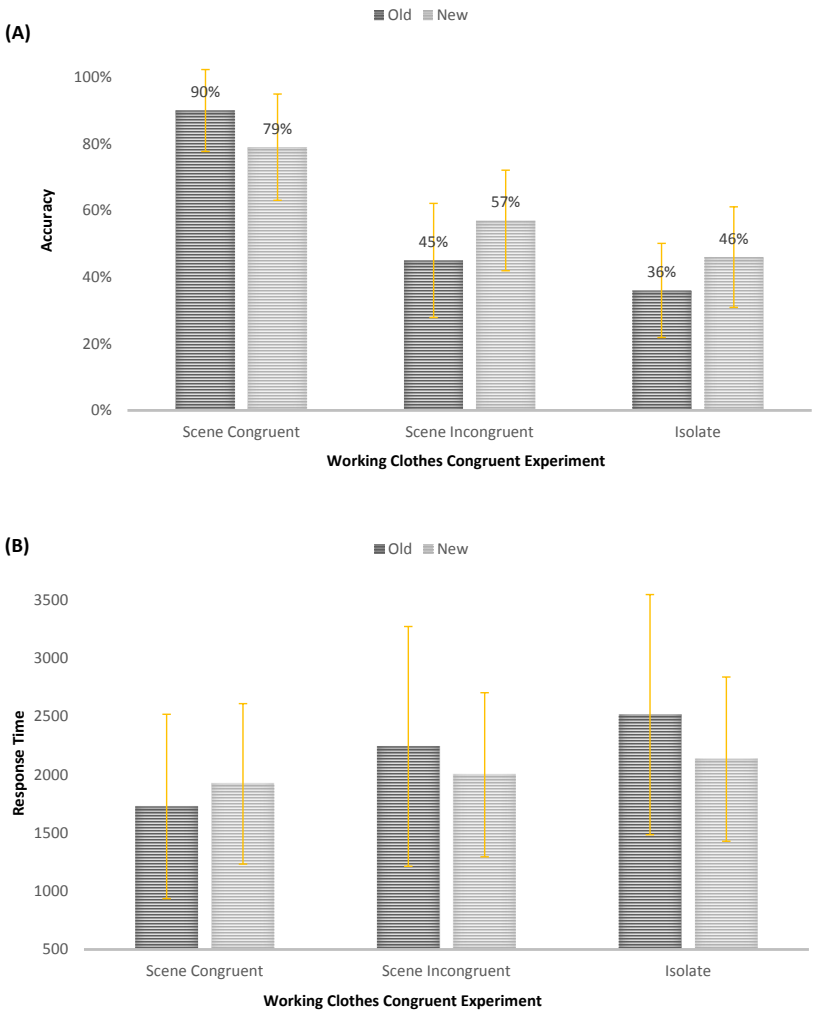

Figure 6.4: (A) Accuracy for old and new faces in the picture for both with and without contextual information condition in the experiment. (B) Shows the response time for old and new faces in the picture for both with and without contextual information condition in the experiment. Error bars indicate standard errors of the mean

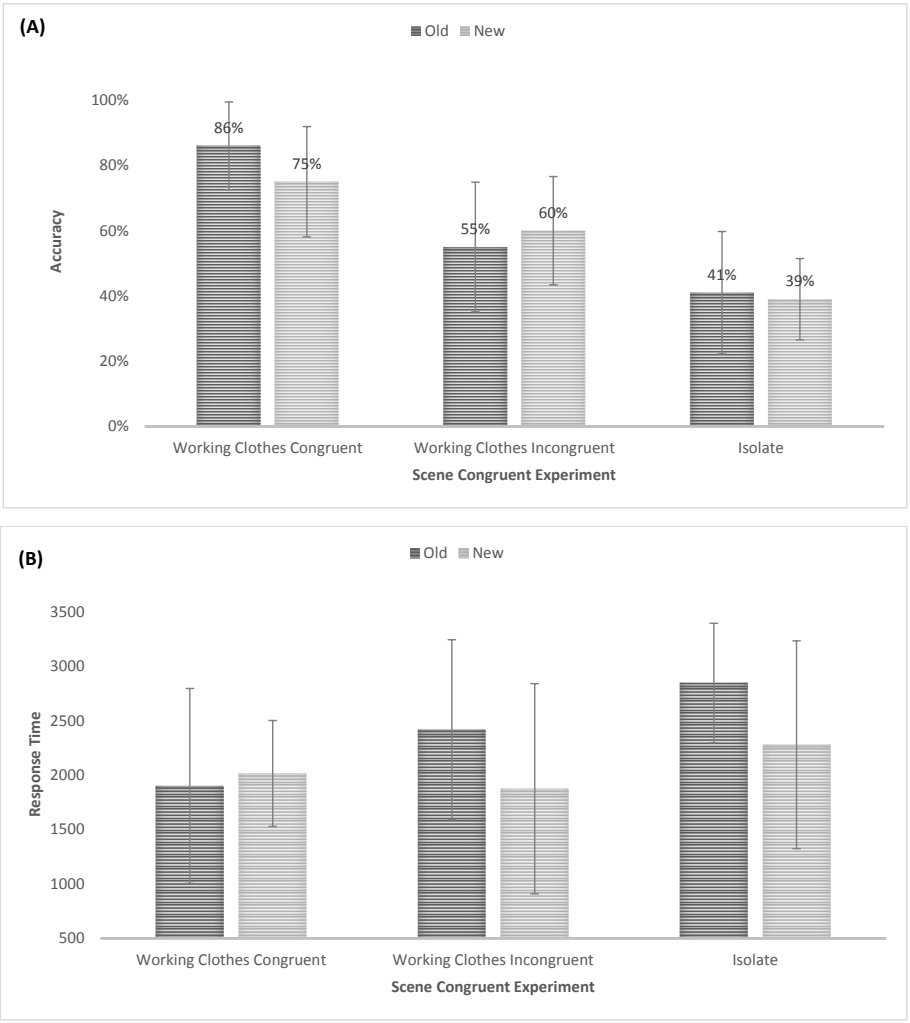

Figure 6.5: (A) Accuracy for old and new faces in the picture for both with and without contextual information condition in the experiment. (B) Shows the response time for old and new faces in the picture for both with and without contextual information condition in the experiment. Error bars indicate standard errors of the mean

6.6 Discussion

When an encoding contextual information is associated with many study episodes, the benefit of reinstatement is diminished. This is because the amount of additional activation that is sent to any single episode node from the activation source of the test-probe in which the context is shared or distributed among all the competing contextual associations. Therefore, manipulations of contextual information should affect the success of recovering the memory trace associated with that context [SM10].

A number of variables affect the ability to associate the encoding context to the presented stimulus, including available working memory resources. When there are insufficient working memory resources, as sometimes occurs with older adults [BFL+11] or in a dual-task setting, it is more difficult to associate context with a stimulus. In that case, memory performance relies more heavily on familiarity than on recollection.

The experiment reported here demonstrates that stimulus familiarity is another variable that also affects the ease of associating a stimulus to its encoding context. The added value of context reinstatement for recognition memory is much greater when the stimuli present preexisting memory representations (e.g., old faces) than when they are new (e.g., first time seen faces). There are several alternative accounts for why faces with contextual information are better remembered than faces without contextual information. One is being easier to generate an elaboration involving a face which has more contextual information (e.g., [ZGKB10]).

Extending this account, one could argue that reinstatement of background should aid memory when the subject has generated an elaboration that involves the background because there would be more features to match the original memory trace. This account (devoid of an assumption of associating the target to a context), however, does not explain why the advantage of reinstatement should be reduced when a background is shared with other faces (New faces with context).

Another possibility is that it is easier to generate a label for a face (e.g., the person's occupation) with context than for a face without contextual information and that having a label will facilitate binding to the long-term memory. Although we agree that part of the advantage for faces in the context is that each face affords a label, we do not believe that merely providing a label for a new face will facilitate recognition for new faces, with or without a context.

In summary, we found contextual information effect for both old and new faces, but this effect was significantly larger for old faces than for new faces, confirming

the hypothesis that faces with contextual information are perceived more holistically than faces without contextual information (cf. Table 6.1). This has been argued elsewhere (e.g., [RVM+13]) that stimuli with stable memory representations are more easily encoded and associated with contextual details, thereby making recollection judgments more successful for those types of stimuli [EZGBM07].

Table 6.1: Response time (in ms) and accuracy for correct judgments as a function of whether stimulus is old or new. Discrimination for each type of stimuli is also shown. Standard deviation are given in next rows

	Working Clothes Congruent						**Scene Congruent**					
	Scene Congurent		Scene Incogruent		Isolate		Working Clothes Congurent		Working Clothes Incongruent		Isolate	
	Old	New	Old	New	Old	New	Old	New	Old	New	Old	New
RT	1728	1922	2243	2000	2516	2134	1900	2014	2416	1874	2845	2278
SD	791	688	1029	704	1031	705	895	486	826	965	548	953
Acc (M)	90	79	45	57	36	46	86	75	55	60	41	39
SD	12	15	17	15	14	15	13	17	19	16	18	12

Note: RT, response time; Acc, accuracy; M, mean; SD, standard deviation.

6.7 Chapter Summary

This chapter presented the effect of contextual information on episodic face recognition, which is our experimental study on human context based face recognition. The experimental study constitutes the fourth and last contribution of this dissertation and allows us to answer research question RQ4 (cf. Figure 1.2).

In Section 6.1, we reviewed human episodic memory studies by way of recognition memory tasks. In Section 6.2, we defined our experimental case studies of the face recognition task that was used to test the contextual information effect on accuracy rate and response time. We presented two test designs working clothes congruent 6.3.1 and scene congruent 6.3.2 in Section 6.3. In Section 6.5 we presented results of our experimental context based face recognition tests by accuracy 6.5.1 and response time 6.5.2. Finally, in Section 6.6 we discussed our finding for context base face recognition.

The next chapter of this dissertation summarizes the research and contributions of this dissertation, presents the conclusions and discusses potential future work.

Chapter 7

Summary and Future Work

This chapter summarizes this dissertation by revisiting our research challenges, goals and contributions. Finally, this chapter concludes this dissertation with a discussion of future work.

7.1 Dissertation Summary

Face recognition systems have greatly matured over the last two decades to the point where many commercial and law enforcement applications use face recognition for person identification, such as smart phone face unlock [Bud18]. Automatic face recognition systems operate on face images acquired in controlled conditions. Technological advancements in automatic face recognition have progressively tackled challenges caused by variations in facial pose, illumination, and expression.

Face recognition in humans is subconsciously associated with contextual information from the environment and social parameters [AT13, OBT16]. Contextual information helps us identify faces in daily social interactions and humans may fail to recognize the observed face without this information [MB13, YB18]. Hence, taking the contextual information into account in real-world face recognition applications is of vital importance to enhance the ability, performance, and reliability of automatic face recognition systems. Contextual information includes information related to the image of the scene surrounding the person, camera context, such as location and image capture time, and the social context that describes the interactions between people.

Further to this cognitive approach, the statistical approach can also be used to tackle the face recognition problem. There is significant statistical correlation between contextual information and image information, which enables statistical operators to achieve a higher recognition rate based on contextual information [Riv14]. In a more general manner, the face recognition problem can be approached with information theory, because real-world face recognition is a challenging problem and contextual information is not redundant with respect to database and image information.

The two motivations that drove this research is the need for (1) improving context awareness and the exploitation of the value of contextual information to enhance the recognition rate in face recognition systems; and (2) improving the dynamic capabilities of adaptivity in face recognition systems by controlling the relevance of contextual information through context collection, analysis and search.

The research problem addressed in this dissertation was to investigate how contextual information aids the face recognition and how to collect and extract contextual information automatically for adaptive learning. Thus, for face recognition systems to become smarter and context aware, these systems need instrumented so that (1) contextual information can be added to exhibit an explicit relationship with the face recognition system; and (2) the resulting face recognition system can support variations in the set of the relevant image database and contextual information entities at runtime.

7.1.1 Addressed Challenges

We classified our research challenges related to context information into three groups: *data collect and context extraction*, *adaptive learning and classification,* and *human visual system context variability*. The challenges that we addressed in this dissertation are summarized as follows:

Data collection and context extraction

RCH1. Developing automatic data collection methods to make an image database equipped with contextual information, such as location, time, and image content.

RCH2. Developing mechanisms to project face images onto a feature space to make the classifiers faster.

Adaptive learning and classification

RCH3. Investigating adaptive learning and classifier method to accommodate variations such as size, view, expression, and light of faces.

RCH4. Decentralizing cloud infrastructure to scale well beyond to millions of concurrent uploads from a dense urban area.

Human visual system context variability

RCH5. Isolating mechanisms responsible for substantial variance in the images.

RCH6. Creating and updating an abstract representation of each individual's facial identity.

7.1.2 Contributions

This section summarizes our contributions. Figure 7.1 relates our contributions to each other through the architecture of the context aware face recognition system. Contribution *C1* corresponds to our location-based face recognition approach. The framework comprises location-centric image databases to recognize faces in images that have been taken at nearby locations frequently visited by individuals. Contribution *C2* constitutes our characterization of context and smart applications for face recognition. Context is broadly characterized as information relevant to something under consideration which can include information from non-face regions of the image, information related to the capture of the image, or the social network context of the interactions between people. Contribution *C3* is our design of a contextual information extraction algorithm and creates cloud base databases with contextual information, as well as our architecture for context aware video based face recognition which decentralizes cloud computing on SAVI network infrastructure. Finally, contribution *C4* demonstrated a design of an experimental study of face recognition by humans. The experimental study provides insights into the nature of cues that the human visual system relies upon for achieving its impressive performance and serves as the building block in the proposed context aware face recognition system.

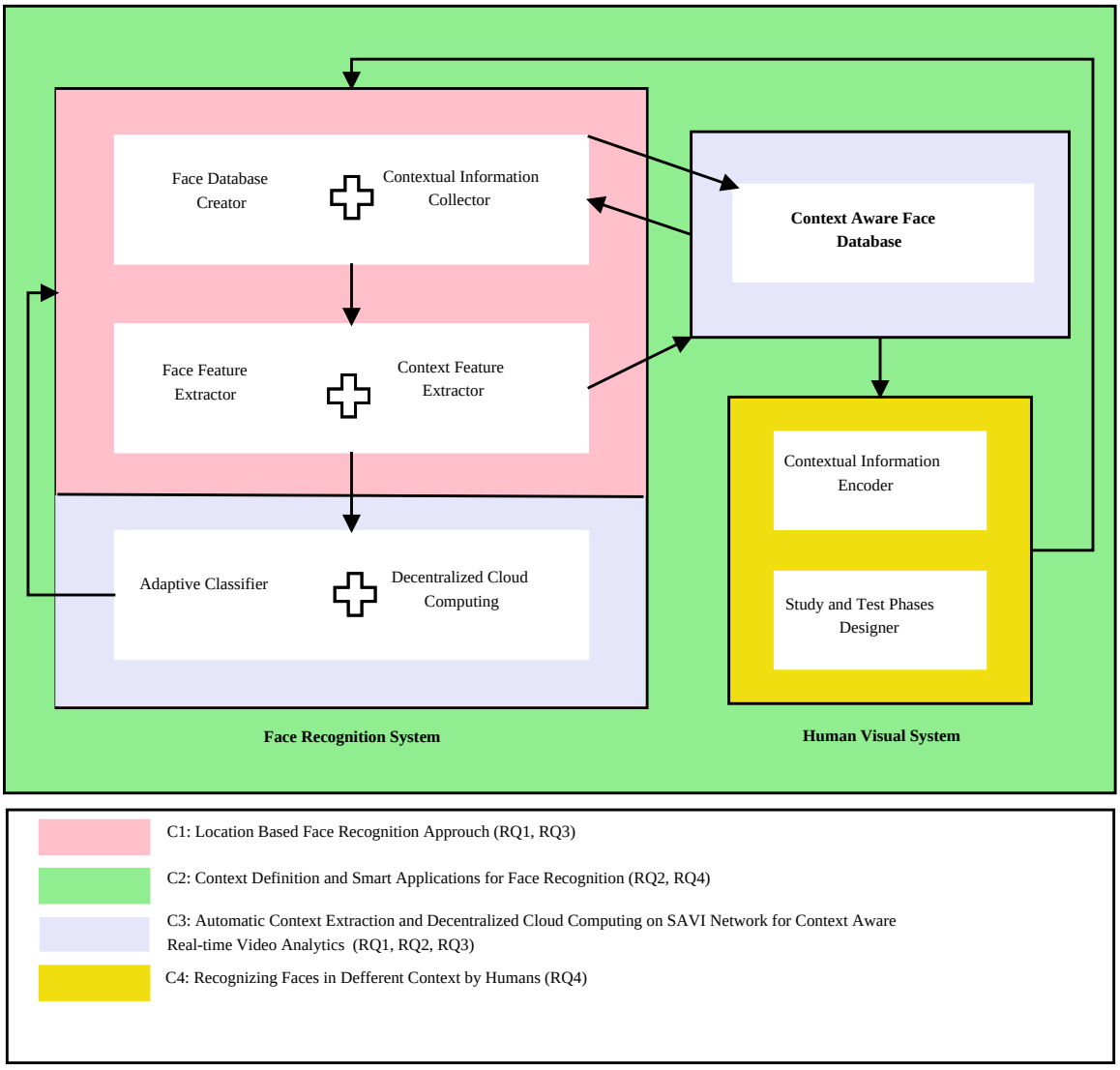

Figure 7.1: Summery of our contributions

C1: Location-based Face Recognition Approach

The framework for location-based face recognition is a detailed description of the method for using location information our face recognition algorithm. The framework comprises location-centric image databases to recognize faces in images that have been taken at nearby locations frequently visited by individuals. The approach is as follows: (1) given a set of known images of faces for training and another set of faces of the same people as a testing set, (2) recognize each face in the testing set, (3) each face image associates with the location information, and (4) creates many clusters of locations from the training set where each location cluster contains a set of individuals, who have images in that location, and images of their interacted people. Finally, (5) the user can take an image and attach the location information, then send it to the system and query for recognizing the face in the image. The system will answer the recognition question and return the identifies of the faces in the image.

C2: Context Characterization and Smart Applications for Face Recognition

This contribution is an architectural design for context aware face recognition systems and their smart applications. Context is broadly characterized as information relevant to something under consideration which can include information from non-face regions of the image, information related to the capture of the image, or the social network context of the interactions between people. The useful contextual information for face recognition is categorized as follows: (1) face context, which includes information about a face, such as anthropometric measurements, skin color, and distance between face parts; (2) pixel context, which includes the non-face regions of the image information such as distinctive clothing, classes, and other faces; (3) sensor context, which includes the capture conditions of an image such as location, time, and brightness; (4) interaction context, which includes the information about social relationship, weak labels, age, and gender. This architectural design is based on contextual information that assists face recognition systems to act smarter. This design demonstrates how face recognition applications can become smarter using the contextual information.

C3: Automatic Context Extraction using Decentralized Cloud Computing on SAVI Network

This contribution is the design of a contextual information extraction algorithm as follows: (1) context aware filters are initially applied on a subset of the selected video frames, then (2) more complex context aware filters are applied that extract features and relevant contexts, to improve face recognition accuracy, (3) detected faces are normalized to the same size and, finally, (4) the detected faces are automatically added to the cloud base database along with contextual information which is the key to adaptability.

Also, this contribution presents an architecture for context aware video based face recognition using decentralized cloud computing on SAVI network infrastructure: (1) a video from an individual mobile device travels as far as its currently associated SAVI edges node, (2) computer vision analytics run on a SAVI node VM in near real time, (3) the Data Manager runs in an individual VM on the SAVI network to manage the storage of the videos and database with the associated contextual information, (4) the data is logically organized as a collection of videos, (5) results of the processing along with contextual information (such as the VM details, location, start time and video duration) are sent to the SAVI core, and (6) the labels and contextual information in the SAVI core can guide and facilitate deeper and more customized searches of the contents of a video while residing on a SAVI node VM.

C4: Recognizing Faces in Different Contexts by Humans

This contribution includes a design of an experimental study of face recognition by humans. This study provides insights into the nature of cues that the human visual system relies upon for achieving its impressive performance and serves as the building block in our context aware face recognition system. We showed that the benefit of reinstatement is diminished when an encoding of contextual information is associated with many study episodes. This experimental study included the following phases: (1) a database of individuals' images was created with and without contextual information (e.g., workplace, working clothes, and generally neutral emotional expressions); (2) design study phase, participants viewed a series of faces from the database with contextual information where participants were instructed to remember these face. (3) Designing the test phase, where faces were presented with and without contextual information. The response indicated whether the face was old or new (i.e., old faces

were previously shown in the study phase, unlike new faces), and (4) analyzing the results of the contextual information effect for both old and new faces on response time and accuracy.

7.2 Future Work

This section presents selected avenues for future work opportunities emerging from our research.

Label Video with Human Action

We processed video contents using our method to obtain labels. In an extended version of our architecture, the analysis of sequences of processed videos can be included so that human-meaningful actions can be tagged, such as running, swimming, or playing.

Person's Properties

An interesting topic is to build estimators to predict persons properties from a person's images. The images in our training database are associated with contextual information such as gender information (or other properties) and could be easily retrieved. People could train a robust gender classifier for the face images in the wild based on this context aware database.

Context Aware Database

We used a benchmark task which is to recognize people in the world from their face images, and link the face to corresponding contextual information. Our face recognition human behavior in recognizing images. We also provide, to the best of our knowledge, the only context base training dataset to facilitate research in the area. Beyond face recognition, our database could inspire other research topics such as people could adopt one of the cutting-edge unsupervised/semi-supervised clustering algorithms on our training dataset, and/or develop new algorithms. Furthermore, there is a need for a robust adaptation scheme incorporating an effective labeling procedure for the input samples. This is supported by our findings related to supervised

versus semi-supervised methods for adaptation where the supervised scheme generalizes better than the semi-supervised one. This indicates that the use of confidently classifying input samples (as used by most of the existing automated systems based on semi-supervised learning) may not be an efficient strategy for adaptation.

Extracted Contextual Information for Smart Systems Development

We designed and implemented context aware feature extraction and learning methods from videos and images independent of the application domain. As our methods are general context feature extraction and learning methods, it is reasonable and interesting to apply them to other smart computer vision applications such as object recognition and visual tracking in the future. It would be interesting to investigate texture feature representations applied to face recognition [DLFZ18]. Furthermore, it would be useful to develop alternative techniques to obtain other geodesic distance approximations for multivariate normal distributions, including Gaussian mixture models.

Conventional Neural Network/Recurrent Neural Network

We designed our context aware face recognition system based on existing Artificial Neural Networks. One avenue for future work to concentrate on best feature selection methods for our context aware face recognition. One approach is to investigate Conventional Neural Networks or Recurrent Neural Networks [ZAZC19].

Adaptivity

Adaptive face recognition is a challenging topic. Although template update methods in adaptive face recognition have shown to be promising, some open issues still need to be addressed particularly investigating the tradeoff between performance enhancement and gallery size maintained when updating. It is worth mentioning, all the template update methods are prone to an impostor's introduction and the attraction of more samples of it may gradually lead to identity creep-in, when the genuine person loses his or her identity.

In other words, adaptive face recognition systems with impostor attacks result in lower performance gain in comparison to those using only genuine samples for adap-

tation. This is an account of updating using impostors as a result of intrinsic failure of the system, i.e., false acceptance rate (FAR); thus increasing the vulnerability to template security and undermining the integrity of the adaptive face recognition systems. To this front, modelling and early stoppage of impostor attack into the updated template set is an important research direction to be pursued. Avoiding impostor intrusion into the updated template set will allow commercial vendors to adopt auto-update procedures into their commercial face recognition products. These results emphasize the need for more robust adaptation schemes that are capable of identifying genuine samples with substantial variations without increasing the vulnerability of impostor intrusion.

References[1]

[AAC16] U. Alegre, J. C. Augusto, and T. Clark. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, 117:55–83, 2016. 1, 47

[AAEF15] Z. Akhtar, A. Ahmed, C. E. Erdem, and G. L. Foresti. Adaptive facial recognition under ageing effect. In *Adaptive Biometric Systems*, pages 97–117. Springer, 2015. 25

[AAL+15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV2015)*, pages 2425–2433, 2015. 19

[AAR+16] S. Anzar, K. Amala, R. Rajendran, A. Mohan, P. Ajeesh, M. Sabeeh, and F. Aziz. Efficient online and offline template update mechanisms for speaker recognition. *Computers & Electrical Engineering*, 50:10–25, 2016. 26, 29

[ADB+99] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing (HUC1999)*, pages 304–307. Springer, 1999. 20

[AHP06] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI2006)*, (12):2037–2041, 2006. 16

[1]The numbers at the end of each bibliography item are the backward references to the pages where it was cited.

[ALGS07] D. Anguelov, K.-c. Lee, S. B. Gokturk, and B. Sumengen. Contextual identity recognition in personal photo albums. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2007)*, pages 1–7. IEEE, 2007. 22

[ANRS07] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007. 15

[AS16] I. E. Allen and J. Seaman. Online report card: Tracking online education in the United States. *Babson Survey Research Group*, 2016. 52

[AT13] M. A. Apps and M. Tsakiris. Predictive codes of familiarity and context during the perceptual learning of facial identities. *Nature Communications*, 4:2698, 2013. 1, 85

[AWR+16] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, et al. Face recognition using deep multi-pose representations. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 15

[BBS+17] S. Bouzefrane, S. Banerjee, F. Sailhan, S. Boumerdassi, and E. Renault. *Third International Conference Mobile, Secure, and Programmable Networking (MSPN2017), Revised Selected Papers*, volume 10566. Springer, 2017. 57

[BFL+11] N. G. Buchler, P. Faunce, L. L. Light, N. Gottfredson, and L. M. Reder. Effects of repetition on associative recognition in young and older adults: Item and associative strengthening. *Psychology and Aging*, 26(1):111, 2011. 81

[BGSV15] R. Bhardwaj, G. Goswami, R. Singh, and M. Vatsa. Harnessing social context for improved face recognition. In *2015 International Conference on Biometrics (ICB)*, pages 121–126. IEEE, 2015. 23

[BJKK15] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti. BackFi: High Throughput WiFi Backscatter. *ACM SIGCOMM Computer Communication Review*, 45(4):283–296, 2015. 64

[BMO+15] M. Busch, E. Mattheiss, R. Orji, A. Marczewski, W. Hochleitner, M. Lankes, L. E. Nacke, and M. Tscheligi. Personalization in serious and persuasive games and gamified interactions. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pages 811–816. ACM, 2015. 52

[Bra17] J. Brannen. *Mixing methods: Qualitative and quantitative research*. Routledge, 2017. 5

[BT16] A. Bergen and N. Taherimakhsousi. Software energy optimization in the cloud. In *Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering (CASCON2016)*, pages 243–249. IBM Corp., 2016. 11

[BTJ+13] A. Bergen, N. Taherimakhsousi, P. Jain, L. Castaneda, and H. A. Müller. Dynamic context extraction in personal communication applications. In *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research (CASCON2013)*, pages 261–273. IBM Corp., 2013. 7, 11, 52

[BTM15] A. Bergen, N. Taherimakhsousi, and H. A. Müller. Adaptive management of energy consumption using adaptive runtime models. In *IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS2015)*, pages 120–126. IEEE, 2015. 11, 57

[Bud18] A. Bud. Facing the future: The impact of apple faceid. *Biometric Technology Today*, 2018(1):5–7, 2018. 85

[BVS14] S. Bharadwaj, M. Vatsa, and R. Singh. Aiding face recognition with social context association rule based re-ranking. In *IEEE International Joint Conference on Biometrics (IJCB2014)*, pages 1–8. IEEE, 2014. 20, 23

[BWL+14] S. Bakkes, S. Whiteson, G. Li, G. V. Vişniuc, E. Charitos, N. Heijne, and A. Swellengrebel. Challenge balancing for personalised game spaces. In *IEEE Games Media Entertainment (GEM)*, pages 1–8. IEEE, 2014. 51

[BZL+13] L. Begeja, E. Zavesky, Z. Liu, D. Gibbon, R. Gopalan, and B. Shahraray. Vidcat: An image and video analysis service for personal media management. In *Multimedia Content and Mobile Devices*, volume 8667, page 86670F. International Society for Optics and Photonics, 2013. 53

[CHL12] Y.-Y. Chen, W. H. Hsu, and H.-Y. M. Liao. Discovering informative social subgraphs and predicting pairwise relationships from group photos. In *Proceedings of the 20th ACM International Conference on Multimedia (ACMMM2012)*, pages 669–678. ACM, 2012. 23

[CK19] N. Christou and N. Kanojiya. Human facial expression recognition with convolution neural networks. In *Third International Congress on Information and Communication Technology*, pages 539–545. Springer, 2019. 51

[CMM+05] A. Chandra, G. M. Martinez, W. D. Mosher, J. C. Abma, and J. Jones. Fertility, family planning, and reproductive health of us women: data from the 2002 national survey of family growth. *Vital and health statistics. Series 23, Data from the National Survey of Family Growth*, page 25:1, 2005. 48

[CSCG16] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero. Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI2016)*, 38(8):1548–1568, 2016. 51

[CSF+13] Q. Chen, Z. Song, R. Feris, A. Datta, L. Cao, Z. Huang, and S. Yan. Efficient maximum appearance search for large-scale object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR013)*, pages 3190–3197, 2013. 20

[CSG+03] I. Cohen, N. Sebe, F. Gozman, M. C. Cirelo, and T. S. Huang. Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2003. 16

[DD18] R. Dwivedi and S. Dey. Score-level fusion for cancelable multi-biometric verification. *Pattern Recognition Letters*, 2018. 25

[DGH+18] A. Das, C. Galdi, H. Han, R. Ramachandra, J.-L. Dugelay, and A. Dantcheva. Recent advances in biometric technology for mobile devices. In *9th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2018)*, 2018. 28

[DGM+16] M. A. A. Dewan, E. Granger, G.-L. Marcialis, R. Sabourin, and F. Roli. Adaptive appearance model tracking for still-to-video face recognition. *Pattern Recognition*, 49:129–151, 2016. 26

[DHWG17] W. Deng, J. Hu, Z. Wu, and J. Guo. Lighting-aware face frontalization for unconstrained face recognition. *Pattern Recognition*, 68:260–271, 2017. 26

[DLFZ18] Y. Duan, J. Lu, J. Feng, and J. Zhou. Context-aware local binary feature learning for face recognition. *IEEE transactions on pattern analysis and machine intelligence (PAMI2018)*, 40(5):1139–1153, 2018. 92

[DlTGSG15] M. De-la Torre, E. Granger, R. Sabourin, and D. O. Gorodnichy. An adaptive ensemble-based system for face recognition in person re-identification. *Machine Vision and Applications*, 26(6):741–773, 2015. 29

[DMS18a] N. Dvornik, J. Mairal, and C. Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV2018)*, pages 364–380, 2018. 18

[DMS18b] N. Dvornik, J. Mairal, and C. Schmid. On the importance of visual context for data augmentation in scene understanding. *arXiv preprint arXiv:1809.02492*, 2018. 18

[DPBB14] A. Das, U. Pal, M. A. F. Ballester, and M. Blumenstein. A new efficient and adaptive sclera recognition system. In *IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM2014)*, pages 1–8. IEEE, 2014. 27

[DSC+05] M. Davis, M. Smith, J. Canny, N. Good, S. King, and R. Janakiraman. Towards context-aware face recognition. In *Proceedings of the 13th Annual ACM International Conference on Multimedia (ACMMM2005)*, pages 483–486. ACM, 2005. 21

[DT16a] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST2016)*, 7(3):37, 2016. 6

[DT16b] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST2016)*, 7(3):37, 2016. 15

[DT18] C. Ding and D. Tao. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):1002–1014, 2018. 15

[EKEY08] R. Ebrahimpour, E. Kabir, H. Esteky, and M. R. Yousefi. View-independent face recognition with mixture of experts. *Neurocomputing*, 71(4-6):1103–1107, 2008. 36, 116, 119

[EZGBM07] U. K. Ecker, H. D. Zimmer, C. Groh-Bordin, and A. Mecklinger. Context effects on familiarity are familiarity effects of context—an electrophysiological study. *International Journal of Psychophysiology*, 64(2):146–156, 2007. 70, 82

[FGMR10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 20

[For14] D. Forsyth. Object detection with discriminatively trained part-based models. *Computer*, (2):6–7, 2014. 20

[GB10] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010. 18

[GC08] A. C. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, pages 1–8. IEEE, 2008. ix, 21, 22

[GC09] A. C. Gallagher and T. Chen. Understanding images of groups of people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2009)*, pages 256–263. IEEE, 2009. ix, 21

[GGDH18] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2018)*, pages 8359–8367, 2018. 19

[GJMG18] R. Gadde, V. Jampani, R. Marlet, and P. V. Gehler. Efficient 2d and 3d facade segmentation using auto-context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1273–1280, 2018. 19

[GMC+10] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 6

[GMSR18] S. Z. Gilani, A. Mian, F. Shafait, and I. Reid. Dense 3D face correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1584–1598, 2018. 15

[GMY17] Y. Gao, J. Ma, and A. L. Yuille. Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Transactions on Image Processing*, 26(5):2545–2560, 2017. 27

[GRSG+18] M. Günther, J. Ruiz-Sarmiento, C. Galindo, J. González-Jiménez, and J. Hertzberg. Context-aware 3D object anchoring for mobile robots. *Robotics and Autonomous Systems*, 110:12–32, 2018. 18

[GTM+16] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260, 2016. 20

[HA15] M. Hassaballah and S. Aly. Face recognition: Challenges, achievements and future directions. *IET Computer Vision*, 9(4):614–626, 2015. 25

[HBRZ19] M. Hassaballah, S. Bekhet, A. A. Rashed, and G. Zhang. Facial features detection and localization. In *Recent Advances in Computer Vision*, pages 33–59. Springer, 2019. 26

[HG14] P. N. Holey and V. T. Gaikwad. Google glass technology. *International Journal*, 2(3), 2014. 14

[HHM+14] J. Hochreiter, Z. Han, S. Z. Masood, S. Fonte, and M. Tappen. Exploring album structure for face recognition in online social networks. *Image and Vision Computing*, 32(10):751–760, 2014. 23

[HJS09] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *IEEE 12th International Conference on Computer Vision*, pages 237–244. IEEE, 2009. 20

[HK08] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision (ECCV2008)*, pages 30–43. Springer, 2008. 18

[HLM14] G. B. Huang and E. Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pages 14–003, 2014. 15

[HTME09] A. Hajiany, N. Taheri Makhsoos, and R. Ebrahimpour. View-independent face recognition with biological features based on mixture of experts. In *Ninth International Conference on Intelligent Systems Design and Applications (ISDA2009)*, pages 1425–1429. IEEE, 2009. 60

[JA09] R. Jafri and H. R. Arabnia. A survey of face recognition techniques. *Jips*, 5(2):41–68, 2009. 15

[JJNH91] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. 116, 119

[JS15] R. E. Jack and P. G. Schyns. The human face as a dynamic tool for social communication. *Current Biology*, 25(14):R621–R634, 2015. 51

[JWVMB11] R. Jenkins, D. White, X. Van Montfort, and A. M. Burton. Variability in photos of the same face. *Cognition*, 121(3):313–323, 2011. 73

[KBLG13] J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia. SAVI testbed: Control and management of converged virtual ICT resources. In *IFIP/IEEE International Symposium on Integrated Network Management*, pages 664–667. IEEE, 2013. 7

[KCT00] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG2000)*, pages 46–53. IEEE, 2000. 6

[KHAB09] A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker. Which faces to tag: Adding prior constraints into active learning. In *IEEE 12th International Conference on Computer Vision (ICCV2009)*, pages 1058–1065. IEEE, 2009. 22

[Kop11] R. Kop. The challenges to connectivist learning on open online networks: Learning experiences during a massive open online course. *The International Review of Research in Open and Distributed Learning*, 12(3):19–38, 2011. 52

[KR16] R. S. Kramer and K. L. Ritchie. Disguising superman: How glasses affect unfamiliar face matching. *Applied Cognitive Psychology*, 30(6):841–845, 2016. 26

[KVS+17] N. Kohli, M. Vatsa, R. Singh, A. Noore, and A. Majumdar. Hierarchical representation learning for kinship verification. *IEEE Transactions on Image Processing*, 26(1):289–302, 2017. 24

[LBB16] J. Luo, M. Boutell, and C. Brown. Pictures are not taken in a vacuum-an overview of exploiting context for semantic scene content understanding. *IEEE Signal Processing Magazine*, 23(2):101–114, 2016. 18

[LBL+16] H. Li, J. Brandt, Z. Lin, X. Shen, and G. Hua. A multi-level contextual model for person recognition in photo albums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*, pages 1297–1305, 2016. 23

[LCPM+19] M. Lado-Codesido, C. M. Pérez, R. Mateos, J. M. Olivares, and A. G. Caballero. Improving emotion recognition in schizophrenia with "voices": An on-line prosodic self-training. *PloS One*, 14(1):e0210816, 2019. 28

[LDB+10] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and Emotion*, 24(8):1377–1388, 2010. x, 72

[LG] A. Leon-Garcia. NSERC Strategic Network on Smart Applications on Virtual Infrastructure. University of Toronto, September 2011. ix, 35

[LHY04] Y. Li, J. Han, and J. Yang. Clustering moving objects. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDDM2004)*, pages 617–622. ACM, 2004. 34

[LKK+19] J. Lee, S. Kim, S. Kim, J. Park, and K. Sohn. Context-aware emotion recognition networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV2019)*, pages 10143–10152, 2019. 21

[LL12] Z. Lei and S. Z. Li. Learning discriminant face descriptor for face recognition. In *Asian Conference on Computer Vision (ACCV2012)*, pages 748–759. Springer, 2012. 16, 22

[LLP14] J. Z. Leibo, Q. Liao, and T. Poggio. Subtasks of unconstrained face recognition. In *International Conference on Computer Vision Theory and Applications (VISAPP2014)*, volume 2, pages 113–121. IEEE, 2014. 15

[LLZ+17] Y. Li, G. Lin, B. Zhuang, L. Liu, C. Shen, and A. van den Hengel. Sequential person recognition in photo albums with a recurrent network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017)*, July 2017. 24

[LLZZ15] J. Lu, V. E. Liong, X. Zhou, and J. Zhou. Learning compact binary face descriptor for face recognition. *IEEE transactions on pattern analysis and machine intelligence (PAMI2015)*, 37(10):2041–2056, 2015. 22

[LVH+17] Y. Li, N. Vishwamitra, H. Hu, B. P. Knijnenburg, and K. Caine. Effectiveness and users' experience of face blurring as a privacy protection for sharing photos via online social networks. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting (HFES2017)*, volume 61, pages 803–807. SAGE Publications Sage CA: Los Angeles, CA, 2017. 59

[LXT+18] J. Li, L. Xu, L. Tang, S. Wang, and L. Li. Big data in tourism research: A literature review. *Tourism Management*, 68:301–323, 2018. 2

[LYRR13] L. A. Libby, A. P. Yonelinas, C. Ranganath, and J. D. Ragland. Recollection and familiarity in schizophrenia: A quantitative review. *Biological Psychiatry*, 73(10):944–950, 2013. 70

[MB13] A. Mike Burton. Why has research in face recognition progressed so slowly? The importance of variability. *The Quarterly Journal of Experimental Psychology*, 66(8):1467–1485, 2013. 1, 85

[MB14] D. H. Mortensen and K. B. Bærentsen. Playing with nonverbal communication: Using grasp and facial direction to create adaptive interaction in a game. *Interacting with Computers*, 26(1):12–26, 2014. 52

[MCC+19] I. Masi, F.-J. Chang, J. Choi, S. Harel, J. Kim, K. Kim, J. Leksut, S. Rawls, Y. Wu, T. Hassner, et al. Learning pose-aware models for pose-invariant face recognition in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):379–393, 2019. 15

[MCRA18a] A. Mhenni, E. Cherrier, C. Rosenberger, and N. E. B. Amara. Adaptive biometric strategy using doddington zoo classification of user's keystroke dynamics. In *14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 488–493. IEEE, 2018. 26

[MCRA18b] A. Mhenni, E. Cherrier, C. Rosenberger, and N. E. B. Amara. Towards a secured authentication based on an online double serial adaptive mechanism of users' keystroke dynamics. In *International Conference on Digital Society and eGovernments (ICDS)*, 2018. 25

[ME14] S. Masoudnia and R. Ebrahimpour. Mixture of experts: A literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014. 7, 116, 119

[MFKC+17] J. Muñoz-Fernández, A. Knauss, L. Castañeda, M. Derakhshanmanesh, R. Heinrich, M. Becker, and N. Taherimakhsousi. Capturing Ambiguity in Artifacts to Support Requirements Engineering for Self-Adaptive Systems. In *RESACS: 3rd International Workshop on Requirements Engineering for Self-Adaptive & Cyber Physical System*, 2017. 12

[MHS+07] J. A. Martin, B. E. Hamilton, P. D. Sutton, S. J. Ventura, F. Menacker, S. Kirmeyer, and M. L. Munson. Births: final data for 2005. *National vital statistics reports*, 56(6):1–103, 2007. 48

[MMX+17] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong. Self-paced co-training. In *International Conference on Machine Learning (ICML2017)*, pages 2275–2284, 2017. 28

[MSG17] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017)*, pages 1396–1404, 2017. 18, 58

[MWY+16] Y. Ma, X. Wu, G. Yu, Y. Xu, and Y. Wang. Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery. *Sensors*, 16(4):446, 2016. 58

[NB05] M. Ní Bhrolcháin. The age difference at marriage in england and wales: a century of patterns and trends. *Population trends*, 120:7–14, 2005. 48

[NBNF17] A. Nambiar, A. Bernardino, J. C. Nascimento, and A. Fred. Context-aware person re-identification in the wild via fusion of gait and anthropometric features. In *12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 973–980. IEEE, 2017. 24

[OBT16] E. Ohn-Bar and M. M. Trivedi. Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):90–104, 2016. 85

[OMR18] G. Orrù, G. L. Marcialis, and F. Roli. An experimental investigation on self adaptive facial recognition algorithms using a long time span data

set. In *Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2018. 26

[OOF+16] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota. Autonomous terrain classification with co-and self-training approach. *IEEE Robotics and Automation Letters*, 1(2):814–819, 2016. 28

[PCMY15] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV2015)*, pages 1742–1750, 2015. 20

[PGDCL16] P. H. Pisani, R. Giot, A. C. De Carvalho, and A. C. Lorena. Enhanced template update: Application to keystroke dynamics. *Computers & Security*, 60:134–153, 2016. 26

[PGGS17] A. Poshtkohi, M. Ghaznavi-Ghoushchi, and K. Saghafi. The parvicursor infrastructure to facilitate the design of grid and cloud computing systems. *Computing*, 99(10):979–1006, 2017. 57

[PJXS16] D. K. Pal, F. Juefei-Xu, and M. Savvides. Discriminative invariant kernel features: a bells-and-whistles-free approach to unsupervised face recognition and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*, pages 5590–5599, 2016. 15

[PKR15] N. Poh, J. Kittler, and A. Rattani. Handling session mismatch by semi-supervised-based co-training scheme. In *Adaptive Biometric Systems*, pages 35–49. Springer, 2015. 28

[PLdC18] P. H. Pisani, A. C. Lorena, and A. C. de Carvalho. Adaptive biometric systems using ensembles. *IEEE Intelligent Systems*, 33(2):19–28, 2018. 26

[PMRR00] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI2000)*, 22(10):1090–1104, 2000. 6

[PPdCL17] P. H. Pisani, N. Poh, A. C. de Carvalho, and A. C. Lorena. Score normalization applied to adaptive biometric systems. *Computers & Security*, 70:565–580, 2017. 26

[PSM10] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV2010)*, pages 143–156. Springer, 2010. 19

[PVZ+15] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 15

[PWLG19] C. Peng, N. Wang, J. Li, and X. Gao. Dlface: Deep local descriptor for cross-modality face recognition. *Pattern Recognition*, 2019. 16

[Rat15] A. Rattani. Introduction to adaptive biometric systems. In *Adaptive Biometric Systems*, pages 1–8. Springer, 2015. 26

[RB09] A. Rabinovich and S. Belongie. Scenes vs. objects: a comparative study of two approaches to context based recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 92–99. IEEE, 2009. 19

[RB17] R. Ramachandra and C. Busch. Presentation attack detection methods for face recognition systems: a comprehensive survey. *ACM Computing Surveys (CSUR)*, 50(1):8, 2017. 15

[RBF+16] D. Ravì, M. Bober, G. M. Farinella, M. Guarnera, and S. Battiato. Semantic segmentation of images exploiting dct based features and random forest. *Pattern Recognition*, 52:260–273, 2016. 19

[RCR15] T. F. Rodriguez, S. Calhoon, and A. M. Reed. Metadata management and generation using perceptual features, February 10 2015. US Patent 8,953,908. 53

[Riv14] D. Rivolta. *Cognitive and Neural Aspects of Face Processing*, pages 19–40. Springer, 2014. 1, 86

[RMR11] A. Rattani, G. L. Marcialis, and F. Roli. Self adaptive systems: An experimental analysis of the performance over time. In *IEEE Workshop*

on Computational Intelligence in Biometrics and Identity Management (CIBIM2011), pages 36–43. IEEE, 2011. 28

[RPP13] O. Rudovic, M. Pantic, and I. Patras. Coupled gaussian processes for pose-invariant facial expression recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI2013)*, 35(6):1357–1369, 2013. 6

[RVM+13] L. M. Reder, L. W. Victoria, A. Manelis, J. M. Oates, J. M. Dutcher, J. T. Bates, S. Cook, H. J. Aizenstein, J. Quinlan, and F. Gyulai. Why it's easier to remember seeing a face we already know than one we don't: Preexisting memory representations facilitate memory formation. *Psychological Science*, 24(3):363–372, 2013. 82

[SAC16] P. Stróżak, D. Abedzadeh, and T. Curran. Separating the fn400 and n400 potentials across recognition memory experiments. *Brain Research*, 1635:41–60, 2016. 70

[Sat17] M. Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017. 56

[SAW94] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *1994 First Workshop on Mobile Computing Systems and Applications (MCSA1994)*, pages 85–90. IEEE, 1994. 1

[SBB02] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FG2002)*, pages 53–58. IEEE, 2002. 46

[Sch15] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. 16

[SD12] F. Siyahjani and G. Doretto. Learning a context aware dictionary for sparse representation. In *Asian Conference on Computer Vision (ACCV2012)*, pages 228–241. Springer, 2012. 21

[SGC15] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition.

IEEE Ttransactions on Pattern Analysis and Machine Intelligence, 37(6):1113–1133, 2015. 15

[SKP15] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 815–823, 2015. 46

[SKR+11] W. J. Scheirer, N. Kumar, K. Ricanek, P. N. Belhumeur, and T. E. Boult. Fusing with context: a bayesian approach to combining descriptive attributes. In *International Joint Conference on Biometrics (IJCB2011)*, pages 1–8. IEEE, 2011. 23

[SL06] Y. Song and T. Leung. Context-aided human recognition–clustering. In *European Conference on Computer Vision (ECCV2006)*, pages 382–395. Springer, 2006. 22

[SLZ+17] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3210–3218, 2017. 27

[SM10] S. M. Smith and I. Manzano. Video context-dependent recall. *Behavior Research Methods*, 42(1):292–301, 2010. 81

[SPG17] M. Sultana, P. P. Paul, and M. L. Gavrilova. Social behavioral information fusion in multimodal biometrics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99):1–12, 2017. 24

[SPMV13] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013. 20

[SRSZ18] D. Sivasankaran, M. Ragab, T. Sim, and Y. Zick. Context-aware fusion for continuous biometric authentication. In *International Conference on Biometrics (ICB)*, pages 233–240. IEEE, 2018. 24

[SS15] R. Shyam and Y. N. Singh. Face recognition using augmented local binary pattern and Bray Curtis dissimilarity metric. In *2nd International*

Conference on Signal Processing and Integrated Networks (SPIN), pages 779–784. IEEE, 2015. 26

[SSGC17] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64:141–158, 2017. 20

[STSG18] N. Sankaran, S. Tulyakov, S. Setlur, and V. Govindaraju. Metadata-based feature aggregation network for face recognition. In *International Conference on Biometrics (ICB)*, pages 118–123. IEEE, 2018. 24

[Sut16] S. Suthaharan. Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification*, pages 207–235. Springer, 2016. 19

[SZD08] Z. Stone, T. Zickler, and T. Darrell. Autotagging facebook: Social network context improves photo annotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008. 22

[TDH16] J. Tang, C. Deng, and G.-B. Huang. Extreme learning machine for multilayer perceptron. *IEEE Transactions on Neural Networks and Learning Systems*, 27(4):809–821, 2016. 16

[TEH09] N. Taherimakhsoos, R. Ebrahimpour, and A. Hajiany. Face recognition based on neuro-fuzzy system. *International Journal of Computer Science and Network*, 9(4):319–326, 2009. 36, 116

[TLLJ10] X. Tan, Y. Li, J. Liu, and L. Jiang. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *European Conference on Computer Vision (ECCV2010)*, pages 504–517. Springer, 2010. 36

[TLZR15] N. Tong, H. Lu, Y. Zhang, and X. Ruan. Salient object detection via global and local cues. *Pattern Recognition*, 48(10):3258–3267, 2015. 18

[TM14] N. Taherimakhsousi and H. A. Müller. Context-based face recognition for smart web tasking applications. In *IEEE World Congress on Services (SERVICES2014)*, pages 21–23. IEEE, 2014. 11, 51

[TM15a] N. Taherimakhsousi and H. A. Müller. Context-aware real-time video analytics. In *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering (CASCON2015)*, pages 223–226. IBM Corp., 2015. 7, 11, 119

[TM15b] N. Taherimakhsousi and H. A. Müller. Location-based face recognition using smart mobile device sensors. In *Proceedings of the International Conference on Computer and Information Science and Technology (CIST2015)*, pages 111–116. Avestia, 2015. 7, 11, 60

[TM16] N. Taherimakhsousi and H. A. Müller. CAVA: Context Aware Video Analytics with Decentralized Cloud on the SAVI Network. In *4th International IBM Cloud Academy Conference (ICACON 2016)*. IBM Corp., 2016. 7, 11

[TMMR15] P. Tuveri, V. Mura, G. L. Marcialis, and F. Roli. A classification-selection approach for self updating of face verification systems under stringent storage and computational requirements. In *International Conference on Image Analysis and Processing (ICIAP2015)*, pages 540–550. Springer, 2015. 28

[TOR01] D. Tsivilis, L. J. Otten, and M. D. Rugg. Context effects on the neural correlates of recognition memory: an electrophysiological study. *Neuron*, 31(3):497–505, 2001. 70, 71

[TP91a] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 16

[TP91b] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 38

[TvSA17] J. Tanha, M. van Someren, and H. Afsarmanesh. Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8(1):355–370, 2017. 28

[TYRW] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), pages=2746–2754, year=2015*. 59

[TYRW14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2014)*, pages 1701–1708, 2014. 15

[VDDP18] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018, 2018. 15

[VJ04] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 36

[VM13] N. M. Villegas and H. A. Müller. The SMARTERCONTEXT ontology and its application to the smart internet: A smarter commerce case study. In *The Personal Web*, pages 151–184. Springer, 2013. 51

[VR14] S. Verma and D. Rastogi. Mobile cloud computing: The potential, challenges & applications. In *International Conference of Advance Research and Innovation (ICARI2014)*, 2014. 52

[VSLC+17] M. F. Valstar, E. Sánchez-Lozano, J. F. Cohn, L. A. Jeni, J. M. Girard, Z. Zhang, L. Yin, and M. Pantic. Fera 2017-addressing head pose in the third facial expression recognition and analysis challenge. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 839–847. IEEE, 2017. 26

[Wan14] Y.-Q. Wang. An analysis of the Viola-Jones face detection algorithm. *Image Processing On Line*, 4:128–148, 2014. 58

[WFHP16] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016. 19

[WGLF10] G. Wang, A. Gallagher, J. Luo, and D. Forsyth. Seeing people in social context: Recognizing people and social relationships. In *European Conference on Computer Vision (ECCV2010)*, pages 169–182. Springer, 2010. 23

[WJ15] X. Wang and Q. Ji. Video event recognition with deep hierarchical context model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 4418–4427, 2015. 19

[WL13] L. Wolf and N. Levy. The SVM-Minus similarity score for video face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2013)*, pages 3523–3530, 2013. 51

[WRKN16] A. Wood, M. Rychlowska, S. Korb, and P. Niedenthal. Fashioning the face: sensorimotor simulation contributes to facial expression recognition. *Trends in cognitive sciences*, 20(3):227–240, 2016. 26

[XYW+16] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma. A hybrid vehicle detection method based on Viola-Jones and HOG+ SVM from UAV images. *Sensors*, 16(8):1325, 2016. 58

[YB18] A. W. Young and A. M. Burton. Are we face experts? *Trends in Cognitive Sciences*, 22(2):100–110, 2018. 85

[YBR06] N. M. Young, M. R. Bashar, and P. K. Rhee. Adaptive context-aware filter fusion for face recognition on bad illumination. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES2006)*, pages 532–541. Springer, 2006. 2, 26, 56

[YLQ+17] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu. Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI2017)*, 39(1):156–171, 2017. 6

[Yon02] A. P. Yonelinas. The nature of recollection and familiarity: A review of 30 years of research. *Journal of Memory and Language*, 46(3):441–517, 2002. 70

[YP04] G. Yovel and K. A. Paller. The neural basis of the butcher-on-the-bus phenomenon: When a face seems familiar but is not remembered. *Neuroimage*, 21(2):789–800, 2004. 70

[ZAZC19] Y. Zhou, L. An, H. Zou, and Z. Cao. Caccnn: context-aware cascaded cnn for face detection. In *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)*, volume 11069, page 110690G. International Society for Optics and Photonics, 2019. 92

[ZBL17] X.-Y. Zhang, Y. Bengio, and C.-L. Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, 61:348–360, 2017. 29

[ZCLZ03] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums. In *Proceedings of the Eleventh ACM International Conference on Multimedia (ACMMM2003)*, pages 355–358. ACM, 2003. 21

[ZCPR03] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458, 2003. 15

[ZDZ18] X. Zhang, S. Du, and Y. Zhang. Semantic and spatial co-occurrence analysis on object pairs for urban scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(8):2630–2643, 2018. 18

[ZGKB10] E. Zion-Golumbic, M. Kutas, and S. Bentin. Neural dynamics associated with semantic and episodic memory for faces: Evidence from multiple frequency bands. *Journal of Cognitive Neuroscience*, 22(2):263–277, 2010. 81

[ZK15] W. Zou and N. Komodakis. Harf: Hierarchy-associated rich features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV2015)*, pages 406–414, 2015. 19

[ZKSE16] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV2016)*, pages 597–613. Springer, 2016. 19

[ZLSR17] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detection. In *Proceedings of*

the IEEE International Conference on Computer Vision (ICCV2017), pages 589–598, 2017. 18

Appendix A

Mixture of Experts

This appendix presents our training algorithm Mixture of multi-layer perceptron experts, as one of the most applied implementations of MoE [ME14, EKEY08, TEH09], for our context aware face recognition.

Mixture of experts (MoE) is one of the most popular and interesting combining methods which have great potential to improve performance in machine learning. MoE was established based on the divide-and-conquer design principle in which the problem space is divided between a few neural network experts, supervised by a gating network [JJNH91]. In this method, the problem space is partitioned stochastically into a number of subspaces through an employed error function, experts become specialized on each subspace. This method uses a gating network to manage this process, which trains together with the experts. The gating network during the training of the experts with respect to differences in the experts' efficiencies in the different sub-spaces simultaneously co-operates in the partitioning of problem. In this method, instead of assigning a set of fixed combinational weights to the experts, the gating network is used to compute these weights dynamically from the inputs, according to the local efficiency of each expert.

Mixture of MLP Experts Training Algorithm

In our proposed mixture of experts' method for context aware face recognition, we use multilayer perceptron (MLP)s instead of linear networks for the experts and a gating network to improve the performance over a conventional MoE. In this implementation, each expert network is an MLP network with one hidden layer that computes an output O_i as a function of the input vector, x and weights of hidden and output

layers and a sigmoid activation function. The weights of MLPs are learned using the error back-propagation (BP) algorithm, in order to maximize the log likelihood of the training data given the parameters. For each expert i, the weights are updated according to the following rules:

$$\Delta w_y = \eta_e h_i (y - O_i)(O_i(1 - O_i))O_{h_i}^T \tag{A.1}$$

$$\Delta w_h = \eta_e h_i w_y^T (y - O_i)(O_i(1 - O_i))O_{h_i}(1 - O_{h_i})x_i \tag{A.2}$$

where η_e is the learning rate for the experts. w_h and w_y are the weights of input to hidden and hidden to output layer for the experts, respectively. $O_{h_i}^T$ is the transpose of O_{h_i} , the outputs of the hidden layer of expert.

$$h_i = \frac{g_i exp(-\frac{1}{2}(y - O_i)^T(y - O_i))}{\sum_j g_j exp(-\frac{1}{2}(y - O_j)^T(y - O_j))} \tag{A.3}$$

Finally, h_i is an estimation of the posterior probability that expert i can generate the desired output y. The error function of the gating network can be written as:

$$E_G = \frac{1}{2}\|(h - O_g)\|^2 \tag{A.4}$$

Based on this error function, the weights of the gating network in the proposed method are determined using the BP error algorithm according to the following rules:

$$\Delta w_{y,g} = \eta_g (h - O_g)(O_g(1 - O_g))O_{h,g}^T \tag{A.5}$$

$$\Delta w_{h,g} = \eta_g w_{y,g}^T (h - O_g)(O_g(1 - O_g))O_{h,g}(1 - O_{h,g})x_i \tag{A.6}$$

where η_g is the learning rate, and $w_{h,g}$ and $w_{y,g}$ are the weights of the inputs to hidden and hidden to output layers of the gating network, respectively. $O_{h,g}^T$ is the transpose of $O_{h,g}$, the outputs of the hidden layer of the gating network. In this learning procedure, the expert networks compete for each input pattern, while the gate network rewards the winner of each competition with stronger error feedback signals. Thus, over time, the gate partitions the input space in response to the expert's performance. The training step in the method is illustrated in Figure A.1.

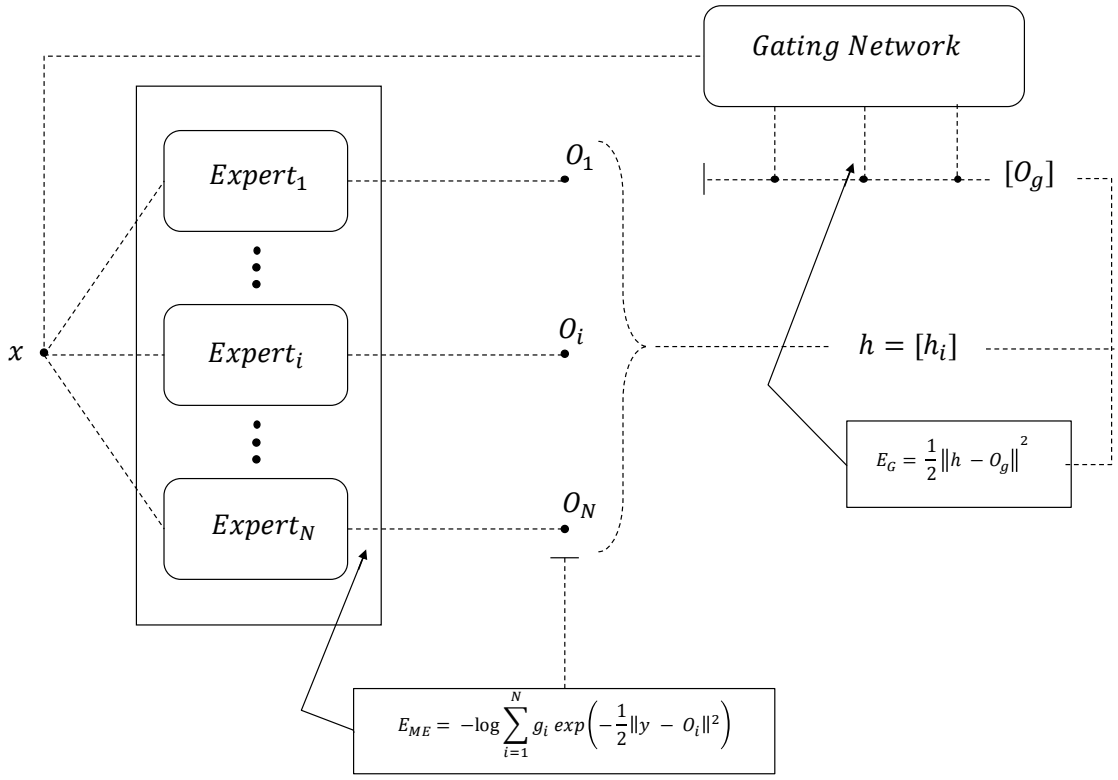

Figure A.1: Diagram for simultaneous training of the experts and gating network through the error functions. The experts compete to learn the training patterns, and the gating network mediates the competition.

We assume that each expert specializes in a specific area of the input space. The gating network assigns a weight g_i to the output of each of the experts, O_i. The gating network determines g_i as a function of the input vector x and a set of parameters such as the weights of its hidden and output layers. The activation function of the gating hidden layer is sigmoid, while in the output layer we use a linear activation function.Hence, it does not have the 0 to 1 range constraints that the sigmoid function does, so that the softmax function applied to the gating network outputs increases the diversity.

The g_i can be interpreted as an estimate of the prior probability that expert i can generate the desired output y. The gating network consists of two layers: the first is an MLP network, and the second is a softmax nonlinear operator. Thus, the gating network computes O_g, which is the output of the MLP layer of the gating network, then applies the softmax function to get:

$$g_i = \frac{exp(O_{gi})}{\sum_{j=1}^{N} exp(O_{gj})} \qquad i = 1, 2, ..., N \tag{A.7}$$

where N is the number of expert networks, and thus g_i are nonnegative and sum to 1. Finally, to combine the experts' outputs, the gate assigns a weight g_i as function of x to each of expert's output O_i, and the final mixed output of the entire network is:

$$O_T = \sum_{i}^{N} O_i g_i \qquad i = 1, 2, ..., N \tag{A.8}$$

Figure A.2 shows the testing step in Mixture of MLP-experts algorithm. As a result of good classification performance and transparency of the Mixture of MLP-experts method, it has been widely employed in many applications [ME14, EKEY08, TM15a] since Jacobs' proposal [JJNH91]. Considering the types of learners and training algorithms employed in the learning of experts and gating.

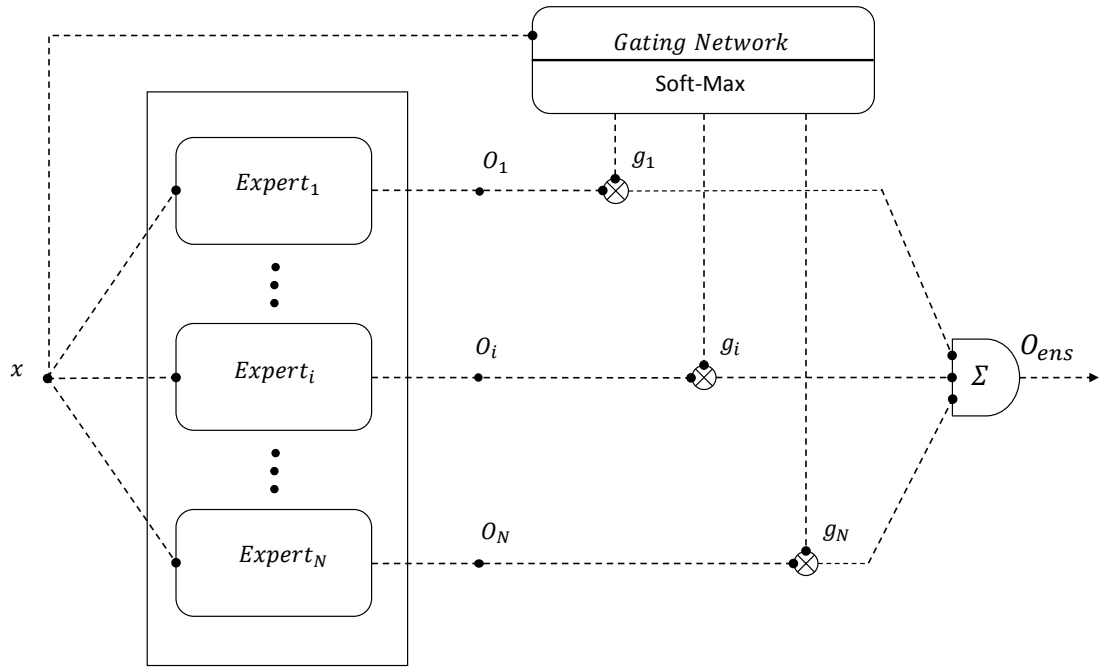

Figure A.2: Diagram for the testing step in mixture of expert method. In this step, the input x is given to the MLP experts and gating network, simultaneously and soft-max function is applied on the outputs of the gating network. The final output of ensemble system is calculated based on the weighted averaging of base MLP experts.

Appendix B

Adaptive Learning

Simplified View of the Source Code for PCA Feature Extraction

```
from sklearn.model\_selection import train\_test\_split
from matplotlib.image import imread
import numpy as np
import os
import pickle

dataset\_path = 'Preprocessed Images/'
dataset\_dir  = os.listdir(dataset\_path)

width  = 92
height = 112

train\_image\_names=dataset\_dir
train\_tensor    = np.ndarray(shape=(len(train\_image\_names)
   -1, height*width), dtype=np.float64)

target = list()
for i in range(len(train\_image\_names)):
        if train\_image\_names[i].endswith('jpg'):
```

```
19                      clas=int((int(os.path.splitext(train\_image\
                            _names[i])[0]))/13)
20                      print(clas)
21                      target.append(clas)
22                      img = imread(dataset\_path + train\_image\
                            _names[i])
23                      train\_tensor[i,:] = np.array(img, dtype='
                            float64').flatten()
24
25  X\_train, X\_test, y\_train, y\_test = train\_test\_split(
        train\_tensor, target, test\_size=0.2)
26  n\_components = 100
27
28  pca = PCA(n\_components=n\_components, whiten=True).fit(X\
        _train)
29
30  \# apply PCA transformation
31  X\_train\_pca = pca.transform(X\_train)
32  X\_test\_pca = pca.transform(X\_test)
33
34  with open('X\_Train.data', 'wb') as filehandle:
35          pickle.dump(X\_train\_pca, filehandle)
36  with open('Y\_Train.data', 'wb') as filehandle:
37          pickle.dump(y\_train, filehandle)
38  with open('X\_Test.data', 'wb') as filehandle:
39          pickle.dump(X\_test\_pca, filehandle)
40  with open('Y\_Test.data', 'wb') as filehandle:
41          pickle.dump(y\_test, filehandle)
```

Simplified View of the Source Code for Mixture of Expert for Context Aware Face Recognition

```
import keras
from keras.datasets import mnist
from keras.layers import Dense, Activation
from keras.models import Sequential
from keras.optimizers import RMSprop
from DenseMoE import DenseMoE

batch_size = 128
num_classes = 10
epochs = 20
num_experts_per_filter = 2

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

input_shape = x_train.shape[1:]

model = Sequential()
```

```
32 model.add(Dense(512, activation='relu', input_shape=
       input_shape))
33 model.add(DenseMoE(512, num_experts_per_filter,
       expert_activation='relu', gating_activation='softmax',
       input_shape=input_shape))
34 model.add(Dense(num_classes))
35 model.add(Activation('softmax'))
36
37 model.summary()
38
39 model.compile(loss='categorical_crossentropy',
40                   optimizer=RMSprop(),
41                   metrics=['accuracy'])
42
43 history = model.fit(x_train, y_train,
44                   batch_size=batch_size,
45                   epochs=epochs,
46                   verbose=1,
47                   validation_data=(x_test, y_test))
48
49 score = model.evaluate(x_test, y_test, verbose=0)
50 print('Test loss:', score[0])
51 print('Test accuracy:', score[1])
```

Simplified View of the Source Code for Dense Tensor for Mixture of Expert Adaptive Model

```
1
2  import numpy as np
3  import tensorflow as tf
4  from keras import backend as K
5  from keras import activations, initializers, regularizers,
       constraints
6  from keras.initializers import RandomUniform
7  from keras.engine.topology import Layer, InputSpec
8
9  class DenseMoE(Layer):
10         """Mixture-of-experts layer.
11 Implements: y = sum_{k=1}^K g(v_k * x) f(W_k * x)
12 # Arguments
13 units: Positive integer, dimensionality of the output space

14 n_experts: Positive integer, number of experts (K)
15 expert_activation:Activation function for the expert model(f)
16 gating_activation:Activation function for the gating model(g)
17 use_expert_bias: Whether to use biases in the expert model
18 use_gating_bias: Whether to use biases in the gating model
19 expert_kernel_initializer_scale: Float, scale of Glorot
       uniform initialization for expert model weights
20 gating_kernel_initializer_scale: Float, scale of Glorot
       uniform initialization for gating model weights
21 expert_bias_initializer: Initializer for the expert biases
22 gating_bias_initializer: Initializer fot the gating biases
23 expert_kernel_regularizer: Regularizer for the expert weights
24 gating_kernel_regularizer: Regularizer for the gating weights
25 expert_bias_regularizer: Regularizer for the expert biases
26 gating_bias_regularizer: Regularizer for the gating biases
27 expert_kernel_constraint: Constraints for the expert weights
```

```
28 gating_kernel_constraint: Constraints for the gating weights
29 expert_bias_constraint: Constraints for the expert biases.
30 gating_bias_constraint: Constraints for the gating biases.
31 activity_regularizer: Activity regularizer.
32 # Input shape
33 nD tensor with shape: (batch_size, ..., input_dim)
34 The most common situation would be a 2D input with shape (
       batch_size, input_dim)
35 # Output shape
36 nD tensor with shape: (batch_size, ..., units)
37 For example, for a 2D input with shape (batch_size, input_dim
       ), the output would have shape (batch_size, units).
38         """
39         def __init__(self, units,
40                 n_experts,
41                 expert_activation=None,
42                 gating_activation=None,
43                 use_expert_bias=True,
44                 use_gating_bias=True,
45                 expert_kernel_initializer_scale=1.0,
46                 gating_kernel_initializer_scale=1.0,
47                 expert_bias_initializer='zeros',
48                 gating_bias_initializer='zeros',
49                 expert_kernel_regularizer=None,
50                 gating_kernel_regularizer=None,
51                 expert_bias_regularizer=None,
52                 gating_bias_regularizer=None,
53                 expert_kernel_constraint=None,
54                 gating_kernel_constraint=None,
55                 expert_bias_constraint=None,
56                 gating_bias_constraint=None,
57                 activity_regularizer=None,
58                 **kwargs):
59
```

```
60          if 'input_shape' not in kwargs and 'input_dim' in
               kwargs:
61                  kwargs['input_shape'] = (kwargs.pop('
                       input_dim'),)
62          super(DenseMoE, self).__init__(**kwargs)
63          self.units = units
64          self.n_experts = n_experts
65
66          self.expert_activation = activations.get(
               expert_activation)
67          self.gating_activation = activations.get(
               gating_activation)
68
69          self.use_expert_bias = use_expert_bias
70          self.use_gating_bias = use_gating_bias
71
72          self.expert_kernel_initializer_scale =
               expert_kernel_initializer_scale
73          self.gating_kernel_initializer_scale =
               gating_kernel_initializer_scale
74
75          self.expert_bias_initializer = initializers.get(
               expert_bias_initializer)
76          self.gating_bias_initializer = initializers.get(
               gating_bias_initializer)
77
78          self.expert_kernel_regularizer = regularizers.get(
               expert_kernel_regularizer)
79          self.gating_kernel_regularizer = regularizers.get(
               gating_kernel_regularizer)
80
81          self.expert_bias_regularizer = regularizers.get(
               expert_bias_regularizer)
82          self.gating_bias_regularizer = regularizers.get(
               gating_bias_regularizer)
```

```
83
84          self.expert_kernel_constraint = constraints.get(
                expert_kernel_constraint)
85          self.gating_kernel_constraint = constraints.get(
                gating_kernel_constraint)
86
87          self.expert_bias_constraint = constraints.get(
                expert_bias_constraint)
88          self.gating_bias_constraint = constraints.get(
                gating_bias_constraint)
89
90          self.activity_regularizer = regularizers.get(
                activity_regularizer)
91          self.input_spec = InputSpec(min_ndim=2)
92          self.supports_masking = True
93
94          def build(self, input_shape):
95                  assert len(input_shape) >= 2
96                  input_dim = input_shape[-1]
97                  expert_init_lim = np.sqrt(3.0*self.
                        expert_kernel_initializer_scale / (max(1.,
                         float(input_dim + self.units) / 2)))
98                  gating_init_lim = np.sqrt(3.0*self.
                        gating_kernel_initializer_scale / (max(1.,
                         float(input_dim + 1) / 2)))
99
100                 self.expert_kernel = self.add_weight(shape=(
                        input_dim, self.units, self.n_experts),
101                 initializer=RandomUniform(minval=-
                        expert_init_lim,maxval=expert_init_lim),
102                 name='expert_kernel',
103                 regularizer=self.expert_kernel_regularizer,
104                 constraint=self.expert_kernel_constraint)
105
```

```
106                 self.gating_kernel = self.add_weight(shape=(
                        input_dim, self.n_experts),
107                 initializer=RandomUniform(minval=-
                        gating_init_lim,maxval=gating_init_lim),
108                 name='gating_kernel',
109                 regularizer=self.gating_kernel_regularizer,
110                 constraint=self.gating_kernel_constraint)
111
112                 if self.use_expert_bias:
113                         self.expert_bias = self.add_weight(
                                shape=(self.units, self.n_experts)
                                ,
114                         initializer=self.
                                expert_bias_initializer,
115                         name='expert_bias',
116                         regularizer=self.
                                expert_bias_regularizer,
117                         constraint=self.
                                expert_bias_constraint)
118                 else:
119                         self.expert_bias = None
120
121                 if self.use_gating_bias:
122                         self.gating_bias = self.add_weight(
                                shape=(self.n_experts,),
123                         initializer=self.
                                gating_bias_initializer,
124                         name='gating_bias',
125                         regularizer=self.
                                gating_bias_regularizer,
126                         constraint=self.
                                gating_bias_constraint)
127                 else:
128                         self.gating_bias = None
129
```

```
130                 self.input_spec = InputSpec(min_ndim=2, axes
                        ={-1: input_dim})
131                 self.built = True
132
133         def call(self, inputs):
134
135                 expert_outputs = tf.tensordot(inputs, self.
                        expert_kernel, axes=1)
136                 if self.use_expert_bias:
137                         expert_outputs = K.bias_add(
                                expert_outputs, self.expert_bias)
138                 if self.expert_activation is not None:
139                         expert_outputs = self.
                                expert_activation(expert_outputs)
140
141                 gating_outputs = K.dot(inputs, self.
                        gating_kernel)
142                 if self.use_gating_bias:
143                         gating_outputs = K.bias_add(
                                gating_outputs, self.gating_bias)
144                 if self.gating_activation is not None:
145                         gating_outputs = self.
                                gating_activation(gating_outputs)
146
147                                 output = K.sum(expert_outputs
                                        * K.repeat_elements(K.
                                    expand_dims(gating_outputs
                                    , axis=1), self.units,
                                    axis=1), axis=2)
148
149                 return output
150
151         def compute_output_shape(self, input_shape):
152                 assert input_shape and len(input_shape) >= 2
153                 assert input_shape[-1]
```

```
154                 output_shape = list(input_shape)
155                 output_shape[-1] = self.units
156                 return tuple(output_shape)
157
158         def get_config(self):
159                 config = {
160                 'units': self.units,
161                 'n_experts':self.n_experts,
162                 'expert_activation': activations.serialize(
                      self.expert_activation),
163                 'gating_activation': activations.serialize(
                      self.gating_activation),
164                 'use_expert_bias': self.use_expert_bias,
165                 'use_gating_bias': self.use_gating_bias,
166                 'expert_kernel_initializer_scale': self.
                      expert_kernel_initializer_scale,
167                 'gating_kernel_initializer_scale': self.
                      gating_kernel_initializer_scale,
168                 'expert_bias_initializer': initializers.
                      serialize(self.expert_bias_initializer),
169                 'gating_bias_initializer': initializers.
                      serialize(self.gating_bias_initializer),
170                 'expert_kernel_regularizer': regularizers.
                      serialize(self.expert_kernel_regularizer),
171                 'gating_kernel_regularizer': regularizers.
                      serialize(self.gating_kernel_regularizer),
172                 'expert_bias_regularizer': regularizers.
                      serialize(self.expert_bias_regularizer),
173                 'gating_bias_regularizer': regularizers.
                      serialize(self.gating_bias_regularizer),
174                 'expert_kernel_constraint': constraints.
                      serialize(self.expert_kernel_constraint),
175                 'gating_kernel_constraint': constraints.
                      serialize(self.gating_kernel_constraint),
```

```
176                 'expert_bias_constraint': constraints.
                      serialize(self.expert_bias_constraint),
177                 'gating_bias_constraint': constraints.
                      serialize(self.gating_bias_constraint),
178                 'activity_regularizer': regularizers.
                      serialize(self.activity_regularizer)
179                                 }
180         base_config = super(DenseMoE, self).get_config()
181         return dict(list(base_config.items()) + list(config.
              items()))
```

Appendix C

Context-based Face Recognition Case Study

Case Study Questionnaire

Context-based Face Recognition

Welcome to this experimental recognition memory study with which we want to investigate the potential of the contextual information on face recognition accuracy and response time. Thank you for filling it all out and participating.
Please note that after filling it out, you are participating to a computer based study.

About you

1. **Your name:**.....
2. **How old are you?*** I am ..... years old.
3. **Gender** .....
4. **Are you in a good mood and healthy right now?** ◯ Yes ◯ No
5. **You are a** ◯ Right-handed ◯ left-handed
6. **Do you have a normal or corrected-to-normal vision?** ◯ Yes ◯ No

Computer based Memory Test Procedure

You are going to participate to two tests. Each test study has a trial included a 1-s gray fixation cross at eye level on a black background followed by a 1.5-s face presentation with contextual information related to the face. Try to remember the face for a subsequent memory tests. Each test trial included 1.5-s fixation, 0.5-s face, fixation, and then response.

Print Name **Signature** **Date**

Simplified View of the Source Code for the Task 1 and 2

```
AssertOpenGL;         % Check if all needed parameters given:
if nargin < 2
        error('Must provide required input parameters "subNo" and "hand"!');
end

rand('state',sum(100*clock));

supported operating systems

KbName('UnifyKeyNames');

advancestudytrial=KbName('n');

if (hand==1)
        oldresp=KbName('c'); % "old" response via key 'c'
        newresp=KbName('m'); % "new" response via key 'm'
else
        oldresp=KbName('m'); % Keys are switched in this case.
        newresp=KbName('c');
end
datafilename = strcat('ContextbasedFaceRecognition_',num2str(subNo),'.dat'); % name of data file to write to
studyfilename = 'studylist.txt';    % study list
testfilename  = 'testlist.txt';     % test list
% files from a previous subject/session (except for subject numbers > 99):
if subNo<99 && fopen(datafilename, 'rt')~=-1
        fclose('all');
```

```
27          error('Result data file already exists! Choose a
               different subject number.');
28  else
29          datafilepointer = fopen(datafilename,'wt'); % open
               ASCII file
30  end
31  %%%%%%%%%%%%%%%%%%%%%
32  % experiment
33  %%%%%%%%%%%%%%%%%%%%%
34  try
35          screens=Screen('Screens');
36          screenNumber=max(screens);
37          % Hide the mouse cursor:
38          HideCursor;
39          % Returns as default the mean gray value of screen:
40          gray=GrayIndex(screenNumber);
41          [w, wRect]=Screen('OpenWindow',screenNumber, gray);
42          Screen('TextSize', w, 32);
43          KbCheck;
44          WaitSecs(0.1);
45          GetSecs;
46          % Set priority for script execution to realtime
               priority:
47          priorityLevel=MaxPriority(w);
48          Priority(priorityLevel);
49          % run through study and test phase
50          for phase=1:2 % 1 is study phase, 2 is test phase
51                  % Setup experiment variables etc. depending
                       on phase:
52                  if phase==1 % study phase
53                          % define variables for current phase
54                          phaselabel='study';
55                          duration=2.000; % Duration of study
                               image presentation in secs.
56                          trialfilename=studyfilename;
```

```
57                  message = 'Study phase ...\nstudy
                        each picture try to remember faces
                         in the picture ... \npress _n_
                        when it disappears ...\n... press
                        mouse button to begin ...';
58              else          % test phase
59                  % define variables
60                  phaselabel='test';
61                  duration= 1.00;  %sec
62                  trialfilename=testfilename;
63                  % write message to subject
64                  str=sprintf('Press _%s_ for OLD face
                        and _%s_ for NEW face\n',KbName(
                        oldresp),KbName(newresp));
65                  message = ['Test phase ...\n' str '
                        ... press mouse button to begin
                        ...'];
66              end
67              DrawFormattedText(w, message, 'center', '
                    center', WhiteIndex(w));
68              % Update the display to show the instruction
                    text:
69              Screen('Flip', w);
70              % Wait for mouse click:
71              GetClicks(w);
72              % Clear screen to background color (our 'gray
                    ' as set at the
73              % beginning):
74              Screen('Flip', w);
75              % Wait a second before starting trial
76              WaitSecs(2.000);
77
78              [ objnumber, objname, objtype ] = textread(
                    trialfilename,'%d %s %d');
79
```

```
80                  % Randomize order of list
81                  ntrials=length(objnumber);  % get number of
                        trials
82                  randomorder=randperm(ntrials);
83                  objnumber=objnumber(randomorder);  % need to
                        randomize each list!
84                  objname=objname(randomorder);
85                  objtype=objtype(randomorder);
86                  % loop through trials
87                  for trial=1:ntrials
88                          % wait a bit between trials
89                          WaitSecs(0.500);
90
91                          [KeyIsDown, endrt, KeyCode]=KbCheck;
92
93                          % read stimulus image into matlab
                                matrix 'imdata':
94                          stimfilename=strcat('stims/',char(
                                objname(trial))); % assume stims
                                are in subfolder "stims"
95                          imdata=imread(char(stimfilename));
96
97                          % make texture image out of image
                                matrix 'imdata'
98                          tex=Screen('MakeTexture', w, imdata);
99
100                         Screen('DrawTexture', w, tex);
101
102                         % Show stimulus on screen at next
                                possible display refresh cycle,
103                         % and record stimulus onset time in '
                                startrt':
104                         [VBLTimestamp startrt]=Screen('Flip',
                                w);
105
```

```
106                 while (GetSecs - startrt)<=duration
107                     if ( phase==2 ) % if test
                            phase
108                             if ( KeyCode(oldresp)
                                    ==1 || KeyCode(
                                    newresp)==1 )
109                                 break;
110                             end
111                             [KeyIsDown, endrt,
                                    KeyCode]=KbCheck;
112                     end
113                     WaitSecs(0.001);
114                 end
115                 Screen('Flip', w);
116                 if ( phase==1 ) % study phase
117                     while (KeyCode(
                            advancestudytrial)==0)
118                             [KeyIsDown, endrt,
                                    KeyCode]=KbCheck;
119                             WaitSecs(0.001);
120                     end
121                 end
122
123                 if ( phase==2 ) % test phase
124                     while ( KeyCode(oldresp)==0
                            && KeyCode(newresp)==0 )
125                             [KeyIsDown, endrt,
                                    KeyCode]=KbCheck;
126                             WaitSecs(0.001);
127                     end
128                 end
129
130                 % compute response time
131                 rt=round(1000*(endrt-startrt));
132
```

```
133                         % compute accuracy
134                         if (phase==1 ) % study phase
135                                 ac=1;
136                         else            % test phase
137                                 ac=0;
138                                 % code correct if old-
                                        response with old stimulus
                                        ,
139                                 % new-response with new
                                        stimulus, or study phase
140                                 if ( (KeyCode(oldresp)==1 &&
                                        objtype(trial)==1) || (
                                        KeyCode(newresp)==1 &&
                                        objtype(trial)==2) )
141                                         ac=1;
142                                 end
143                         end
144                         resp=KbName(KeyCode); % get key
                                pressed by subject
145                         % Write trial result to file:
146                         fprintf(datafilepointer,'%i %i %s %i
                                %s %i %s %i %i %i\n', ...
147                                 subNo, ...
148                                 hand, ...
149                                 phaselabel, ...
150                                 trial, ...
151                                 resp, ...
152                                 objnumber(trial), ...
153                                 char(objname(trial)), ...
154                                 objtype(trial), ...
155                                 ac, ...
156                                 rt);
157                 end % for trial loop
158         end % phase loop
159         % priority:
```

```
160             Screen('CloseAll');
161             ShowCursor;
162             fclose('all');
163             Priority(0);
164             % End of experiment:
165             return;
166     catch
167             Screen('CloseAll');
168             ShowCursor;
169             fclose('all');
170             Priority(0);
171             % Output the error message that describes the error:
172             psychrethrow(psychlasterror);
173     end
```