

# Face De-identification for Privacy Protection

Zongji Sun

A thesis submitted to the University of Hertfordshire  
in partial fulfilment of the requirements for the degree of  
*Doctor of Philosophy*

October 2018



## Acknowledgements

Many people have supported me during my PhD research, and I would like to thank them all.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr Lily Meng, for her continuous guidance, inspiration and support during my PhD research. I appreciate her valuable advice on my research and this thesis.

I would like to thank the members in my supervision team, Prof Aladdin Ariyaeinia and Dr Georgios Pissanidis, for their helpful discussions and advice.

I am deeply grateful to Prof Zheng-Hua Tan and Xiaodong Duan for their arrangement and collaboration to complete the STSM during my visit at Aalborg University.

I would like to thank ICT COST Action IC1206 “De-identification for privacy protection in multimedia content” for supporting me to attend conferences, group meetings and training schools.

I also owe much to my kind friends and colleagues at the Engineering and Technology department including Dr Hock Gan, Dr Yangjun Chen, Longsheng Yu and Rowan Karadaghi.

Last but not least, I am grateful to my dear family. No words can express my gratitude to them.



## Abstract

The ability to record, store and analyse images of faces economically, rapidly and on a vast scale brings people's attention to privacy. The current privacy protection approaches for face images are mainly through masking, blurring or black-out which, however, removes data utilities along with the identifying information. As a result, these ad hoc methods are hardly used for data publishing or in further researches. The technique of de-identification attempts to remove identifying information from a dataset while preserving the data utility as much as possible. The research on de-identify structured data has been established while it remains a challenge to de-identify unstructured data such as face data in images and videos. The  $k$ -Same face de-identification was the first method that attempted to use an established de-identification theory,  $k$ -anonymity, to de-identify a face image dataset. The  $k$ -Same face de-identification is also the starting point of this thesis.

Re-identification risk and data utility are two incompatible aspects in face de-identification. The focus of this thesis is to improve the privacy protection performance of a face de-identification system while providing data utility preserving solutions for different application scenarios. This thesis first proposes the  $k$ -Same-furthest face de-identification method which introduces the wrong-map protection to the  $k$ -Same-M face de-identification, where the identity loss is maximised by replacing an original face with the face that has the least similarity to it.

The data utility of face images has been considered from two aspects in this thesis, the dataset-wise data utility such as data distribution of the data set and the individual-wise data utility such as the facial expression in an individual image. With the aim to preserve the diversity of a face image dataset, the  $k$ -Diff-furthest face de-identification method is proposed, which extends the  $k$ -Same-furthest method and can provide the wrong-map protection.

With respect to the data utility of an individual face image, the visual quality and the preservation of facial expression are discussed in this thesis. A method to merge the isolated de-identified face region and its original image background is presented. The described method can increase the visual quality of a de-identified face image in

terms of fidelity and intelligibility. A novel solution to preserving facial expressions in de-identified face images is presented, which can preserve not only the category of facial expressions but also the intensity of face Action Units.

Finally, an integration of the Active Appearance Model (AAM) and Generative Adversarial Network (GAN) is presented, which achieves the synthesis of realistic face images with shallow neural network architectures.

# Table of contents

<b>List of figures</b>	<b>xi</b>
<b>List of tables</b>	<b>xv</b>
<b>Glossary</b>	<b>xvii</b>
<b>Acronyms</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	4
1.2 Research outputs . . . . .	5
1.3 Thesis outline . . . . .	6
<b>2 Background</b>	<b>7</b>
2.1 De-identification on structured data . . . . .	7
2.2 Ad hoc face de-identification . . . . .	9
2.3 <i>k</i> -Same face de-identification . . . . .	15
2.3.1 <i>k</i> -Same . . . . .	15
2.3.2 Model-based <i>k</i> -Same . . . . .	19
2.3.3 <i>k</i> -Same variants for utility preservation . . . . .	19
2.4 Other face de-identification approaches . . . . .	22
<b>3 Related work</b>	<b>25</b>
3.1 Face appearance models . . . . .	25
3.1.1 Principal component analysis . . . . .	25
3.1.2 Statistical models for face appearance . . . . .	26
3.2 Facial landmark detection . . . . .	32
3.3 Face recognition . . . . .	35
3.3.1 Feature matching . . . . .	35

3.3.2	Face features . . . . .	36
3.4	Facial expression synthesis . . . . .	41
<b>4</b>	<b>Face de-identification with cluster swapping</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	$k$ -Same-furthest . . . . .	44
4.3	Experiments . . . . .	51
4.3.1	Dataset . . . . .	51
4.3.2	Test design . . . . .	51
4.3.3	Results and discussions . . . . .	51
4.4	Conclusions . . . . .	54
<b>5</b>	<b>Distinguishable face de-identification</b>	<b>55</b>
5.1	Introduction . . . . .	55
5.2	$k$ -Diff-furthest face de-identification . . . . .	56
5.2.1	The proposed algorithm . . . . .	56
5.2.2	Wrong-map protection in $k$ -Diff-furthest . . . . .	58
5.3	Discussion on single-member clusters . . . . .	61
5.3.1	Avoid the pair of single-member clusters . . . . .	61
5.3.2	Generate random feature vectors . . . . .	62
5.4	Experiments . . . . .	63
5.4.1	Dataset . . . . .	63
5.4.2	Re-identification risk of $k$ -Diff-furthest . . . . .	64
5.4.3	Diversity of the de-identified face set . . . . .	65
5.5	Conclusions . . . . .	68
<b>6</b>	<b>Visual quality and re-identification risk in real-world application</b>	<b>69</b>
6.1	Introduction . . . . .	69
6.2	Merging a de-identified face with its original background . . . . .	70
6.3	Re-identification risk test . . . . .	74
6.3.1	Types of attacks . . . . .	75
6.3.2	Evaluation setup . . . . .	75
6.3.3	Results and discussion . . . . .	77
6.4	Image background attack to face de-identification system . . . . .	78
6.5	Conclusions . . . . .	79



<b>7</b>	<b>Open-set de-identification of faces in videos with preservation of expressions</b>	<b>81</b>
7.1	Introduction . . . . .	81
7.2	Facial expression transfer . . . . .	82
7.2.1	Semantic analogies in the face feature space . . . . .	83
7.2.2	Face de-identification with facial expression preservation . . . . .	84
7.3	Face de-identification in videos . . . . .	85
7.3.1	Calculation of identity shift . . . . .	86
7.3.2	Feature normalisation and retainment of the original head pose . . . . .	87
7.3.3	Implementing texture transfer in pixel space . . . . .	88
7.4	Evaluate face de-identification with FET on still images . . . . .	89
7.4.1	Evaluation of re-identification risk . . . . .	90
7.4.2	Evaluation of data utility . . . . .	91
7.5	Evaluate face de-identification with FET on video data . . . . .	92
7.5.1	Preservation of facial expressions . . . . .	94
7.5.2	Identity consistency of the de-identified videos . . . . .	96
7.5.3	Privacy protection performance . . . . .	96
7.6	Conclusions . . . . .	98
<b>8</b>	<b>Appearance model-based GAN for face de-identification</b>	<b>99</b>
8.1	Introduction . . . . .	99
8.2	Generative Adversarial Networks . . . . .	100
8.3	Appearance Model-based GAN . . . . .	104
8.3.1	Network architecture . . . . .	105
8.3.2	Appearance model parameter pre-processing . . . . .	105
8.3.3	Attribute-controlled face synthesis . . . . .	106
8.3.4	Generating the identity pool with an AMGAN . . . . .	108
8.4	Experiments . . . . .	109
8.4.1	Datasets . . . . .	109
8.4.2	Generating random faces with the AMGAN without conditions . . . . .	110
8.4.3	Generating face images with attributes . . . . .	112
8.4.4	Re-identification risks . . . . .	114
8.5	Conclusions . . . . .	114
<b>9</b>	<b>Conclusions and future work</b>	<b>121</b>
9.1	Summary . . . . .	121
9.2	Conclusions . . . . .	123

9.3 Future work . . . . .	124
<b>Bibliography</b>	<b>127</b>

# List of figures

2.1	Ad hoc face de-identification examples . . . . .	11
2.2	Face detection rates on the de-identified face images with ad hoc methods . . . . .	13
2.3	Examples of face detection on de-identified face images with ad hoc methods . . . . .	14
2.4	Re-identification risk on toy data . . . . .	17
2.5	The clustering results of $k$ -Same and MDAV-generic . . . . .	18
2.6	Face de-identification examples of $k$ -Same and $k$ -Same-M . . . . .	20
2.7	Comparison of data utility of smiling between $k$ -Same and $k$ -Same-Select	20
3.1	PCA variances of face shape models used in following experiments .	27
3.2	Face shape of 111 triangles defined by 68 vertices . . . . .	28
3.3	Face shape models . . . . .	29
3.4	Face texture models . . . . .	30
3.5	The comparison of original faces and their reconstructed faces from the corresponding appearance model parameters . . . . .	31
3.6	Comparison of AAM face features and CNN learned face features . .	41
4.1	An iteration of the $k$ -Same-furthest face de-identification process with an example data set . . . . .	45
4.2	Example face images from the IMM dataset . . . . .	51
4.3	Re-identification risks of de-identified faces . . . . .	52
4.4	De-identification results of the proposed algorithm with data utility preservation. . . . .	53
4.5	De-identification results of the proposed algorithm without data utility preservation. . . . .	53

5.1	The $k$ -Diff-furthest face de-identification swaps original faces between a pair of clusters in order to retain the diversity of the original face set in the de-identified face set. . . . .	57
5.2	Illustration of Theorem 4 in a 2D space . . . . .	58
5.3	Examples of face images in the testing set of re-identification test . . . . .	64
5.6	Example faces showing relationships between the face feature distance and the visual difference . . . . .	68
6.1	Overview of merging a de-identified face with its original background	70
6.2	Illustration for guided interpolation notations . . . . .	71
6.3	Blending results of de-identified face regions . . . . .	73
6.4	Examples of faces (FERET dataset) used in re-identification tests . . . . .	76
6.5	Example of original and de-identified faces of one person with 200 pixels and 300 pixels cropping sizes . . . . .	77
6.6	Examples of inverse crop face images used in background attack experiment . . . . .	79
7.1	An example showing the AUs associated with the expression of pain	83
7.2	An example of transfer the expression from the original face to its de-identified face . . . . .	85
7.3	Overview of face de-identification with facial motion and expression preservation in videos . . . . .	86
7.4	Comparison of texture transfer in feature and pixel space . . . . .	89
7.5	Re-identification risk comparison of $k$ -Same-M-Select and $k$ -Same-furthest with FET . . . . .	90
7.6	Comparison of facial expression classification on the de-identified faces	91
7.7	Comparison of AU detection on original faces and de-identified faces	93
7.8	Examples of the cropped and resized face images used in the identity consistency test . . . . .	96
8.1	Overview of the AMGAN . . . . .	104
8.2	Data distribution of 500 random samples in the first two dimensions of the shape and texture models . . . . .	111
8.3	Example face images reconstructed from LSGAN generated appearance model parameters . . . . .	112
8.4	Example images of the generated identity pool . . . . .	114
8.5	Re-identification risks of UNBC-McMaster video frames de-identified with GAN generated identity pool. . . . .	115

---

8.6	Example face images when linear interpolation of ‘male’ attribute (−1 to 1) . . . . .	116
8.7	Example face images when linear interpolation of ‘youth’ attribute (−1 to 1) . . . . .	117
8.8	Example face images when linear interpolation of ‘smiling’ attribute (−1 to 1) . . . . .	118
8.9	Shape parameters of cGAN using LSGAN, WGAN and WGAN-GP losses trained on CelebA and LFW dataset . . . . .	119
8.10	Shape parameters of GAN using LSGAN, WGAN and WGAN-GP losses with auxiliary loss trained on CelebA and LFW dataset . . . . .	120



# List of tables

3.1	Types of similarity metrics that are used in feature matching . . . . .	35
5.1	Feature distances statistics . . . . .	66
6.1	Re-identification risk of the de-identified face with and without background deformation . . . . .	74
6.2	Key parameters of the face recognition methods used in the evaluation experiments . . . . .	74
6.3	Re-identification risk of $300 \times 300$ inverse cropped face images . . . . .	78
6.4	Naïve recognition rates of original 'fb' faces and de-identified 'fb' faces against original 'fa' . . . . .	80
6.5	Reverse recognition rates of original 'fb' faces and de-identified 'fb' faces against original 'fa' . . . . .	80
7.1	Average absolute difference in AU intensity between original frames and corresponding de-identified frames . . . . .	95
7.2	Rank-1 re-identification risk of the de-identified video frames . . . . .	98





# Glossary

## **aggregated data**

Statistical data about several individuals that has been combined to show general trends or values without identifying individuals within the data.

## **de-identification**

“General term for any process of removing the association between a set of identifying data and the data subject.” ISO/TS 25237:2008(E).

## **de-identified information**

“records that have had enough PII removed or obscured such that the remaining information does not identify an individual and there is no reasonable basis to believe that the information can be used to identify an individual” (SP800-122).

## **identification**

“Process of using claimed or observed attributes of an entity to single out the entity among other entities in a set of identities” (ISO/TS 25237:2008).

## **identifier**

“Information used to claim an identity, before a potential corroboration by a corresponding authenticator” (ISO/TS 25237:2008).

## **personal data**

“Any information relating to an identified or identifiable natural person (data subject)” (ISO/TS 25237:2008).

## **personally identifiable information (PII)**

“Any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual’s

identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information." (SP800-122).

**privacy**

"Freedom from intrusion into the private life or affairs of an individual when that intrusion results from undue or illegal gathering and use of data about that individual" (ISO/IEC 2382-8:1998, definition 08-01-23).

**re-identification**

The process of analysing data or combining it with other data with the result that individuals become identifiable.

**trusted data recipient**

An entity that has limited access to the data that it receives as a result of being bound by some administrative control such as a law, regulation, or data use agreement.

# Acronyms

**AAM** Active Appearance Model

**AC-GAN** Auxiliary Classifier GAN

**AMGAN** Appearance Model-based GAN

**ASM** Active Shape Model

**AU** Action Unit

**cGAN** Conditional GAN

**CLM** Constrained Local Model

**CLNF** Constrained Local Neural Fields

**CNN** Convolutional Neural Network

**DCGAN** Deep Convolutional Generative Adversarial Network

**FACS** Facial Action Coding System

**FET** Facial Expression Transfer

**GAN** Generative Adversarial Network

**HIPAA** Health Insurance Portability and Accountability Act

**HOG** Histogram of Oriented Gradient

**LBP** Local Binary Patterns

**LDA** Linear Discriminant Analysis

**LPQ** Local Phase Quantisation

**LSGAN** Least Squares GAN

**MLP** Multilayer Perceptron

**MLS** Moving Least Squares

**PCA** Principal Component Analysis

**PDM** Point Distribution Model

**PHI** Protected Health Information

**PII** Personally Identifiable Information

**SDM** Supervised Descent Method

**SVD** Singular Value Decomposition

**SVM** Support Vector Machine

**VAE** Variational Autoencoder

**WGAN** Wasserstein GAN

**WGAN-GP** Wasserstein GAN with gradient penalty

# Chapter 1

## Introduction

With the development of information technology and the reduced cost of data storage, large amount of personal information is collected. In recent years, a lot of new equipment, such as smartphones, wearable devices and drones are expanding the way how personal information is collected. Personal information can be easily collected, stored, shared and even analysed as this information has great values for business, public medical service, academic research and many more. For example, Amazon Go achieves checkout-free stores based on data collected by their surveillance system; face tracking becomes the new tool for personalised advertising; and facial expressions can be used as diagnosis information in Parkinson's disease. However, when such data is shared, personal information or privacy could be leaked to the public or unauthorised parties.

People started the modern discussion on the right to privacy in late 19<sup>th</sup> century [1]. It was drafted in the European Convention on Human Rights (ECHR) by the Council of Europe in 1950. The protection of personal information or personal data was involved in the laws in the 1990s. In 1996, the United States Congress enacted the Health Insurance Portability and Accountability Act (HIPAA) [2] and the HIPAA Privacy Rule [3] was promulgated which establishes national regulations for the use and disclosure of Protected Health Information (PHI) including individuals' medical records and other personal health information. Meanwhile, the European Data Protection Directive (Directive 95/46/EC) [4] was adopted in 1995, which demands the deployment of appropriate technical and organisational measures to protect private information in the course of transferring or processing such data. It has been updated and replaced by Regulation (EU) 2016/679 of the European Parliament and of the Council, also known as General Data Protection Regulation (GDPR) [5].

Another fact is that biometric information especially face biometrics has been widely used in access control, academic research and commercial products. It is not only because of the recent breakthrough in computer vision which has boosted the face recognition accuracy to the human level, but also the face data can be acquired easily even without the coordination from the data subjects. The high performance of the face recognition system can establish a link from a face data to the identity of the data subject quickly and economically. However, this has led to ethical issues when storing and using the face data. Therefore, it is necessary to have a model to facilitate privacy-preserving use of face-related data.

There can be different approaches to preserving privacy in face (biometric) data depending on the data use case. For example, Google Street View blurs the pedestrians' faces that stops a further analyse on the face region [6]; Cancellable biometrics [7] do not register users with their real biometrics information but scrambled or encrypted information to reduce the risk in case there is a data leak from the biometric database. However, these methods destroy the nature/shape of the face data and the data utility is highly downgraded so that they can barely support any further face related analyses. Therefore, face de-identification technique is needed to protect privacy while preserving data utility for further usage or data publishing.

The face data are commonly collected from data subjects and released by a data publisher. The current practice in face dataset releasing relies mostly on agreements on the use of the published data. This approach alone may lead to insufficient protection. Some policies and guidelines, e.g. HIPAA Privacy Rule, limit the distribution of face data, which make it hard for researchers to have enough data to model the real-world scenario. Academics have proposed two models for using personal information in a database [8]:

#### **Privacy-Preserving Data Mining (PPDM) [9]**

In this model, the original data are not released, but the results extracted from the original data through data mining are released. The released results need to be accurate and the model should provide a reconstruction procedure that can estimate the distribution of the original data.

#### **Privacy-Preserving Data Publishing (PPDP) [10]**

This model aims to provide data that have high utility without revealing the identity of the data subjects. The published data are processed with

---

de-identification or synthetic data generation methods. With such data, other researchers could perform novel analyses.

These two privacy-preserving models have been investigated and applied to structured data [11]. However, challenges remain for preserving privacy with unstructured data such as visual data. “A big difference between ordinary data privacy and video data privacy is the amorphous nature of the latter and the difficulty in processing it automatically to extract useful information” [12]. In other words, preserving privacy in visual data requires the algorithm to detect, segment and extract target information precisely. Thanks to the development of machine learning techniques more and more approaches have been proposed to preserve privacy in unstructured data [8], [13].

Research in de-identification aims to not only protect privacy in the de-identified data but also maintain the utility of data. The early approaches to face de-identification are called ad hoc face de-identification because their primary purpose is to hide the identity information in the image while the preservation of the data utility is not considered in the design. The  $k$ -Same face de-identification method is the first attempt that applied  $k$ -anonymity to face data. Its variants  $k$ -Same-Select and model-based  $k$ -Same ( $k$ -Same-M) have been proposed to improve the preservation of data utility and intelligibility of the de-identified face image respectively. More face de-identification methods are introduced in Chapter 2.

The model-based  $k$ -Same is the basis of the face de-identification systems proposed in this thesis. The proposed face de-identification methods attempt to further reduce the re-identification risk of the  $k$ -Same-M method, as well as preserve the data utility of the de-identified face data. Its key technique is manipulating the face data in the face feature space and reconstructing synthetic faces as the de-identified faces. It can adapt to different use cases and preserve specific data utilities. The data utilities are discussed in this thesis including the diversity of a face dataset, facial expressions in image and video data. An end-to-end face de-identification system for both closed-set and open-set is proposed in this thesis. This thesis also discusses that the de-identified face data should have the nature of a face data and proposes a method that can guarantee the de-identified face image’s visual quality in terms of the fidelity and the intelligibility. The proposed face de-identification system is intelligible to both human vision and existing computer vision system, because the realistic de-identified face image is accepted by a human observer and can be used directly in existing face related computer vision system as input.

## 1.1 Contributions

This thesis addresses several limitations of the existing face de-identification methods and makes following contributions to the research field of face de-identification:

- A novel  $k$ -Same-furthest face de-identification method introducing the wrong-map protection to the model-based  $k$ -Same face de-identification to further reduce the one-to-one re-identification risk (see Chapter 4);
- Another low-risk face de-identification method,  $k$ -Diff-furthest, which is modified from  $k$ -Same-furthest with the aim of maintaining the data diversity in the de-identified face dataset (see Chapter 5);
- An approach to merging the isolated de-identified faces with their original backgrounds to increase both fidelity and intelligibility of the de-identified face images for human observer in real-world applications (see Chapter 6);
- Evaluation of the re-identification risk caused by the image background and demonstrate the high risk of potential attacks using information in the image background (see Chapter 6);
- A novel Facial Expression Transfer (FET) process which has been combined with face de-identification to preserve the data utility of facial expressions. It can map the facial expressions from an original face image to its de-identified face image. Furthermore, it can also be used to preserve the head motions in the video frames (see Chapter 7);
- Introduce the identity pool to face de-identification, which is a set of person-specific neutral frontal face images playing the role of identity references. The identity pool can help to adapt the closed-set face de-identification methods to open-set face de-identification (see Chapter 7 and 8);
- An application of end-to-end face de-identification in video sequences, which combines all the above-mentioned contributions, including de-identification for face image dataset, FET, identity shift and background merging (see Chapter 7);
- A novel approach to synthesising realistic-looking face image using the face appearance model and the Generative Adversarial Network (GAN) (see Chapter 8);



- An application of using the proposed appearance model-based GAN to generate the identity pool for face de-identification. To this end, cascaded filters for selecting frontal neutral faces is proposed (see Chapter 8).

## 1.2 Research outputs

- L. Meng and Z. Sun, “Face De-identification with perfect privacy protection,” in *Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014, pp. 1234–1239. (Chapter 4)
- Z. Sun, L. Meng, and A. Ariyaeinia, “Distinguishable de-identified faces,” in *Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015, vol. 04, pp. 1–6. (Chapter 5)
- Z. Sun, L. Meng, A. Ariyaeinia, X. Duan, and Z.-H. Tan, “Privacy protection performance of De-identified face images with and without background,” in *Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 1354–1359. (Chapter 6)
- L. Meng, Z. Sun, A. Ariyaeinia, and K. L. Bennett, “Retaining expressions on de-identified faces,” in *Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014, pp. 1252–1257. (Chapter 7)
- L. Meng, Z. Sun, and O. Collado, “Efficient approach to de-identifying faces in videos,” *IET Signal Process.*, vol. 11, no. 9, pp. 1039–1045, Dec. 2017. (Chapter 7)

### 1.3 Thesis outline

The thesis is structured as follows. Chapter 2 reviews and compares various de-identification methods with a focus on the methods developed for face images. Chapter 3 reviews research advances related to this thesis, including face appearance model, facial landmark detection methods, face recognition methods and the approaches to facial expression synthesis. Chapter 4 presents the  $k$ -Same-furthest face de-identification method which introduces the wrong-map protection to the model-based  $k$ -Same face de-identification. Chapter 5 proposes the  $k$ -Diff-furthest face de-identification method which aims to maintain the diversity of the de-identified face image dataset. Chapter 6 discusses the data utility and the re-identification risk of the original image background in the face de-identification. An approach to merging the de-identified face region to its original background as well as a potential attack using the background information are presented. Chapter 7 presents an approach to map the facial attribute changes to a de-identified face including facial expression transfer, identity shift and head motion transfer between video frames. An end-to-end approach to face de-identification in videos is also presented in Chapter 7. Chapter 8 introduces the Appearance Model-based GAN for face image generation and presents the training strategies for with/without facial attribute conditions. The generated face images are integrated with the proposed face de-identification system and the face de-identification performance are presented.

# Chapter 2

## Background

The concepts of de-identification have been well established for tabular data, especially health data records [8], [11], [14], [15]. The research in health data de-identification shares concepts and has inspired the investigation of de-identification in other types of data including de-identification in multimedia data [13]. Section 2.1 briefly introduces the existing de-identification methods on structured data, and the other sections of this chapter present a review of the latest research in face de-identification and the commonly adopted techniques.

### 2.1 De-identification on structured data

In de-identification literature [8], [10], [14], [16], [17], the attributes of a data record are divided into two categories, namely direct identifiers and quasi-identifiers (or indirect identifier). The direct identifiers can be directly used to identify individuals, for example, name, phone number, social security number, etc. Quasi-identifiers are the identifiers which hardly identify an individual when being used alone but can be linked with other information to identify a record, for example, date of birth, postcode, gender, etc. Sweeney demonstrated a re-identification attack achieved by linking shared attributes (quasi-identifiers) cross datasets [17]. Direct identifiers can be anonymised by masking techniques, and quasi-identifiers by de-identification methods [16]. Here are some commonly used data anonymisation methods:

#### **Data masking**

Data masking could be achieved by applying partial data removal or data quarantining. Partial data removal, especially the removal of direct personal identifiers [8], is

the most common and a necessary approach to reducing the re-identification risk. Data quarantining technique is usually applied to a controllable environment where a trusted data recipient is unlikely or unable to gain access to the other data needed to facilitate re-identification.

Data masking is a relatively high-risk technique because the anonymised data still exists in an individual-level form.

### **Pseudonymisation**

In pseudonymisation, a coded reference or pseudonym is attached to a record to allow the data to be associated with an individual without the individual being identified. If multiple records are linked, to support certain types of data analysis, the same original values are always replaced by the same pseudonym, and it is called deterministic modification.

The same as data masking, data anonymised by pseudonymisation exists in an individual-level form and hence has a relatively high-risk.

### **Aggregation**

Generally speaking, aggregation technique replaces original data with their totals, so no data identifying an individual remains after de-identification. Depending on the data type or the statistical characteristics of the data, different settings need to be considered to preserve data utility. In real-world applications, several variants can be employed, e.g. inference control, rounding, sampling, synthetic data, etc.

Aggregation techniques are relatively low-risk techniques because it is difficult to associate an aggregated data with an individual. Aggregated data cannot support individual-level research but can be sufficient for analysing social trends on a regional basis.

One of the microaggregation approaches named  $k$ -anonymity [17], also known as  $k$ -partition [11], has been proposed to anonymise quasi-identifiers. It has also provided theoretical support to the  $k$ -Same face de-identification method. Partition and aggregation are carried out in the process of  $k$ -anonymity. Each partition contains at least  $k$  records which they are clustered based on similarity. Then the  $k$  records are replaced by their average, so that if the source records are matched against the de-identified data, there will be at least  $k$  de-identified records having the same similarities.  $\ell$ -diversity [18] adds requirements for diversity of the sensitive

attributes within each class to  $k$ -anonymity. Later,  $t$ -closeness [19] requires that the de-identified data are statistically close to the original data.

### Derived data items and banding

Banding produces coarser-grained descriptions of the source data. The derived data are values that reflect some higher-level characteristics of the source data. In most cases, this is a “summary” of the source data, e.g. using partial postcode or replacing a date of birth by its year or the age.

Banding is a relatively low-risk technique because it is hard to establish one-to-one matching between the derived data and the source data. Nevertheless, certain individual-level analysis can be done with the derived data.

### Data swapping

Data swapping exchanges certain parts of a record with the corresponding parts of another record. It adds uncertainty to the links between the identities and the records without changing data distribution. Several factors can affect the de-identification level, including selection of swapping candidates, similarity among swapping attributes and swapping rate [20].

## 2.2 Ad hoc face de-identification

According to HIPAA Privacy Rule Safe Harbor standard [3], face image data must be removed from a health dataset.<sup>1</sup> However, datasets used in some application such as facial expression analysis and surveillance-based systems only contain images/videos of human faces. In such cases, the only task is to de-identify the face instances in the dataset.

Ad hoc face de-identification is the method which applies image processing techniques to a source image with the aim to blur or distort the identifying information in the image. The motivation of these ad hoc methods is to prevent the target faces being recognised by human or machines. The research of ad hoc face de-identification focuses on improving the face detection techniques and designing a better image masking technique to satisfy specific purposes [6], [21]–[23]. Ad hoc methods have been wildly used in anonymising photos of public-oriented

---

<sup>1</sup> The HIPAA Privacy Rule Safe Harbor standard lists 18 specific identifiers including “(P) Biometric identifiers, including finger and voice prints; (Q) Full face photographs and any comparable images.”

scenes, e.g. images/videos used in news and street view photos. In these scenarios, the removal of facial information does not affect too much to the storytelling or the expression of the key information. Comparing to other face de-identification techniques, these methods are ad hoc because they did not consider much about the unique characteristics of human faces in the design.

Ad hoc face de-identification usually starts with a face detection process. The detected face area is then processed by one of the following methods [24]:

**Blurring.**

Apply a 2D convolution to the image with a flat kernel or a Gaussian kernel. The kernel size and the standard deviation  $\sigma$  of the Gaussian kernel are usually used to control the de-identification performance.

**Pixilation.**

Slide a pixel block across the image, strip size = block size. The average value replaces the original pixel values in the block. The block size is used to control the de-identification performance.

**Censor bar.**

Cover part of the face area with a bar-shaped or T-shaped single colour pattern. Usually, censor bar covers only the region of eyes as shown in Fig. 2.1d. There is no parameter in this method, but the area of the censor bar can be used to measure the de-identification performance. When the bar area equals to the detected face area, the particular method is named *blackout*, which removes all the information in the detected area.

**Random noise.**

Choose random pixel locations and replace the values in each colour channel with a random value between 0 and 255. The probability of whether change the value(s) of a pixel location or not is set to control the performance of de-identification.

**Threshold.**

Set all the pixel values to 0 or 255 depending on a threshold value. The threshold value is the parameter to control the de-identification performance. It can be either an integer value between 0 and 255 or an intensity value between 0 and 1.

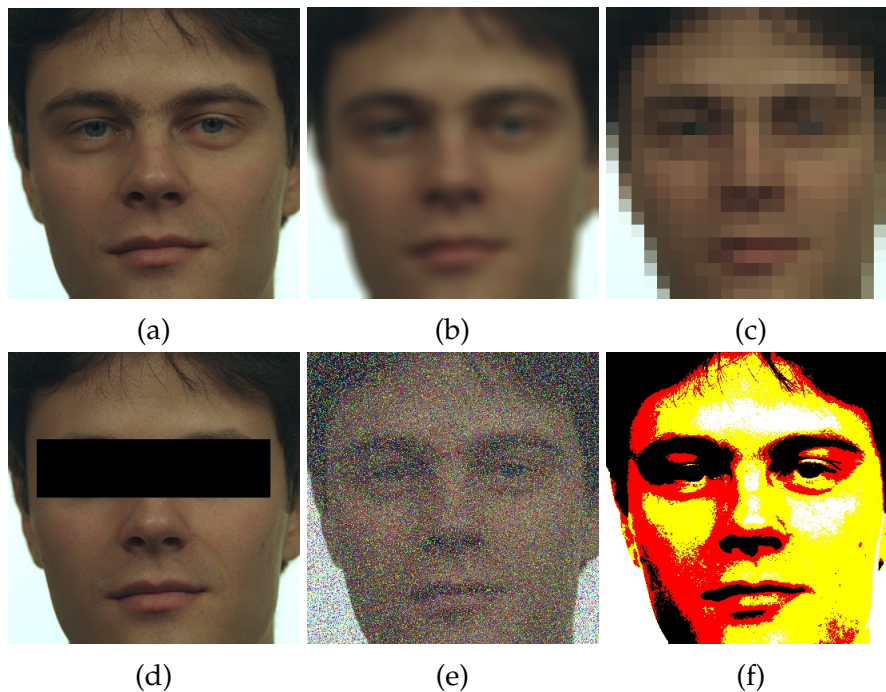


Fig. 2.1 Ad hoc face de-identification examples. (a) original face region detected by Viola and Jones face detector [26]; (b) blurring with Gaussian filter; (c) pixelation; (d) censor bar; (e) random noise; (f) threshold.

### Warping.

Warping is yet another image processing technique which can distort information in images. It has been mentioned in [7], [25], but it is rarely seen in practice. A set of selected key points are shifted randomly to obtain the transformation matrix. There are relatively more parameters in this method. The maximum shift distance of the key points is one of the parameters used to control the de-identification level. The warping process is revisable, so the original image can be re-generated when the de-identification level is low.

Fig. 2.1 illustrates some common ad hoc face de-identification methods. The advantage of these methods is their operands are pixels, and they only rely on face detection techniques, so they are easy to implement and could be highly automatic. However, they have few disadvantages on both re-identification risk and data utility folds. In terms of preservation of data utility, they are destructive. Most of the ad hoc face de-identification methods treat each pixel equally. As a result, they tend to destroy the utility of the image data for the purpose of reducing the re-identification risks. For example, the facial expression becomes undistinguishable when a high level of de-identification is implemented. They can even make de-identified images

lost the nature of face images so that a face detection software cannot detect faces from such de-identified images. Automatic face detection is the first step of the most face-related analysis. If the face in a de-identified face image fails to be detected, then it is hard to use this image for further analysis, unless the data publisher provides the face detection boundary boxes or facial landmarks. A face detection test is shown below to compare the data utility of de-identified face image for three popular face detection frameworks with their default settings:

- OpenCV face detector is an implementation of Viola-Jones detection, which is a Haar feature-based cascade classifiers [26]. OpenCV provides several trained models. In the following test, the model `haarcascade_frontalface_alt.xml` was used.
- Dlib provides a face detector which trained with HOG features combined with linear Support Vector Machine (SVM). It is another popular object detection proposed by Dalal et al. [27].
- MTCNN is a relatively new face detector. It cascades three Convolutional Neural Networks to conduct a coarse-to-fine face detection [28]. It is more robust in the head pose, illumination and tiny size face than previous two face detectors.

962 frontal faces from FERET data are used in this face detection test. Fig. 2.2 shows the success detection rate and de-identification examples of a face image are shown in Fig. 2.3. The results show that the success detection rates are decreasing when the protection levels are increasing. Face detectors are not sensitive to blur, while pixelation, random noise, threshold and blackout methods can make the face detectors fail to detect faces in the de-identified images when protection levels are increasing.

In terms of reducing re-identification risk, these ad hoc methods may fail to serve their goal of privacy protection. Because the image processing techniques in these methods are easy to re-implement, attackers can use parrot attack to crack the de-identification settings. For details of parrot attack (see Chapter 6). It has been demonstrated in [24], [29], [30] that the re-identification risk of face images de-identified by bar masking or pixelation is 100%. Recent research shows that a deep neural network with Variational Autoencoder (VAE) or GAN architecture has the ability to restore a blurred face image or obscured face images [31]–[33].



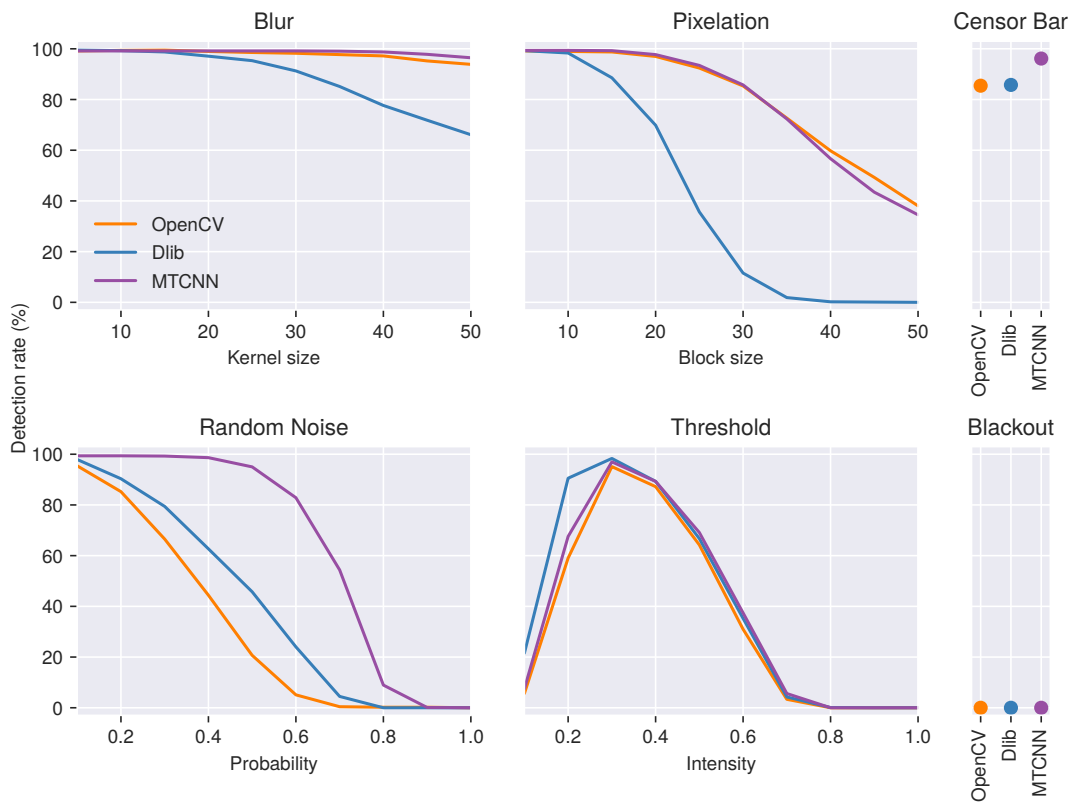


Fig. 2.2 Face detection rates on the de-identified face images with ad hoc methods

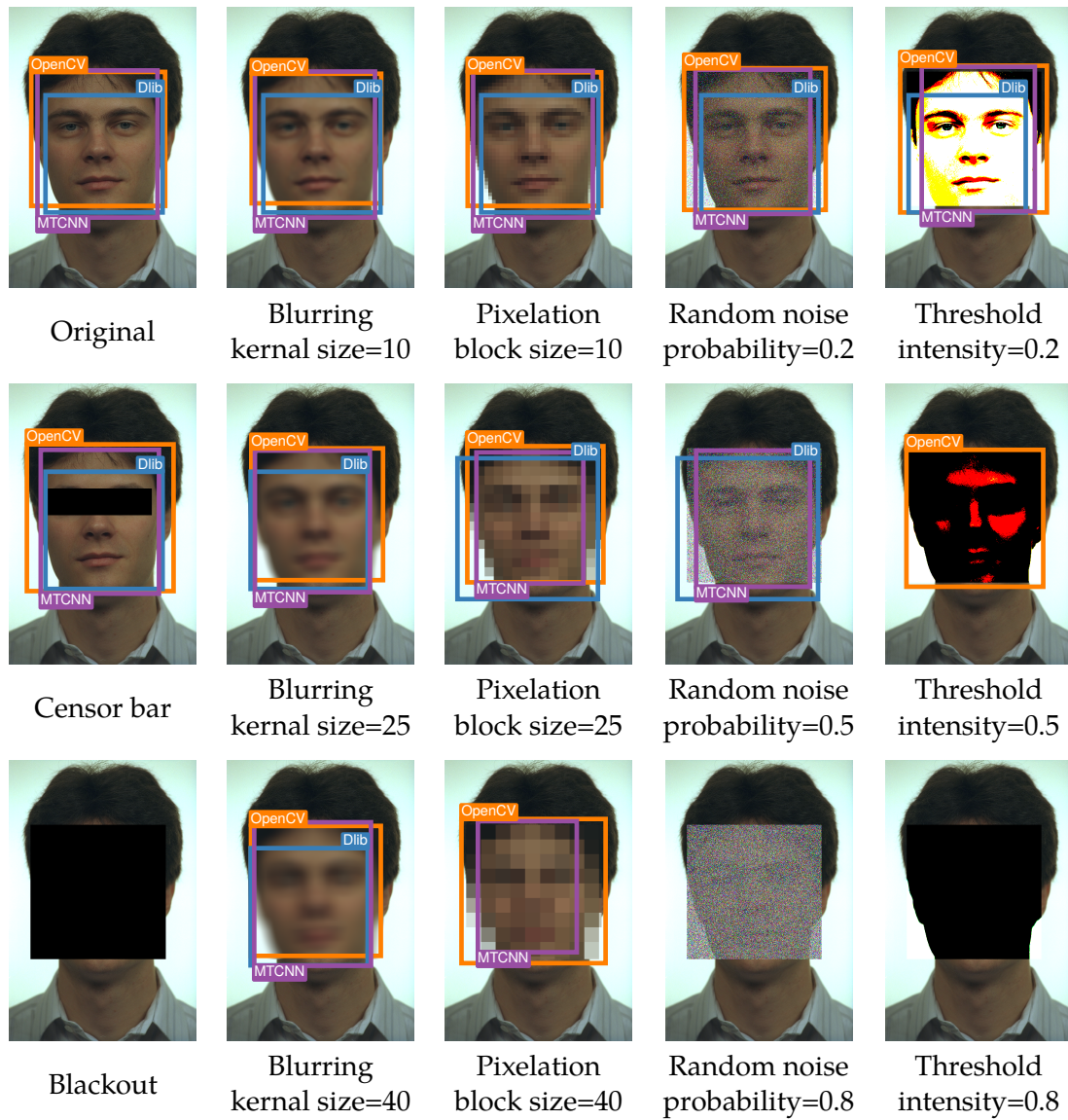


Fig. 2.3 Examples of face detection on de-identified face images with ad hoc methods, where the example image is from FERET dataset [34].

## 2.3 $k$ -Same face de-identification

$k$ -Same and its variants implement  $k$ -anonymity de-identification to a face dataset. Here are some definitions commonly used in the context of  $k$ -Same face de-identification.

**Definition 2.1** (Face Set). A face set is a set of  $m$  face images  $\{I_i : I_i \in \mathbb{R}^{h \times w \times c}, i = 1, \dots, m\}$  with height  $h$ , width  $w$  and  $c$  colour channels.

**Definition 2.2** (Person-Specific Face Set). Let  $H$  be a face set of  $m$  images  $\{I_1, \dots, I_m\}$ .  $H$  is person-specific if and only if each person has only one image in the set  $H$  and each  $I \in H$  relates to only one person.

**Definition 2.3** ( $k$ -anonymity on Face Images). Given a person-specific face set  $H$ , a set of de-identified face images  $H_d$ ,  $|H| > 1$ , a de-identification function  $f : H \rightarrow H_d$ . If for each  $I \in H$  there exists  $I_d \in H_d$  where  $f(I) = I_d$  and for each  $I_d \in H_d$ , there are at least  $k$  identical copies of  $I_d \in H_d$ , then  $H_d$  adheres to  $k$ -anonymity. It is said that  $H_d$  is  $k$ -anonymised over  $H$ .

### 2.3.1 $k$ -Same

Ad hoc methods operate at the pixel level and provide privacy protection through blurring (or completely black out) image pixels. However, when applying face de-identification to an image set, privacy protection can be achieved by aggregating Personally Identifiable Information (PII) through a  $k$ -anonymity process. The  $k$ -anonymity based face de-identification methods are able to maintain the original resolution of the face and hence provide a valid foundation for data utility preservation.  $k$ -Same [24] method is the first face de-identification method, which was designed to operate on a person-specific face set.

In  $k$ -Same, the original face set is divided into clusters of size  $k$ . Each of the  $k$  records in a cluster is replaced/de-identified by the centroid of its cluster so that the  $k$ -Same face de-identification method could achieve  $k$ -anonymity.  $k$ -Same face de-identification can apply microaggregation on either face image pixels or their Eigenface features. Details of the  $k$ -Same method are described in Algorithm 1.

The outcomes of the  $k$ -Same method depend on its clustering result. The optimal clustering solution leads to the minimal information loss, i.e. the minimum data utility loss in the de-identified dataset. Because when the members in a cluster are closer to each other geometrically the cluster centroid becomes more representative of

**Algorithm 1:**  $k$ -Same pixel/Eigen

---

**input** : Person-specific face set  $H : \{x\}$ , where  $x$  is a vector of pixel values or image PCA features; Privacy constant  $k$ , with  $|H| \geq k$ .  
**output**: De-identified face set  $H_d$ .

```

1  $H_d \leftarrow \emptyset$ ;
2 for  $\exists x \in H$  do
3   if  $|H| < 2k$  then
4      $k \leftarrow |H|$ ;
5   end
6    $\{x_1, \dots, x_k\} \leftarrow \text{kNN}(H, x, k)$ ; // k-Nearest Neighbours
7    $\bar{x} \leftarrow \frac{1}{k} \sum_{m=1}^k x_m$ ;
8   Add  $k$  copies of  $\bar{x}$  to  $H_d$ ;
9    $H \leftarrow H \setminus \{x_1, \dots, x_k\}$ ;
10 end

```

---

**Algorithm 2:** MDAV-generic for face de-identification

---

**input** : Person-specific face set  $H : \{x\}$ , where  $x$  is a vector of pixel values or image PCA features; Privacy constant  $k$ , with  $|H| \geq k$ .  
**output**: De-identified face set  $H_d$ .

```

1  $H_d \leftarrow \emptyset$ ;
2 while  $|H| \geq 3k$  do
3    $\bar{x} \leftarrow \frac{1}{|H|} \sum x, \quad \forall x \in H$ ;
4   Find the most distant record  $x_r^{(1)}$  to  $\bar{x}$ ;
5   Find the most distant record  $x_s^{(1)}$  from the record  $x_r^{(1)}$ ;
6    $\{x_r^{(1)}, \dots, x_r^{(k)}\} \leftarrow \text{kNN}(H, x_r, k)$ ; // k-Nearest Neighbours
7    $H \leftarrow H \setminus \{x_r^{(1)}, \dots, x_r^{(k)}\}$ ;
8    $\{x_s^{(1)}, \dots, x_s^{(k)}\} \leftarrow \text{kNN}(H, x_s, k)$ ;
9    $H \leftarrow H \setminus \{x_s^{(1)}, \dots, x_s^{(k)}\}$ ;
10 end
11 if  $|H| \geq 2k$  then
12    $\bar{x} \leftarrow \frac{1}{|H|} \sum x, \quad \forall x \in H$ ;
13   find the most distant record  $x_r$  from  $\bar{x}$ ;
14    $\{x_r^{(1)}, \dots, x_r^{(k)}\} \leftarrow \text{kNN}(H, x_r, k)$ ;
15   form another cluster containing the rest of records;
16 else
17   form a new cluster with the remaining records;
18 end

```

---

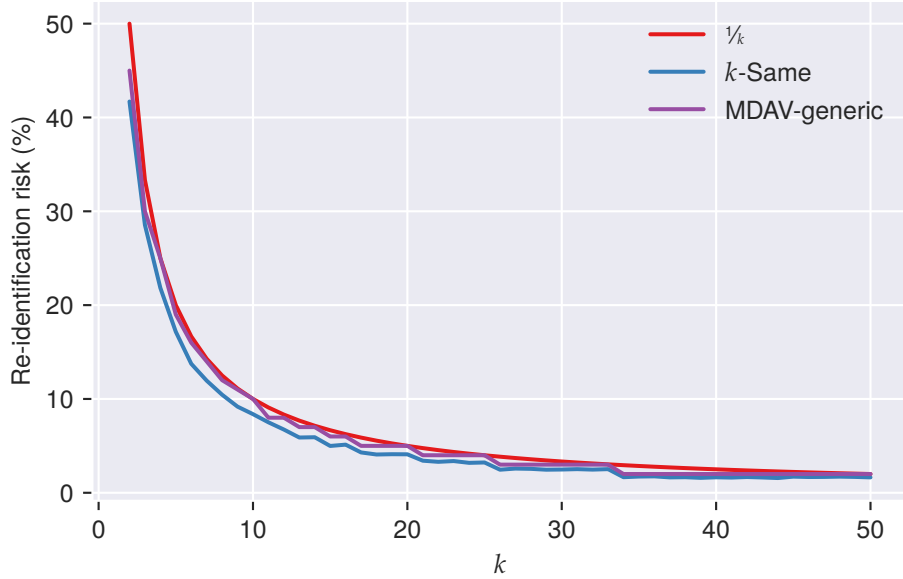


Fig. 2.4 Re-identification risk on toy data

the cluster members. It has shown that the optimal solution to the microaggregation is an NP-hard problem [35], [36]. However,  $k$ -Same algorithm is not designed to find the optimal partition of the input data. In most cases, the face image data has a near Gaussian distribution. Hence, in  $k$ -Same, the random selected clustering seed  $x$  have a high probability to be in a high-density neighbourhood. As a result, a “hole” often appears in the data space after several iterations. This will make the centroids of later formed clusters lose the ability to represent and maintain the data utility of their members (see Fig. 2.5a). MDAV-generic [37] is a near-optimal solution to microaggregation, which also satisfies  $k$ -anonymity. At each iteration, the method forms clusters with the two records that have the largest distance to the average of all records. As a result, the records which are far from other records have a higher priority to be processed and “hole” in data/feature space are avoided. A comparison of  $k$ -Same and MDAV-generic is shown in Fig. 2.4 and 2.5. The toy dataset used in this comparison study was randomly generated with  $\mathcal{N}(x; \mu, \Sigma)$  where  $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$  and  $\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix}$ . As shown in Fig. 2.4, the re-identification risks of both methods are lower than  $1/k$ . The re-identification risk of MDAV-generic is closer to the theoretical upper bound  $1/k$ , which means MDAV-generic gives more expected partition results. The cluster centroids are more representative of its own cluster members than members in the other clusters.

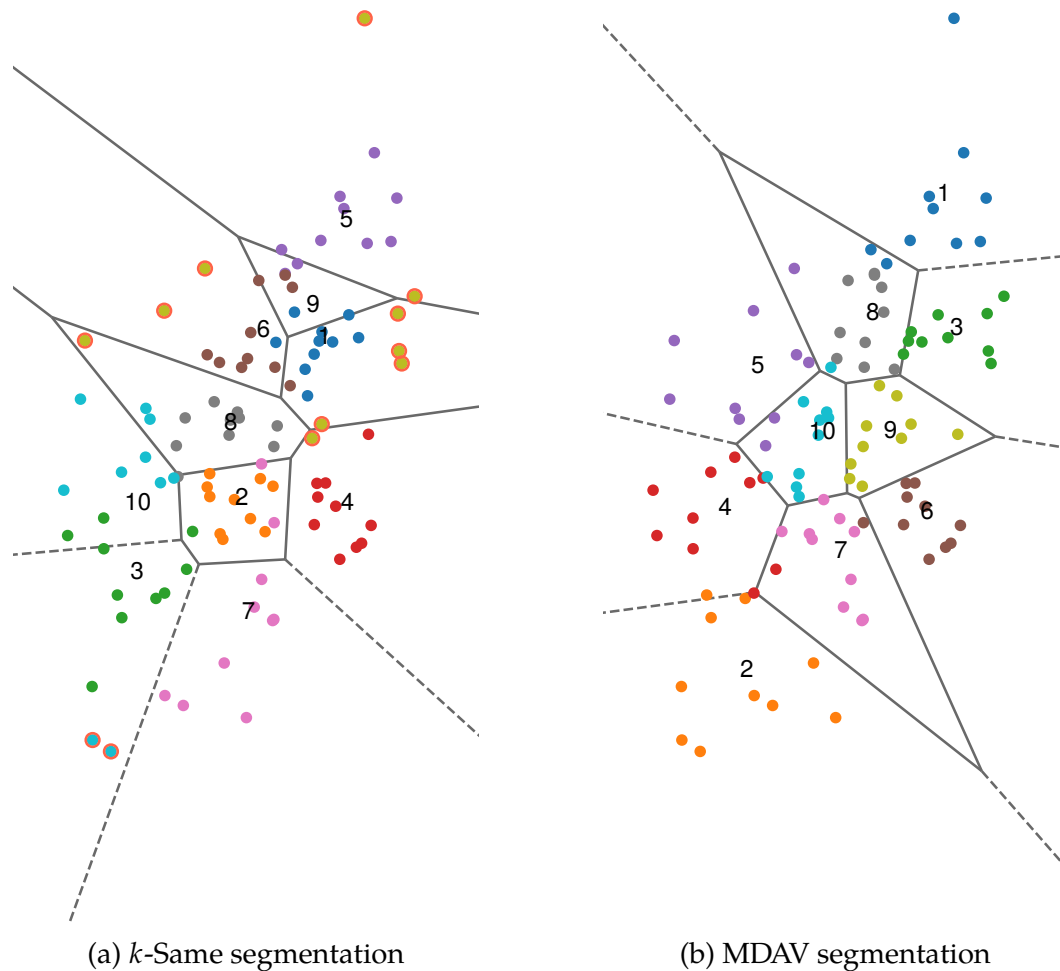


Fig. 2.5 The clustering results of  $k$ -Same and MDAV-generic. The cluster index numbers are displayed at the locations of their cluster centroids. In (a), the highlighted records of cluster 9 and cluster 10 are far from their cluster centroids. It is obvious that after the formation of clusters 1, 5, 6 and 8, there is a “hole” in the data/feature space. Cluster 9 can only take the remaining records and thus spans over many existing clusters.

A face image contains various data utility information, e.g. genders, face shapes, expressions and age. *k*-Same de-identification is normally applied to the face images directly after face detection without any furthest analysis. Although *k*-Same face de-identification could guarantee a re-identification risk of de-identified faces lower than  $1/k$ , the data utility information is often lost along with the aggregation. There are several limitations of applying microaggregation on pixels or their Eigenface parameters directly, including: (a) loss of data utility; (b) ghost artefacts on the de-identified images (see Fig. 2.6); and (c) limited privacy protection.

### 2.3.2 Model-based *k*-Same

One of the limitations of *k*-Same de-identification is visual quality in the sense that the de-identified faces may have ghost artefacts on them. The cause of this problem is the misalignment among faces when calculating their mean. Automatic face alignment is a general problem in face recognition [38], and the state-of-the-art solution is based on the annotations of the facial landmarks. Facial landmarks can be detected by a regression model such as Active Appearance Model (AAM), Constrained Local Model (CLM) [39] and Constrained Local Neural Fields (CLNF) [40]. In 2006, Gross et al. [30] proposed model-based face de-identification, namely *k*-Same-M, which is an integration of AAM and their previous face de-identification work. The experimental results showed that the *k*-Same-M de-identification offers a similar protection performance as *k*-Same, i.e. a re-identification risk lower than  $1/k$ , while generating de-identified faces with a much better visual quality (see Fig. 2.6). In *k*-Same-M, a statistical appearance model [30], [41], [42] is used to represent the faces. The same model has been used in the rest of this thesis. This appearance model contains a shape representation and a texture representation. The shape of a face is defined by its facial landmarks, and the face texture is the mean-shaped face appearances which are warped from the pixels inside its face shape. For the details of the appearance model (see Section 3.1).

### 2.3.3 *k*-Same variants for utility preservation

Microaggregation in pixel/Eigenface space not only aggregates the identity information but also other face attributes such as facial expressions. Gross et al. [29] demonstrated that ad hoc methods fail to preserve data utility at high obfuscation levels. To integrate data utility to the de-identified face images, they proposed the *k*-Same-Select face de-identification method [29]. *k*-Same-Select applies data utility

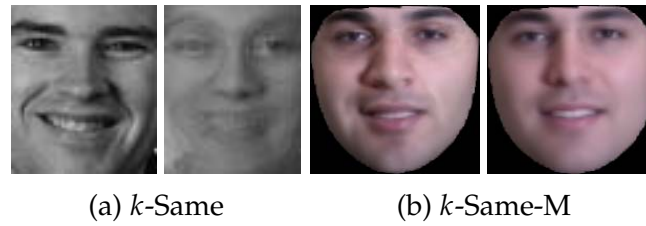


Fig. 2.6 Face de-identification examples of  $k$ -Same and  $k$ -Same-M [30]. The original face shows on the left and the de-identified face on the right. The  $k$  value in both examples is 10. There are some heavy ghost artefacts shown in the result of  $k$ -Same.

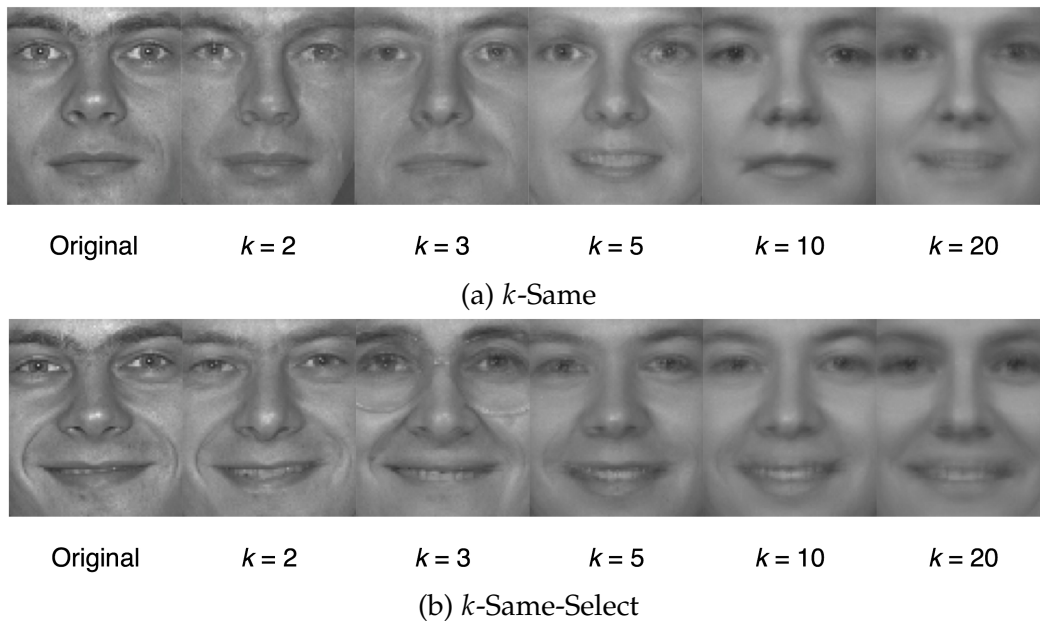


Fig. 2.7 Comparison of data utility of smiling between  $k$ -Same and  $k$ -Same-Select (adapted from [29])

classifiers on the original image dataset to divide it into subsets. The face images in each subset have the same or similar data utility so that the de-identified faces maintain the expected data utility after applying the  $k$ -Same face de-identification process on each subset (see Fig. 2.7).

In [43], Gross et al. mentioned three types of mapping mechanism for face de-identification, i.e.

**$\epsilon$ -map** can be described as “make the de-identified face look like no one”.

**wrong-map** can be described as “make the de-identified face look like someone else”.

**$(k, \epsilon)$ -map** can be described as “make the de-identified face look like everyone”.



They also pointed out that *k*-Same face de-identification was designed for a closed dataset. However, in the real-world scenario, the subject to be de-identified is not always in the dataset. Even though the subject is in the dataset, the illumination and expression variant can be different from his/her reference in the dataset. Later, a multi-factor model for face de-identification was proposed by Gross et al. [44], [45]. The framework combined linear, bilinear and quadratic data models. It is a generative multi-factor model which is able to factorise the face image into identity and non-identity factors. The de-identification then be applied to the combined factors. Experiments showed that this algorithm protects privacy while preserving data utility on an expression variant face dataset.

Du et al. proposed GARP-Face framework with an aim to preserve specific facial attributes including, gender, age and race of the original faces [46]. Their approach preserves the data utility in a way similar to *k*-Same-select. A reference face gallery is required and, the gallery is divided into several subsets based on the combination of the attributes, then the utility-specific AAMs are trained on each subset. The de-identified face is an aggregated face computed within each subset, which satisfies *k*-anonymity. Experiments show that GARP-Face outperforms the general AAM approach.

Attribute preserved face de-identification (APFD) was proposed by Jourabloo et al. [47]. This work extended the work of *k*-Same-M, and added attribute classifiers to preserve data utility. The main novelty of this work is that the de-identified face is not the average of *k* faces but a weighted sum of *k* faces. The initial weights of shape parameters and texture parameters are  $1/k$ , then the weights are optimised by gradient descent to minimise a joint objective function, which is constituted by a term of attribute classification error and a term of face verification likelihood. Experiments showed that with the same *k* value APFD has a lower re-identification risk than *k*-Same-M.

Most recently, Meden et al. used generative neural network (GNN) to generate synthesised faces to de-identify face images [48]. The GNN is controllable to generate a face image with given facial attributes. Later, this GNN face de-identification pipeline was extended with the idea of *k*-anonymity and became *k*-Same-Net [49]. Different from the above mentioned *k*-Same family, this approach involves a proxy image set which is a set of the face images used to train the GNN. First, a one-to-one correspondence was established between a cluster of *k* faces from the input image set and the images in the proxy set. The identity vector for GNN was then calculated with the *k* corresponding faces in the proxy set. To preserve the expression

information, the non-identity-related parameters for GNN were computed from the input images. Because the identity-related information of the de-identified faces are synthesised from the proxy set (the GNN training set) through the GNN, the re-identification risks are close to random matching no matter what  $k$  value is. To train such a decoder-like neural network with a good understanding of latent space is challenging. Image generation is a hot research topic and later proposed image generation models can improve the image quality of the GNN.

## 2.4 Other face de-identification approaches

In [50], AAM was adopted to de-identify faces in different poses. First, AAM fitting obtains the face shapes, and then the face appearance was used to search in a frontal face gallery to find the proper candidates for face swapping. The method was named  $q$ -far, where the  $q$  denotes the rank of similarity to the input face. This approach trades-off between visual quality and re-identification risk. If the similarity of the candidate is too close to the original face, the de-identified face looks natural, but the re-identification risk is high. Otherwise, if the candidate is too different from the original face, it will make the de-identified face have lower naturalness but higher security level. Thus, a large candidate gallery was suggested when applying this approach. In the same year, we proposed a facial expression transfer approach to achieve preservation of facial expressions and also to address the different head pose problem in face de-identification [51].

Apart from using AAMs, other approaches were proposed to de-identify face images. In 2014, Mosaddegh et al. [52] proposed a face de-identification approach aiming to preserve the visual quality of output images while reducing the re-identification risk. In their approach, based on the results of facial landmark detection, the face was segmented into four facial components, namely, eyebrow, eye, nose & cheek and mouth & chin. Each component is then replaced with the same component from a donor. Four different donors are used to replace the four components on an original face. Each de-identified face generated by this method is an aggregation of four different faces and hence presents a low re-identification risk.

Letournel et al. [53] extended the ad hoc blurring method and applied different blurring levels to individual face regions to better preserve target facial expression. It relied on the facial landmark detection technique, and a weights map for each pixel was computed using a variational model based on the facial landmarks. The

face regions that contribute more to emotional analysis, e.g. eyes, gaze, lips, etc. will be given a reduced blurring level.

In [54], Chriskos et al. discussed two de-identification methods that could fail the machine but not human. The first method extracts face features through Singular Value Decomposition (SVD), and then the SVD coefficients are manipulated to achieve face de-identification. The second method normalises the face projections to have the same distance to their mean vector so that all the face projections lie on the surface of a hypersphere. Because face alignment did not involve in this work, ghost artefacts appeared on the de-identified faces. The visual quality cannot compete with model-based approaches.

In [55], we pointed out that soft- and non-biometric identifiers, e.g. hair colour/style, ears, clothing, etc. could also significantly increase the re-identification risk. Later, Brkić et al. proposed a face de-identification system to address on face, hairstyle and clothing colour [56]. In their system, the face area is segmented from the background. Based on the face detection and segmentation information, a synthetic face generated by DCGAN is used to replace the original face, and a segmented hairstyle image then is put on the image to replace the original hair. Finally, a clothing/background recolour scheme is used to preserve the skin colour while altering the colour of the clothing.



# Chapter 3

## Related work

### 3.1 Face appearance models

In all the work presented in this thesis, statistical appearance models have been used as the face representation model. This section starts with an introduction to principal component analysis, which is the key technique in the AAM framework for dimension reduction. The section then moves on to a mathematical description of the appearance models.

#### 3.1.1 Principal component analysis

The Principal Component Analysis (PCA) is an important algorithm in modern data analysis. It is non-parametric, and it is categorised to unsupervised learning in machine learning field. PCA is an effective way to reduce a complex dataset to a lower dimension to reveal the simplified data structures. The process of PCA is guided by the assumptions including linearity, the large variances have important structure and the principal components are orthogonal.

PCA can be solved by eigenvector decomposition or SVD. Consider a zero-mean dataset  $\mathbf{X}$ , represented as an  $m \times n$  matrix, and its covariance matrix

$$\mathbf{C}_X = \frac{1}{n-1} \mathbf{X} \mathbf{X}^\top. \quad (3.1)$$

According to the assumptions, the principal components  $\mathbf{P}$  of  $\mathbf{X}$  can be found through  $\mathbf{Y} = \mathbf{P} \mathbf{X}$  with the condition that  $\mathbf{C}_Y = \frac{1}{n-1} \mathbf{Y} \mathbf{Y}^\top$  is a diagonal matrix. It is proved that  $\mathbf{P}$  diagonalises  $\mathbf{C}_Y$  if  $\mathbf{P}$  is the transpose of eigenvectors of  $\mathbf{C}_X$ . Furthermore, the

eigenvectors are re-ordered depending on their eigenvalues  $\lambda$  so that large variances have importance structure and lower variances represent noise.

SVD is a powerful factorisation method which converts any arbitrary  $n \times m$  matrix  $A$  to two orthogonal matrices and a diagonal matrix.

$$A = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \quad (3.2)$$

where  $\mathbf{U}$  is an  $n \times n$  orthogonal matrix,  $\mathbf{V}$  is an  $m \times m$  orthogonal matrix which is the eigenvectors of  $A^\top A$ , and  $\mathbf{\Sigma}$  is a diagonal matrix whose diagonal is the singular values of  $A$ .  $\mathbf{V}$  is ordered so that the corresponding eigenvalues  $\lambda$  satisfy  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ . Furthermore, the diagonal entries of  $\mathbf{\Sigma}$  are singular values  $\sigma_i = \sqrt{\lambda_i}$ . It is intimately related to PCA, so the principal components of  $\mathbf{X}$  can be solved by SVD. Let

$$A = \frac{1}{\sqrt{n-1}} \mathbf{X}^\top \quad (3.3)$$

imply that

$$A^\top A = \frac{1}{n-1} \mathbf{X}\mathbf{X}^\top = \mathbf{C}_X \quad (3.4)$$

Therefore,  $\mathbf{V}$  is the eigenvectors of  $\mathbf{C}_X$ , and its columns are the principal components of  $\mathbf{X}$ . The first few principal components could retain majority variance of  $\mathbf{X}$ .

$$\sum_{i=1}^k \lambda_i \geq (1 - \epsilon) \sum_{i=1}^m \lambda_i, \quad 1 \leq k \leq m; 0 \leq \epsilon \leq 1, \quad (3.5)$$

$$\hat{\mathbf{X}} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k] \mathbf{Y}, \quad \mathbf{Y} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k]^\top \mathbf{X}, \quad (3.6)$$

where  $\mathbf{p}_i \in \mathbb{R}^m$  is the  $i$ th principal component of  $\mathbf{X}$  and  $k$  is the number of principal components to retain a certain amount of variance. In other words,  $x_i$  in  $\mathbf{X}$  could be represented by  $k$  PCA parameters approximately (3.6). This is the reason why PCA can achieve dimension reduction. Fig. 3.1 shows that in a face model, the top 10 principal components can maintain more than 95% variance of the shape information.

### 3.1.2 Statistical models for face appearance

Statistical face appearance models are widely used in face image editing, face synthesis, face alignment and face de-identification [30], [57]–[59]. Recently, this

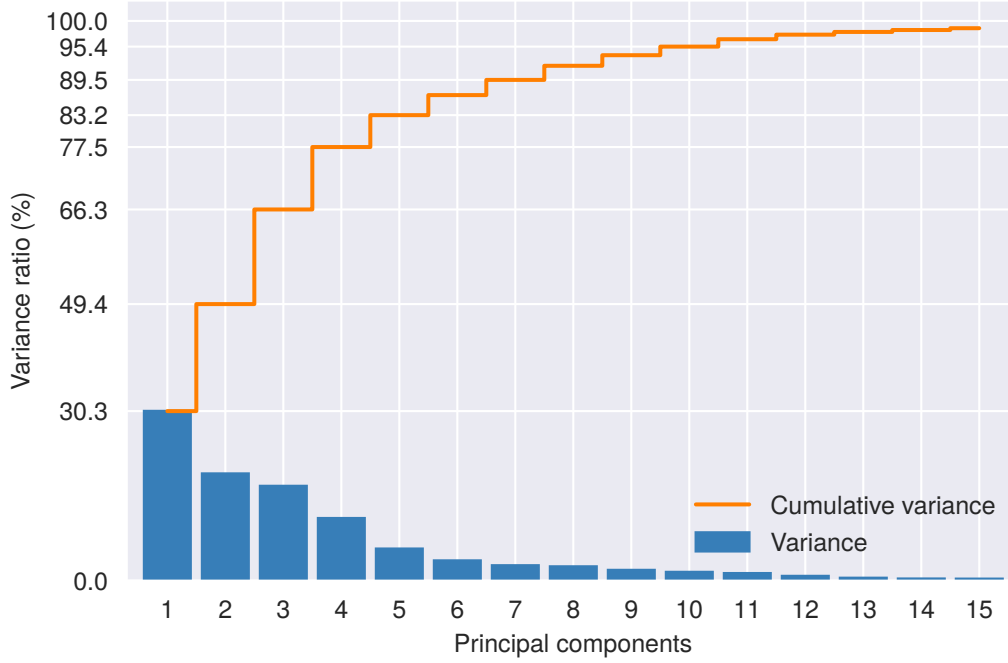


Fig. 3.1 PCA variances of face shape models used in following experiments.

model has been used to decode face appearance from neural population responses in the primate brain [60]. It is a linear model consisting of a model of face shape variances, Point Distribution Model (PDM) [61], and a model of shapeless facial texture variances. The shape  $s$  is defined by  $n$  facial landmarks. Then a facial triangle mesh can be constructed with the shape points (see Fig. 3.2). The facial landmarks are annotated manually or detected automatically by algorithms. The techniques to detect facial landmarks are introduced in the next section.

$$s = [x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T \in \mathbb{R}^{2n}. \quad (3.7)$$

Applying PCA to a training set of  $s$ , we obtain the eigenvectors  $\Phi_s \in \mathbb{R}^{2n \times k_s}$  and eigenvalues  $\lambda_s \in \mathbb{R}^{k_s}$ . We use a proportion of the total variation of  $\lambda$  to decide the number of components  $k_s$ . The model represents a given shape  $s$  as

$$\hat{s} = \bar{s} + \Phi_s \alpha, \quad (3.8a)$$

$$\alpha = \Phi_s^T (s - \bar{s}), \quad (3.8b)$$

where  $\bar{s}$  is the mean shape and  $\alpha \in \mathbb{R}^{k_s}$  is the shape parameters.

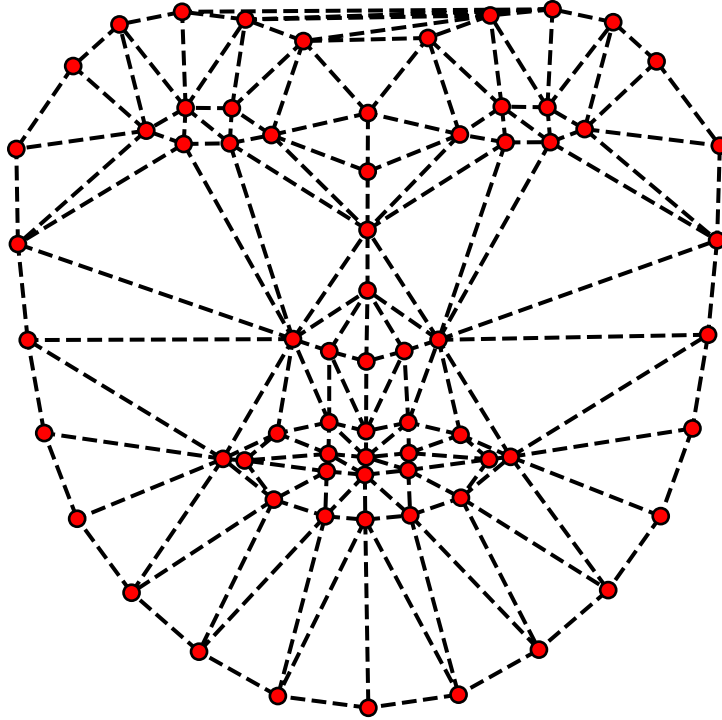


Fig. 3.2 Face shape of 111 triangles defined by 68 vertices. This 68 facial landmark scheme is the most popular facial landmark scheme, which was used in CMU Multi-PIE [62] and IBUG 300-W [63] datasets.

The shapeless texture  $\mathbf{t} \in \mathbb{R}^{3m}$  is a vector of  $m$  pixel values which is obtained by warping all the pixels inside a convex hull around  $\mathbf{s}$  to the mean shape  $\bar{\mathbf{s}}$ . The image warping process applies piecewise affine warp on each triangle in the triangle mesh. The same as shape, PCA is applied to a set of  $\mathbf{t}$  to obtain the texture principal components  $\Phi_t \in \mathbb{R}^{3m \times k_t}$  and eigenvalues  $\lambda_t \in \mathbb{R}^{k_t}$ . A given texture  $\mathbf{t}$  is represented in the model as

$$\hat{\mathbf{t}} = \bar{\mathbf{t}} + \Phi_t \boldsymbol{\beta}, \quad (3.9a)$$

$$\boldsymbol{\beta} = \Phi_t^\top (\mathbf{t} - \bar{\mathbf{t}}), \quad (3.9b)$$

where  $\bar{\mathbf{t}}$  is the mean texture and  $\boldsymbol{\beta} \in \mathbb{R}^{k_t}$  is the texture parameters.

Semantic meanings are observed in the first several principal components in both shape models and texture models (see Fig. 3.3 and 3.4). These semantic meanings are used in the pre-processing stage of face de-identification process. To keep the shape model parameters and texture model parameters separate, shape parameters and texture parameters are merely concatenated to construct appearance parameters



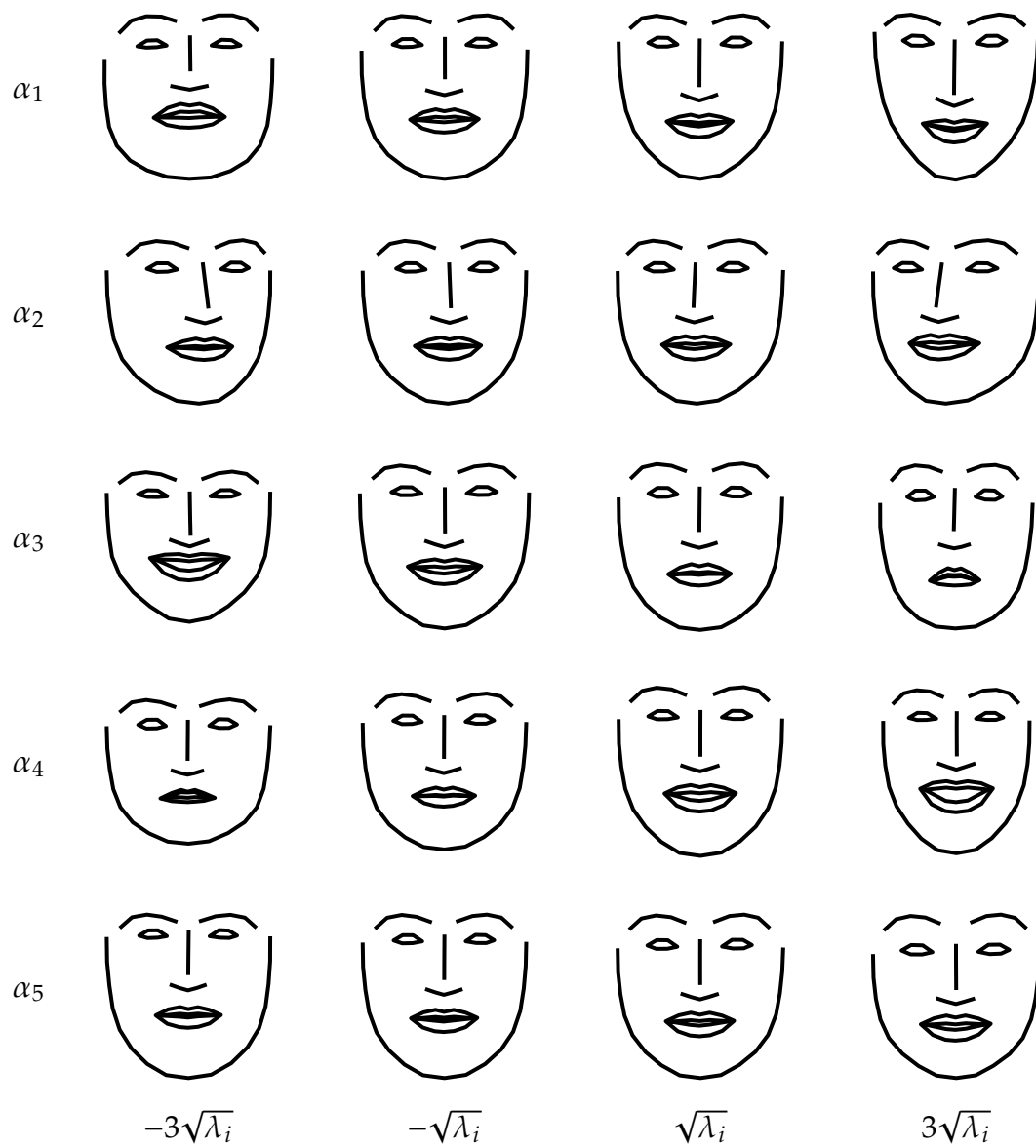


Fig. 3.3 Face shape models



Fig. 3.4 Face texture models

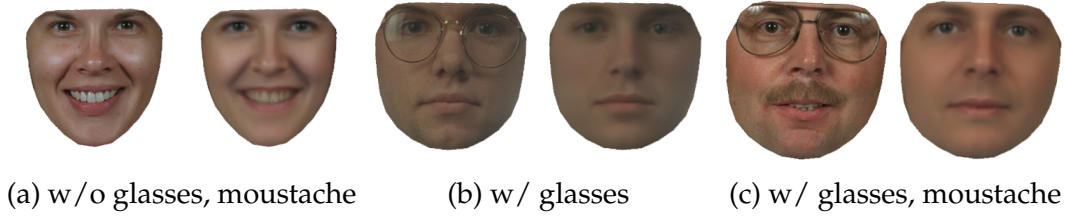


Fig. 3.5 The comparison of original faces and their reconstructed faces from the corresponding appearance model parameters. Original face on the left and reconstructed face on the right.

without further reduction.<sup>1</sup> Shape parameters and texture parameters have different units. To make  $\alpha$  and  $\beta$  commensurate, weights need to apply to one set of the parameters. The choice of parameter weights has been discussed in [41], and it suggests a simple method to compute the shape weights  $W_s$ :

$$W_s = rI, \quad r^2 = \frac{\|\lambda_t\|_1}{\|\lambda_s\|_1} \quad (3.10)$$

$$\gamma = \begin{bmatrix} W_s \alpha \\ \beta \end{bmatrix} \quad (3.11)$$

In the following sections, this face appearance model is used to represent each face as a vector  $\gamma$  which is the concatenation of shape parameters and texture parameters.

In the experiments, glasses, beard and moustache are regarded as unexpected features as they cannot be appropriately represented in the appearance models. PCA assumes data following Gaussian distribution. However, facial hairs and facial accessories are not Gaussian distributed in a dataset. To achieve the goal of reducing the variances of the unexpected facial components in the PCA, the images in the appearance model training set are manually selected to make the unexpected facial components rarely appear in the training set, so that they are regarded as noise and ignored by the model. Fig. 3.5 shows that the glasses, beard and moustache are ignored by the model with minor effect on other facial components.

<sup>1</sup>In [41] the appearance models are obtained by applying a (third) PCA to the concatenation of shape model parameters and texture model parameters.

## 3.2 Facial landmark detection

Facial landmark localisation is a key step of face alignment in face recognition [64], [65] and face de-identification [30], [47], [66]. It also plays an important role in many other face related applications including face animation [67], face tracking [68]–[70] and face modelling [57], [59], [64], [71]. As mentioned in the previous section, the facial landmark coordinates can define the shape of a face.

Various approaches have been proposed to achieve automatic facial landmarks detection. These approaches build models based on the face appearance or features. In general, the model can be a generative model such as AAMs [58], [72]–[74], or a discriminative model such as Active Shape Models (ASMs) [61], CLMs [39], [75] or Deformable Part Models (DPMs) [76]. The facial landmarks are allocated through the model fitting. It can be a challenging task because of the large variances in the head pose, unexpected obscure or uneven illumination on the image. Recently, more deep neural network approaches [28], [77], [78] have been proposed to address the facial landmark detection task.

Cootes et al. proposed ASM [61] to solve the problem of locating examples of known objects in images. ASM adds a fitting process directly to the PDM which is the shape model mentioned in Section 3.1.2. The ASM fitting process starts by placing a model instance at an approximate position, and the fitting process updates the position, orientation, scale and the shape model parameters of the instance until convergence. The location of each model point is updated along profiles normal to the model boundary until the model boundary corresponds to an edge in the image. The performance of this method is limited because it requires a relatively simple structure of the object in the image.

ASM only uses the shape information of an object while the later proposed AAM [58] represents both shape information and texture information in an appearance model as described in Section 3.1.2. Unlike ASM which minimises the distance between model points and the corresponding points in the image, AAM fitting process minimises the difference between the synthesised model image and the input target image. However, solving such an optimisation problem is computationally expensive. It is inefficient using the standard gradient descent algorithms in AAM fitting because the partial derivatives and Hessian need to be recomputed in each iteration. In [58], Cootes et al. assume that the relationship between error image and additive increments to the parameters are linear, so the gradients are calculated only once in their algorithm. Furthermore, Matthews and Baker [42] extended the inverse

compositional algorithm for AAMs to import the fitting performance. AAM require the appearance model can generate an appearance close to the input image while using one appearance model sometimes is not accurate. As a result, view-based AAMs [79], person-specific AAMs [80] were proposed for specific purposes.

In 2008, Cristinacce and Cootes [39] showed an efficient and robust method of locating a set of landmark points in an object of interest. From a training set, they constructed a joint model of the appearance of each landmark point together with their relative positions. The method of building the CLM is similar to the AAM. The main difference is that instead of modelling the whole object region CLM models a set of local feature templates based on the known spatial relationship between the local features. It is an alternative method of AAM. In CLM the detector is constrained by a face shape template. Thus the detector will not search randomly in the image but a particular interest area. It makes the CLM algorithm more robust and more accurate than the AAM search method. The processing time of CLM is 4 fps for static images. However, when tracking face with CLM the processing time is able to achieve 25 fps, which is fast enough to process a video in real-time.

In 2011, Sauer et al. [81] showed the superior generalisation performance of random forest within a Sequential AAM framework. The Random Forest can be as efficient as a boosting procedure without significant reduction in accuracy. Later, Cootes et al. [82] demonstrated that combining the Random Forest regression with a statistical shape model (e.g. CLM, ASM or Pictorial Structure Matching) is significantly faster and more accurate than equivalent discriminative or boosted regression-based methods trained on the same dataset. The proposed hybrid took 27 ms (37 fps) on a single core to search on a face image [82]. This approach is fast enough to track faces in real-time. Another contribution of this approach is on the robustness because the coarseness of the sampling step can be adjusted to balance between speed and accuracy.

Baltrusaitis et al. [40] proposed the CLNF for facial landmark detection and tracking in 2013. This model includes two main novelties. First, it introduced a probabilistic landmark detector that can learn non-linear and spatial relationships between the input pixels and the probability of a landmark being aligned. Secondly, this model is optimised using a novel non-uniform Regularised Landmark Mean-Shift optimisation technique, which considers the reliabilities of each patch expert. Because this approach considers the unseen illumination and in-the-wild data, the experimental results showed that its accuracy is higher than CLM. It is a complex

system, and the main drawback is that the processing speed is not fast enough to achieve a real-time process.

Most of the landmark detection methods minimise the error functions through second-order descent methods. However, such optimisation schemes have two main drawbacks: (a) the error function might not be differentiable; and (b) the Hessian might be large and not positive definite. To overcome these drawbacks, Xiong and De la Torre [83] proposed Supervised Descent Method (SDM) for minimising a non-linear least square function. This method learns a set of descent directions through supervised training. The learned descent directions are used in the test phase to minimise the objective function without computing the Jacobian nor the Hessian. This method is fast and able to achieve real-time processing in face landmark detection. The results in [83] showed that it outperformed the person-specific AAM in terms of accuracy.

Furthermore, the research on facial landmark detection is not only in 2D but also 3D. Sangineto published a paper of pose and expression independent facial landmark localisation in 2012 [84]. It presented an approach to automatic localisation of facial feature points which deals with pose, expression, and identity variations combining 3D shape models with local image patch classification. A SURF-like feature, which called Dense Upright SURF (DU-SURF) is used in this approach and compared with other modern features. The experiment results showed that the accuracy of the proposed approach is higher than CLM.

In recent years, Convolutional Neural Network (CNN) is the state-of-the-art approach to extract features from images. Face landmark detection methods also adopt CNN to improve their robustness and accuracy. In [85], Zhou et al. designed a coarse-to-fine deep CNN cascade which is able to detect a large number of facial landmarks. Four deep CNN levels are designed to form the network cascade. The proposed system can detect 51 inner points (mouth, nose, eyes and eyebrows) and 17 contour points from a human face.

Trigeorgis et al. [77] proposed the convolutional recurrent neural network architecture which can be trained in an end-to-end manner. The convolutional module extracts features and the recurrent module facilitates the optimisation of the regressors. It is shown that the learned features from the convolutional module outperform the hand-crafted features for landmark detection. In contrast to traditional cascaded regression frameworks, the recurrent module acts as a memory unit which shares information across all cascade levels. The accuracy of this method outperforms the state-of-the-art methods including SDM and the cascade CNN.

Table 3.1 Types of similarity metrics that are used in feature matching

Metric	Formula
Euclidean distance	$\left(\sum_{i=1}^n (x_i - y_i)^2\right)^{1/2}$ or $\ \mathbf{x} - \mathbf{y}\ $
City block distance	$\sum_{i=1}^n  x_i - y_i $ or $\ \mathbf{x} - \mathbf{y}\ _1$
Minkowski distance	$\left(\sum_{i=1}^n  x_i - y_i ^p\right)^{1/p}$ or $\ \mathbf{x} - \mathbf{y}\ _p$
Cosine similarity	$1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\ \mathbf{x}\  \ \mathbf{y}\ }$
Chi-squared distance	$\sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i + y_i}$

### 3.3 Face recognition

There are two types of face recognition tasks:

**Verification** (one to one matching) — “is the person whom he/she claims to be”.

**Identification** (one to many matching) — “who the person is”.

A face recognition process usually includes face detection, face alignment (see Section 3.2), feature extraction and feature matching, in which feature extraction and feature matching are the two key steps. Verification and identification can share the first three steps, and the only difference can be in the last feature matching step. Typically, both verification and identification compare the similarity of the face features; while sometimes identification on a close set can get the probability distribution from a classifier directly, e.g. using a softmax function in neural networks.

#### 3.3.1 Feature matching

The similarities can be measured using many metrics. Table 3.1 shows the commonly used metrics. Euclidean distance and city block distance are two particular cases of Minkowski distance when  $p = 1, 2$ . Euclidean distance is the most common metric used in feature matching and squared Euclidean distance,  $\|\mathbf{x} - \mathbf{y}\|_2^2$ , is often implemented for efficiency. Unlike other distances, cosine similarity measures the angular difference between two vectors. It is also used for feature matching [86], [87]. Chi-squared distance can be regarded as weighted square Euclidean distance. It usually be used to compare the similarity of two histograms.

### 3.3.2 Face features

Face features are the low dimensional representation of face images in a face manifold or subspace. They keep the discriminative information which is useful for distinguishing the faces from different identities and reduce the redundant information which is not related to identities to improve the robustness and efficiency of face recognition system. Face features can be divided into three categories, depending on the manner in which they are constructed.

**Holistic approach** analyses the whole image or region(s) of interest. The feature represents the most desired characteristics of the image. For example, applying linear methods such as PCA, ICA and LDA to the image.

**Local representation approach** uses local image (texture) descriptor, e.g. Local Binary Patterns (LBP), Local Phase Quantisation (LPQ), Histogram of Oriented Gradient (HOG), SIFT, SURF, etc. Then the concatenation or the histogram of the local representations is used as a feature vector.

**Deep neural network approach** is an end-to-end approach that learns the face features through optimising a loss function. In contrast to hand-crafted features mentioned above, the learned features tend to be more robust and suitable for the face recognition task, although they are hard to explain.

#### Holistic approaches

At the early stage, face recognition used to employ a local approach where the features were facial geometric information. In [88], the facial landmarks such as the position of eyes, nose, ears, etc. were used to construct a 16 dimension feature vector which included the ratio of landmark distances and angles. The similarity of vectors was measured with Euclidean distance. The advantage of using facial geometric information is that it is insensitive to the illumination. Nevertheless, automatic facial landmark detection techniques are not always reliable even now.

Later the colour intensity was used to compare the similarities between face images. However, the vector of pixels in one image has a large dimensionality. The dimension reduction methods such as PCA was therefore employed in face recognition. The first application of PCA to face recognition was proposed by Turk and Pentland [89]. This method decomposes a face image into a mean face plus a weighted sum of principal components under the assumption that the face manifold is linear. As each principal component has a face shape like structure, this method



is also known as “Eigenfaces” technique where each principal component is an Eigenface.

The PCA method selects the face subspace which contains the most variation in the training set. However, they are not necessarily related to the identities. For example, Fig. 3.6a illustrates the distribution of the PCA-based face features. The large variation in head pose or face expressions can introduce a shift to the face features that is higher than the difference between identities. This makes PCA-based face recognition systems noticeably sensitive to illumination, head pose and expressions. Linear and nonlinear approaches are proposed to address this problem. The most straightforward way is to find a subspace in the PCA subspace via Fisher’s Linear Discriminant (FLD) or Linear Discriminant Analysis (LDA). This method was proposed by Belhumeur et al. [90] and named as “Fisherfaces”. It is a supervised learning process to find the linear subspace  $\Phi$ , which maximises the ratio

$$\arg \max_{\Phi} \frac{|\Phi^T S_b \Phi|}{|\Phi^T S_w \Phi|}, \quad (3.12)$$

where  $S_b$  denotes the between-class (inter-class) scatter matrix and  $S_w$  denotes the within-class (intra-class) scatter matrix. The face features will be further reduced to  $c - 1$  dimension, where  $c$  is the number of classes. Experiments in [90] showed the Fisherface technique outperformed Eigenface in handling variation in illumination and facial expressions.

Edwards et al. [91] demonstrated AAM is able to improve the stability of face identification and tracking when there is high variation in pose, expression and illumination. Same as PCA features, the appearance model features are not guaranteed to be related to the identities. In their method, LDA was used to find the ID-subspace [92]. The extracted model parameters are projected to the ID-subspace to facilitate face identification.

### Local representation approaches

Apart from using the holistic appearance information, local representation of the image is also used in face recognition. Gabor filter was used in early face recognition literature. In [93], Lades et al. proposed a Gabor based object recognition system using dynamic link architecture. Later, Wiskott et al. extended the work and proposed the Gabor wavelet-based elastic bunch graph matching (EBGM) to recognise face [94]. In EBGM, the nodes in the graph are located at few important facial landmark points

such as the position of the eyes and the tip of the nose. 40 complex coefficients (5 frequencies  $\times$  8 orientations) are obtained on each node. The similarities of feature vectors are measured in cosine similarity.

LBP is a simple yet powerful image texture descriptor proposed by Ojala et al. [95], [96]. A binary sequence is obtained by comparing the differences among the centre pixel and its neighbourhoods. The binary pattern of a centre pixel  $g_c$  with  $p$  neighbourhoods in the radius  $r$  is calculated by

$$\text{LBP}_{p,r} = \sum_{i=0}^{p-1} s(g_i - g_c) 2^i, \quad (3.13)$$

where

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.14)$$

Furthermore, [96] defined the rotation invariance with uniform patterns through counting the number of spatial transitions (bitwise 0/1 changes) in the patterns. The uniform patterns have at most two spatial transitions, and they are denoted as  $\text{LBP}_{p,r}^{u2}$ . For example, the patterns  $1111111_2$  (0 transition) and  $0011000_2$  (2 transitions) are uniform patterns whereas the patterns  $11001100_2$  (4 transitions) and  $10101100_2$  (6 transitions) are not. The histogram of uniform patterns shows a better discrimination than the histogram using all patterns. Essentially, the uniform patterns represent the relatively low frequency information in an image that can only be found at edges, corners and flat parts.

The application of using LBP in face recognition was proposed by Ahonen et al. [97]. In their work, the face features are constructed by the histogram of LBP from each block in a  $7 \times 7$  grid on the image. Different weights are applied to different blocks. For example, the eye regions have a higher weight value as they are more important for face recognition than many other regions. The similarities between the face features are compared in a weighted chi-squared distance. The experiment results in [97] showed that the LBP outperformed the PCA and EBGGM methods.

LPQ [98] is another local descriptor. The LPQ descriptor uses the 2D discrete Fourier transform to extract local phase information in a window for every image position. The phases of the four low-frequency coefficients are decorrelated and uniformly quantised as an eight-bit binary. LPQ features are invariant to blurring, which has been confirmed by the experimental results in [99].

### Deep neural network approach

CNN has been used in the early research of face recognition [100]. It did not become a popular approach to face recognition and other challenges in computer vision field until recent years, mainly due to the limitation of computational power and lack of data. Large enough training set is essential to a successful tuning of the vast number of parameters in a deep neural network. Zhou et al. [101] achieved a 99.5% recognition accuracy on the Labelled Faces in the Wild (LFW) benchmark using a ten layer CNN which was trained with 5 million faces from 20 000 identities. The network was trained as a multi-class classifier with a cross-entropy loss. In the test phase, features from the last hidden layer were reduced through PCA and then the similarities between features were measured in Euclidean distance.

In [64], a 3D face model was employed for face alignment, and a nine-layer neural network was trained for extracting features. This method achieved a 97.35% accuracy on LFW, and it was one of the top results in 2014. Later on, more face recognition research focused on the latent representations of the face images and made the training for face recognition task different from other multiclass classification tasks.

Face recognition is not precisely a multiclass classification problem, because in most applications the number of classes (identities) is vast and sometimes not fixed. The bottleneck layer feature embeddings are usually used to facilitate a face verification or open set face identification task. It is required that the learned features are discriminative, i.e. they have less difference within the same identity and more significant difference between different identities (see Fig. 3.6). However, the features trained with a softmax classifier cannot guarantee this requirement. Metric learning was then considered to address this problem. Metric learning in face recognition aims to shorten the intra-class distance while increasing the inter-class distance. Starting from DeepID2 [102], the DeepID family [103], [104] have made use of a joint loss which contains an identification signal and a verification signal. The identification signal is a cross-entropy loss for different identities, and the verification signal is a contrastive loss for the images of the same identity.

$$L_{\text{Verif}} = \begin{cases} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, & \text{if } y_i = y_j \\ \frac{1}{2} \max(0, m - \|\mathbf{x}_i - \mathbf{x}_j\|_2)^2, & \text{otherwise} \end{cases} \quad (3.15)$$

where  $\mathbf{x}$  is the DeepID vector extracted from the input image and  $m$  denotes a margin between two identities.

Furthermore, Schroff et al. proposed FaceNet [105] which removes the softmax classifier from the network architecture. The aim of the neural network to extract unified feature embeddings which can support both recognition and clustering tasks. The embeddings are constrained on a  $d$  dimensional hypersphere via  $L^2$  norm. The network is trained with the triplet loss defined as

$$L = \sum_{i=1}^n \left[ \left\| \mathbf{x}_i^a - \mathbf{x}_i^p \right\|_2^2 - \left\| \mathbf{x}_i^a - \mathbf{x}_i^n \right\|_2^2 + \alpha \right]_+ \quad (3.16)$$

where  $x$  denotes the embedding of the input image,  $a$  denotes an anchor sample,  $p$  and  $n$  denote the positive (intra-class) and the negative (inter-class) samples to the anchor sample and  $\alpha$  denotes a margin that is enforced between positive and negative pairs. FaceNet achieved 99.63% accuracy on LFW dataset. However, it was trained on a 200 million face image dataset and the selection of triplets  $(a, p, n)$  is tricky.

Wen et al. proposed another method to extract discriminative features using CNN architecture [106]. They introduced the centre loss to the CNN training

$$L_C = \frac{1}{2} \sum_{i=1}^m \left\| \mathbf{x}_i - \mathbf{c}_{y_i} \right\|_2^2, \quad (3.17)$$

where  $\mathbf{x}_i$  denotes the  $i$ th deep feature,  $y_i$  denotes the class label of  $i$ th  $x$  and  $\mathbf{c}_{y_i} \in \mathbb{R}^d$  denotes the centre of the  $y_i$ th class. The neural network was trained with a joint loss of softmax cross-entropy loss and centre loss

$$\begin{aligned} L &= L_S + \lambda L_C \\ &= - \sum_{i=1}^m \log \frac{\exp(\mathbf{W}_{y_i}^\top \mathbf{x}_i + b_{y_i})}{\sum_{j=1}^n \exp(\mathbf{W}_j^\top \mathbf{x}_i + b_j)} + \frac{\lambda}{2} \sum_{i=1}^m \left\| \mathbf{x}_i - \mathbf{c}_{y_i} \right\|_2^2 \end{aligned} \quad (3.18)$$

Training on a relatively small dataset, this method achieved a 99.28% accuracy on LFW dataset. Fig. 3.6 is the data visualisation of face features from 10 identities aiming to compare the traditional PCA-based features and CNN learned features. Fig. 3.6a demonstrates the distribution of the AAM face features, which are PCA-based features. Fig. 3.6b demonstrates the distribution of the CNN learned face features trained with centre loss. In contrast to PCA-based face features, the features learned by CNN are more discriminative and more useful for feature matching task.

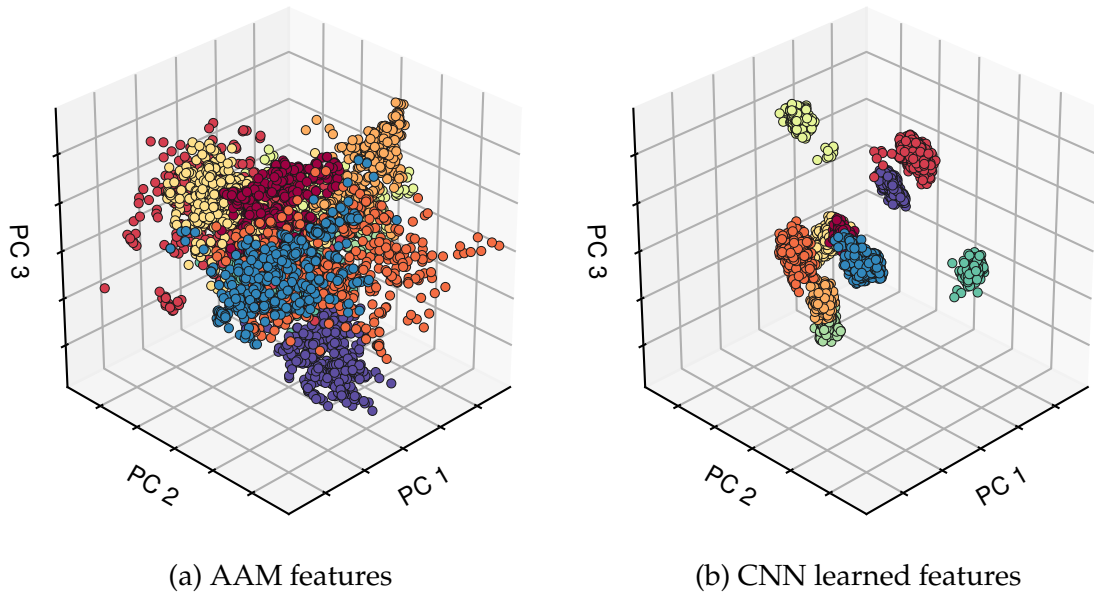


Fig. 3.6 Comparison of AAM face features and CNN learned face features. Top 3 PCA of both features from 10 randomly selected identities in the UNBC-McMaster dataset. UNBC-McMaster dataset is not used in the training phase of either model. CNN learned face features are more discriminative because the intra-class distances are small, and the variations of head-poses and expressions are ignored.

The CNN architectures in face recognition keep updating with the state-of-the-art network architectures, i.e. the networks become deeper and wider. The VGG building blocks and more advanced inception blocks and residual blocks have been employed in recent face recognition work [105], [107], [108].

### 3.4 Facial expression synthesis

Examples of data utility that are often desired and should be preserved include gender, age and facial expressions, among many others depending on the purpose of the application. The data utility that this thesis most concentrate on is the facial expressions. The facial expression contains rich information about emotion and intention, which makes the facial expression recognition a challenging task. The category of facial expression sometimes is coarse for analysis, and the intensity of the facial expression needs to be detected. Facial expression intensities are quantified results of pre-defined expressions, e.g. Facial Action Coding System (FACS) [109]. Analysis and synthesis of facial expressions can be traced back to the 1970s [110]. Since then significant research efforts have been made to generate realistic synthesised

facial expressions. These have led to the advancement of FET, which has been widely implemented in computer graphics and vision, character animation, computer games and advertising. The movie industry has long been refining motion capture and the transfer of facial expressions from an actor to a CGI-generated movie character [111], [112]. Furthermore, the MPEG-4 Face and Body Animation (MPEG-4 FBA) is the part of MPEG-4 Standard (ISO14496), which uses a set of points to represent the face and body movement of animation characters.

In recent years, AAM [57] has been broadly used for building non-rigid deformable models. In face biometrics, this model provides a compact statistical representation of the shape and appearance variation of the face as measured in 2D images. AAM derived facial representations were employed in a prior study [113], and its experimental results proved representations using this method to be highly useful for the task of facial action recognition.

A relevant work [114] focuses on real-time dynamic facial expression transfer using AAMs, generating realistic talking faces in real-time at a low computational cost. The proposal assessed how a fitted expression from one AAM could be used to synthesise the same expression realistically onto another person or animated character in a separate AAM. The procedure has been seen to produce video sequences that are smooth and seemingly acceptable. Similarly, the study in [115] described techniques for manipulating facial gestures and global head movements in video sequences of people engaged in conversation. These operations can be applied blindly to participants interacting through a video conference. Such techniques operate in real time at video frame-rate due to the simple mapping of parameters between AAMs, which is performed without requiring high-level semantic information about the facial expressions. A more recent implementation [116] presented ad hoc control of the facial expressions of a target actor by cloning the facial expressions from an actor in a source video, which is in real-time as well. This method maximises photometric consistency between the input and the re-rendered output video so that the synthesised expressions are virtually indistinguishable from a real video.

# Chapter 4

## Face de-identification with cluster swapping

### 4.1 Introduction

It has been mentioned in Chapter 2 that the  $k$ -Same and model-based  $k$ -Same methods form clusters of similar original face images and synthesise the de-identified face images for each cluster using the average face of the cluster members. These methods comply with the  $k$ -anonymity theorem and hence can guarantee a re-identification risk lower than  $1/k$ . However, the theoretical upper bound  $1/k$  means that a large value of  $k$  is required in order to achieve a low re-identification risk.

The most common attack on the de-identified face dataset is that the attacker uses the de-identified face image to match with a gallery of real face images and tries to find the real identity of the de-identified face image. If there always exists a gallery image which has a shorter distance to the de-identified face image than its true original identity, then the attacker will always mismatch the de-identified faces. This is also known as wrong-map protection. Data swapping is one approach to provide wrong-map protection. However, unlike tabular data which can swap multiple attributes, a face dataset can only swap the face images against their identity labels and vice versa. This means when wrong-map protection is applied to the face dataset directly, the face images in the original dataset still have identical copies in the de-identified dataset. The de-identified images are still in a high-risk status, although the links to their real labels are broken.

This chapter introduces a novel face de-identification method named  $k$ -Same-furthest, which aims to further reduce the re-identification risk and remove the

dependence of re-identification risk on  $k$ . The proposed method applies a wrong-map to the  $k$ -Same clustering results to further reduce the re-identification risks. Specifically, based on the  $k$ -Same-M framework, the proposed method adds data swapping processing to the  $k$ -Same-M de-identification method. It selects a pair of clusters that have the longest distance between each other in the feature subspace and swaps the mean values of the pair of clusters. In terms of reducing the re-identification risk, the proposed method gathers the advantages from both  $k$ -Same and wrong-map protection.

- 1) There are at least  $k$  identical copies of a face image in the de-identified face set and a face recognition software will associate all  $k$  copies to one particular identity although they originally associate to  $k$  different identities.
- 2) The face recognition software tends to recognise to a wrong identity because the de-identified faces have a large distance to their original faces in the face feature space.

On the other hand, the  $k$ -Same-furthest face de-identification method itself cannot provide the preservation of the data utility. It assumes that the desired data utility is not represented or has a low variance in the face feature space. Nevertheless, the  $k$ -Same-furthest face de-identification can combine with other approaches to achieve data utility preservation. For example, apply it to a data utility biased subset such as in  $k$ -Same-select [29] or transfer data utility information from original face to the de-identified face (see Chapter 7).

## 4.2 $k$ -Same-furthest

The proposed face de-identification method,  $k$ -Same-furthest, is a  $k$ -Same solution, which repeats each de-identified face at least  $k$  times in the de-identified face set  $H_d$ . To maximise the removal of PII from the original faces, the  $k$ -Same-furthest method uses wrong-map and de-identifies each original face  $I \in H$  with the centroid of the cluster that is, identity-wise, furthest away from it. In contrast to the proposed  $k$ -Same-furthest method, the original  $k$ -Same method will be described as  $k$ -Same-closest in the following descriptions.

The  $k$ -Same-furthest calculates the average as the aggregate of  $k$  faces, although other measures can be used to perform the aggregation. The identity similarity can be measured in the pixel space, or a projected feature subspace such as the Eigenface subspace [89] or the feature space constructed by an AAM [42], [57]. To prevent ghost



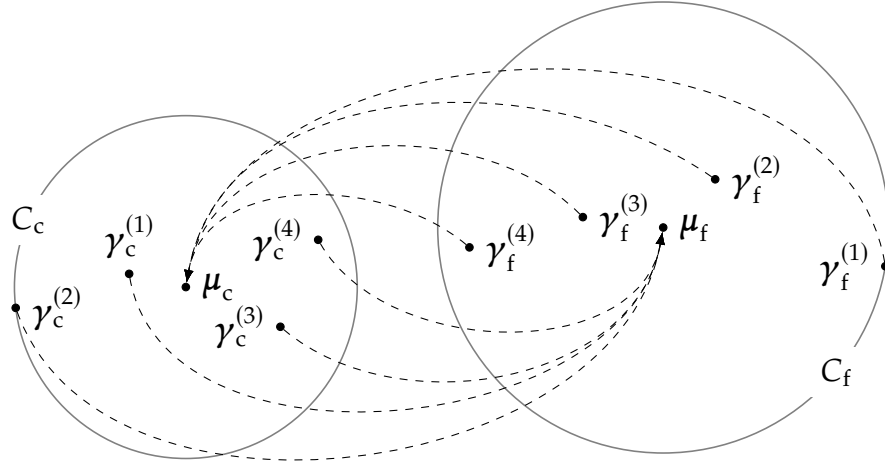


Fig. 4.1 An iteration of the  $k$ -Same-furthest face de-identification process with an example data set, where original samples  $\gamma_f^{(i)}$  in the cluster  $C_f$  are de-identified as  $\mu_c$  (the centroid of the cluster  $C_c$ ) and original samples  $\gamma_c^{(i)}$  in  $C_c$  are de-identified as  $\mu_f$  (the centroid of cluster  $C_f$ ).  $i = 1, 2, \dots, k$  with  $k = 4$  here.

artefacts in the de-identified faces,  $k$ -Same-furthest performs averaging of faces in the AAM feature space which is the same as the model-based  $k$ -Same method [30].

The  $k$ -Same-furthest de-identification process is iterative. In each iteration, two clusters are formed hierarchically from two seeds, and members of a cluster are de-identified with the centroid of the other cluster. The clustering process in  $k$ -Same-furthest ensures the two seeds have a maximum Euclidean distance between each other in the AAM feature space.

For a given face set  $H$ , the proposed  $k$ -Same-furthest projects  $H$  to the feature space constructed by a trained AAM and generates  $\Gamma$  which is a set of appearance model features of the original faces  $\gamma$ . The feature set is stored as a matrix with model features as the row vectors. For a person-specific dataset, each row represents one unique identity. All the subsequent operations of de-identification are performed in the AAM model feature space. Each iteration of the proposed de-identification process deals with at least  $2k$  faces remaining in  $\Gamma$  and remove them from  $\Gamma$  at the end of each iteration.  $\Gamma$  becomes an empty set when the de-identification process is completed.

When there are more than  $2k$  faces in  $\Gamma$ , the  $k$ -Same-furthest algorithm de-identifies  $2k$  original faces. Fig. 4.1 illustrates the results of a de-identification iteration for an example data set when  $k = 4$ . To simplify illustration and ease of

understanding, a 2D dataset of scalars is used in Fig. 4.1.  $\gamma_c^{(1)}$  is the feature vector of a face that triggers the de-identification process and  $\gamma_f^{(1)}$  is the furthest feature vector to  $\gamma_c^{(1)}$ .  $C_c$  and  $C_f$  denote the clusters which is closest and furthest to  $\gamma_c^{(1)}$ . A high dimensional sphere in the feature space is defined by the cluster centroid  $\mu$  and radius  $r$ .

$$\mu = \frac{1}{|C|} \sum_{i=1}^{|C|} \gamma_i \quad (4.1)$$

$$r = \max_{\gamma \in C} \|\gamma - \mu\| \quad (4.2)$$

The cluster centroid  $\mu$  and the cluster sphere radius  $r$  are updated whenever a new member is added to the cluster.

Cluster  $C_c$  is formed by selecting the closest faces to  $\mu_c$  from  $\Gamma$  (the remaining original face features) and hence the closest cluster of features to  $\gamma_c^{(1)}$  that is available in  $\Gamma$ . Cluster  $C_f$  is formed by selecting from  $\Gamma$  the closest faces to  $\mu_f$  and hence the furthest cluster of features to  $\gamma_c^{(1)}$  that is available in  $\Gamma$ . Cluster  $C_c$  may not be the closest cluster of face features to  $\gamma_c^{(1)}$  in the whole face set but it is the closest among the remaining original faces that have not been de-identified or added to a cluster. The same holds for cluster  $C_f$ . To apply the wrong-map to the de-identification results, the proposed  $k$ -Same-furthest method de-identifies the members in a cluster with the centroid of the other cluster. Since cluster  $C_c$  is identify-wise the closest cluster to  $\gamma_c^{(1)}$  and  $C_f$  the furthest, they are identity-wise the furthest away from each other, meaning the de-identified face images have a low risk to be associated with their original face images.

To avoid members of a cluster have a higher chance to be recognised as their original identity after data swapping, *overlapping* must be avoided between  $C_c$  and  $C_f$ . In the proposed method, two clusters *overlap* when

$$r_c + r_f \geq \|\mu_c - \mu_f\| \quad (4.3)$$

Overlapping between two clusters weakens the identity loss when data swap is applied to the cluster centroids. Furthermore, an original feature can even be the closest one to its de-identified feature after data swapping, when the centroid of the opposite cluster moves into the cluster sphere where the original feature is in.

Whenever a new member is added to a cluster,  $k$ -Same-furthest checks to see whether overlapping is caused by this new member. If so, this new member is moved back to  $\Gamma$  and the clustering loop for both  $C_c$  and  $C_f$  is terminated, as this new

member is the closest to the cluster and therefore adding any other remaining face to  $C_c$  or  $C_f$  would even shorten the distance between the two clusters. If clustering is stopped before  $C_c$  and  $C_f$  has been assigned  $k$  faces each, faces closest to the centroid of each cluster are selected from  $\Gamma$  and added to the cluster to fill up the gaps. However, the centroids of  $C_c$  and  $C_f$  are frozen during this process, i.e. the cluster centroids are calculated using those members before overlapping appears.

When  $|H|$  is not a multiple of  $2k$ , there will be fewer than  $2k$  faces remaining in  $\Gamma$  after  $\lfloor \frac{|H|}{2k} \rfloor$  iterations. The remaining faces will be associated to one of the last two clusters depends on their distances to the latest  $\mu_c$  and  $\mu_f$ . For example, the faces closer to  $\mu_c$  will be associated with  $C_c$  and then de-identified with the centroid of the opposite cluster  $\mu_f$ . Again, the values of  $\mu_c$  and  $\mu_f$  are not updated during this process.

**Theorem 1.** If  $H$  is a person-specific face set where  $|H| \geq 2k$ ,  $k$  is a privacy constraint and  $H_d = k\text{-Same-furthest}(H, k)$ ,  $k > 1$ , then  $H_d$  satisfies  $k$ -anonymity.

*Proof.* The proposed algorithm de-identifies original faces as the centroids of various face clusters. There are  $2 \cdot \lfloor \frac{|H|}{2k} \rfloor$  clusters formed in the clustering process. For each cluster centroid calculated, at least  $k$  copies of the same de-identified face images are added to  $H_d$ , making the  $k$  or more copies in  $H_d$  indistinguishable. In other words,  $H_d$  satisfies  $k$ -anonymity and always guarantees a recognition rate less than  $1/k$ . Furthermore, the de-identified face images in  $H_d$  has a one-to-one correspondence to an original face in  $H$ . Therefore,  $|H_d| = |H|$  and for each  $I \in H$  there exists  $I_d \in H_d$ .  $\square$

**Lemma 2.** Let  $\mu_n$  be the centroid of a cluster  $C_n$  of  $n$  face features and  $\mu_{n+1}$  the centroid of  $C_{n+1}$  which consists of  $C_n$  and an additional face feature  $x \in \Gamma$ .  $C_n \cap \Gamma = \emptyset$ . If  $x$  is the closest feature in set  $\Gamma$  to  $\mu_n$ , there cannot exist any other face feature in  $\Gamma$  that is closer to  $\mu_{n+1}$  than  $x$ .

*Proof.* Given  $x$  is the closest feature to the centroid  $\mu_n$  in  $\Gamma$ ,

$$x = \arg \min_{\gamma \in \Gamma} \|\gamma - \mu_n\| \quad (4.4)$$

The distances from  $\mu_n$  and  $\mu_{n+1}$  to  $x$  are

$$d_n = \min_{\gamma \in \Gamma} \|\gamma - \mu_n\| = \|x - \mu_n\| \quad (4.5)$$

$$d_{n+1} = \|x - \mu_{n+1}\| \quad (4.6)$$

The centroids  $\mu_n$  and  $\mu_{n+1}$  can be calculated by (4.1) and imply that

$$\mu_{n+1} = \frac{1}{n+1} (n\mu_n + x) \quad (4.7)$$

Therefore, (4.6) can be re-written as

$$\begin{aligned} d_{n+1} &= \left\| x - \frac{1}{n+1} (n\mu_n + x) \right\| \\ &= \frac{n}{n+1} \|x - \mu_n\| \\ &= \frac{n}{n+1} d_n < d_n = \min_{\gamma \in \Gamma} \|\gamma - \mu_n\|, \quad n \in \mathbb{N}^+ \end{aligned} \quad (4.8)$$

Moreover, the update of centroid  $\Delta\mu = \frac{1}{n+1} (x - \mu_n)$  have the same direction as  $x - \mu_n$ , which means the distance decrement of other face features  $\gamma \in \Gamma$  are at most the same as  $x$ .

Thus,  $x$  is closer to the updated centroid  $\mu_{n+1}$  than any other face features  $\gamma \in \Gamma$ , if  $x$  is the closest feature in set  $\Gamma$  to  $\mu_n$ .  $\square$

Note that there can be face features in  $C_n$  that are closer to  $\mu_{n+1}$  than  $x$ .

**Definition 4.1** (wrong-map). Given a person-specific face set  $H$ , a de-identification function  $f : H \rightarrow H_d$  provides wrong-map protection if

$$p(I^{(j)} | I_d^{(i)}) > p(I^{(i)} | I_d^{(i)}), \quad \forall I_d^{(i)} \in H_d.$$

**Theorem 3.** If  $H$  is a person-specific face set,  $k$  is a privacy constraint,  $k > 1$ ,  $|H| \geq 2k$ ,  $H_d = k\text{-Same-furthest}(H, k)$ ,  $k\text{-Same-furthest}(H, k)$  uses  $\text{dist}(\gamma_1, \gamma_2) = \|\gamma_1 - \gamma_2\|$  to measure the identity distance between any two faces  $\gamma_1$  and  $\gamma_2$ , then  $H_d$  satisfies wrong-map protection with the distance metric  $\text{dist}(\gamma_1, \gamma_2)$ .

*Proof.* Since  $C_{n+1} = C_n \cup \{x\}$  and **Lemma 2** states that no other face feature in  $\Gamma$  can be closer to  $\mu_{n+1}$  than  $x$ , the feature that is closest to  $\mu_{n+1}$  must be a member of  $C_{n+1}$  when clusters are formed in the way described in **Lemma 2**, i.e. the cluster adds the face feature that is closest to the current centroid. Furthermore,  $k\text{-Same-furthest}()$  forms two clusters simultaneously, meaning  $C_n$  in **Lemma 2** can be either  $C_c$  or  $C_f$ .  $k\text{-Same-furthest}()$  de-identifies the members of a cluster as the centroid of the opposite cluster. A member in  $C_f$  can only be the closest face to the centroid of  $C_c$  after swapping when there is no *overlapping* between  $C_f$  and  $C_c$ , and vice versa.

**Case 1** (no overlapping between two clusters)

Given a cluster  $C_c$  and its opposite cluster  $C_f$ . According to **Lemma 2**, there always exists a member in  $C_f$  closer to  $\mu_f$  than any member in  $C_c$ ,

$$\min_{\gamma \in C_f} \|\gamma - \mu_f\| < \min_{\gamma \in C_c} \|\gamma - \mu_f\|$$

Using the opposite centroid  $\mu_f$  to de-identify all the face features in  $C_c$  gives that

$$p(\gamma_f^{(j)} \mid \mu_f) > p(\gamma_c^{(i)} \mid \mu_f), \quad \exists j \in \{1, \dots, k\}, \forall i \in \{1, \dots, k\}.$$

The same holds for  $C_f$ .

**Case 2** (two clusters do not overlap while the size of each cluster is less than  $k$ )

Let the maximum number of members of both clusters before overlapping be  $k'$ , where  $1 \leq k' < k$ . Because the centroids are only calculated with the first  $k'$  members. The first  $k'$  members in both clusters satisfy **Case 1**, and the rest members in  $C_c$  are selected from  $\Gamma$  which cannot be the closest face feature to  $\mu_f$ , according to **Lemma 2**. Therefore,

$$p(\gamma_f^{(j)} \mid \mu_f) > p(\gamma_c^{(i)} \mid \mu_f), \quad \exists j \in \{1, \dots, k'\}, \forall i \in \{1, \dots, k\}.$$

The same holds for  $C_f$ .

**Case 3** (remaining features in  $\Gamma$  are less than  $2k$ )

This case is similar to **Case 2**. Because all the remaining face features in  $\Gamma$  can be neither the closest to  $\mu_c$  nor  $\mu_f$ . Assume that  $\mu_c$  and  $\mu_f$  of the latest formed clusters are calculated with  $k'$  members and the final size of the last cluster is  $m$ , where  $1 \leq k' \leq k$  and  $m \geq k$ . For  $C_c$ , it can be found that

$$p(\gamma_f^{(j)} \mid \mu_f) > p(\gamma_c^{(i)} \mid \mu_f), \quad \exists j \in \{1, \dots, k'\}, \forall i \in \{1, \dots, m\}.$$

The same holds for  $C_f$ .

In all cases, the de-identified faces will be matched with the original faces in the opposite cluster rather than their corresponding original faces, as long as the matching process uses the same distance measure as  $k$ -Same-furthest(). Thus, the described face de-identification method satisfies wrong-map protection.  $\square$

**Algorithm 3:**  $k$ -Same-furthest

---

**input** : Person-specific face set  $H$ ;  
 Privacy constant  $k$ , whit  $|H| \geq 2k$ ; Trained appearance models  $\mathcal{M}$ .  
**output**: De-identified face set  $H_d$

```

1  $H_d \leftarrow \emptyset$ ;
2  $\Gamma \leftarrow \mathcal{M}(H), \Gamma_d \leftarrow \emptyset$ ;
3 for  $\exists \gamma \in \Gamma$  do
4   if  $|\Gamma| \geq 2k$  then
5     Select  $\gamma_c^{(1)}$  randomly;
6      $\gamma_f^{(1)} \leftarrow \arg \max_{\gamma \in \Gamma} \text{dist}(\gamma_c^{(1)}, \gamma)$ ;
7     Add  $\gamma_c^{(1)}$  to  $C_c$ ; Add  $\gamma_f^{(1)}$  to  $C_f$ ; Remove  $\gamma_c^{(1)}$  and  $\gamma_f^{(1)}$  from  $\Gamma$ ;
8      $\mu_c \leftarrow \gamma_c^{(1)}$ ;  $\mu_f \leftarrow \gamma_f^{(1)}$ ;
9     while  $|C_c| < k$  and  $|C_f| < k$  do
10       $\gamma_f \leftarrow \arg \min_{\gamma \in \Gamma} \text{dist}(\mu_f, \gamma)$ ;
11       $\gamma_c \leftarrow \arg \min_{\gamma \in \Gamma} \text{dist}(\mu_c, \gamma)$ ;
12      Add  $\gamma_c$  to  $C_c$ ; Add  $\gamma_f$  to  $C_f$ ;
13      Remove  $\gamma_c$  and  $\gamma_f$  from  $\Gamma$ ;
14      Update centroids  $\mu_c \leftarrow \frac{1}{|C_c|} \sum_{i=1}^{|C_c|} \gamma_c^{(i)}$ ,  $\mu_f \leftarrow \frac{1}{|C_f|} \sum_{i=1}^{|C_f|} \gamma_f^{(i)}$ ;
15      if overlap between  $C_c$  and  $C_f$  then
16        Move the last added members of both  $C_c$  and  $C_f$  cluster to  $\Gamma$ ;
17        Update  $\mu_c$  and  $\mu_f$ ;
18        break; // Break from while loop
19      end
20    end
21    if  $|C_c| < k$  or  $|C_f| < k$  then
22       $\{\gamma_f^{(i)}\} \leftarrow \text{kNN}(\Gamma, \mu_f, k - |C_f|)$ ;
23      Add  $\{\gamma_f^{(i)}\}$  to  $C_f$ , Remove  $\{\gamma_f^{(i)}\}$  from  $\Gamma$ ;
24       $\{\gamma_c^{(i)}\} \leftarrow \text{kNN}(\Gamma, \mu_c, k - |C_c|)$ ;
25      Add  $\{\gamma_c^{(i)}\}$  to  $C_c$ , Remove  $\{\gamma_c^{(i)}\}$  from  $\Gamma$ ;
26    end
27    Add  $k$  copies of  $\mu_f$  to de-identify the faces in  $C_c$ ;  $C_c \leftarrow \emptyset$ ;
28    Add  $k$  copies of  $\mu_c$  to de-identify the faces in  $C_f$ ;  $C_f \leftarrow \emptyset$ ;
29  else
30    if  $\text{dist}(\gamma, \mu_f) > \text{dist}(\gamma, \mu_c)$  then
31      Add  $\mu_f$  to  $\Gamma_d$  to de-identify  $\gamma$ ;
32    else
33      Add  $\mu_c$  to  $\Gamma_d$  to de-identify  $\gamma$ ;
34    end
35  end
36 end
37  $H_d \leftarrow \mathcal{M}^{-1}(\Gamma_d)$ ;

```

---



Fig. 4.2 Example face images from the IMM dataset

## 4.3 Experiments

### 4.3.1 Dataset

Experiments were conducted with the IMM dataset [117], which contains face images of 40 subjects with different facial expression and head pose. Only the images with a near-frontal pose have been used, which includes a neutral, a happy and an arbitrary expression face images from each subject. As shown in Fig. 4.2, There is variation in head pose among the neutral as well as the happy faces, and there are variations in both pose and lighting among the arbitrary expression faces.

### 4.3.2 Test design

The re-identification risk of the proposed  $k$ -Same-furthest algorithm is measured through comparing the similarities of the face features in the Eigenface subspace and the appearance model subspace which is the same feature space used in  $k$ -Same-furthest face de-identification. The rank-1 recognition rate is used as the re-identification risk. The face images used in the experiments are aligned and cropped based on the AAM detected face landmarks. The images are masked and showing only the region inside the contour of the face shapes. The cropped original face images from 70% subjects in the IMM dataset have been used to train the PCA used in the re-identification experiment as well as the AAM in the  $k$ -Same-furthest method. In the test phase, all cropped original images with various expressions are used as the gallery and the de-identified images as the probes. All results reported are the average results based on ten runs with different training data from the IMM dataset.

### 4.3.3 Results and discussions

Fig. 4.3 shows the rank-1 re-identification risk for the cropped original faces against the faces de-identified using either  $k$ -Same-M or the  $k$ -Same-furthest. As shown

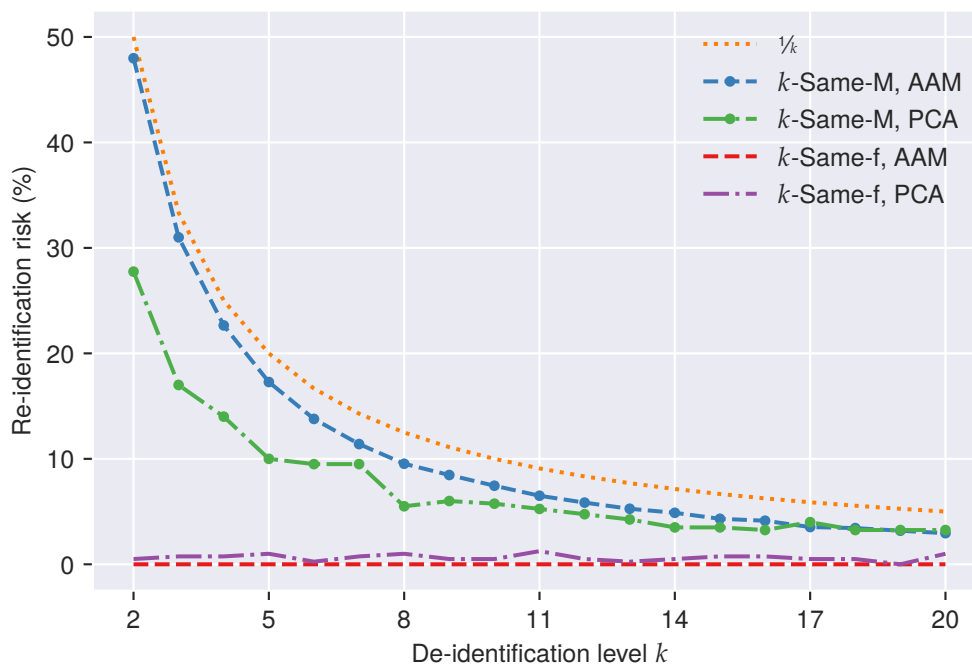


Fig. 4.3 Re-identification risks of de-identified faces

in Fig. 4.3, when the face de-identification and re-identification using the same feature space, the  $k$ -Same-furthest always produces a re-identification risk of zero while  $k$ -Same-M remain just lower than the theoretical upper bound  $1/k$  by around 2–3%. When PCA representation of face images is used in the re-identification, the re-identification risk of the  $k$ -Same-furthest is slightly above zero whilst stays far lower than the risk of  $k$ -Same-M. Regardless of which feature space the recognition software uses to represent face images, the re-identification risk of the  $k$ -Same-M de-identified faces reduces along with the theoretical upper bound  $1/k$ . It requires  $k$ -Same-M to use large  $k$  value in order to achieve decent privacy protection. However, with  $k$ -Same-furthest, the wrong-map protection keeps the re-identification risk staying at zero or slightly above zero even with a small  $k$  value.

Fig. 4.4 displays the de-identification results of the proposed  $k$ -Same-furthest algorithm with three different expression faces from the same individual. The facial expression is preserved by forming clusters only with the original faces with the same expression label as the target face. Fig. 4.5 displays the de-identification results of the  $k$ -Same-furthest algorithm where no preservation of data utility is implemented and hence the entire image gallery containing faces with various expressions and





Fig. 4.4 De-identification results of the proposed algorithm with data utility preservation.

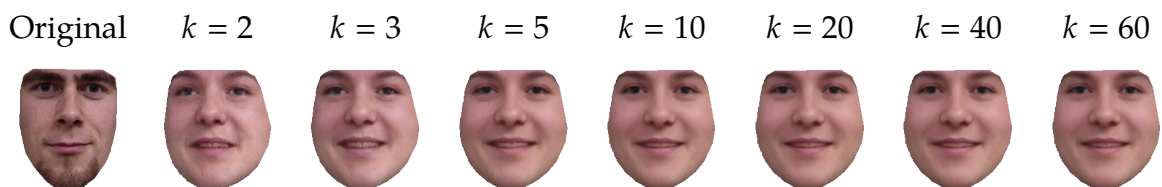


Fig. 4.5 De-identification results of the proposed algorithm without data utility preservation.

various head poses has been used to form the face clusters. The de-identified faces display various expressions and head poses. When  $k$  is small, the de-identified faces tend to display an expression and a head pose which have a larger difference to the target face image. As  $k$  increases, the cluster on which the de-identified face is based becomes more diverse. As a result, the de-identified faces tend to appear as generic average faces which have low variations on facial expressions and head poses.

As shown in Fig. 4.4 and 4.5, the de-identified faces for each original appear significantly different from their corresponding originals. In the meantime, Because the face appearance model is used in the proposed face de-identification method, the de-identified face images have high visual quality.

## 4.4 Conclusions

This chapter proposed a novel face de-identification method, *k*-Same-furthest, which introduces wrong-map protection to the *k*-Same face de-identification. In contrast to *k*-Same-closest algorithms, the proposed *k*-Same-furthest algorithm de-identifies faces based on the faces that are furthest away from them in a feature space that has been constructed to maximise the variation among identities and hence maximises the identity loss. It assumes that the desired data utility is not represented or has a low variance in the face feature space. Nevertheless, the *k*-Same-furthest face de-identification can be applied to a data utility biased subset to preserve the data utility, and the results are demonstrated in Fig. 4.4. In terms of reducing the re-identification risk, the proposed method inherits the advantages from both *k*-Same and wrong-map protection. For a simple re-identification attack, the face recognition software tends to recognise to a wrong identity regardless the value of *k*, because the de-identified faces have a large distance to their original faces in the face feature space. For more advanced re-identification attack, at least *k* original identities will share an identical de-identified face image, and this will restrict the re-identification risk by any attack to  $1/k$ .

# Chapter 5

## Distinguishable face de-identification

### 5.1 Introduction

The family of  $k$ -Same face de-identification methods de-identify a face dataset by dividing the dataset into clusters of at least  $k$  face images, then generating an aggregate face image for each cluster and finally replacing all images in a cluster with the same aggregated image. The advantage of making the de-identified dataset to have at least  $k$  identical copies of every aggregated face is that no face recognition software can build a one-to-one link with such data. On the other side, one drawback of these  $k$ -Same approaches is that the identical copies of face images in the dataset reduce the diversity of a dataset and the de-identified data cannot support certain tasks. For example, using the same de-identified face for multiple individuals make it impossible to track an individual in a de-identified video.

Consider a face dataset  $H$  and its de-identified dataset  $H_d$  that has been generated by a  $k$ -Same approach. The number of unique face images in  $H$  is  $|H|$  and in  $H_d$  is  $\lfloor \frac{|H|}{k} \rfloor$ . This information loss is due to a decrease in data diversity. Nevertheless,  $k$ -anonymity requires microaggregation in order to guarantee a re-identification risk lower than  $1/k$ .

To address the loss of diversity issue with the  $k$ -Same face de-identification methods, a novel face de-identification method named  $k$ -Diff-furthest is proposed and presented in this Chapter. Instead of using the same face to de-identify a group of faces, the  $k$ -Diff-furthest generate distinguishable de-identified faces and therefore is able to support a larger variety of applications, especially when tracking of individuals is required. The proposed  $k$ -Diff-furthest face de-identification is modified from the  $k$ -Same-furthest method. It has been discussed in Chapter 4 that

if the desired data utility has low variance in the face feature space, increasing the distance between the original face feature and the de-identified face feature can reduce the re-identification risk. Like the  $k$ -Same-furthest method, the  $k$ -Diff-furthest face de-identification method selects the two groups of faces that have the longest distance to each other to achieve a large identity loss. Unlike the  $k$ -Same-furthest method, each group of face features is not replaced by the centroid of its opposite group but shifted towards it in the feature space. This new approach generates a unique (different) de-identified face for each of the  $k$  original faces in a cluster. It is hence named  $k$ -Diff-furthest.

## 5.2 $k$ -Diff-furthest face de-identification

### 5.2.1 The proposed algorithm

The proposed  $k$ -Diff-furthest algorithm transforms the given person-specific face set  $H$  from the RGB pixel space to a pre-trained AAM feature space to ensure the alignment of face textures. The AAM representation of the original face set  $H$  is denoted as  $\Gamma$ , and its corresponding de-identified appearance model feature set  $\Gamma_d$ .

Similar to the process of the  $k$ -Same-furthest method,  $k$ -Diff-furthest is iterative and completes two tasks in each iteration. The first task is to form two clusters  $C_c$  and  $C_f$  in  $\Gamma$  with a given original face  $\gamma_c^{(1)}$ , where  $\gamma_c^{(1)}$  is the trigger of the current iteration and  $C_c$  is formed with faces closest to  $\gamma_c^{(1)}$  while  $C_f$  with those furthest from it. Once an original face is assigned to a cluster, it is removed from  $\Gamma$ . The second task of each iteration is to generate a de-identified face  $\gamma_d$  for each original in  $C_c$  and  $C_f$ .

In order to guarantee a privacy protection level better than  $1/k$ , all the  $k$ -Same-closest methods demand that each cluster formed in the de-identification process must contain at least  $k$  members. Furthermore,  $k$ -Same-furthest adds wrong-map protection to further reduce the re-identification risk, by preventing *overlapping* between the two clusters formed in each iteration.  $k$ -Diff-furthest follows the same approach and adopts wrong-map protection. The definition of *overlapping* and other variables such as cluster centroid  $\mu$  and cluster sphere radius  $r$  are the same as those used in the Chapter 4.

Whenever the clusters  $C_c$  and  $C_f$  receives a new member each,  $k$ -Diff-furthest checks whether *overlapping* is caused by the two new members. If so, both new members are removed from their clusters and the cluster formation for both  $C_c$  and

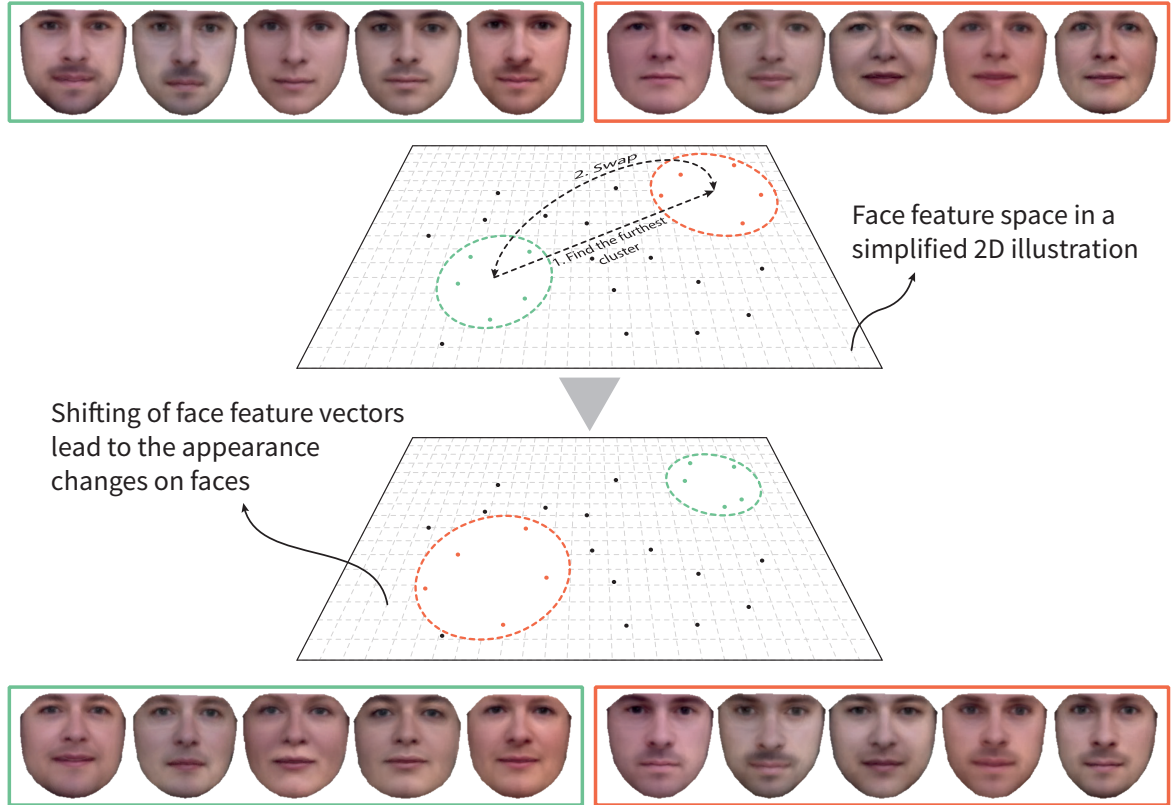


Fig. 5.1 The  $k$ -Diff-furthest face de-identification swaps original faces between a pair of clusters in order to retain the diversity of the original face set in the de-identified face set.

$C_f$  is terminated to avoid *overlapping* in the feature space between the two clusters. As a result, the size of the clusters formed in the  $k$ -Diff-furthest process might be less than  $k$ .

To provide the wrong-map protection,  $k$ -Same-furthest de-identifies the originals in  $C_c$  and  $C_f$  by swapping the cluster centroids. Whilst the same approach is adopted in  $k$ -Diff-furthest, the two methods differ in the way the de-identified faces are computed. The  $k$ -Same-furthest algorithm implements  $k$ -anonymity and uses the centroid of  $C_c$  as the de-identified face for all the faces in  $C_f$  and vice versa. In the  $k$ -Diff-furthest, for each pair of  $C_c$  and  $C_f$ , the de-identified feature vectors are computed by replacing the centroid with the centroid of the opposite cluster (see Fig. 5.1). More specifically,

$$\gamma_d = \begin{cases} \gamma - \mu_c + \mu_{f'} & \gamma \in C_c \\ \gamma - \mu_f + \mu_{c'} & \gamma \in C_f \end{cases} \quad (5.1)$$

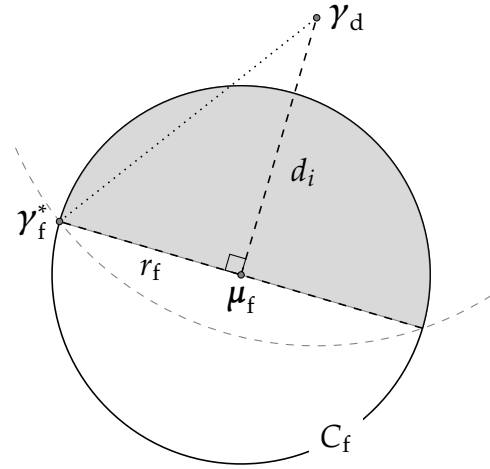


Fig. 5.2 Illustration of Theorem 4 in a 2D space

The de-identification step in  $k$ -Diff-furthest is equivalent to moving original faces in  $C_c$  to their new centroid  $\mu_f$  with their relative locations to the centroid unchanged, i.e.

$$\gamma - \mu_c = \gamma_d - \mu_f \quad (5.2)$$

The same applies to the face features in  $C_f$ .

Through this approach,  $k$ -Diff-furthest generates a unique de-identified face for each original face in  $H$  and retains the diversity of  $H$  in  $H_d$ . As it is assumed that original faces in  $H$  are distinguishable,  $k$ -Diff-furthest ensures that the de-identified faces in  $H_d$  have similar distances among each other as their original faces and are therefore equally distinguishable.

### 5.2.2 Wrong-map protection in $k$ -Diff-furthest

**Theorem 4.** Given a privacy constraint  $k > 1$ ; a person-specific face set  $H$  with  $|H| \geq 2k$ ; and a face set  $H_d = k\text{-Diff-furthest}(H, k)$ .  $k\text{-Diff-furthest}()$  uses  $\text{dist}(\gamma_1, \gamma_2) = \|\gamma_1 - \gamma_2\|$  to measure the identity distance between any two faces  $\gamma_1$  and  $\gamma_2$ , the  $k\text{-Diff-furthest}()$  can provide wrong-map protection as long as the de-identification and the re-identification are using the same distance measure.

*Proof.* As long as the centroid  $\mu_f$  is measured in Euclidean space, as shown in Fig. 5.2, the shaded half of  $C_f$  must contain at least one original face  $\gamma_f$ . Within the shaded

**Algorithm 4:**  $k$ -Diff-furthest

---

**input** : Person-specific face set  $H$ , cluster limiter  $k$   
**output**: De-identified face set  $H_d$

- 1  $H_d \leftarrow \emptyset$ ;
- 2  $\Gamma \leftarrow \mathcal{M}(H), \Gamma_d \leftarrow \emptyset$ ;
- 3 **for**  $\exists \gamma \in \Gamma$  **do**
- 4 Select  $\gamma_c^{(1)}$  randomly;
- 5  $\gamma_f^{(1)} \leftarrow \arg \max_{\gamma \in \Gamma} \text{dist}(\gamma_c^{(1)}, \gamma)$ ;
- 6  $C_c \leftarrow \{\gamma_c^{(1)}\}; C_f \leftarrow \{\gamma_f^{(1)}\}$ ;
- 7  $\mu_c \leftarrow \gamma_c^{(1)}; \mu_f \leftarrow \gamma_f^{(1)}$ ;
- 8  $\Gamma \leftarrow \Gamma \setminus \{\gamma_c^{(1)}, \gamma_f^{(1)}\}$ ;
- 9 **while**  $|C_c| < k$  **do**
- 10  $\gamma_f \leftarrow \arg \min_{\gamma \in \Gamma} \text{dist}(\mu_f, \gamma)$ ;
- 11  $C_f \leftarrow C_f \cup \{\gamma_f\}$ ;
- 12  $\gamma_c \leftarrow \arg \min_{\gamma \in \Gamma} \text{dist}(\mu_c, \gamma)$ ;
- 13  $C_c \leftarrow C_c \cup \{\gamma_c\}$ ;
- 14 Update  $\mu_c, \mu_f, r_c, r_f$ ;
- 15 **if**  $\text{dist}(\mu_c, \mu_f) < r_c + r_f$  **or**  $\gamma_c = \gamma_f$  **then**
- 16  $C_f \leftarrow C_f \setminus \{\gamma_f\}; C_c \leftarrow C_c \setminus \{\gamma_c\}$ ;
- 17 Update  $\mu_f$  and  $\mu_c$ ;
- 18 Break; // Break from while loop
- 19 **end**
- 20  $\Gamma \leftarrow \Gamma \setminus \{\gamma_c, \gamma_f\}$ ;
- 21 **end**
- 22  $\Delta\gamma \leftarrow \mu_c - \mu_f$ ;
- 23 **for**  $\gamma_c \in C_c$  **do**
- 24  $\gamma_d \leftarrow \gamma_c - \Delta\gamma$ ;
- 25 Add  $\gamma_d$  to  $\Gamma_d$  to de-identify  $\gamma_c$ ;
- 26 **end**
- 27 **for**  $\gamma_f \in C_f$  **do**
- 28  $\gamma_d \leftarrow \gamma_f + \Delta\gamma$ ;
- 29 Add  $\gamma_d$  to  $\Gamma_d$  to de-identify  $\gamma_f$ ;
- 30 **end**
- 31 **end**
- 32  $H_d \leftarrow \mathcal{M}^{-1}(\Gamma_d)$ ;

---

half of  $C_f$ , the furthest position to  $\gamma_d$  is at  $\gamma_f^*$ , meaning that

$$\sup \left\{ \min_{\gamma_f \in C_f} \|\gamma_d - \gamma_f\| \right\} \leq \|\gamma_d - \gamma_f^*\| \quad (5.3)$$

According to the Triangle Inequality Theorem,

$$\|\gamma_d - \gamma_f^*\| < r_f + \|\gamma_d - \mu_f\| \quad (5.4)$$

where, as defined in the algorithm of  $k$ -Diff-furthest( $H, k$ ),

$$\|\gamma_d - \mu_f\| = \|\gamma - \mu_c\| \leq r_c. \quad (5.5)$$

Therefore,

$$\|\gamma_d - \gamma_f^*\| < r_f + r_c \quad (5.6)$$

Because overlapping between two spheres is avoided in  $k$ -Diff-furthest(),

$$r_c + r_f \leq \|\mu_c - \mu_f\| \quad (5.7)$$

Combining (5.3), (5.6) and (5.7) gives

$$\min_{\gamma_f \in C_f} \|\gamma_d - \gamma_f\| < \|\mu_c - \mu_f\| \quad (5.8)$$

According to (5.1),

$$\|\gamma - \gamma_d\| = \|\mu_c - \mu_f\|. \quad (5.9)$$

Therefore, for any face  $\gamma_c \in C_c$  and its de-identified feature vector  $\gamma_d$ , there is

$$p(\gamma_f | \gamma_d) > p(\gamma_c | \gamma_d), \quad \exists \gamma_f \in C_f \quad (5.10)$$

The same holds  $\forall \gamma_f \in C_f$ . This means that a  $k$ -Diff-furthest de-identified face has a higher similarity with at least one member in the opposite cluster than its original. This method provides wrong-map protection, as long as the face recognition software of the re-identification attack uses the same face representation and distance measurement as the de-identification process.  $\square$



## 5.3 Discussion on single-member clusters

Single-member clusters will appear in two situations: (a) when there are only one or two left in  $\Gamma$ ; and (b) clustering is terminated due to a violation of the non-overlapping condition defined in (5.7). Although swapping two single-member clusters can provide wrong-map protection, it does not change the member values. There is a high risk when some original data have identical copies in the de-identified dataset. This section gives two potential solutions to the pair of single-member clusters problem and they can be simple add-ons to the  $k$ -Diff-furthest algorithm.

---

### Algorithm 5: Avoid single-member cluster pair

---

```

input : Two new formed clusters  $C_c$  and  $C_f$ ;
        A set of remaining face features  $\Gamma$ .

1 if  $|C_c| = 1$  and  $|C_f| = 1$  then
2    $\gamma_c \leftarrow \arg \min_{\gamma \in \Gamma} \text{dist}(\mu_c, \gamma)$ ;
3    $C_c \leftarrow C_c \cup \{\gamma_c\}$ ;
4    $\Gamma \leftarrow \Gamma \setminus \{\gamma_c\}$ ;
5   Update  $\mu_c, r_c$ ;
6 end
7 if  $|\Gamma| \leq 2$  then
8   for  $\gamma \in \Gamma$  do
9     if  $\text{dist}(\gamma, \mu_c) < \text{dist}(\gamma, \mu_f)$  then
10       $C_c \leftarrow C_c \cup \{\gamma\}$ ;
11    else
12       $C_f \leftarrow C_f \cup \{\gamma\}$ ;
13    end
14  end
15   $\Gamma \leftarrow \emptyset$ ;
16 end

```

---

### 5.3.1 Avoid the pair of single-member clusters

One potential solution is to include at least one new member into one of the last formed clusters to break the pair of single-member clusters situation. Each time when the non-overlapping condition terminates the cluster forming process, an additional checking process as shown in Algorithm 5 will take place. This process will check whether the two new clusters are single-member clusters and add an

extra member to one cluster if so. If less than three members are remaining in  $\Gamma$ , this process will force add the remaining members to their nearest clusters.

It is worth mentioning that the updated  $C_c$  and  $C_f$  may not comply with the non-overlapping condition in (5.7). When two clusters do not satisfy the condition in (5.7), the de-identified feature vectors may not guarantee the wrong-map protection and it will lead a correct matching to their originals. This situation occurs to the last two original face features in  $\Gamma$  when they cannot join  $C_c$  or  $C_f$  without breaking condition in (5.7).

### 5.3.2 Generate random feature vectors

The second solution does not add a new member to the single-member clusters but generates a random face sample within the neighbourhood of each single member. However, generating a face sample is a complex task with the following difficulties:

- 1) The feature manifold is usually non-linear and non-continuous.
- 2) The small number of existing samples gives a large sample error to estimate the distribution in a local neighbourhood.

The random sample generation method discussed in this section assumes that the face feature manifold is linear and continuous in a small local neighbourhood. The randomly generated face sample is constrained within the sphere with the given face as the centroid.

For an arbitrary  $n$  dimensional vector  $x \in \mathbb{R}^n$  in Euclidean space,  $\frac{x}{\|x\|}$  scales  $x$  onto the surface of the unit sphere. Random sample a feature vector  $\gamma^*$  around given centroid  $\mu$  within a radius  $r$

$$\gamma^* = r \cdot u \cdot \frac{x}{\|x\|} + \mu \quad (5.11)$$

where  $u$  is a random number generated uniformly from  $(0, 1]$ . Equation (5.11) generates random samples whose distance to the centroid is uniform distributed, but the vectors are not distributed uniformly in the sphere. To random sample a vector within the  $n$ -dimensional sphere uniformly, the radius should follow the  $u^{\frac{1}{n}}$  distribution, which gives

$$\gamma^* = r \cdot u^{1/n} \cdot \frac{x}{\|x\|} + \mu \quad (5.12)$$

Both (5.11) and (5.12) can be adopted for the random face feature generation. The distances of the vectors generated by (5.11) to the centroid have a uniform distribution. It means the generated vector may have a high chance to be close to the centroid. This can be solved by increasing the lower bound of  $u$ . The face samples generated by (5.12) tend to have large distances to the centroid when the dimension  $n$  is high. However, when the distance gets larger, it may break the assumption and generate feature vectors at non-continuous neighbours. As a result, unnatural face appearance will be reconstructed from these feature vectors. The value for radius  $r$  has been selected through trial and error. This work uses  $r = \frac{1}{4} \|\mu_c - \mu_f\|$ .

## 5.4 Experiments

### 5.4.1 Dataset

Experiments in this work were conducted with face images from the IMM [117] and the LFPW datasets. The 68-point facial landmarks as defined in Fig. 3.2 are annotated manually for the IMM dataset, and the landmark annotations for the LFPW dataset are provided by the 300 Faces In-the-Wild (300-W) Challenge [63] (see Fig. 5.3). The LFPW dataset contains faces with uncontrolled head pose, facial expressions and illumination, whereas the IMM dataset contains face images of 40 individuals captured in a controlled environment with six images per individual (frontal neutral, frontal happy, left rotated neutral, right rotated neutral and freestyle). A subset of 783 24-bit colour face images from the LFPW dataset and all the 40 frontal neutral face images in the IMM dataset were used to train/construct the AAM feature space in this work.

The face appearance which is the cropped face images showing only the region inside the contour of face shape was used in the experiments. The contour of a face is defined by facial landmarks (see Fig. 3.2). Each face appearance is represented as a 59-dimensional feature vector in a trained AAM feature space, with 9 face shape components and 50 face texture components. The face images in the LFPW dataset were captured under various lighting conditions. As shown in Fig. 5.3, the three LFPW images vary in overall image brightness, and two of them are significantly brighter than those from the IMM dataset. As mentioned, this work trains an AAM feature space with the 783 LFPW face images plus the 40 IMM face images. Due to the large variation in overall image brightness across the two datasets, the trained AAM feature space has the most dominant component of its face texture feature

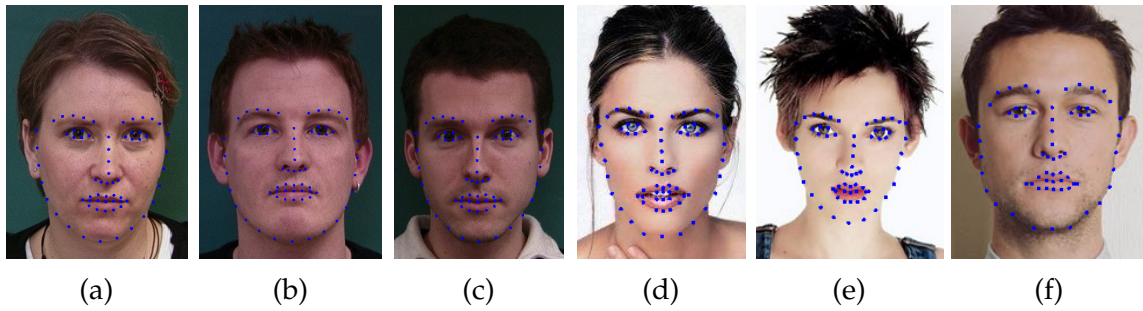


Fig. 5.3 Examples of face images in the testing set of re-identification test. (a–c) are from the IMM dataset [117] and (d–f) from the LFPW dataset.

representing the overall image brightness. Considering that the overall brightness of a face image does not contain any identifying information whereas clustering of face images in the face de-identification process should be carried out based on the identity features of the faces, this work sets the first texture model component to zero for all the faces to be de-identified. This is equivalent to the brightness/histogram equalisation procedure that is typically applied as part of the image pre-processing process in face recognition.

There is a consensus that the neutral frontal faces can best represent identity information, and most  $k$ -Same face de-identification algorithms (including  $k$ -Same-Pixel/-Eigen [24],  $k$ -Same-M [30], and  $k$ -Same-furthest [66]) have only reported experiments with neutral frontal faces. Therefore, all images used in the experimental evaluation of the  $k$ -Diff-furthest algorithm are neutral frontal faces. The testing set of this work consists of the 37 colour images from the IMM dataset and another three randomly selected images from the LFPW dataset (see Fig. 5.3d–f). The reason for replacing the three greyscale face images in the IMM dataset is to remove the impact of the colour format on the visual quality of the de-identified faces.

#### 5.4.2 Re-identification risk of $k$ -Diff-furthest

The privacy protection ability of the proposed  $k$ -Diff-furthest algorithm is measured and compared against the performance of  $k$ -Same-M through re-identification experiments using the AAM face appearance features, where the 40 original face images from the testing set are de-identified, and their de-identified faces (probe) are then matched against all the original faces (gallery) in the testing set. Both de-identification and re-identification are conducted in AAM feature space using Euclidean distance. In the de-identification process, the original face that triggers

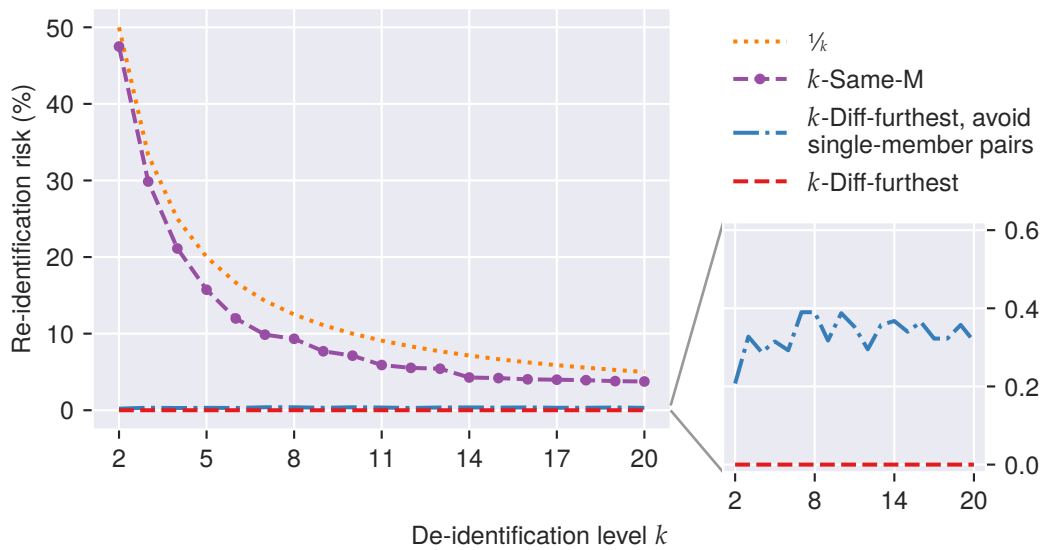


Fig. 5.4 Re-identification risks of  $k$ -Diff-furthest

each iteration is randomly selected. All results reported are based on running the identification process 1000 times for each value of  $k$ .

Fig. 5.4 shows the rank-1 re-identification risks of the de-identified faces against their original faces. In Fig. 5.4, as a  $k$ -Same solution, the re-identification risk of the  $k$ -Same-M always stays synchronized with and just below the theoretical upper bound  $1/k$ . The same experimental results of re-identification risk have been reported for all the other  $k$ -Same face de-identification methods in [24], [29], [30]. The re-identification risk of  $k$ -Diff-furthest is significantly lower than the risk of the  $k$ -Same-M. Fig. 5.4 confirms that when single-member cluster pair is allowed, the de-identified faces generated by  $k$ -Diff-furthest always yield a recognition rate of zero regardless of the value of  $k$ . When single-member cluster pair is not allowed the overlapping of clusters in the feature space may lead an increasement on the re-identification risk. However, as shown in Fig. 5.4, the increasement of the risk is lower than 0.5%.

### 5.4.3 Diversity of the de-identified face set

To measure the diversity of the resulting face dataset, the Euclidean distance between each image and every other image in the set is computed in the AAM feature space. The smaller the distance between two face feature vectors, the more difficult to distinguish the two faces in the images.

Table 5.1 Feature distances statistics

Face set	De-id method	Min	Max	Median	Mean	Std
Original	–	7.836	50.502	24.634	25.743	7.319
De-id	$k$ -Diff-furthest	12.764	50.101	26.073	26.799	6.554
	$k$ -Same-closest	0.000	18.467	13.558	12.641	2.169
	$k$ -Same-furthest	0.000	31.171	19.658	19.098	4.999

Fig. 5.5 shows the histogram distribution of pair-wise face feature distances among the original testing face images as well as their de-identified faces generated by  $k$ -Same-closest,  $k$ -Same-furthest and the proposed  $k$ -Diff-furthest when  $k = 5$ . There are 40 face images in the testing set, meaning each histogram in Fig. 5.5 shows the distribution of  $\binom{40}{2} = 780$  pair-wise face feature distances. Table 5.1 lists the statistics of all the results. The calculation of standard deviation for both  $k$ -Same-closest and  $k$ -Same-furthest has excluded the distance values of zero as this distance value is given by repetitions of the same de-identified face.

As shown in Fig. 5.5, the distance distributions of the original faces and the  $k$ -Diff-furthest de-identified faces have similar outlines, indicating that the diversity of faces in terms of their facial features is kept through the  $k$ -Diff-furthest face de-identification process and hence the  $k$ -Diff-furthest de-identified faces are equally distinguishable as their original faces among individuals. The same conclusion can be drawn with the results in Table 5.1, where the two sets of face images have similar mean, std and maximum distances. In contrast to those of the original and the  $k$ -Diff-furthest de-identified faces, the distance distributions for both the  $k$ -Same-closest and the  $k$ -Same-furthest faces are more discrete. This reflects the fact that  $k$ -Same methods de-identify a cluster of  $k$  original faces using the same de-identified face, reducing the number of unique faces in the dataset from  $|H|$  to  $\lfloor \frac{|H|}{k} \rfloor$ . This is also indicated by the spike at zero in both histograms. In addition, the de-identified faces generated by the  $k$ -Same methods are the centroids of clusters. The averaging effect of these de-identified faces has led to a smaller maximum distance and a narrower distribution diagram for each  $k$ -Same method. Furthermore, the results in Fig. 5.5 and Table 5.1 shows that the de-identified face features by  $k$ -Same-closest are more compact than those de-identified by  $k$ -Same-furthest. It is because the  $k$ -Same does not guarantee an optimal clustering solution and the formed clusters are heavily overlapped (see Section 2.3.1).

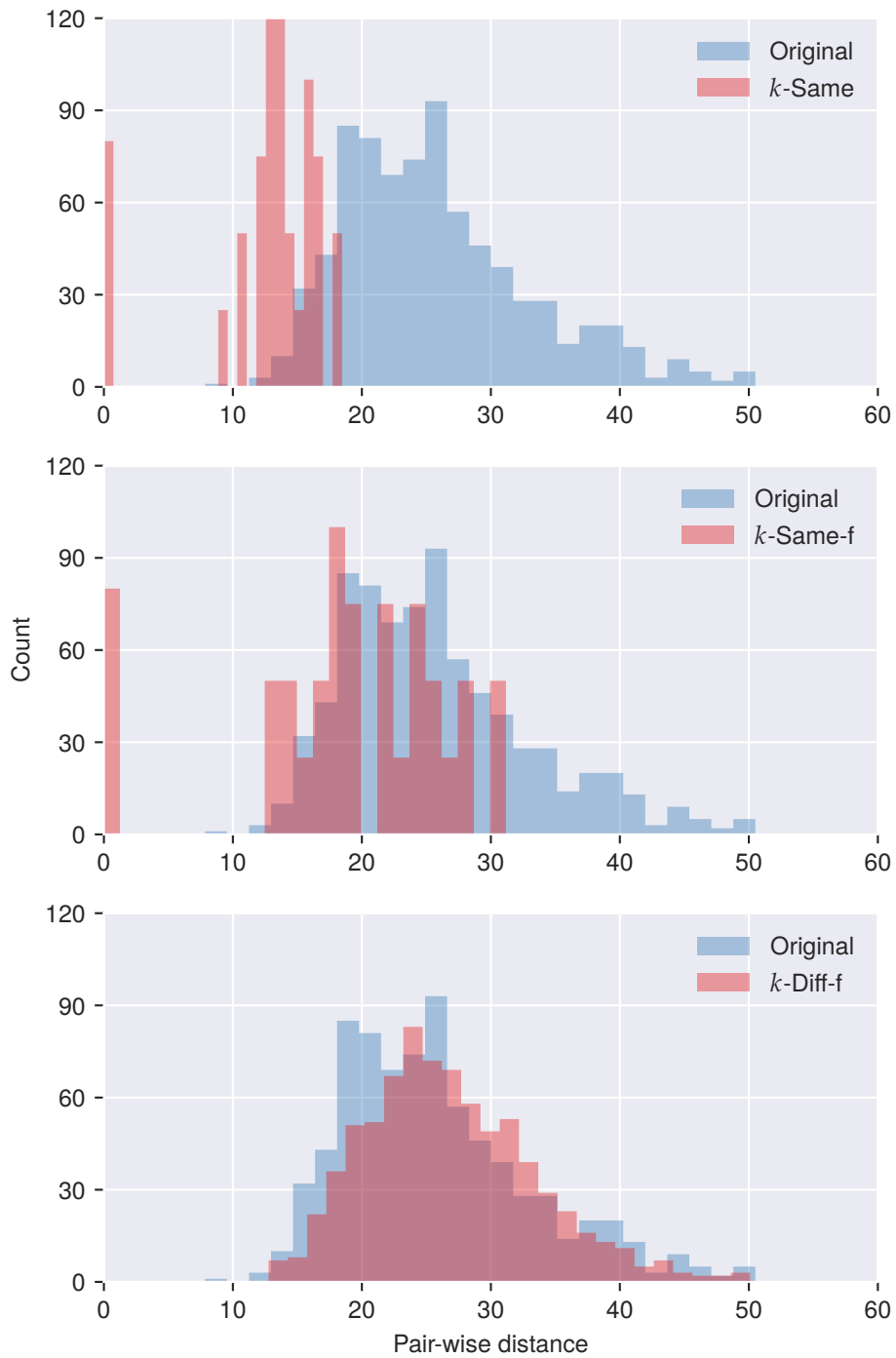


Fig. 5.5 Histogram of feature distances distribution of original faces and de-identified faces when  $k=5$ .

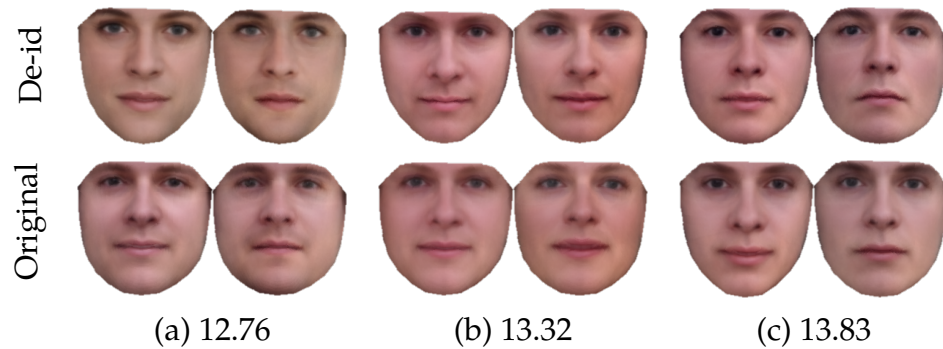


Fig. 5.6 Example faces showing relationships between the face feature distance and the visual difference. The computed face feature distances are given in the labels for each pair of example face images.

Fig. 5.6 illustrates the relationships between the computed face feature distances and the visual difference displayed in the face images. Because the group of face features get the same incremental updates, the information of the relative position in the group is kept. The differences between original faces can also be observed in their de-identified faces, which is highly useful for maintaining certain network graph information in a group, such as the kinship.

## 5.5 Conclusions

This chapter presents a new approach to generating distinguishable de-identified faces, named  $k$ -Diff-furthest. It maximises the loss of identity information by shifting the face feature to a position which is far from its original. The proposed  $k$ -Diff-furthest method provides wrong-map protection hence the re-identification risk of the de-identified dataset is low. Even when the avoiding single-member cluster pair condition breaks the wrong-map protection, the re-identification risk remains at a low level. In addition, the de-identified face dataset generated by  $k$ -Diff-furthest maintain more information from the original dataset. In contrast to the  $k$ -Same face de-identification methods,  $k$ -Diff-furthest maintains the diversity of the dataset and the relative position of a face feature in its neighbourhood.



# Chapter 6

## Visual quality and re-identification risk in real-world application

### 6.1 Introduction

The preservation of data utility and the reduction of re-identification risk conflict with each other in face de-identification. Depending on the applications, the de-identification methods choose different equilibrium points to compromise between utility and risk. In the real-world face de-identification applications, there are even more challenges from both data utility and re-identification risk aspects. The performance evaluation experiments presented in Chapters 4 and 5 are not complete for a real-world application. On the data utility fold, the visual quality of the de-identified face is an essential requirement in real-world applications. The visual quality can be measured by the fidelity and the intelligibility of the images [118]. The output images from previous chapters are isolated from their original background, which have relatively low fidelity and intelligibility to a human observer. On the re-identification risk fold, the de-identified face images are also facing a lot of challenges as the face recognition software is becoming more and more powerful. In Chapters 4 and 5, AAM features [58] and PCA features [89] are used as face descriptors and the similarities are measured with Euclidean distance. However, AAM features and PCA features are homogeneous subspace features, and they share similar advantages and drawbacks.

Facing the challenges from the real-world applications and aiming to increase the fidelity and intelligibility of the de-identified face images, this chapter presents an approach to merging a de-identified face region with its original background. In

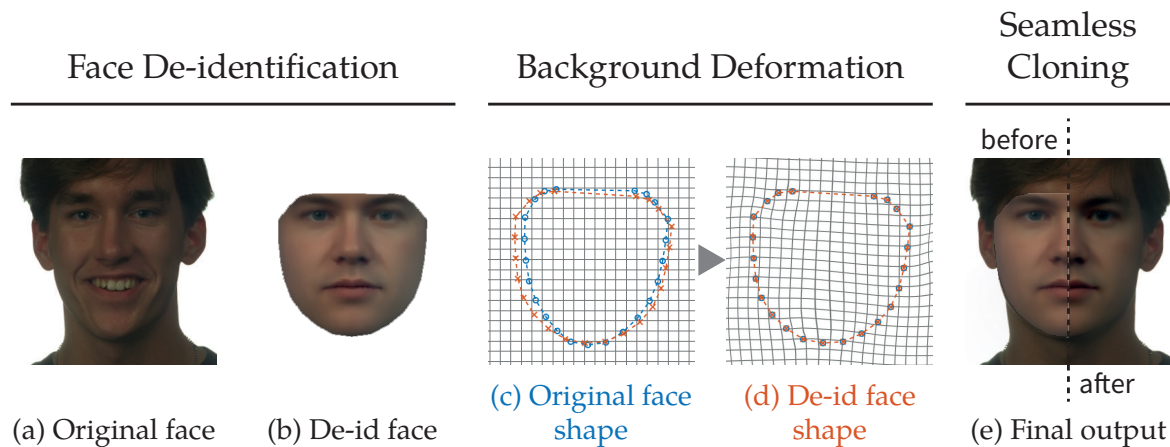


Fig. 6.1 Overview of merging a de-identified face with its original background

the experiment part, to evaluate the re-identification risk of the de-identified face images more thoroughly, several local image features such as LBP, HOG, and LPQ are used in the re-identification risk testing. This chapter presents the evaluation experiments conducted on face images with and without a background. The face dataset used in this chapter is larger than those used in the previous chapters. The experimental results reveal that further de-identification measures must be applied to the background of a face image as it presents a high re-identification risk.

## 6.2 Merging a de-identified face with its original background

To date, most of the published face de-identification methods have focused on the isolated face region in the original images [30], [43], [66], [119]. The result images presented in the publications are composed of a de-identified face region and a blank background (see Fig. 6.1b), which excludes hair, the ears, the forehead, the neck, the rest of the human body and the shooting environment. However, real-world applications always prefer a face with a background. As stated, the background here means the rest of the human body and the shooting environment that outside the face region. In the case of face de-identification, this leads to the demand of blending the de-identified face region back onto its original image background.

One of the main challenges in this task is given by the noticeable differences between the original and the de-identified faces regarding skin tone, illumination condition, etc. Previous research in the fields of face swapping and image editing

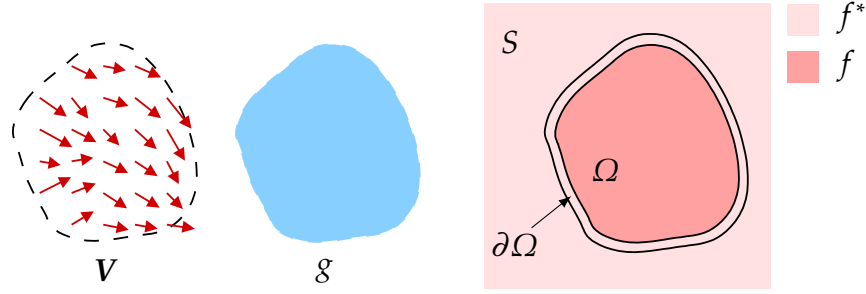


Fig. 6.2 Illustration for guided interpolation notations (adapted from [122])

has investigated similar problems and has provided several useful solutions to this challenge. The study on face swapping in [120] used one recolouring method followed by one relighting method to adjust the skin tone. Impett et al. [121] used histogram matching in the RGB space to allow real-time operation. As the real-time operation is not a priority at this stage, the more powerful but more time-consuming method of Poisson seamless cloning [122] has been used in this work to achieve a better visual quality of the blended images.

In the Poisson image editing [122], the source image can be interpolated to the destination with a guided vector field. Let  $S \in \mathbb{R}^2$  be the image definition domain and let  $\Omega$  be a closed subset of  $S$  with boundary  $\partial\Omega$ .  $g$  denotes the source image and  $V$  denotes a guidance vector field. The image blending task is to find an unknown function  $f$  that interpolates in domain  $\Omega$  the destination function  $f^*$ , under the guidance of vector field  $V$ . The unknown function  $f$  minimises

$$\min_f \iint_{\Omega} |\nabla f - V|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (6.1)$$

More specifically, in the task of margining the de-identified face to the background,  $g$  is the isolated face image and  $S$  is the target background. To achieve the seamless cloning, the guidance field  $V$  is a gradient field taken directly from  $g$  and (6.1) becomes

$$\min_f \iint_{\Omega} |\nabla f - \nabla g|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (6.2)$$

which can be solved with the following Poisson equation with Dirichlet boundary conditions:

$$\Delta f = \Delta g \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (6.3)$$

where  $\Delta \cdot = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  is the Laplace operator.

Fig. 6.3 shows the results of two example faces. Fig. 6.3c also illustrates an unexpected defect caused by the misalignment between the boundary of the source image and the boundary of the destination area  $\partial\Omega$ . The face de-identification process changes not only the texture of a face but also the shape. Consequently, the shape of the generated face might not fit with the original image background. This means a simple replacement would not generate satisfying results. One approach to this challenge could be warping the de-identified face texture to the original face shape, where the shape of the original face would be recovered in the de-identified image. However, the shape of a face contains rich personally identifiable information. Bringing back the original face shape would significantly degrade the privacy protection performance of a face de-identification system. Therefore, the new shape of the de-identified face must be maintained in the final de-identified image. This means that the background of the original face image has to be deformed to align with the new shape of the face region. As described in Chapter 2, a face texture is obtained through piece-wise affine warping, which triangulates the face image based on the facial landmarks (see Fig. 3.2). Nevertheless, there are not well-defined points in the background. In [47], the triangles of the background was defined with the landmarks of the facial contour and eight additional anchor points on the edges and corners of the image. This is an ad hoc approach because the transitions in between pieces are discontinuous. It becomes noticeable when the area of the background is increased.

The Moving Least Squares (MLS) image deformation method [123] is adopted in this work to deform the image backgrounds, which can avoid triangulating the input image and produce deformations that are globally smooth. MLS was used to warp the background by solving the best affine transformation  $l_v(x)$  that minimises, for each given point  $v$ ,

$$\sum_i w_i |l_v(p_i) - q_i|^2 \quad (6.4)$$

where  $\{p_i\}$  is a set of original points and  $\{q_i\}$  is the target deformed positions of  $\{p_i\}$ .

$$w_i = \frac{1}{|p_i - v|^{2\alpha}} \quad (6.5)$$

In the task of background deformation,  $\{p_i\}$  and  $\{q_i\}$  each is a set of facial contour landmarks as defined in [62], [63] (see also Fig. 3.2).  $\{p_i\}$  are the contour landmarks of the face in the original image, and  $\{q_i\}$  are the landmark positions of the de-identified face. Prior to background deformation, the landmarks of the de-identified face were

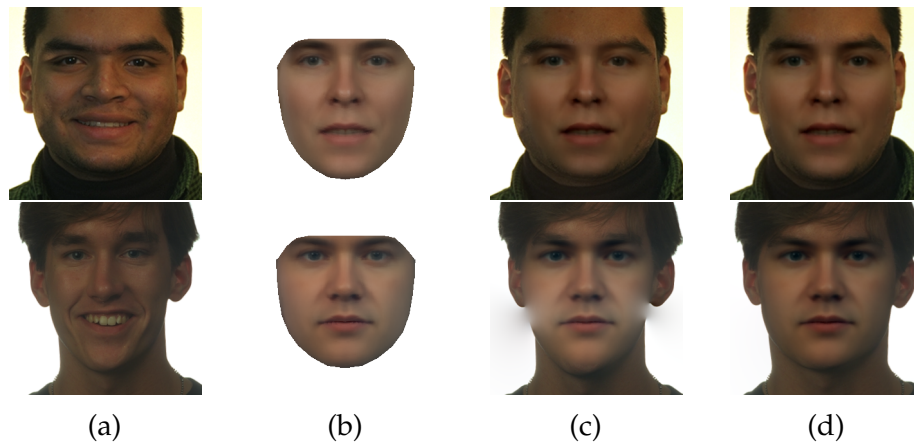


Fig. 6.3 Blending results of de-identified face regions. (a) Original face images. (b) De-identified face region. (c) De-identified face image without background deformation. (d) De-identified face image with background deformation.

firstly aligned to the original landmark through Procrustes analysis based on the positions of the inner corners of the eyes and tip of the nose.

The employment of background deformation has two contributions to a face de-identification system:

- **Better visual quality** of the de-identified face image. Poisson image blending removes the noticeable colour difference in between the de-identified face foreground and original image background. It increases the fidelity of the result images. Fig. 6.3 shows the blending results of two de-identified faces without and with background deformation. The last column shows the de-identified faces with background deformation, where each de-identified face has a shape different from its the original face while maintaining a similar skin tone and illumination condition as its original face. When the de-identified face region is narrower than its original face region (see the first row of Fig. 6.3), the visual quality of both with and without background deformation are acceptable in terms of the global colour tone and illumination condition. When the de-identified shape boundary is wider than the original face shape boundary, the guidance vector field is applied on the image background. As a result, the original background fades into the face region following the gradient of the de-identified face. As shown in the second row of Fig. 6.3, unexpected white spots appear on the sides of the face.

Table 6.1 Re-identification risk (%) of the de-identified face with and without background deformation

Background deformation	Crop size	
	300 × 300	200 × 200
w/o	71.24	10.80
w/	54.83	1.25

Table 6.2 Key parameters of the face recognition methods used in the evaluation experiments

Feature	Parameter values	Distance measurement
PCA	–	Euclidean distance
LBP	$radius = 1, neighbours = 8$	Chi-squared distance
HOG	$cell = 10 \times 10, orientations = 16$	Cosine distance
LPQ	$cell = 10 \times 10$	Cosine distance

- **Lower re-identification risk** of the de-identified face images. When the original face shape is retained, the de-identified image contains more PII information that can be matched with the original image. An experiment has been conducted to compare the re-identification risk of the resulting images with and without background deformation. The image LBP feature was used in this experiment. More details of the experiment and settings of parameters are described in Section 6.3. Experimental results are given in Table 6.1, which shows that the re-identification risks of images with background deformation are noticeably lower than those without background deformation.

### 6.3 Re-identification risk test

To evaluate the privacy protection performance of the proposed method thoroughly, further evaluation experiments have been conducted with various face representation models and distance measurements, including PCA (also known as Eigenface) [89], LBP [97], HOG [27] and LPQ [99] features. The classifier in this re-identification risk test is a  $k$ NN classifier which returns the rank-1 candidate with a given distance measurement. Some key parameters of the face recognition methods used are shown in Table 6.2.

### 6.3.1 Types of attacks

In a closed environment where only the original images and their corresponding de-identified images are presented, there are three types of re-identification attacks that can be used to test the protection performance of a de-identification method [24].

**Naïve recognition** The original face images are used as the gallery and de-identified face images as the probes.

**Reverse recognition** The de-identified face images are used as the gallery and the original face images as the probes.

**Parrot recognition** When the de-identification process can be imitated by the attacker and the set of de-identified face images generated by the attacker is used as probes to match with the released version of the de-identified face image set.

The following experiments evaluate the re-identification risks with naïve recognition and reverse recognition. The parrot recognition does not work on either  $k$ -Same-furthest or  $k$ -Diff-furthest method. As mentioned in Chapter 4 and 5, in each iteration of  $k$ -Same/Diff-furthest face de-identification two clusters are formed based on a randomly selected face. This means it is highly unlikely to repeat the same random selections and produce the same set of de-identified faces.

### 6.3.2 Evaluation setup

A subset of the FERET face dataset [34] containing 962 subjects has been used in the experiments. This subset was chosen from the available images of 994 subjects to ensure that each subject has two colour frontal face images ('fa' and 'fb'). 'fa' and 'fb' of each subject are from the same shooting session with slightly different facial expressions. All the 'fa' faces were used as the gallery in the re-identification tests to match against either the original version and the de-identified version of the 'fb' faces. Fig. 6.4 shows the original 'fa' (the first row), the original 'fb' (the second row) and the de-identified 'fb' (the last row) images for five subjects from the chosen FERET subset.

In this experiment, all the de-identified faces were generated with the same AAM, which was trained on another subset from the FERET dataset. The AAM training set was different from the gallery and probe set mentioned before. It contains 1952 colour images of frontal faces, and all of them are without glasses, beard or moustache. The exclusion of such features has enabled the automatic removal of

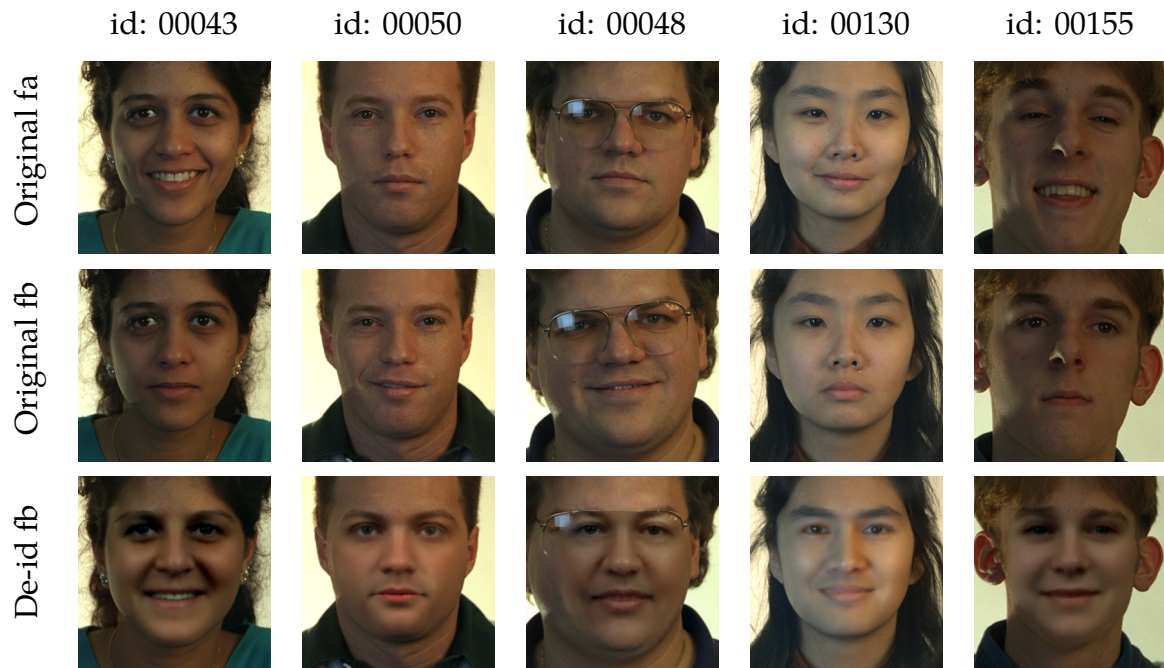


Fig. 6.4 Examples of faces (FERET dataset) used in re-identification tests

such non-identity related features through AAM representation (see Section 3.1.2 on page 26). As a result, no faded glasses frame, beard or moustache would appear on the resulting de-identified face images to degrade the visual quality of the image. In addition, it was observed that the most significant texture feature in the trained AAM describes merely the lighting condition of the images. To calibrate the lighting condition, the most significant variance in the texture model was set to zero for all the images represented in the face appearance model.

All face recognition systems isolate the face region from the background through cropping and then extract the face features. Different crop settings are there for face region cropping. Some systems crop the face region with a rectangular box while some systems define the face region with facial contour landmarks, e.g. the landmarks on the eyebrows and the jawline. In the re-identification tests, the face images were first aligned to the face mean shape whose size is approximately  $200 \times 200$  pixels and then cropped by a rectangular box which is concentric with the mean shape. The same crop setting was applied to both face images with background and without background. Two different cropping sizes of either  $200 \times 200$  or  $300 \times 300$  pixels were applied on the images. Therefore, there are four different cropped images for each face image in both the gallery and the probe set. Fig. 6.5 shows some example face images used in the following re-identification tests.



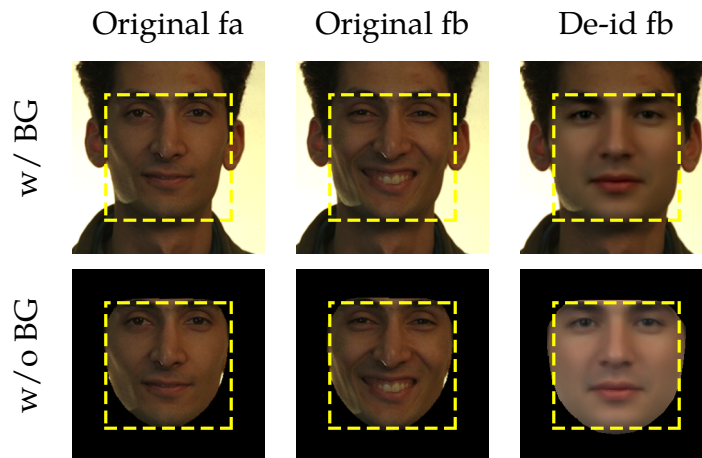


Fig. 6.5 Example of original and de-identified faces of one person with 200 pixels and 300 pixels cropping sizes

### 6.3.3 Results and discussion

Table 6.4 presents the re-identification risk of the  $k$ -Diff-furthest method measured in naïve recognition attacks using four different face recognition methods. Because of the randomness in the face de-identification method, all the re-identification risks in Table 6.4 are averages over 10 runs of  $k$ -Diff-furthest and are presented in the format of mean accuracy  $\pm$  standard deviation. Table 6.4a shows that without merging to the original background the de-identified face regions generated by the  $k$ -diff-furthest method always present a near zero re-identification risk for all the face recognition methods tested.

Table 6.4b shows the re-identification risk increases when the de-identified face region is blended with its original background. This is mainly because the Poisson blending process has brought low-frequency information such as the skin-tone and illumination of the original face image back to the de-identified face image. Furthermore, the more the original background in a cropped face image, the higher the re-identification risk will be. The risk values are under 7% and still acceptable when the face images were cropped with a  $200 \times 200$  square. However, the level of re-identification risk increases rapidly when the face images were cropped with a  $300 \times 300$  square, indicating that the background areas around the face region may also contain personally identifiable information and de-identification must also be applied to these image areas to achieve complete privacy protection. Further experiments have been conducted to evaluate the re-identification risk presented by the background area alone. Results of these experiments are presented and discussed in the next section.

Table 6.3 Re-identification risk (%) of  $300 \times 300$  inverse cropped face images

Feature	Probe set	
	Original	De-id
PCA	56.39	$31.82 \pm 0.89$
LBP	<b>78.19</b>	<b><math>55.62 \pm 0.68</math></b>
HOG	53.27	$27.07 \pm 1.09$
LPQ	60.44	$32.88 \pm 1.06$

Table 6.5 shows the results of reverse recognition attacks, which gives the same trends as the results of naïve recognition attacks. All the de-identified face images yield a near zero re-identification risk when the background is excluded from the image but a much higher risk when the background is included.

## 6.4 Image background attack to face de-identification system

The re-identification tests in Section 6.3 show that the background area of an original face image presents personal identifiable information and can increase the re-identification risk when being blended with the de-identified face region. The background area may contain not only the background environment of the original image but also the hairstyle, the ear, the neck and the dressing style presented in the original image.

The following experiment has been conducted to investigate the possibility of using merely the background area of the original face image to attack a face de-identification system. Fig. 6.6 shows some example images used in this experiment. Background attack is a general attack to any face de-identification method that modifies the face region only. The implementation of this attack is straightforward. All the face images were applied with an inverse crop based on their facial landmarks so that only the image area outside the face region was kept. The inverse crop has been used in face recognition to compare the recognition performance between human and computer [124], [125]. After inverse crop, the images were cropped into the size of  $300 \times 300$  for face recognition. The experimental results are shown in Table 6.3. Comparing with the results shown in Table 6.4, it is clear that the background area of the original image is the main contributor to the increase in the re-identification risk.

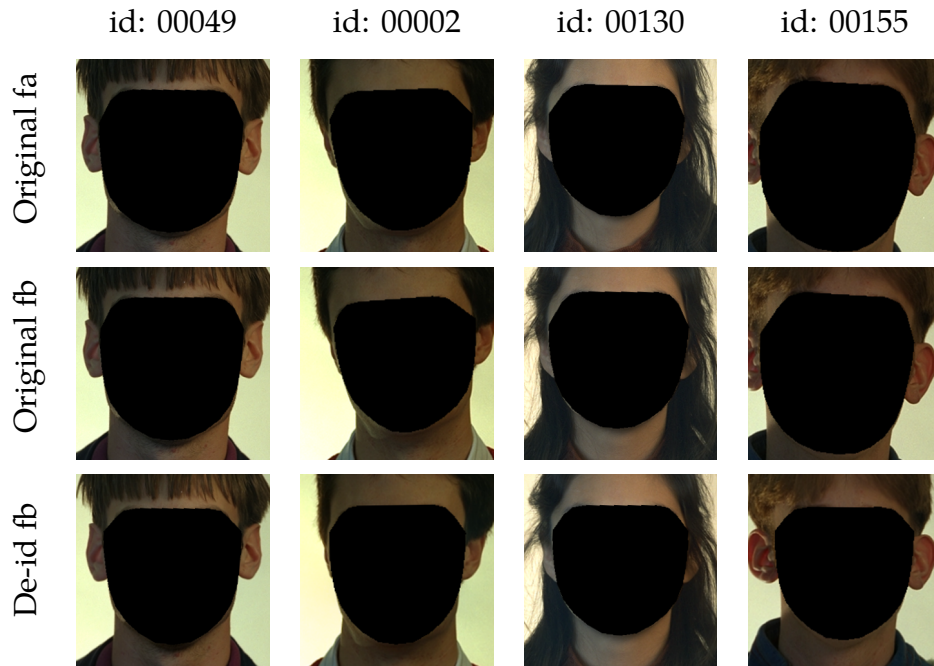


Fig. 6.6 Examples of inverse crop face images used in background attack experiment

## 6.5 Conclusions

To ensure the visual quality of a de-identified face image, the de-identified face region has to be merged with its original background. This chapter investigated the impact of this extract step on the re-identification risk.

Firstly, an approach was proposed to merge the isolated de-identified face with its original image background. This approach can increase the fidelity and intelligibility of the de-identified face images. The re-identification risk experiment results show that the  $k$ -Diff-furthest face de-identification method provides high privacy protection within the face region. However, blending the de-identified face to its original background increases the re-identification risk. Although a face recognition software focuses on the cropped face region, but the information contained in the background area around the face region (e.g. hair colour, hairstyle, and dressing style) also contribute to identify a person. The results of the re-identification attack using only the background information confirmed that face region is sufficient but not necessary for identifying a person. Face region information is the most efficient information to facilitate face recognition. However, only de-identify this information cannot stop the attacker from establishing re-identification using information such as the background.

Table 6.4 Naïve recognition rates (%) of original ‘fb’ faces and de-identified ‘fb’ faces against original ‘fa’

Feature	(a) w/o background				(b) w/ background			
	200 × 200		300 × 300		200 × 200		300 × 300	
	Original	De-id	Original	De-id	Original	De-id	Original	De-id
PCA	47.25	0.13 ± 0.16	42.37	0.07 ± 0.09	54.83	4.10 ± 0.58	61.27	39.13 ± 0.55
LBP	<b>74.25</b>	0.11 ± 0.10	<b>63.03</b>	0.13 ± 0.11	<b>83.39</b>	1.30 ± 0.24	<b>87.23</b>	55.12 ± 0.56
HOG	47.14	0.21 ± 0.14	18.38	0.17 ± 0.20	74.87	<b>6.09 ± 0.48</b>	78.09	56.93 ± 0.70
LPQ	53.27	<b>0.25 ± 0.18</b>	47.04	<b>0.25 ± 0.11</b>	80.48	4.42 ± 0.61	82.66	<b>59.14 ± 0.81</b>

Table 6.5 Reverse recognition rates (%) of original ‘fb’ faces and de-identified ‘fb’ faces against original ‘fa’

Feature	(a) w/o background				(b) w/ background			
	200 × 200		300 × 300		200 × 200		300 × 300	
	Original	De-id	Original	De-id	Original	De-id	Original	De-id
PCA	45.38	<b>0.33 ± 0.19</b>	41.53	<b>0.30 ± 0.21</b>	53.48	<b>9.41 ± 0.53</b>	61.37	49.06 ± 0.61
LBP	<b>72.59</b>	0.23 ± 0.14	<b>63.14</b>	0.21 ± 0.11	<b>80.48</b>	1.59 ± 0.41	<b>86.60</b>	59.28 ± 1.63
HOG	46.52	0.24 ± 0.16	17.03	0.07 ± 0.09	74.77	5.92 ± 0.70	76.95	<b>59.81 ± 0.68</b>
LPQ	50.88	0.23 ± 0.15	46.31	0.20 ± 0.14	80.06	4.35 ± 0.58	82.87	58.36 ± 0.48

# Chapter 7

## Open-set de-identification of faces in videos with preservation of expressions

### 7.1 Introduction

Chapter 4 and 5 have presented two novel face de-identification methods for the protection of privacy. In the meantime, there are various needs for preserving data utility of the de-identified faces in different real-world application scenarios such as the monitoring of patients' health status based on their facial expressions. This chapter proposes a unified approach to address three challenges from the real-world applications of face de-identification.

**The first challenge** is to maintain the facial expressions on the de-identified faces because the facial expressions are the most important non-identity related attributes on the face images and convey rich information. Face data are not only saved as still images but also video sequences. When dealing with videos, it is necessary to preserve not only the category but also the intensity of the facial expressions during de-identification to ensure a nature display of face dynamics afterwards.

**The second challenge** is an additional requirement from video data, where the consistency of identities in a video must be maintained after de-identification.

**The third challenge** is towards open-set face de-identification. The closed-set scenario performs face de-identification to a set of face images and uses the given image set as a gallery to train the feature space used by the de-identification method. In contrast, the original identities of the face images to be de-identified in an open-set scenario are not seen in the training phase [126]. Majority of the real-world applications are dealing with an open-set problem while the  $k$ -Same like face de-identification methods were designed for closed-set.

This chapter presents novel solutions to address all the three challenges mentioned above. In Section 7.2, a FET method is proposed to transfer the facial expression through face feature subspace. This FET method can be combined with previously described face de-identification methods and retain the facial motion and expression from the original face to the de-identified face with minor impact on the re-identification risk. In Section 7.3, an end-to-end face de-identification system for video sequences is proposed. To this end, the FET method is extended to video data to address additional identify consistency and open-set challenges. Finally, evaluations on data utilities and re-identification risks of the de-identified image and video data are presented in this chapter quantitatively and qualitatively. Section 7.4 evaluates the performance of FET in face de-identification with datasets of still images, and Section 7.5 tests the proposed end-to-end face de-identification system with video data to evaluate its performance in terms of Action Unit (AU) intensity preservation, identity consistency and privacy protection performance.

## 7.2 Facial expression transfer

Facial expression is one of the most important data utilities of face images which often convey emotions. Ekman proposed six basic emotions which are interpreted through facial expressions, namely anger, fear, disgust, happiness, sadness and surprise [127]. However, the facial expression is not equivalent to emotion, and facial expressions are not limited to six categories. In fact, the facial expression is dynamic and complex, and it is a combination of facial muscle actions. It is necessary to be accurate to the action level of each facial muscle to describe facial expressions. The FACS [109] is one of the most common tools used to describe facial muscle actions. The FACS defines atomic facial muscle actions as AUs. In FACS, facial expressions are described as a combination of AUs (see Fig. 7.1). In the study of facial expression analysis, AUs on human faces have been widely used to describe and analyse facial expressions [128]–[131].

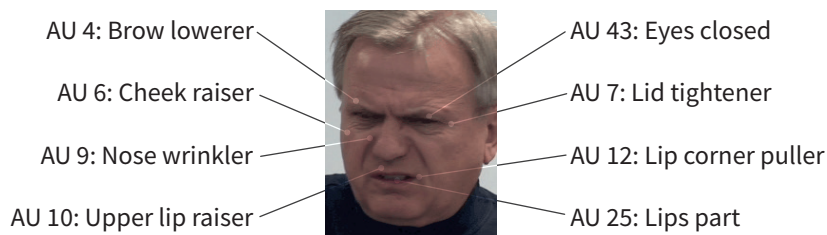


Fig. 7.1 An example showing the AUs associated with the expression of pain

Any procedure that blurs the facial expression in an image will degrade the data utility of the images for machines and human observers. Examples of such procedures include the ad hoc face de-identification methods (see Section 2.2) and the  $k$ -Same-Pixel/Eigen method [24] due to the ‘ghost artefact’.  $k$ -Same-Select [29] attempted to integrate utility preservation into face de-identification where data utility classifiers are adopted, e.g. a gender or an expression classifier. The utility classifier is used to split the original face set into several data utility biased subsets. The standard de-identification process is then performed on each biased subset separately. As a result, the de-identified images have the same utility bias as the input images so that the information of the desired data utility is preserved. The main drawback of this method is that it can preserve only the bias of the desired data utility but not the variance. For binary data utility such as gender, this method has been demonstrated to be adequate. For data utilities such as facial expressions, it is necessary to preserve not only the category but also the intensity, and  $k$ -Same-Select would fail in this circumstance. Furthermore, with the increasing number of data utilities, the number of subsets increase exponentially.

The facial expression preservation is a challenging task because it needs to not only guarantee the correct expression category but also retain the correct AU values. It is even more challenging with video footages since any discontinuous facial motions will look unnatural and can be easily spotted by a human observer.

### 7.2.1 Semantic analogies in the face feature space

As introduced in Section 3.1, the facial appearance model consists of a linear shape model and a linear texture model of the face appearance. More specifically, both shape and texture are PCA models and the principal components are orthogonal within each model. Each dimension in the model feature space reflects a certain variation in the model training set.

There is a path can be found in the feature manifold corresponds to certain change of the face appearance in the pixel space. When the face features of different identities move towards the same direction in the feature space, their facial appearances in the pixel space tend to have similar visual changes. This implies that the semantic analogy of “*king – man + woman = queen*” can be found in the feature spaces of the face appearance model.

Consider the transfer of facial expressions between two identities, where A is the source identity and B is the target identity.  $\gamma_A$  and  $\gamma_B$  are two reference faces of A and B which have the same expressions (e.g. neutral expression). Given the representation of an arbitrary face appearance of identity A in the feature space  $\gamma'_A$ , the facial expression of  $\gamma'_A$  can be mapped to  $\gamma_B$  through

$$\gamma'_B = \gamma'_A - \gamma_A + \gamma_B \quad (7.1)$$

Equation (7.1) can also be applied to transfer other facial attributes, e.g. gender, head pose, illumination condition. It is worth pointing out that, even for an attribute such as gender, Equation (7.1) will not only transfer the category of the attribute (i.e. male or female) but also the intensity of the attribute (i.e. how masculine/feminine a face is).

### 7.2.2 Face de-identification with facial expression preservation

To achieve the preservation of facial expression in a de-identified face image. The proposed FET process transfers the facial expression from an original face to its de-identified face using the semantic analogy defined in (7.1). In this case, the original face is the source and its de-identified face the target. This work always uses neutral frontal faces as the references since all face de-identification processes proposed in Chapter 4 and 5 deal with neutral frontal faces only. The FET process can be described as follows

$$\gamma'_d = \gamma' - \gamma + \gamma_d \quad (7.2)$$

where  $\gamma'$  is the feature vector of an input (original) expressive face image and  $\gamma$  is the feature vector of the neutral frontal face of the same identity.  $\gamma_d$  is the feature vector of the de-identified neutral frontal face generated by a model-based face de-identification method such as *k*-Same-M, *k*-Same-furthest and *k*-Diff-furthest.  $\gamma'_d$  is the output from the de-identification system which is a de-identified face with the same expression as  $\gamma'$ . Fig. 7.2 shows an example of *k*-Same-furthest face



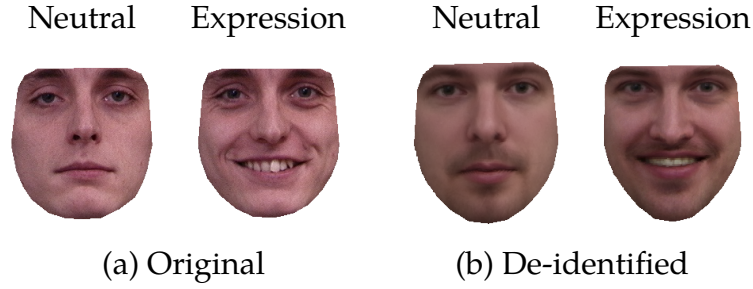


Fig. 7.2 An example of transfer the expression from the original face to its de-identified face

de-identification with FET. As demonstrated in Fig. 7.2, details of expression have been transferred without altering the new identity.

### 7.3 Face de-identification in videos

Comparing with still images, the video footages contains dynamic information about facial expressions and facial motions. This section proposes an approach that employs the FET process to preserve facial expressions in the de-identified videos.

Let  $\gamma'^{(i)} \in S$  be the face instances in a video sequence that are related to the same identity. When the FET process defined in (7.2) is applied to the individual face instances  $\gamma'^{(i)}$ , (7.2) can be re-written as

$$\gamma'_d{}^{(i)} = \gamma'^{(i)} + (\gamma_d - \gamma), \quad (7.3)$$

where the feature vector  $\gamma$  represents the neutral frontal face of the original identity defined by  $\gamma'^{(i)}$  and  $\gamma_d$  represents the de-identified version of  $\gamma$  that is also a neutral frontal face. The term  $(\gamma_d - \gamma)$  is referred to as the *identity shift* of the given original identity as it reflects the changes of the given identity's neutral face when it shifts from the original identity to a new identity.

As stated, all  $\gamma'^{(i)} \in S$  relate to the same identity and hence have the same  $\gamma$ . Obviously, for identity consistency in the de-identified videos all  $\gamma'^{(i)} \in S$  must have an identical new identity  $\gamma_d$ . This means that to de-identify face instances of a given identity, the face de-identification process only needs to calculate the *identity shift* of the source identity once and apply the same *identity shift* to all the face instances of the same source identity. In such way, face de-identification of the given identity in a video can be achieved through a simple addition of the input face instances and the calculated *identity shift*. In addition, the face de-identification

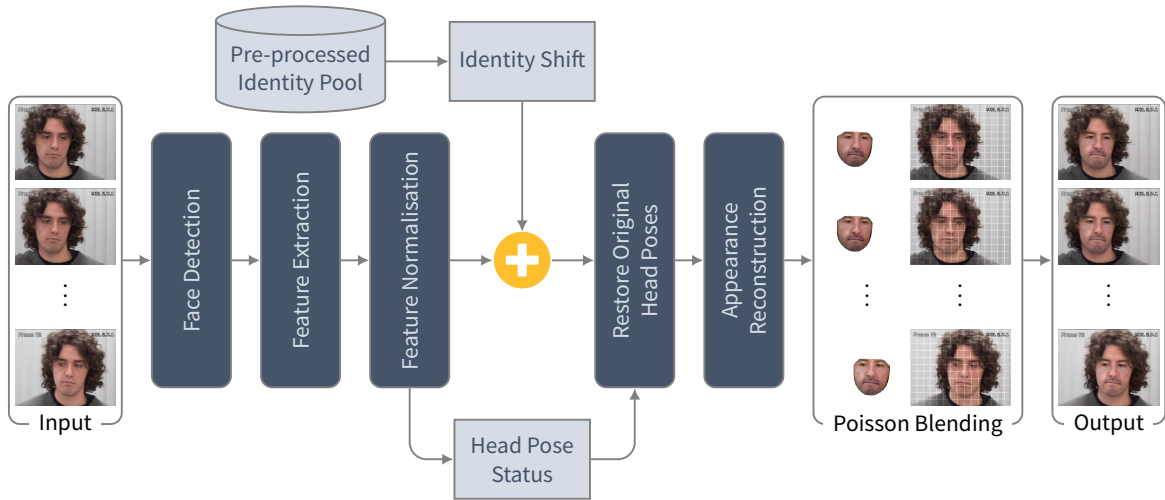


Fig. 7.3 Overview of face de-identification with facial motion and expression preservation in videos

approach for video defined in (7.3) is able to achieve both privacy protection and preservation of facial expressions simultaneously: (a) the *identity shift* calculated by an appropriate face de-identification method which guarantees the new identity has a low re-identification risk; and (b) only the neutral frontal faces are used as the identity reference which makes sure the FET process can transfer the facial expression from original face to the de-identified face. Furthermore, this approach inherently ensures identity consistency in the video after face de-identification as the same *identity shift* is applied to the same person in each video frame.

Fig. 7.3 shows an end-to-end face de-identification system with an example video input. Apart from the critical operations defined in (7.3), additional operations such as background deformation and blending which mentioned in Chapter 6 have been employed in the system to achieve better visual quality with the de-identified video. The rest of this section describes the functional blocks in Fig. 7.3.

### 7.3.1 Calculation of identity shift

The *identity shift* defines the difference between the original neutral face  $\gamma$  and its de-identified version  $\gamma_d$ . All the existing  $k$ -Same methods as well as the  $k$ -Diff-furthest method operate with a set of face images and is not viable with an individual input face image. Furthermore, the original neutral face  $\gamma$  is not always available in the gallery in real-world applications (open-set) and searching for a neutral frontal face in a video sequence is complicated and can be time-consuming. To resolve these two issues, an additional person-specific set of neutral frontal faces is introduced in

the proposed face de-identification system. The person-specific set contains only neutral frontal face images which is used to provide identity references for unseen face images, and it is named as the identity pool. For open-set applications where the original neutral frontal face  $\gamma$  may be unavailable, the *identity shift* is calculated using the following equation instead:

$$\Delta\gamma = \hat{\gamma}_d - \hat{\gamma}, \quad (7.4)$$

where  $\hat{\gamma}$  is the nearest face to  $\gamma$  found in the identity pool. The corresponding de-identified face set of the identity pool is generated offline using the  $k$ -Diff-furthest method in a single pass. Although the de-identified identity pool can be generated using any face de-identification method, the  $k$ -Diff-furthest has been used here to maintain the diversity among the de-identified neutral faces. The *identity shift*  $\Delta\gamma$  for each face (or identity) in the identity pool is also calculated offline to increase the process efficiency of the proposed system. To minimise the initial delay of the system in a time-critical application such as video streaming, the de-identification system uses the first available video frame of a person's face to establish a link with the identity pool.

### 7.3.2 Feature normalisation and retainment of the original head pose

The proposed approach uses a generic face appearance model to make sure it is capable of accurately representing any face instance in the input video. The generic face appearance model is trained with face images of various head poses, facial expressions and illumination conditions. As a result, in the trained model, the first two shape components represent the pitch and yaw rotations of a face; while the first three texture components represent the illumination conditions of a face image. Considering that illumination and head pose have a noticeable impact on the accuracy of face recognition, the illumination, pitch and yaw of  $\gamma'^{(1)}$  (the first instance of  $\gamma'^{(i)}$ ) is normalised before searching for its nearest face  $\hat{\gamma}$  in the identity pool. This normalisation is achieved by setting the parameters of the above-mentioned model feature components to zero.

The same normalisation has also been applied to the identity pool such that the *identity shift* generated using (7.4) will not alter illumination, pitch and yaw rotations of a face. Hence, through the calculation of (7.3), all these characteristics of an original face  $\gamma'^{(i)}$  will be automatically mapped to its de-identified version of  $\gamma'_d{}^{(i)}$ .

Two of the head motions are not mentioned in the feature normalisation, which are translation and roll. In this work, the translation and roll of  $\gamma^{(i)}$  are restored in  $\gamma_d^{(i)}$  through Procrustes analysis based on three facial landmarks, explicitly the inner corners of the eyes and the tip of the nose. The retainment of original illumination and head pose makes the whole video frame look much smoother and more natural when the de-identified face region is merged with the original image background.

### 7.3.3 Implementing texture transfer in pixel space

In previous sections, the FET process is executed in the model feature space. However, the PCA process for establishing the face appearance model is not lossless. As a result, the information which the face appearance model cannot represent will be lost when a face appearance is represented in the model space. For example, there are not eye gaze variations in the face appearance model training set in the following experiments, so the appearance model cannot represent eye gaze. The eye gaze information will lose when the FET process is conducted in the feature space (see Fig. 7.4b).

In the face appearance model, the shape and texture of a face can be represented and processed separately. Therefore, considering (3.11), the *identity shift*  $\Delta\gamma$  is able to be split into shape shift  $\Delta\alpha$  and texture shift  $\Delta\beta$ . According to (3.8a), the shape of the de-identified face appearance  $s'_d$  is given by

$$s'_d = \bar{s} + \Phi_s(\alpha' + \Delta\alpha) \quad (7.5)$$

Similarly, according to (3.9a), the de-identified face texture through FET in feature space is given by

$$t'_d = \bar{t} + \Phi_t(\beta' + \Delta\beta) \quad (7.6)$$

The texture information loss happens when using  $\beta'$  representing  $t'$ . If the FET is executed in pixel space, all the texture detail will be retained and (7.6) becomes

$$t'_d = t' + \Phi_t \cdot \Delta\beta \quad (7.7)$$

where  $\Phi_t \cdot \Delta\beta$  converts the feature *identity shift* into image pixels and only needs to be calculated once when de-identifying a video sequence.

Fig. 7.4 compares the results of the FET when conducting texture transfer in feature space and pixel space with three example video frames. As shown, the de-identified video frame generated through texture transfer in the pixel space



Fig. 7.4 Comparison of texture transfer in feature and pixel space. (a) the original video frame; (b) the de-identified video frame whose texture has been transferred in the model feature space; (c) the de-identified video frame whose texture has been transferred in raw pixel space. (b) and (c) are using the same de-identified face shape which is different from (a). (c) contains more texture details of (a), e.g. wrinkles and gaze.

looks more natural than the one generated in the feature space. Because the second approach does not filter the original face textures with a face appearance model, any details such as a mole or a scar on the face will be passed from the original image to its de-identified image and increase the re-identification risk. The re-identification risk of the de-identified images produced by these two different approaches is evaluated and compared in Section 7.5.

## 7.4 Evaluate face de-identification with FET on still images

The experiments in this section extend the re-identification risk evaluation in Chapter 4 and evaluates the proposed FET process with the IMM dataset [117]. The dataset contains still images of 40 subjects. Only images with a near-frontal head pose were used. These include a neutral, a happy and an arbitrary expression

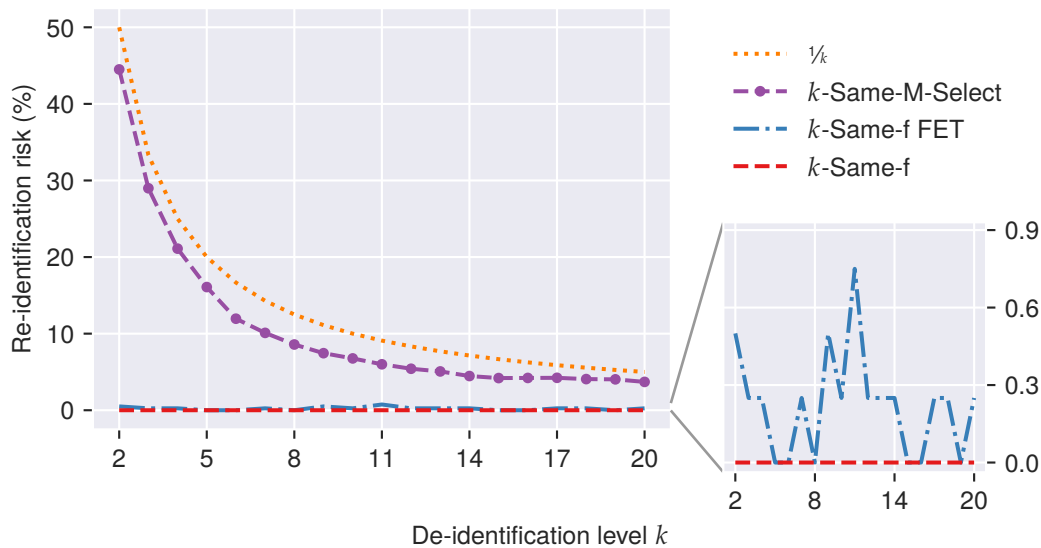


Fig. 7.5 Re-identification risk comparison of  $k$ -Same-M-Select and  $k$ -Same-furthest with FET

face images per subject. The following experiments aim to compare the facial expression preservation performance between  $k$ -Same-Select [29] and the proposed  $k$ -Same-furthest with FET. Because all the face de-identification methods in this experiment are face appearance model-based, the name  $k$ -Same-M-Select is used here to distinguish from the original  $k$ -Same-Select which is the Pixel/Eigen-based solution.

#### 7.4.1 Evaluation of re-identification risk

The re-identification risk was evaluated through face recognition experiments measuring Euclidean distance in the AAM feature space. In each run of the experiments, 70% of the subjects were randomly selected and the near-frontal face images of these selected subjects were cropped and then used to train an AAM face appearance model. In the test phase, the original near-frontal face images of all the subjects were cropped and used as the gallery and their de-identified images as the probes. All the results reported here are averages over ten runs of the experiment.

Fig. 7.5 shows the rank-1 re-identification risks of the cropped original faces against their de-identified faces generated by model-based  $k$ -Same-Select and by  $k$ -Same-furthest with and without FET. The re-identification risk of  $k$ -Same-M-Select stay slightly below the theoretical upper bound  $1/k$ . As proved and tested in

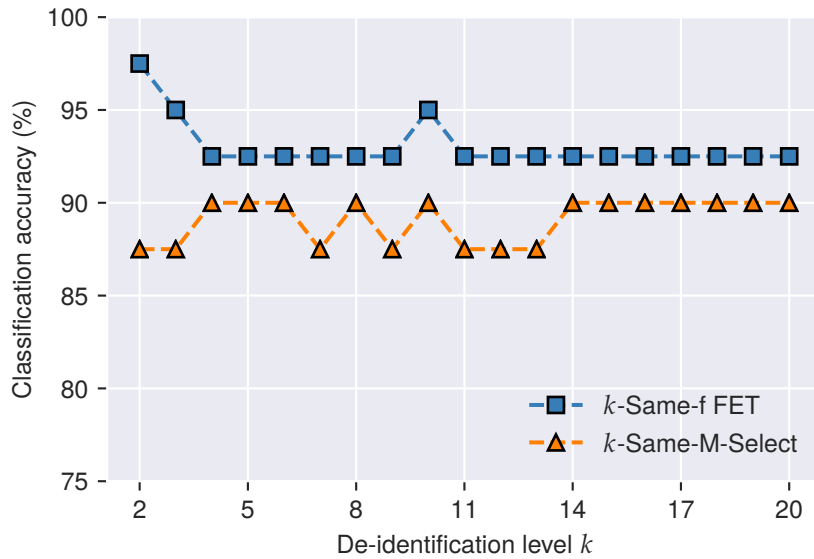


Fig. 7.6 Comparison of facial expression classification on the de-identified faces

Chapter 4,  $k$ -Same-furthest without the FET process produces a re-identification risk of zero in the appearance model feature space regardless of the value of  $k$ . The re-identification risk of the  $k$ -Same-furthest with FET is near zero regardless of  $k$ , indicating that the FET process after  $k$ -Same-furthest de-identification has hardly any impact on the privacy protection performance of the proposed system.

### 7.4.2 Evaluation of data utility

Following the evaluation of  $k$ -Same-Select in [29], the data utility preservation ability of the  $k$ -Same-furthest with FET method was measured in terms of the accuracy of a facial expression classifier. A LDA model was employed as the expression classifier. The LDA classifier was trained with original face images displaying neutral and happiness expressions. This classifier was then tested with the de-identified happy faces generated by model-based  $k$ -Same-Select and the  $k$ -Same-furthest with FET, respectively. Fig. 7.6 compares the classification accuracies of the two approaches. As shown, the  $k$ -Same-furthest with FET method can retain more expression information than the  $k$ -Same-Select method. In addition, the  $k$ -Same-Select does not distinguish the intensity of the facial expressions, while different levels of happiness may present on different faces. Synthesising the de-identified face as an aggregation, the expression information on different faces may cancel out each other and be lost from their de-identified faces. This leads an information loss on the facial

expressions. In contrast, the proposed approach transfers the original expression after the de-identification process, aiming to make the output face display an identical expression to the original. Hence when the output faces are classified with a classifier trained with the original expressions, it is more likely for the classifier to detect the expression cloned by  $k$ -Same-furthest-FET. Fig. 7.6 confirms that applying FET after  $k$ -Same-furthest face de-identification preserves expression category better than  $k$ -Same-Select.

## 7.5 Evaluate face de-identification with FET on video data

The evaluation experiments on face de-identification with FET was conducted with the video sequences from the UNBC-McMaster Shoulder Pain Expression Archive Database [132]. The UNBC-McMaster database contains 200 video sequences of the faces of 25 subjects, where faces in each video all relate to the same subject. 184 video sequences were selected from the database to make sure a complete face with the full set of 68 facial landmarks can be detected in each frame. There are 238 frames per video on average.

To enable accurate representation of the faces in the identity pool as well as those in the test video sequences, a generic appearance model was trained with 1952 near frontal faces from the FERET dataset. The appearance model training data present various facial expressions, illumination and small head poses and are without glasses or heavy facial hair. In the trained appearance model, 8 shape components are kept to represent 90% of the shape variance within the training set and 59 texture components for 90% of the texture variance. Furthermore, the identity pool was composed of 780 near neutral faces from the FERET dataset [34] that are also near frontal. Before applying face de-identification to video sequences, de-identification of the identity pool is carried out offline in a single pass. The evaluations on data utility, identity consistency and re-identification risks of the proposed face de-identification method in the video are presented in the following subsections.



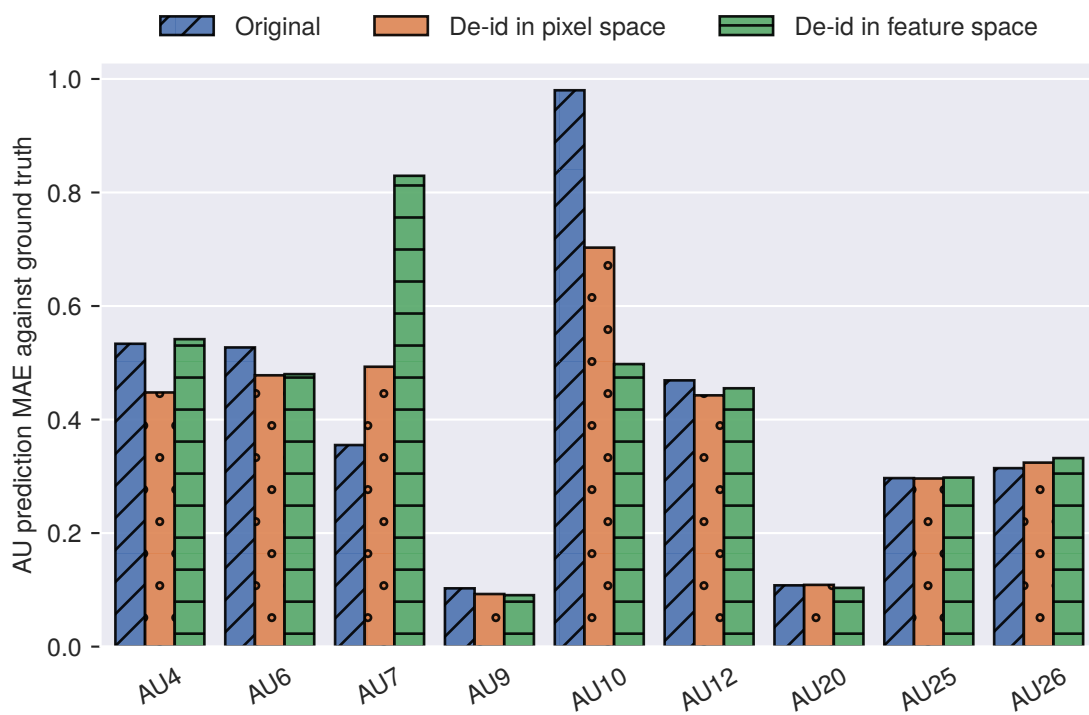


Fig. 7.7 The mean absolute errors of OpenFace AU detection results of the original frames and the de-identified frames, in comparison with the AU intensity ground truth provided by UNBC-McMaster dataset [132].

### 7.5.1 Preservation of facial expressions

The UNBC-McMaster dataset contains the ground truth of 9 AUs including brow-lowering (AU4), cheek-raising (AU6), eyelid tightening (AU7), nose wrinkling (AU9), upper-lip raising (AU10), oblique lip raising (AU12), horizontal lip stretch (AU20), lips parting (AU25) and jaw-dropping (AU26), where the intensity of each AU has been scored manually as an integer from 0 to 6 inclusively. To test the expression preservation performance of the proposed system, the intensity level of each AU is compared between the original video frames and their corresponding de-identified frames. The AU intensities of a video frame were measured by OpenFace [131], which generates AU intensity as real numbers.<sup>1</sup>











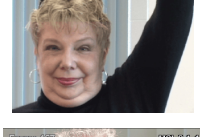
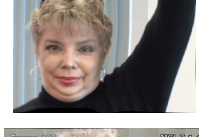






Fig. 7.7 shows the mean absolute difference between the AU intensities calculated by OpenFace and the ground truth for the original video frames as well as the de-identified frames generated in the pixel space and the model feature space, respectively. The difference values are averaged over all the 43 734 test video frames. Fig. 7.7 shows that, apart from AU7 eyelid tightener and AU10 upper lip raiser, the AU intensity values of the de-identified frames have remained almost the same as their original frames. For AU10, the AU intensity values predicted from the de-identified video frames are closer to the ground truth than their original video frames. Even in the worst case of AU7, the AU intensity has remained within the same integer level after face de-identification.

As shown by the blue bars in Fig. 7.7, OpenFace is not always accurate, and there are some AUs harder to predict than the others. Therefore, a further AU comparison is performed to test with only the frames that OpenFace can predict correctly, i.e. when the rounded-up value of OpenFace’s prediction matches with the ground truth. Table 7.1 shows the comparison results. Again, there is hardly any change in AU intensity for AU9 nose wrinkler and AU20 lip stretcher. The highest difference is still with AU7 but at a negligible level of 0.3 out of 6. Example video frames are included in Table 7.1 to demonstrate the visual impact of the average intensity difference obtained for each AU. As seen from these example frames, there is hardly any visual difference in the facial expressions before and after applying face de-identification.

---

<sup>1</sup><https://github.com/TadasBaltrusaitis/OpenFace>

Table 7.1 Average absolute difference in AU intensity between original frames and corresponding de-identified frames, with example frames demonstrating that the average difference with each AU.

AU intensity difference		Example frames	
		Original	De-identified
AU4	0.148		
AU6	0.232		
AU7	0.302		
AU9	0.060		
AU10	0.260		
AU12	0.215		
AU20	0.077		
AU25	0.135		
AU26	0.166		

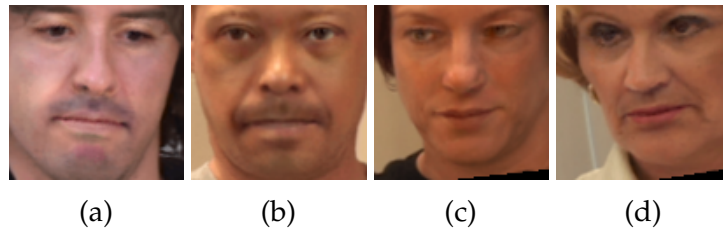


Fig. 7.8 Examples of the cropped and resized face images used in the identity consistency test

### 7.5.2 Identity consistency of the de-identified videos

Each UNBC-McMaster video sequence contains face instances of only one person. The following experiment aimed to evaluate the identity consistency within each de-identified video, i.e. to test whether the face instances within a de-identified video still relate to the same identity. The face images used in this experiment have been cropped out, aligned with the inner corners of the eyes and the tip of the nose, and resized to  $100 \times 100$  (see Fig. 7.8). The face recognition used  $LBP_{8,2}^{u2}$  face feature [97] and conducted a 10-fold cross validation on the de-identified faces.

The identity consistency was tested sequence-wise firstly. Each video sequence was de-identified independently. The result shows that 99.98% of the de-identified faces were matched with another de-identified face from the same video sequence at rank-1. The identity consistency was then tested across videos. The original videos were first grouped according to their identities given by the dataset and then the same *identity shift* was used to de-identify all the videos in the same group. The 10-fold cross validation result shows that 99.99% of the de-identified faces are top-rank matched as another de-identified face from the same identity group, where 97.57% of the correct matches are found within the same video sequence and 2.43% from another video sequence of the same person.

### 7.5.3 Privacy protection performance

The de-identification process of the identity pool can be performed with any face de-identification method and three face de-identification methods were tested, including the  $k$ -Diff-furthest method,  $k$ -Same method and face swapping method. The face swapping method followed the work of [50], [120], where Rank- $i$  face swapping replaces an original face with its  $i$ th closest face chosen from an identity pool of size  $n$ . The identity pool consists of  $n = 780$  person-specific faces from the FERET dataset. All 43 734 original faces extracted from the UNBC-McMaster videos plus the identity

pool were used to form the face gallery in the re-identification test. The identity pool was included in the gallery to increase the number of subjects. Otherwise, the re-identification risk even by random matching would be 1 out of the 25 subjects of the UNBC-McMaster database. Eigenfaces [89] and LBP [97] face recognition methods were used to match the de-identified faces of the UNBC-McMaster videos with those in the face gallery. For each face de-identification method being tested, all the 43 734 de-identified faces before Poisson blending and background merging have been used as the probe images. Again, the face regions are cropped out, aligned and resized to  $100 \times 100$ .

Table 7.2 shows the rank-1 re-identification risk of the de-identified faces and compares the privacy protection performance of the three face de-identification methods mentioned above as well as the two texture transfer approaches described in Section 7.3.3. Rank-1 re-identification risk in Table 7.2 relates to the cases when a de-identified video frame has been matched with any frame from its original video sequence.

Results in Table 7.2 show that when implementing *identity shift* in the feature space all the tested face de-identification methods have been able to provide sufficient privacy protection against similarity-based face recognition software. Transferring face texture in the pixel space gives the de-identified faces more original visual details. However, as expected, it sacrifices the privacy protection performance as more original face texture details such as wrinkles and facial hair are also transferred to the de-identified faces.

As shown in Table 7.2, the *k*-Diff-furthest method has outperformed the *k*-Same method in both FET through the feature space and the pixel space. The performance of Rank-1 face swapping is comparable to that of *k*-Same. With the identity pool of 780 faces, the performance of *k*-Diff-furthest is comparable to Rank-200 face swapping. Rank-*n* face swapping has generated the lowest re-identification risk. However, Rank-*n* face swapping tends to choose the few outliers in the identity pool to replace the original faces. In the experiments, Rank-*n* has used only 14 identities to de-identify the entire identity pool of 780 identities with 625 identities (80%) sharing the same 5 new identities. The mechanism of the *k*-Diff-furthest method guarantees that each original face is replaced with a unique de-identified face and the set of de-identified faces remains as diverse and distinguishable as the original set. Furthermore, both *k*-Same and *k*-Diff-furthest methods replace an original face with a synthesised face while the other three face swapping methods use a natural face appearance chosen from an identity pool.

Table 7.2 Rank-1 re-identification risk (%) of the de-identified video frames

Texture transfer in	Re-id feature	De-identification method				
		$k$ -Same	$k$ -Diff-f	Rank-1	Rank-200	Rank- $n$
feature space	PCA	3.83	2.56	4.78	2.67	0.51
	LBP	2.66	1.29	1.98	1.74	0.95
pixel space	PCA	23.28	16.76	21.20	19.29	4.66
	LBP	20.95	13.59	19.36	13.09	8.33

## 7.6 Conclusions

This chapter introduces a FET process to the face de-identification system to preserve facial motions and expressions. The proposed FET can map the dynamic changes on the original faces to the de-identified faces so that not only the categories of the facial expression but also the intensities of the facial AUs can be preserved. The expression preservation performance of the proposed  $k$ -Same-furthest with FET outperforms the  $k$ -Same-Select. The FET is an additional step to the face de-identification method, and the experimental results show that the FET has a negligible impact on the re-identification risk. The FET can help extend the face de-identification methods that have been designed for image set to video data. The second part of this chapter proposed an efficient approach that achieves privacy protection and preservation of facial expressions simultaneously through the simple operation of adding a pre-calculated *identity shift* to the original face instances in the input video. The use of the same *identity shift* for each subject in the original videos guarantees identity consistency in the de-identified video sequences. It also allows the dynamics of facial expressions presented in an original video to be preserved in the de-identified video. Last but not least, as the neutral frontal face of the original face identities are not always available, this chapter introduced the new concept of identity pool which is used to estimate the neutral frontal face of an unseen identity using an existing face image dataset. Such a method can extend face de-identification from the closed-set to open-set.

# Chapter 8

## Appearance model-based GAN for face de-identification

### 8.1 Introduction

In Chapter 7, an efficient face de-identification system has been proposed for videos. The proposed system achieves both privacy protection and preservation of facial expressions and head motions through a simple approach of calculating the identity shift caused by de-identifying the neutral frontal face of a subject and then adding the same identity shift to all the face instances of the same subject in the given video sequence(s). This approach was then extended to open-set face de-identification where the neutral frontal face of the subject to be de-identified is unseen (i.e. not available in the system gallery of known subjects). This extension to open-set has been achieved by finding an estimate neutral frontal face of the target subject and using the identity shift of the estimate in the calculation, where the estimate is chosen from an identity pool. Obviously, to ensure a close resemblance can be found for any given subject, the identity pool must contain an adequate number of neutral frontal faces and there must be rich diversity among these faces. Establishing such an identity pool tends to be difficult due to the requirements mentioned above and the ethical reasons and is meant to be time-consuming. However, generative models have provided an effective and economical way to solve this problem by generating face images of artificial identities based on existing data.

Face image synthesis is an effective approach to investigate the face feature subspace. It is also a pre-processing step in many face recognition systems for head pose or facial expression normalisation. Majority work has investigated the feature

manifold (subspace). The statistical models such as AAM [58] and 3DMM [59] show that different face images can be reconstructed by manipulating the low-dimensional features in the model feature space. Huang et al. [133] proposed a manifold estimation method to synthesise face images with different head poses from one single image. Sagonas et al. [134] proposed a face frontalisation model using a statistical model of frontal face images.

With the rapid development and success of deep neural networks in recent years, the task of generating face image is now mainly completed by employing neural network-based approaches such as a VAE or a GAN architecture.

In this Chapter, the GAN architecture is employed to generate neutral frontal face images for the formation of a diverse identity pool. In general, the generator in a GAN architecture is trained to map a noise distribution to the distribution of face image pixels. Due to the complexity of this task, the performance of a GAN is not always ideal. The output images often present a distorted human face with low resolution despite the complex network architecture and the massive training set used. In order to address these issues, this Chapter introduces an Appearance Model-based GAN (AMGAN), where a statistical face appearance model is employed at the input of the GAN with the aim to reduce the complexity of the GAN learning task through transforming the input data from the pixel space to the appearance model feature space. As a result, the GAN is able to generate the required appearance model parameters using shallow Multilayer Perceptrons (MLPs) and the generated model parameters can be used to construct  $200 \times 200$  high-quality face images.

Furthermore, facial attributes have been used as conditions during the training procedure. By doing so, the generator is able to generate a face appearance from given semantic descriptions, i.e. the generated face images can have certain facial attributes such as facial expression as specified in the given description. An identity pool has been constructed with the proposed AMGAN and applied to the face de-identification system described in Chapter 7. The re-identification risk experiment shows that the generated data can replace the real data and provides comparable protection performance.

## 8.2 Generative Adversarial Networks

Goodfellow et al. proposed the concept of GANs [135] and it has become the most popular training strategy for generative models. As shown in (8.1), GANs contain two adversarial models, a generative model  $G$  and a discriminative model  $D$ . In



the context of GANs, both  $G$  and  $D$  are neural networks.  $D$  examines data samples to determine whether they are real or fake, while  $G$  tries to fool  $D$  by producing random samples that comply with the distribution of the given real ones. The training process of a GAN is a zero-sum or minimax two-player game between its  $G$  and its  $D$ , which is typically defined by

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] , \quad (8.1)$$

where  $x$  is the real data complying with distribution  $p_{\text{data}}$  and  $z$  is random noise sampled from distribution  $p_z$ . The models  $G$  and  $D$  are trained through alternative optimisation, where parameters of  $D$  are optimised to maximise the objective function (i.e. increase the probability of classifying a real/fake sample correctly) and parameters of  $G$  are optimised to minimise the objective function (i.e. decrease the probability of classifying a fake sample correctly). Both models get optimised until an equilibrium is established. The equilibrium is a saddle point of the objective function.

One of the most popular applications of adversarial networks is the creation of realistic-looking images. The first proposed GAN architecture [135] is able to create images of simple objects such as hand-written digits. However, for more complicated objects such as animals, these early GANs create fuzzy images when learning from the CIFAR-10 dataset. The work of Deep Convolutional Generative Adversarial Network (DCGAN) [136] proposed by Radford et al. was a breakthrough on image generation using GANs, where CNN architecture was used for both the generator and the discriminator, and it has become a popular architecture that has been used in many following models. However, GANs are often trained using gradient decent methods that have been designed to minimise a loss function. When used to search for a Nash equilibrium, these methods make the training of a GAN unstable and fail to converge. Salimans et al. [137] have provided suggestions to improve the training convergence and the quality of the generated data, including minibatch discrimination, one-sided label smoothing and inception scores. Another contribution of their work was that they further divided the set of real samples into  $k$  classes. Instead of a binary (real or fake) classification, they trained their discriminator to perform a  $k + 1$  classification and suggested that using the class labels of real samples on the discriminator side to conduct the semi-supervised learning could enable the model to learn recognisable features of the classes and help improve the visual quality of the generated data.

Often there is a need to control specific attributes of the synthesised samples, e.g. gender or head pose of the synthesised faces. Conditional GANs (cGANs) [138] has been proposed for this purpose. Equation (8.1) can be extended to cGANs by adding data labels as conditions  $c$  to both the generator and the discriminator, as shown in (8.2).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x, c \sim p_{\text{data}}(x, c)} [\log D(x, c)] + \mathbb{E}_{z \sim p_z(z), c \sim p_{\text{data}}(c)} [\log(1 - D(G(z, c), c))] \quad (8.2)$$

The Auxiliary Classifier GAN (AC-GAN) [139] uses the class labels of the real data to train an auxiliary classifier on the outputs of the discriminator so that the discriminator can not only examine the real/fake data but also classify the given samples. Instead of class labels, images have been used as the conditional inputs to the multiple generators in the LAPGAN network [140].

There have several GAN variants work been proposed to generate face images. They have been demonstrated their performance in generating face images from random noise distribution [135], [136], [141]–[145]. More recently, Karras et al. [146] generated high-quality facial images with an incremental growth GAN. Furthermore, data labels have been used to control the outputs of a GAN, e.g. certain attributes of the output face images. Larsen et al. [147] used the VAE/GAN model to learn the latent representation of an input face image. In their work, the attributes are manipulated in the latent space, and a decoder generates the image for the discriminator. Invertible Conditional GANs [148] use the cGANs architecture. In addition, they trained two encoders to map the input face images to latent space and attribute labels. Age-cGAN proposed by Antipov et al. [149] is able to generate face images of 6 difference age groups for the same identity. Huang et al. [150] generate frontal faces from profile faces with a Two-Pathway GAN in which the generator takes both global facial image and local patches.

Apart from investigating on the network architectures, researchers have also worked on the training loss in order to achieve a better training performance, e.g. to solve the gradient vanishing problem. Equation (8.1) is used for theoretical analysis. In practice, the discriminator and the generator optimise their loss function  $L^{(D)}$  and  $L^{(G)}$  defined in (8.3), respectively.  $L^{(G)}$  is not in the flip sign form of  $L^{(D)}$  but a flip

target form. Thus, the training process is no longer a zero-sum game.

$$\begin{aligned} L^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) &= -\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \log D(\mathbf{x}) - \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \log(1 - D(G(\mathbf{z}))) \\ L^{(G)} &= -\frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \log D(G(\mathbf{z})), \end{aligned} \quad (8.3)$$

Least Squares GAN (LSGAN) proposed by Mao et al. [142] minimises the loss function in Pearson  $\chi^2$  divergence instead of KL divergence, where loss functions of  $D$  and  $G$  are defined as

$$\begin{aligned} L_{\text{LSGAN}}^{(D)} &= \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[ (D(\mathbf{x}) - b)^2 \right] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \left[ (D(G(\mathbf{z})) - a)^2 \right] \\ L_{\text{LSGAN}}^{(G)} &= \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \left[ (D(G(\mathbf{z})) - c)^2 \right], \end{aligned} \quad (8.4)$$

where  $a$  and  $b$  are the labels for fake data and real data, respectively. Moreover,  $c$  is the value that  $G$  wants  $D$  to believe for fake data, in our experiments they are set as  $a = 0$  and  $b = c = 1$ .

Arjovsky et al. proposed Wasserstein GAN (WGAN) [143] which optimises the loss functions in the Earth-Mover distance and the loss function can be written as Equation (8.5). Weight clipping is applied to the weights during training, to satisfy the condition that the discriminator must lie within the space of 1-Lipschitz functions,

$$\begin{aligned} L_{\text{WGAN}}^{(D)} &= -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} [D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} [D(G(\mathbf{z}))] \\ L_{\text{WGAN}}^{(G)} &= -\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} [D(G(\mathbf{z}))], \end{aligned} \quad (8.5)$$

However, Gulrajani et al. [144] demonstrated that weight clipping might cause undesired behaviour, and they proposed Wasserstein GAN with gradient penalty (WGAN-GP) to address this problem. The loss functions for  $G$  and  $D$  is shown in (8.6).

$$\begin{aligned} L_{\text{WGAN-GP}}^{(D)} &= -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} [D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} [D(G(\mathbf{z}))] + \underbrace{\lambda \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}} \left[ (\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2 \right]}_{\text{gradient penalty}} \\ L_{\text{WGAN-GP}}^{(G)} &= -\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} [D(G(\mathbf{z}))], \end{aligned} \quad (8.6)$$

where  $\hat{\mathbf{x}}$  is sampled randomly between real data  $\mathbf{x}$  and generated data, and  $\lambda$  is the penalty coefficient.

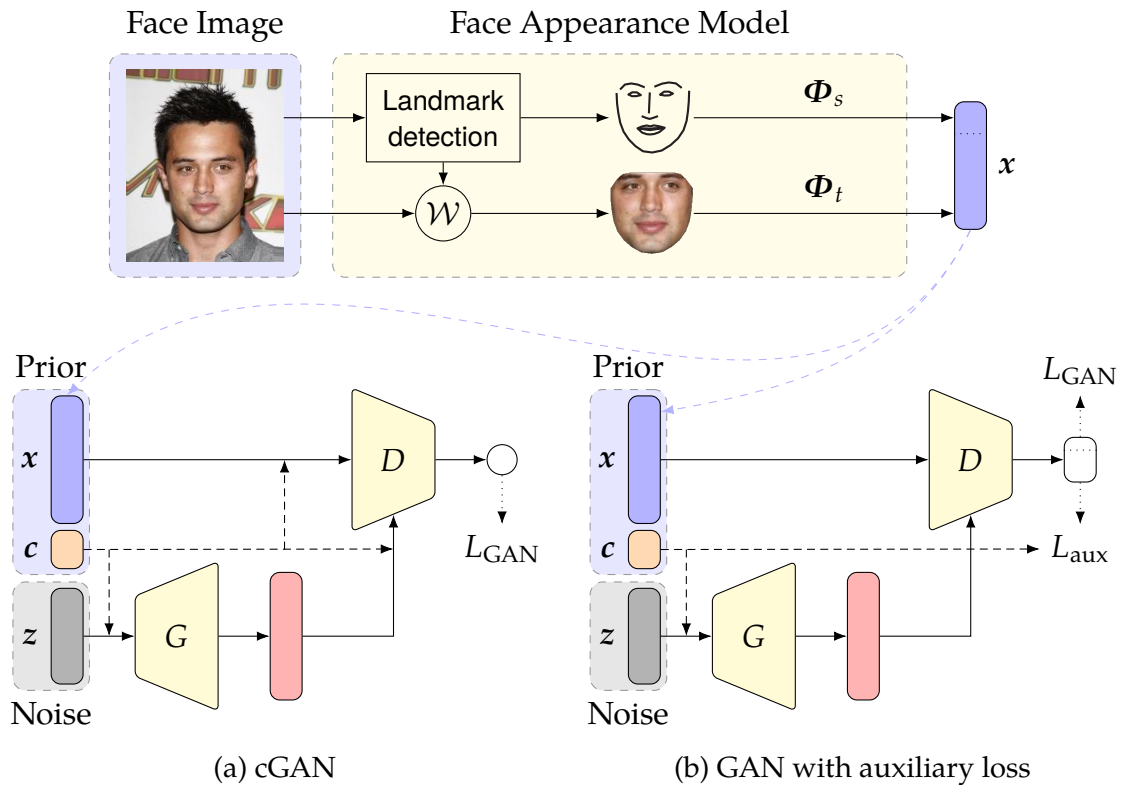


Fig. 8.1 Overview of the AMGAN

### 8.3 Appearance Model-based GAN

Using a DCGAN architecture can generate realistic and high-quality face images, but it requires a significant amount of computational resource and time [146]. To simplify the face image generating task, the prior work of face landmark detection is used to allocate the face region and build a face appearance model through linear transformations. Instead of using images, the appearance model parameters are used for the following GAN learning. Because the dimension of inputs is significantly reduced through PCA, the adversarial training task can be completed with two shallow networks. An AMGAN is proposed in this section. Furthermore, two GAN architectures and training schemes are discussed with the aim of controlling certain attributes of the generated face images. At the end of this section, the proposed AMGAN is used to generate the identity pool of face images for the face de-identification system.

### 8.3.1 Network architecture

As shown in Fig. 8.1, during the training phase, all the pixel-based face images are first fed into a pre-trained appearance model and represented in the model feature space as a set of parameters. The model parameters of all the real images in the training set are used to form the prior probability distribution  $p_{\text{data}}(\mathbf{x})$ . The generator in the system is trained to map a noise distribution  $p_z(\mathbf{z})$  to the prior distribution  $p_{\text{data}}(\mathbf{x})$ . The generator and the discriminator in the proposed GAN system are two shallow MLP networks because the dimensionality of the network input data has been significantly reduced through the appearance model. The generator takes a uniformly distributed noise as the inputs while the input to the discriminator can be either the scaled appearance model parameters of real face images or the generated parameters from the generator. In cGAN architecture, both the  $G$  and  $D$  take a condition vector  $\mathbf{c} \in \mathbb{R}^n$  as a second input. The discriminator distinguishes whether the input samples are from the real data prior or the distribution estimated by  $G$ . In the GAN with auxiliary loss architecture, the output of  $D$  is not a single value but a  $n + 1$  dimensional vector. The first value of the output is used to compute a regular GAN loss and the rest to compute an auxiliary loss. As mentioned, all the data samples used in the AMGAN training are not at the image pixel level but the appearance model parameter level. When the training is completed, the model parameters generated by the trained  $G$  are projected back into the pixel space to produce images.

### 8.3.2 Appearance model parameter pre-processing

The face appearance model represents face images in model subspace as face features. Each real data sample  $\mathbf{x}$  for the GAN is a concatenation of normalised model parameters. The shape and texture model parameters are firstly normalised with the corresponding eigenvalues  $\lambda$  and then downscaled with a scalar  $a$ . In a normal distribution, 99.7% of the data are within  $\pm 3\sqrt{\lambda}$ . A value greater than 3 for  $a$  is recommended to downscale the feature value to around  $[-1, 1]$ . Overall, the rescaled feature vector  $\mathbf{x}$  of a real image is calculated as

$$\mathbf{x} = \frac{1}{a} \text{diag} \left( \begin{bmatrix} \lambda_s \\ \lambda_t \end{bmatrix}^{-\frac{1}{2}} \right) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (8.7)$$

Then the rescaled feature vectors of the real images are used along with random noise samples  $z$  to conduct the training of the GAN.

### 8.3.3 Attribute-controlled face synthesis

The aim of including GAN in this project is to overcome the lack of data problem and generate an artificial identity pool for the face de-identification system. To control certain attributes of the outputs of a generator, data labels are required to feed into the generator as additional information. Some face dataset provides facial attribute labels which describe certain characteristics on the corresponding face images, e.g. gender, age, expression, and face shape, etc. Two types of face attribute labels are considered in the following experiments, namely the categorical label and the continuous label.

**Categorical label:** The labels provided by the dataset which contains binary classes of each attribute, e.g. the CelebA dataset [151] provides 40 binary face attributes per image. Positive is marked as 1 and negative is marked as  $-1$ .

**Continuous label:** The labels are not binary but continuous intensity values of each class. e.g. LFW dataset [152] provides the intensity values of 73 face attributes which are detected by a facial attribute classifier [153].

As mentioned, the appearance model parameters are normalised and rescaled in the way described in Section 8.3.2. It is necessary to normalise the attribute labels to the same range, e.g. the interval of  $[-1, 1]$  is used in the following experiments. Pre-processed label values are used in the both of generator and discriminator as the condition values.

#### Conditional GAN

One approach to using the conditions in the GAN training is the cGAN architecture [138] which is shown in Fig. 8.1a. The cGAN architecture combines the condition values  $c \in \mathbb{R}^n$  with the inputs to  $G$  and  $D$  while the outputs of  $G$  and  $D$  are same as when train the GAN without conditions. The objective function of cGAN is shown in (8.2), while in practice, it has been split into the generator loss and discriminator loss and adapted with LSGAN, WGAN and WGAN-GP. The  $x$ ,  $z$  and  $G(z)$  in (8.4), (8.5) and (8.6) are replaced by  $\begin{bmatrix} x \\ c \end{bmatrix}$ ,  $\begin{bmatrix} z \\ c \end{bmatrix}$  and  $\begin{bmatrix} G(z) \\ c \end{bmatrix}$  respectively.

Essentially, the discriminator is modelling a joint distribution of real/generated data and conditions.

### Auxiliary losses for conditions

Another approach to using the conditions is that the discriminator does not take the conditions as the inputs but an auxiliary classification/regression task to help the model learn the conditional values. It is similar to AC-GAN architecture, but the auxiliary loss needs to be redefined instead of using multi-class classification loss. The discriminator outputs  $n + 1$  logits where  $n$  is the size of the condition values and  $\mathbf{y} \in \mathbb{R}^n$  are the logits which are used in the auxiliary task in the following descriptions. The overall losses of generator and discriminator are composed by two parts,

$$L = L_{\text{GAN}} + \lambda L_{\text{aux}} \quad (8.8)$$

where  $L_{\text{GAN}}$  represents both generator loss and discriminator loss in all above mentioned GAN losses.  $L_{\text{aux}}$  is the auxiliary loss and it is computed differently depend on the type of the attribute label.  $\lambda$  is the auxiliary loss coefficient and it is set to  $\lambda = 1$  in the following experiments.

- For the categorical label, the auxiliary task is essentially a multi-label classification. Both the logits  $\mathbf{y}$  and the condition values  $\mathbf{c}$  need to be mapped to the interval of  $[0, 1]$  using

$$\begin{aligned} \mathbf{p} &= H(\mathbf{c}) \\ \hat{\mathbf{p}} &= S(\mathbf{y}) \end{aligned} \quad (8.9)$$

where  $H$  is the Heaviside step function and  $S$  is the sigmoid function. The auxiliary loss for categorical label is

$$L_{\text{aux}} = -\frac{1}{n} \sum_{i=1}^n [p_i \log \hat{p}_i + (1 - p_i) \log(1 - \hat{p}_i)] \quad (8.10)$$

- For the continuous label, the auxiliary loss is the mean squared error between the logits  $\mathbf{y}$  from the discriminator and the conditional values  $\mathbf{c}$ ,

$$L_{\text{aux}} = \frac{1}{2n} \|\mathbf{y} - \mathbf{c}\|^2 \quad (8.11)$$

### 8.3.4 Generating the identity pool with an AMGAN

As mentioned in Chapter 7, the identity pool for the face de-identification system must consist of neutral frontal face images ideally with appropriate illumination. Although attribute labels have been used to control certain attributes of the face images produced by the generator, some attributes such as head pose and image brightness are not controlled. In addition, the real images used to train the AMGAN are from in-the-wild image sets and present various head poses and illumination conditions. This means the generated face images cannot be added directly into the identity pool without selection. A group of two filters have been designed for automatic identity pool candidate selection. The filters examine the generated model parameters in two aspects, explicitly, head pose and image fidelity. The identity pool only accepts the samples which can pass both filters.

**Head pose:** As mentioned, in the trained shape model the first two dimensions represent the head pitch angles and yaw angles. The frontal face should have small parameter values along these two dimensions. Therefore, a generated model parameter vector is dropped if any of its first two shape parameters is greater than the square root of their corresponding eigenvalues  $\sqrt{\lambda_{s1}}$  and  $\sqrt{\lambda_{s2}}$ .

**Image fidelity:** The classification scores from the discriminator are used to filter the generated data to ensure identity pool only contains the images with high visual quality. Although the discriminator in the GAN is not guaranteed to be the optimal solution for distinguishing real and fake data, the high score samples tend to provide realistic looking face images and hence used as the valid candidates for the identity pool.



## 8.4 Experiments

### 8.4.1 Datasets

LFW [152] and CelebA [151] datasets have been considered as the training set for the experiments in this chapter. For experiments on cGANs, both datasets have been used in the tests as they provide different types of attribute labels. For experiments on GANs without conditional inputs, only CelebA dataset has been used as the training set as both datasets contain data of similar natures while CelebA dataset contains more images than LFW dataset. All the facial landmarks were detected by CLNF [40], [131] face landmark detector since neither of these datasets provides facial landmark ground truth.<sup>1</sup> Images that CLNF has failed to detect a full set of 68 landmarks are excluded from the respective training set. As a result, the LFW training set has all the 13 114 images in the original LFW dataset and the CelebA training set has 198 765 out of 202 599 images from the original CelebA dataset. All the images in the training set are aligned with a  $200 \times 200$  mean shape based on the facial landmarks. The same as previous chapters, the appearance model used in the AMGAN has been trained with the same 1952 frontal face images from the FERET dataset. The trained appearance model contains 10 shape components and 147 texture components, representing 95% shape variance and 95% texture variance. It is worth noting that the training data of the appearance model are the frontal face images which are different from the in-the-wild face images used to train the AMGAN. Therefore, when transferring the in-the-wild image data into the pre-trained appearance model space, some dimensions would present higher data variances than other dimensions. For example, the LFW/CelebA image dataset has a variance higher than the corresponding eigenvalue in the second dimension of the shape model, which represents the head rotation (see Fig. 8.2). In this experiment, the rescaling coefficient  $a$  in (8.7) is set to 5, which downscales the appearance model parameters of most images in both GAN training sets to the interval of  $[-1, 1]$ .

---

<sup>1</sup>The CLNF facial landmark detector is provided by the OpenFace toolkit which is available on <https://github.com/TadasBaltrusaitis/OpenFace>.

### 8.4.2 Generating random faces with the AMGAN without conditions

The experiment presented in this subsection has been conducted to test whether the proposed AMGAN architecture could map the uniformly distributed noise samples to the appearance model parameter distribution of real face images. As mentioned, a subset of CelebA dataset was used as the training set in this experiment. Both the generator and the discriminator are three-layer MLPs which include an input layer, a hidden layer and an output layer. The size of the hidden layer in both neural networks is 128. The Leaky ReLU activation function is used in the hidden layer with  $\alpha = 0.2$ . The input and output layer sizes of the generator are 128 and 157; the input and output layer sizes of the discriminator are 157 and 1. There is no activation function in the output layer of the generator, because the real data is not completely constrained to  $[-1, 1]$ . The AMGAN has been trained with LSGAN, WGAN and WGAN-GP loss functions, respectively. The learning rate of the generator has been set to  $10^{-3}$  and the learning rate of the discriminator  $10^{-4}$ . In each iteration, the discriminator updates 2 times while the generator updates once. For LSGAN and WGAN loss functions, the RMSProp method has been used to optimise the networks. For WGAN-GP, the Adam optimisation method has been used, where  $\beta_1 = 0.5, \beta_2 = 0.999$  and the gradient penalty  $\lambda = 1$ .

Fig. 8.2 compares the data distributions of real face images against those produced by the trained  $G$ . The first row in Fig. 8.2 shows the distributions of the first two dimensions of face shape parameters and the second row in Fig. 8.2 shows the distributions of the first two face texture parameters. As shown in both figures, with the same network architectures, loss functions LSGAN, WGAN and WGAN-GP can all generate a data distribution that closely resembles the real data distribution. The GAN models trained with all three loss functions can learn the bias and variance in each dimension, e.g. the second dimension of shape parameters have a higher variance than other dimensions as shown in Fig. 8.2. There was no mode collapse observed from the results.

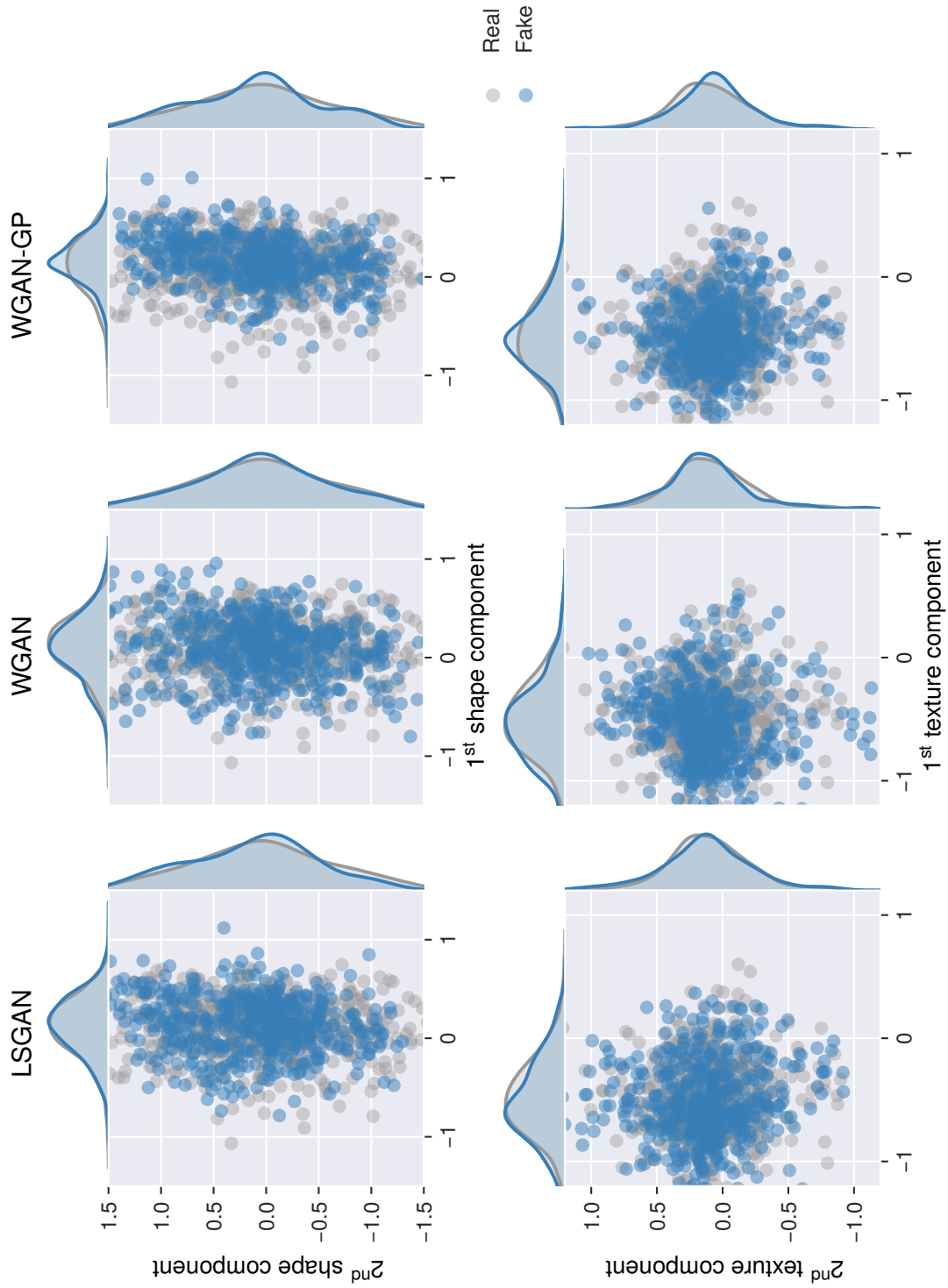


Fig. 8.2 Data distribution of 500 random samples in the first two dimensions of the shape and texture models. All samples in the figure have been randomly generated by our trained generator.



Fig. 8.3 Example face images reconstructed from LSGAN generated appearance model parameters. Each row illustrates the face images corresponding to a linear interpolation between two random input noise samples.

Fig. 8.3 shows some face image examples reconstructed from the model parameters generated by the LSGAN model. Each row illustrates the face images corresponding to a linear interpolation between two random input noise samples. On the one hand, it shows that the generated faces have a high visual quality in the resolution of  $200 \times 200$ . On the other hand, it shows that the continuous change in the noise dimension has been mapped to the appearance model feature manifold.

### 8.4.3 Generating face images with attributes

The aim of this experiment is to exam the performance of the proposed AMGAN with attribute control and select the more suitable model for the generation of an identity pool. From the provided attribute labels, three common attributes have been selected for CelebA and LFW datasets, namely male, youth and smiling. The images from these two datasets were used to train the AMGAN with the three loss functions described in Section 8.4.2. Both cGAN architecture and GAN with auxiliary loss architecture are used for comparison. In the training phase, the network training hyperparameters are specified in Section 8.4.2. In the test phase, every attribute in  $c$  has been examined by linear interpolating the corresponding condition value independently while setting the rest of the condition values to zero. The example synthetic images are shown in Fig. 8.6–8.8 and the values in the noise vector  $z$  of all the shown examples have been fixed to zero. Regardless of the loss functions and

network architectures, all the GANs trained with the CelebA dataset have produced expected results in the sense that the generated faces are associated with the same identity while the facial attributes vary in correspondence with the given attribute values. In contrast, as shown on the examples generated by a GAN trained on the LFW dataset, the change of the attributes value can affect the identities of the generated images. The output examples from the GAN trained with LFW data indicate that the generator could not distinguish  $z$  and  $c$  well enough when being given continuous labels. Furthermore, it is obvious that when the given attribute is close to  $-1$  or  $1$ , the GANs trained with continuous labels may generate face images with low visual quality.

Further investigation on the smiling attribute has been conducted and the changing path of shape model parameters on third and fourth dimension when smiling attribute changes from  $-1$  to  $1$  are shown on Fig. 8.9 and 8.10. The third dimension of shape parameters represents the positions of the corners of the mouth and the fourth dimension of shape parameters represents the mouth openness (see also Fig. 3.3). Because the CelebA dataset describes each face attribute with a categorical label, the trained generator has estimated a more linear transition between the attribute values  $-1$  and  $1$ . In contrast, the GANs trained with the LFW dataset and its continuous labels, have shown a better understanding of the non-linear transition of smiling embedded in the manifold. However, the learned smiling path is not associated with one identity but across the dataset. Apart from the different types of labels, the size of CelebA dataset is 15 times larger than the LFW dataset, which means a GAN can learn a better estimation of the real face distribution.

All the generator trained with CelebA dataset can provide the expected results, and the model trained with LSGAN using cGAN architecture was selected to generate the identity pool. More specifically, all the identity pool candidates were generated and selected with the following conditions: (a) the smiling attribute value has been set to  $-0.9$ ; (b) the generated samples are selected only if the first two shape model parameters is within one standard deviation of each dimension; and (c) the score of the discriminator is higher than  $0.6$ . Some face images from the constructed identity pool are shown in Fig. 8.4.



Fig. 8.4 Example images of the generated identity pool

#### 8.4.4 Re-identification risks

The following experiment extends the re-identification risk test in Section 7.5.3 and replace the manually selected identity pool with the GAN generated identity pool. In addition, the state-of-the-art face recognition method is used in this experiment, which uses the neural network learned features rather than handcrafted features. The Inception-ResNet-v1 architecture [154] is used to extract the 128-dimensional face features, and the network was trained on MS-Celeb-1M dataset [155] with centre loss [106]. The trained model is provided by the facenet project.<sup>2</sup>

The re-identification risks are evaluated with the UNBC-McMaster dataset 10 times for different proportions of the generated identity pools. The mean  $\pm$  standard deviations of the risks are shown in Fig. 8.5. Several conclusions can be drawn from the results in Fig. 8.5. First of all, replacing the identity pool of real face images with a GAN generated identity pool has a negligible impact on the privacy protection performance of the face de-identification system. The results also show that the re-identification risks are not sensitive to the size of the identity pool, which means the lack of data does not limit the construction of the identity pool, and the identity pool can have a dynamic range and be updated on-demand. Last but not least, the state-of-the-art face recognition method yields a higher re-identification risk, the re-identification risks of using FaceNet features are two times higher than using LBP features.

## 8.5 Conclusions

With the aim to solve the lack of data problem, this chapter proposes the AMGAN to generate face images. The AMGAN is a combination of appearance model and GAN. It can be controlled to generate frontal neutral face images for face de-identification.

<sup>2</sup><https://github.com/davidsandberg/facenet>

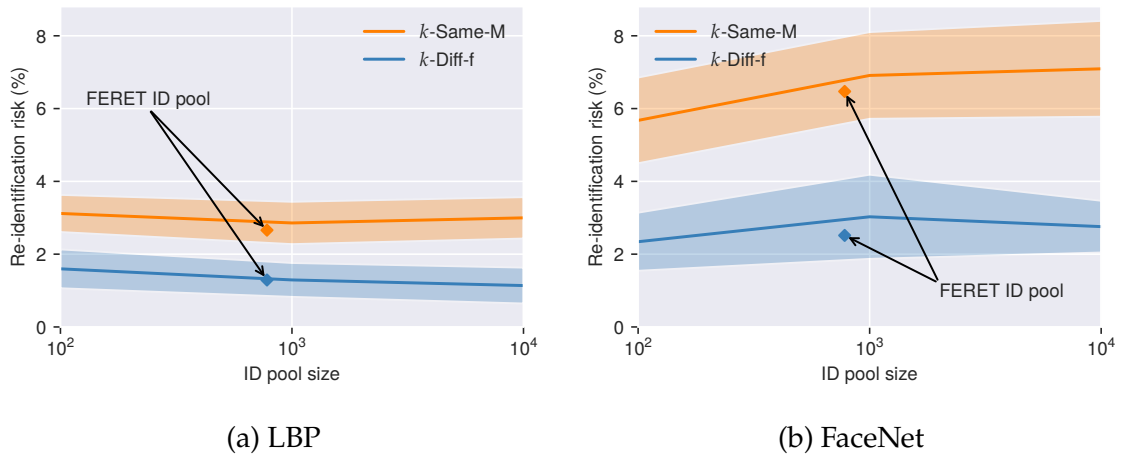


Fig. 8.5 Re-identification risks of UNBC-McMaster video frames de-identified with GAN generated identity pool.

The proposed AMGAN trains a generator that can map a noise distribution to the prior appearance model parameter distribution. Because the dimensionality of the target data distribution is highly reduced, two shallow MLP networks are used in the proposed AMGAN as the generator and the discriminator. The experiment results show that the proposed AMGAN architecture can generate realistic-looking face images and the facial attributes of the outputs can be controlled by adding conditions as inputs to the generator. Finally, an identity pool was generated by the proposed AMGAN and connected with the face de-identification system introduced in Chapter 7. The re-identification risks show that the GAN generated identity pool provides a comparable privacy protection performance as the identity pool of manually selected real images.

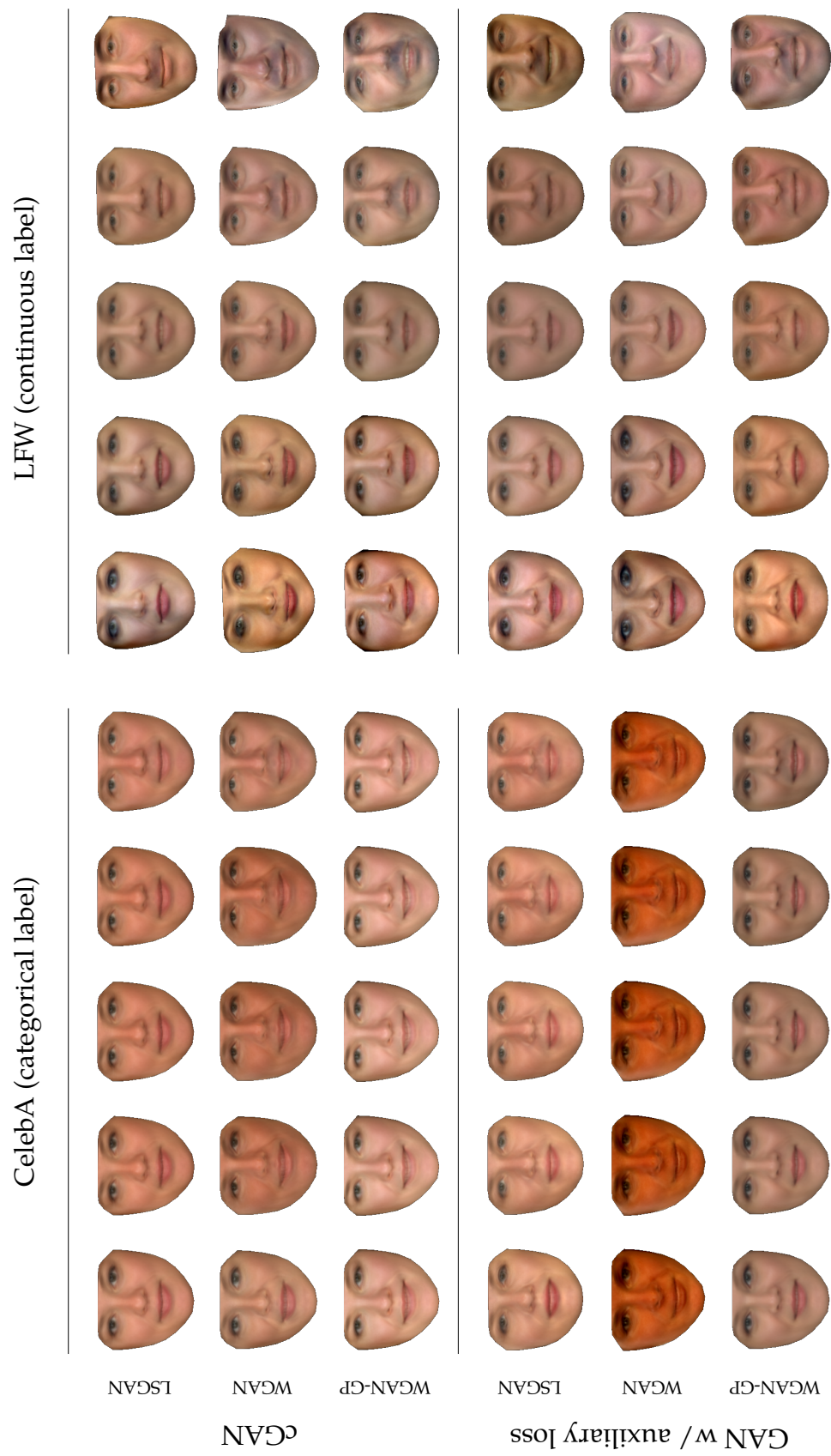


Fig. 8.6 Example face images when linear interpolation of 'male' attribute (-1 to 1)



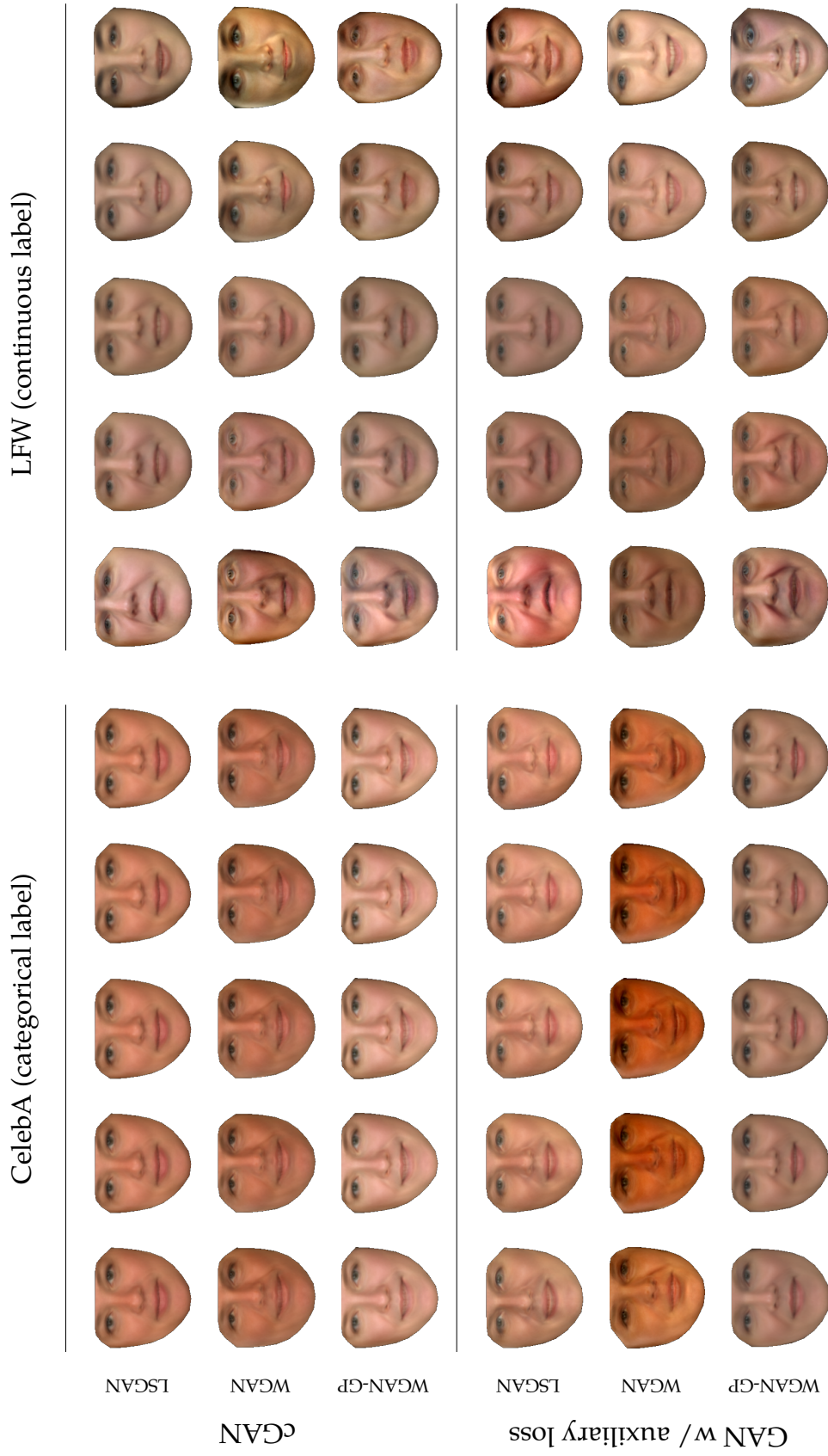


Fig. 8.7 Example face images when linear interpolation of 'youth' attribute (-1 to 1)

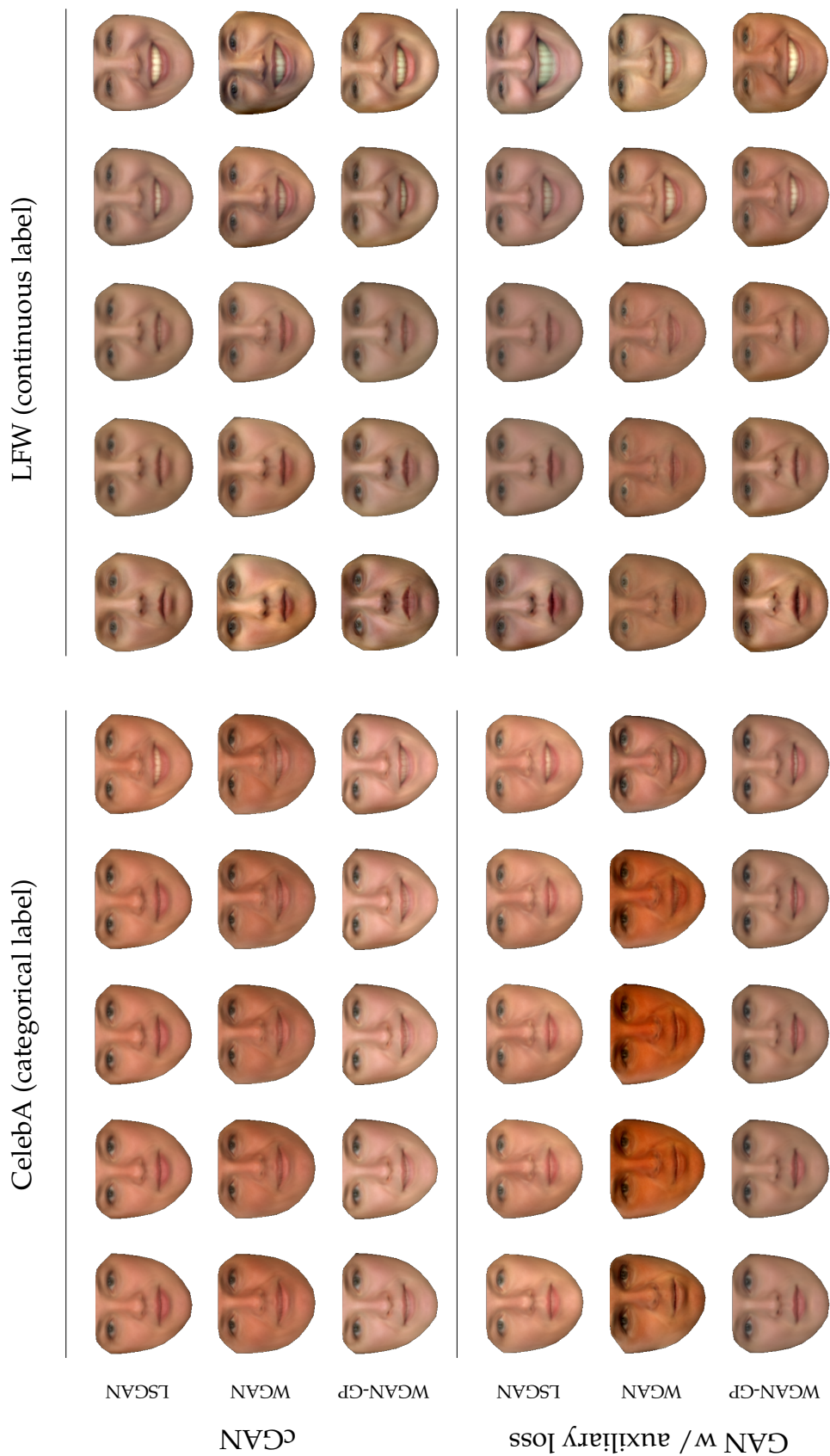


Fig. 8.8 Example face images when linear interpolation of 'smiling' attribute (-1 to 1)

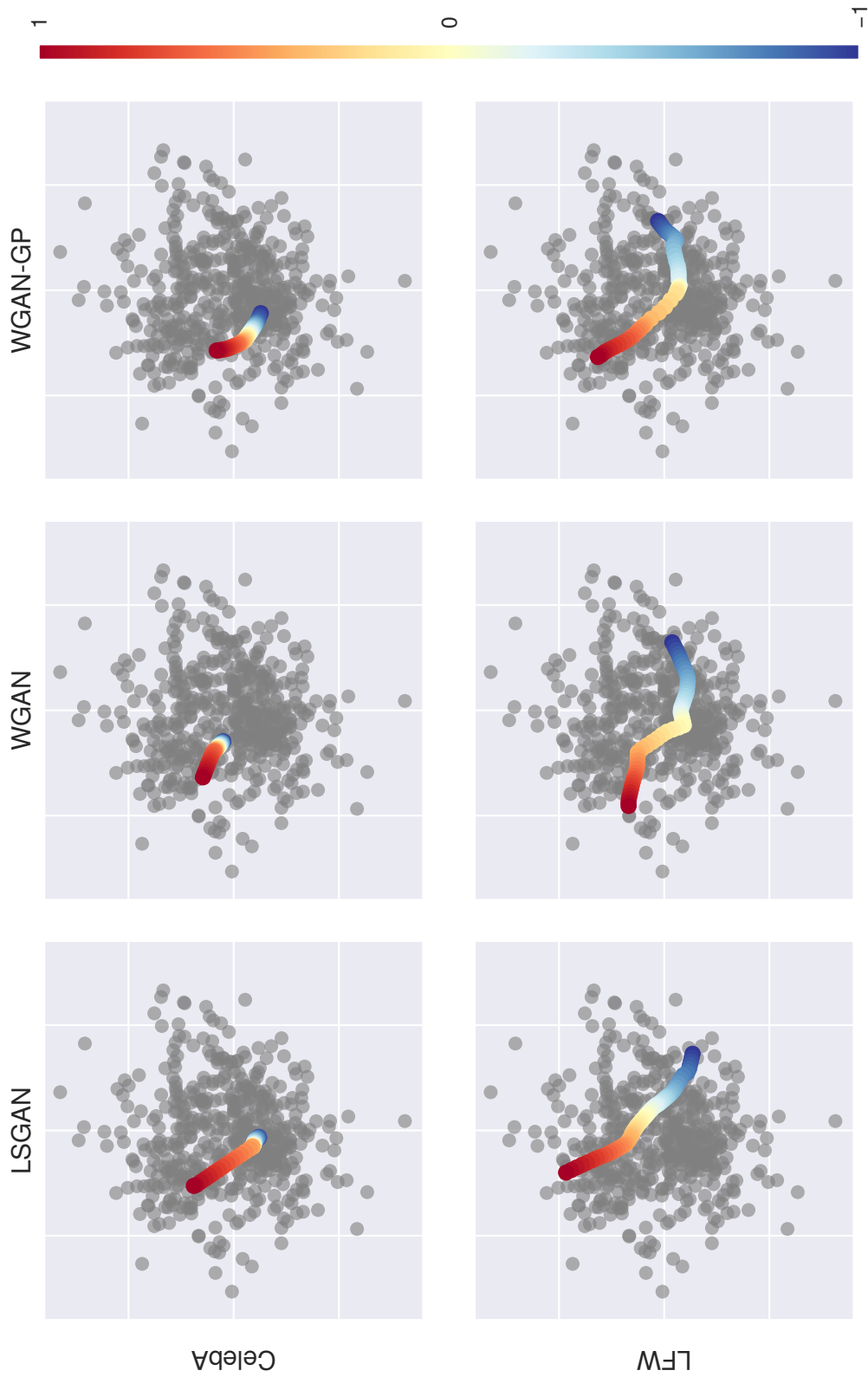


Fig. 8.9 Shape parameters of cGAN using LSGAN, WGAN and WGAN-GP losses trained on CelebA and LFW dataset. The reconstructed images can be found in Fig. 8.8.

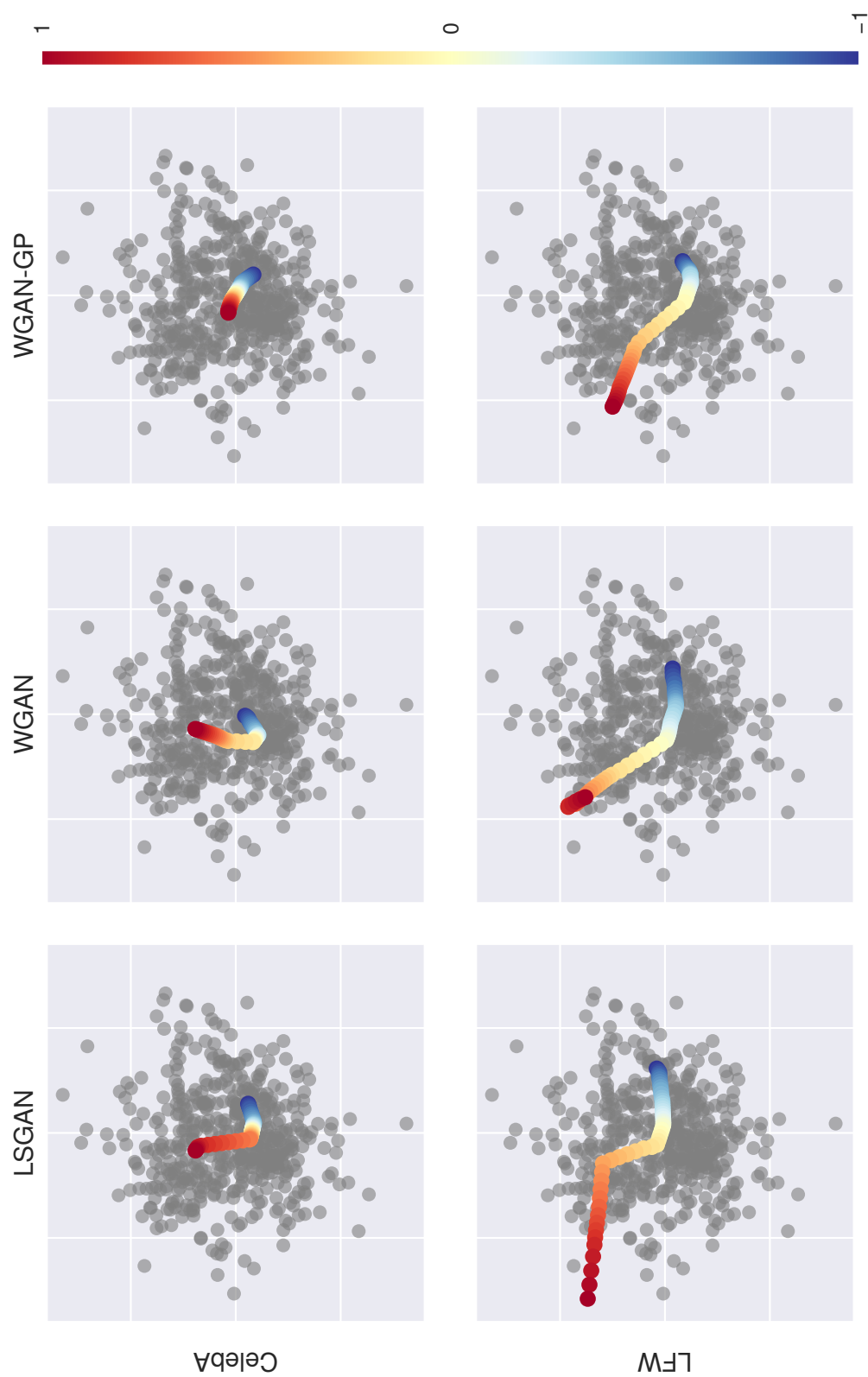


Fig. 8.10 Shape parameters of GAN using LSGAN, WGAN and WGAN-GP losses with auxiliary loss trained on CelebA and LFW dataset. The reconstructed images can be found in Fig. 8.8.

# Chapter 9

## Conclusions and future work

### 9.1 Summary

This thesis first introduced the  $k$ -Same-furthest face de-identification method which includes the wrong-map protection to the  $k$ -Same-M face de-identification. It showed as long as the dataset has a low variance on the expected data utility, the de-identification method can replace the original face with the face having the largest similarity distance to maximise the identity loss. The  $k$ -Same-furthest can provide two safeguards against re-identification attacks:

- 1) Face recognition software tends to recognise the given identity as a wrong identity;
- 2) At least  $k$  identities sharing the same copy of the de-identified face in a dataset guarantees a re-identification risk of at most  $1/k$ .

Based on the  $k$ -Same-furthest face de-identification, data utility in different application scenarios has been discussed. Data utility was considered in two aspects, the dataset-wise data utility (e.g. the data distribution within a data set) and the individual-wise data utility (e.g. the expression on an individual face). The  $k$ -Same family uses an identical face image to replace a group of faces which destroys the statistical characteristic of the original dataset. This thesis attempts to answer the question that *could we generate a de-identified face dataset that has a similar data distribution to the original?* To this end, the  $k$ -Diff-furthest and the AMGAN methods were proposed. Both methods synthesis non-exist face images using face appearance model.

The  $k$ -Diff-furthest was modified from  $k$ -Same-furthest and the microaggregation step in the  $k$ -Same-furthest was removed. It can provide wrong-map protection

meanwhile maintains the diversity of the dataset and the relative position of a face feature in its neighbourhood. The proposed AMGAN is a combination of appearance model and GAN. The generator in the AMGAN was trained to generate appearance model parameters that have the same distribution as a real face dataset. It has been discussed in Chapter 8 that the generated face images can be controlled with given conditions (face attributes). This was adopted to generate the identity pool for face de-identification. The re-identification risks show that the AMGAN generated identity pool provides a comparable privacy protection performance as the identity pool of manually selected real images.

The face de-identification methods were proposed for face datasets, while the data utility of individual face image was also discussed in this thesis. Preserving facial expression, the most important non-identity information in the face images, was discussed in this thesis and a FET process was introduced to the face de-identification system to preserve facial motions and expressions. The proposed FET can map the dynamic changes on the original faces to the de-identified faces so that not only the categories of the facial expression but also the intensities of the facial AUs can be preserved. It is an additional step to a face de-identification method, and the experimental results showed that the FET has a negligible impact on the re-identification risk.

The transfer manner in FET can help extend the face de-identification methods that have been designed for the image set to video data. The concept of identity pool has been introduced to design the face de-identification system for open-set scenario. Chapter 7 showed an efficient manner that can transfer the knowledge of a de-identified face dataset to open-set data through an identity shift process.

The proposed background merging process for face de-identification can blend the de-identified face image with its original background seamlessly and maintain the de-identified face shapes. It increases the visual quality of the de-identified face images in terms of fidelity and intelligibility.

## 9.2 Conclusions

The goal of this thesis was to investigate the face de-identification methods that can eliminate the identifying information while preserving the usability of the face image/video data. The proposed face de-identification system attempted to generate images with high visual qualities which is intelligible to both human and computer vision system. In this proposed system, the frontal neutral faces were used as identity references, and the system firstly removes the personally identifying information from the target images then add the original data utility information back to the target images. The preservation of data utility was considered in different application scenarios, and this thesis showed the trade-off between re-identification risk and data utility. Each method for face de-identification described in this thesis has its advantages and disadvantages.

- The  $k$ -Same/Diff-furthest face de-identification methods provide wrong-map protection which can reduce the re-identification risk to near zero. However, these two methods require low data utility variance in the target face dataset.
- Additionally, it is a double-edged sword that the  $k$ -Diff-furthest method generates de-identified face images without face image aggregation. The de-identified face dataset maintains the diversity of the original dataset and some local relationships among data subjects. Such information can support tasks, e.g. face tracking and kinship recognition. However, the retaining information could also help an attacker restore an original face from a de-identified face.
- The background merging method helps to generate high visual quality de-identified face images in terms of fidelity and intelligibility. The image blending method restores the low-frequency information from the original image which can increase the re-identification risk. Experimental results reveal that the image de-identification need to attach importance to image background, although the image background deformation method can help to reduce the re-identification risk from the image background.
- The FET method maps the facial expressions from the original face to de-identified faces not only in categorical level but also in intensity level. This method was used to remedy the data utility protection limitations on  $k$ -Same/Diff-furthest face de-identification methods. However, the process of FET does not satisfy the  $k$ -anonymity protection or wrong-map protection.

- The method proposed for face de-identification in videos can de-identify video sequence efficiently by using identity shift. The identity shift only need be calculated once at the first frame for each video or each data subject, and it can guarantee the identity consistency of the de-identified videos. However, it requires the face region and facial landmarks can be fully detected in each frame.
- The AMGAN was proposed to generate frontal neutral face images for face de-identification. The AMGAN learns the distribution of the appearance model feature space instead of image pixel space. It highly reduces the computational complexity and power to synthesise high-quality face images by using two shallow MLPs. However, because the appearance model reconstructs the output images, synthesised face images are limited to these faces which can be represented by the appearance model.

### 9.3 Future work

While with the work presented in this thesis, privacy preservation performance of the face de-identification system has been improved and solutions have been proposed for different application scenarios regarding the preservation of face-related data utility. There are several unsolved problems to be addressed in future work.

A face de-identification method is not competent without extracting usable information from a face image. It needs accurate face detection and face feature extraction. These are sharing tasks and challenges with face recognition. Research in these areas has developed rapidly in recent years. Integrating the state-of-the-art face recognition techniques to face de-identification can help face de-identification generate promising results.

In this thesis, the face images are de-identified in appearance model feature space. The appearance model is a PCA based model and it provides reliable statistical features. It is a linear model and provides an easy way to extract features and reconstruct data through Eigenvectors. Some semantic information has been found in the feature space of the current model, e.g. the global illumination, head poses and mouth openness, but they are limited. A more powerful semantic face model through supervised learning can help the face de-identification system maintain data utility more precisely.

In terms of increasing the visual quality, a 3D face model can be considered. There are more 3D face data publicly available and the cost of graphics rendering



has reduced. Furthermore, the knowledge of anatomies such as skull and facial musculature model can be adopted in reconstructing a face physically.

In Chapter 6, the re-identification results show that the proposed face de-identification method provides high privacy protection within the face region. However, blending the de-identified face to its original background increases the re-identification risk noticeably. Although a face recognition software focuses on the cropped face region, the information contained in the background area around a face region (e.g. hair colour, hairstyle, and dressing style) also contribute to the identification of a person. Experimental results of the re-identification attack using only the background information confirmed that face region in an image is sufficient but not necessary for identifying a person. Face region information is the most efficient information to facilitate face recognition. However, only de-identify this information cannot stop the attacker from establishing re-identification using other information such as the background. If the problem is not constrained in preserving privacy in face region but preserving the privacy of an image with faces in, de-identification must be applied to not only the face region but also all the image regions that contain personally identifiable information.



# Bibliography

- [1] S. D. Warren and L. D. Brandeis, "The Right to Privacy," *Harvard Law Review*, vol. 4, no. 5, p. 193, 1890. doi: 10.2307/1321160.
- [2] United States Congress, *Health Insurance Portability and Accountability Act of 1996*, 1996.
- [3] HHS Office for Civil Rights, "Standards for privacy of individually identifiable health information. Final rule.," *Federal register*, vol. 67, no. 157, pp. 53 181–273, 2002.
- [4] "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data," *Official Journal of the European Communities*, vol. L 281, pp. 31–50, 1995.
- [5] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC," *Official Journal of the European Union*, vol. L 119, pp. 1–88, 2016.
- [6] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in Google Street View," in *2009 IEEE 12th International Conference on Computer Vision*, Kyoto: IEEE, 2009, pp. 2373–2380. doi: 10.1109/ICCV.2009.5459413.
- [7] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001. doi: 10.1147/sj.403.0614.
- [8] S. L. Garfinkel, "De-identification of personal information," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., 2015. doi: 10.6028/NIST.IR.8053.
- [9] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 439–450, 2000. doi: 10.1145/335191.335438.
- [10] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, pp. 1–53, 2010. doi: 10.1145/1749603.1749605.
- [11] C. Graham, "Anonymisation: managing data protection risk code of practice," *Information Commissioner's Office*, p. 106, 2012.

- [12] A. W. Senior and S. Pankanti, "Privacy Protection and Face Recognition," in *Handbook of Face Recognition*, London: Springer London, 2011, pp. 671–691. doi: 10.1007/978-0-85729-932-1\_27.
- [13] S. Ribaric, A. Ariyaeinia, and N. Pavesic, "De-identification for privacy protection in multimedia content: A survey," *Signal Processing: Image Communication*, vol. 47, pp. 131–151, 2016. doi: 10.1016/j.image.2016.05.020.
- [14] K. El Emam, "Methods for the de-identification of electronic health records for genomic research," *Genome Medicine*, vol. 3, no. 4, p. 25, 2011. doi: 10.1186/gm239.
- [15] G. Zuccon, D. Kotzur, A. Nguyen, and A. Bergheim, "De-identification of health records using Anonym: effectiveness and robustness across datasets.," *Artificial intelligence in medicine*, vol. 61, no. 3, pp. 145–51, 2014. doi: 10.1016/j.artmed.2014.03.006.
- [16] K. El Emam and L. Arbuckle, *Anonymizing Health Data: Case Studies and Methods to Get You Started*. O'Reilly Media, Inc., 2013.
- [17] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002. doi: 10.1142/S0218488502001648.
- [18] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian, "L-Diversity: Privacy Beyond k-Anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 3–es, 2007. doi: 10.1145/1217299.1217302.
- [19] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, IEEE, 2007, pp. 106–115. doi: 10.1109/ICDE.2007.367856.
- [20] G. T. Duncan, M. Elliot, and J.-J. Salazar-González, *Statistical Confidentiality*, 1st ed. New York, NY: Springer New York, 2011. doi: 10.1007/978-1-4419-7802-8.
- [21] P. Agrawal and P. J. Narayanan, "Person De-Identification in Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 299–310, 2011. doi: 10.1109/TCSVT.2011.2105551.
- [22] D. Chen, Y. Chang, R. Yan, and J. Yang, "Protecting Personal Identification in Video," in *Protecting Privacy in Video Surveillance*, London: Springer London, 2009, pp. 115–128. doi: 10.1007/978-1-84882-301-3\_7.
- [23] A. Senior, "Privacy Protection in a Video Surveillance System," in *Protecting Privacy in Video Surveillance*, London: Springer London, 2009, pp. 35–47. doi: 10.1007/978-1-84882-301-3\_3.
- [24] E. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, 2005. doi: 10.1109/TKDE.2005.32.
- [25] P. Korshunov and T. Ebrahimi, "Using warping for privacy protection in video surveillance," in *2013 18th International Conference on Digital Signal Processing (DSP)*, IEEE, 2013, pp. 1–6. doi: 10.1109/ICDSP.2013.6622791.

- [26] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, IEEE Comput. Soc, 2001, pp. 511–518. doi: 10.1109/CVPR.2001.990517.
- [27] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, IEEE, 2005, pp. 886–893. doi: 10.1109/CVPR.2005.177.
- [28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. doi: 10.1109/LSP.2016.2603342. arXiv: 1604.02878.
- [29] R. Gross, E. Airoidi, B. Malin, and L. Sweeney, "Integrating Utility into Face De-identification," in *Proceedings of the 5th International Conference on Privacy Enhancing Technologies*, ser. PET'05, Berlin, Heidelberg: Springer-Verlag, 2005, pp. 227–242. doi: 10.1007/11767831\_15.
- [30] R. Gross, L. Sweeney, F. de la Torre, and S. Baker, "Model-Based Face De-Identification," in *Proceedings of Conference on Computer Vision and Pattern Recognition Workshop*, New York, USA: IEEE, 2006, pp. 161–161. doi: 10.1109/CVPRW.2006.125.
- [31] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative Face Completion," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 5892–5900. doi: 10.1109/CVPR.2017.624. arXiv: 1704.05838.
- [32] A. Bora, E. Price, and A. G. Dimakis, "AmbientGAN: Generative models from lossy measurements," in *International Conference on Learning Representations*, 2018.
- [33] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017. doi: 10.1145/3072959.3073659.
- [34] P. Phillips, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000. doi: 10.1109/34.879790.
- [35] A. Oganian and J. Domingo-ferrer, "On the complexity of optimal microaggregation for statistical disclosure control," *Statistical Journal of the United Nations Economic Commission for Europe*, vol. 4, pp. 345–353. 2001.
- [36] C.-C. Chang, Y.-C. Li, and W.-H. Huang, "TFRP: An efficient microaggregation algorithm for statistical disclosure control," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1866–1878, 2007. doi: 10.1016/j.jss.2007.02.014.
- [37] J. Domingo-Ferrer and V. Torra, "Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 195–212, 2005. doi: 10.1007/s10618-005-0007-5.
- [38] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*. London: Springer London, 2011. doi: 10.1007/978-0-85729-932-1.

- [39] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008. doi: 10.1016/j.patcog.2008.01.024.
- [40] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Constrained Local Neural Fields for Robust Facial Landmark Detection in the Wild," in *2013 IEEE International Conference on Computer Vision Workshops*, Sydney, Australia: IEEE, 2013, pp. 354–361. doi: 10.1109/ICCVW.2013.54.
- [41] T. F. Cootes, C. J. Taylor, *et al.*, "Statistical models of appearance for computer vision," Tech. Rep., 2004, p. 121.
- [42] I. Matthews and S. Baker, "Active Appearance Models Revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004. doi: 10.1023/B:VISI.0000029666.37597.d3.
- [43] R. Gross and L. Sweeney, "Towards Real-World Face De-Identification," in *2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, IEEE, 2007, pp. 1–8. doi: 10.1109/BTAS.2007.4401915.
- [44] R. Gross, L. Sweeney, F. de la Torre, and S. Baker, "Semi-supervised learning of multi-factor models for face de-identification," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Ieee, 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587369.
- [45] R. Gross, L. Sweeney, J. Cohn, F. de la Torre, and S. Baker, "Face De-identification," in *Protecting Privacy in Video Surveillance*, A. Senior, Ed., London: Springer London, 2009, pp. 129–146. doi: 10.1007/978-1-84882-301-3\_8.
- [46] L. Du, M. Yi, E. Blasch, and H. Ling, "GARP-face: Balancing privacy protection and utility preservation in face de-identification," in *IEEE International Joint Conference on Biometrics*, IEEE, 2014, pp. 1–8. doi: 10.1109/BTAS.2014.6996249.
- [47] A. Jourabloo, X. Yin, and X. Liu, "Attribute preserved face de-identification," in *2015 International Conference on Biometrics (ICB)*, IEEE, 2015, pp. 278–285. doi: 10.1109/ICB.2015.7139096.
- [48] B. Meden, R. C. Mallı, S. Fabijan, H. K. Ekenel, V. Štruc, and P. Peer, "Face deidentification with generative deep neural networks," *IET Signal Processing*, vol. 11, no. 9, pp. 1046–1054, 2017. doi: 10.1049/iet-spr.2017.0049.
- [49] B. Meden, Ž. Emeršič, V. Štruc, and P. Peer, "k-Same-Net: k-Anonymity with Generative Deep Neural Networks for Face Deidentification," *Entropy*, vol. 20, no. 2, p. 60, 2018. doi: 10.3390/e20010060.
- [50] B. Samarzija and S. Ribaric, "An approach to the de-identification of faces in different poses," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2014, pp. 1246–1251. doi: 10.1109/MIPRO.2014.6859758.
- [51] L. Meng, Z. Sun, A. Ariyaeenia, and K. L. Bennett, "Retaining expressions on de-identified faces," in *Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2014, pp. 1252–1257. doi: 10.1109/MIPRO.2014.6859759.

- [52] S. Mosaddegh, L. Simon, and F. Jurie, "Photorealistic Face De-Identification by Aggregating Donors' Face Components," in *Computer Vision – ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds., Cham: Springer International Publishing, 2015, pp. 159–174. doi: 10.1007/978-3-319-16811-1\_11.
- [53] G. Letournel, A. Bugeau, V.-T. Ta, and J.-P. Domenger, "Face de-identification with expressions preservation," in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 4366–4370. doi: 10.1109/ICIP.2015.7351631.
- [54] P. Chriskos, O. Zoidi, A. Tefas, and I. Pitas, "De-identifying facial images using singular value decomposition and projections," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3435–3468, 2017. doi: 10.1007/s11042-016-4069-8.
- [55] Z. Sun, L. Meng, A. Ariyaeinia, X. Duan, and Z.-H. Tan, "Privacy protection performance of De-identified face images with and without background," in *Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia: IEEE, 2016, pp. 1354–1359. doi: 10.1109/MIPRO.2016.7522350.
- [56] K. Brkić, T. Hrkać, Z. Kalafatić, and I. Sikirić, "Face, hairstyle and clothing colour de-identification in video sequences," *IET Signal Processing*, vol. 11, no. 9, pp. 1062–1068, 2017. doi: 10.1049/iet-spr.2017.0048.
- [57] G. Edwards, C. Taylor, and T. Cootes, "Interpreting face images using active appearance models," English, in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Comput. Soc, 1998, pp. 300–305. doi: 10.1109/AFGR.1998.670965.
- [58] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001. doi: 10.1109/34.927467.
- [59] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, New York, New York, USA: ACM Press, 1999, pp. 187–194. doi: 10.1145/311535.311556.
- [60] L. Chang and D. Y. Tsao, "The Code for Facial Identity in the Primate Brain," *Cell*, vol. 169, no. 6, pp. 1013–1028.e14, 2017. doi: 10.1016/j.cell.2017.05.011.
- [61] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995. doi: 10.1006/cviu.1995.1004.
- [62] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE.," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010. doi: 10.1016/j.imavis.2009.08.002.
- [63] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 Faces In-The-Wild Challenge: database and results," *Image and Vision Computing*, vol. 47, pp. 3–18, 2016. doi: 10.1016/j.imavis.2016.01.002.

- [64] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014, pp. 1701–1708. doi: 10.1109/CVPR.2014.220.
- [65] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 3025–3032. doi: 10.1109/CVPR.2013.389.
- [66] L. Meng and Z. Sun, "Face De-identification with perfect privacy protection," in *Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2014, pp. 1234–1239. doi: 10.1109/MIPRO.2014.6859756.
- [67] E. B. Roesch, L. Tamarit, L. Reveret, D. Grandjean, D. Sander, and K. R. Scherer, "FACSGen: A Tool to Synthesize Emotional Facial Expressions Through Systematic Manipulation of Facial Action Units," *Journal of Nonverbal Behavior*, vol. 35, no. 1, pp. 1–16, 2011. doi: 10.1007/s10919-010-0095-9.
- [68] M. Zhou, L. Liang, J. Sun, and Y. Wang, "AAM based face tracking with temporal matching and face segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 701–708. doi: 10.1109/CVPR.2010.5540146.
- [69] D. Cristinacce and T. F. Cootes, "Feature Detection and Tracking with Constrained Local Models," in *Proceedings of the British Machine Vision Conference 2006*, British Machine Vision Association, 2006, pp. 95.1–95.10. doi: 10.5244/C.20.95.
- [70] P. A. Tresadern, M. C. Ionita, and T. F. Cootes, "Real-Time Facial Feature Tracking on a Mobile Device," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 280–289, 2011. doi: 10.1007/s11263-011-0464-9.
- [71] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, IEEE, 2009, pp. 296–301. doi: 10.1109/AVSS.2009.58.
- [72] K. Ramnath, S. Koterba, J. Xiao, C. Hu, I. Matthews, S. Baker, J. Cohn, and T. Kanade, "Multi-view AAM fitting and construction," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 183–204, 2008.
- [73] Y. Cui and Z. Jin, "AAM-Based Face Modeling and Landmark Selection," in *2010 Third International Conference on Information and Computing*, vol. 2, IEEE, 2010, pp. 281–284. doi: 10.1109/ICIC.2010.166.
- [74] G. Tzimiropoulos and M. Pantic, "Optimization Problems for Fast AAM Fitting in-the-Wild," in *2013 IEEE International Conference on Computer Vision*, IEEE, 2013, pp. 593–600. doi: 10.1109/ICCV.2013.79.
- [75] T. Baltrusaitis, P. Robinson, and L. Morency, "3D Constrained Local Model for rigid and non-rigid facial tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2610–2617. doi: 10.1109/CVPR.2012.6247980.



- [76] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2879–2886. doi: 10.1109/CVPR.2012.6248014.
- [77] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic Descent Method: A Recurrent Process Applied for End-to-End Face Alignment," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 4177–4187. doi: 10.1109/CVPR.2016.453.
- [78] A. Bulat and G. Tzimiropoulos, "How Far are We from Solving the 2D & 3D Face Alignment Problem? (and a Dataset of 230,000 3D Facial Landmarks)," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 1021–1030. doi: 10.1109/ICCV.2017.116. arXiv: 1703.07332.
- [79] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor, "View-based active appearance models," *Image and Vision Computing*, vol. 20, no. 9-10, pp. 657–664, 2002. doi: 10.1016/S0262-8856(02)00055-0.
- [80] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 12, pp. 1080–1093, 2005. doi: 10.1016/j.imavis.2005.07.009.
- [81] P. Sauer, T. Cootes, and C. Taylor, "Accurate Regression Procedures for Active Appearance Models," in *Proceedings of the British Machine Vision Conference 2011*, British Machine Vision Association, 2011, pp. 30.1–30.11. doi: 10.5244/C.25.30.
- [82] T. F. Cootes, M. C. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, PART 7, vol. 7578 LNCS, 2012, pp. 278–291.
- [83] X. Xiong and F. De la Torre, "Supervised Descent Method and Its Applications to Face Alignment," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, USA: IEEE, 2013, pp. 532–539. doi: 10.1109/CVPR.2013.75.
- [84] E. Sangineto, "Pose and Expression Independent Facial Landmark Localization Using Dense-SURF and the Hausdorff Distance.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 624–638, 2012. doi: 10.1109/TPAMI.2012.87.
- [85] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive Facial Landmark Localization with Coarse-to-Fine Convolutional Network Cascade," in *2013 IEEE International Conference on Computer Vision Workshops*, IEEE, 2013, pp. 386–391. doi: 10.1109/ICCVW.2013.58.
- [86] H. V. Nguyen and L. Bai, "Cosine Similarity Metric Learning for Face Verification," in *ACCV 2010*, R. Kimmel, R. Klette, and A. Sugimoto, Eds., Springer Berlin Heidelberg, 2011, pp. 709–720. doi: 10.1007/978-3-642-19309-5\_55.
- [87] G. Shakhnarovich and B. Moghaddam, "Face Recognition in Subspaces," in *Handbook of Face Recognition*, London: Springer London, 2011, pp. 19–49. doi: 10.1007/978-0-85729-932-1\_2.
- [88] T. Kanade, "Picture Processing System by Computer Complex and Recognition of Human Faces," PhD thesis, Kyoto University, 1973.

- [89] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991. doi: 10.1162/jocn.1991.3.1.71.
- [90] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997. doi: 10.1109/34.598228.
- [91] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Face recognition using active appearance models," in *Computer Vision—ECCV'98*, Springer Berlin Heidelberg, 1998, pp. 581–595.
- [92] G. Edwards, A. Lanitis, C. Taylor, and T. Cootes, "Statistical models of face images — improving specificity," *Image and Vision Computing*, vol. 16, no. 3, pp. 203–211, 1998. doi: 10.1016/S0262-8856(97)00069-3.
- [93] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, 1993. doi: 10.1109/12.210173.
- [94] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997. doi: 10.1109/34.598235.
- [95] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996. doi: 10.1016/0031-3203(95)00067-4.
- [96] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002. doi: 10.1109/TPAMI.2002.1017623.
- [97] 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*, vol. 28, no. 12, pp. 2037–2041, 2006. doi: 10.1109/TPAMI.2006.244.
- [98] V. Ojansivu and J. Heikkilä, "Blur Insensitive Texture Classification Using Local Phase Quantization," in *Image and Signal Processing*, ser. Lecture Notes in Computer Science, A. Elmoataz, O. Lezoray, F. Nouboud, and D. Mammass, Eds., vol. 5099, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 236–243. doi: 10.1007/978-3-540-69905-7.
- [99] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, "Recognition of blurred faces using Local Phase Quantization," in *2008 19th International Conference on Pattern Recognition*, IEEE, 2008, pp. 1–4. doi: 10.1109/ICPR.2008.4761847.
- [100] S. Lawrence, C. Giles, Ah Chung Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997. doi: 10.1109/72.554195.
- [101] E. Zhou, Z. Cao, and Q. Yin, "Naive-Deep Face Recognition: Touching the Limit of LFW Benchmark or Not?," 2015. arXiv: 1501.04690.

- [102] Y. Sun, X. Wang, and X. Tang, "Deep Learning Face Representation by Joint Identification-Verification," in *NIPS*, Montreal, Canada, 2014, pp. 1988–1996. arXiv: 1406.4773.
- [103] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 2892–2900. doi: 10.1109/CVPR.2015.7298907. arXiv: 1412.1265.
- [104] Y. Sun, D. Liang, X. Wang, and X. Tang, "DeepID3: Face Recognition with Very Deep Neural Networks," 2015. arXiv: 1502.00873.
- [105] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 815–823. doi: 10.1109/CVPR.2015.7298682. arXiv: 1503.03832.
- [106] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," in *Computer Vision – ECCV 2016*, Springer, Cham, 2016, pp. 499–515. doi: 10.1007/978-3-319-46478-7\_31.
- [107] O. M. Parkhi, A. Vedaldi, A. Zisserman, A. Vedaldi, K. Lenc, M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, K. Lenc, *et al.*, "Deep face recognition," in *Proceedings of the British Machine Vision*, 2015.
- [108] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy, "Pose-Robust Face Recognition via Deep Residual Equivariant Mapping," 2018. arXiv: 1803.00839.
- [109] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto: Consulting Psychologists Press, 1978.
- [110] F. I. Parke, "Computer generated animation of faces," in *Proceedings of the ACM annual conference on - ACM'72*, vol. 1, New York, USA: ACM Press, 1972, pp. 451–457. doi: 10.1145/800193.569955.
- [111] S. Bouaziz, Y. Wang, and M. Pauly, "Online modeling for realtime facial animation," *ACM Transactions on Graphics*, vol. 32, no. 4, 40:1–40:9, 2013. doi: 10.1145/2461912.2461976.
- [112] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Transactions on Graphics*, vol. 33, no. 4, 43:1–43:10, 2014. doi: 10.1145/2601097.2601204.
- [113] S. Lucey, I. Matthews, Changbo Hu, Z. Ambadar, F. de la Torre, and J. Cohn, "AAM Derived Face Representations for Robust Facial Action Recognition," in *Proceedings of 7th International Conference on Automatic Face and Gesture Recognition*, vol. 2006, Southampton, UK: IEEE, 2006, pp. 155–162. doi: 10.1109/FGR.2006.17.
- [114] M. de la Hunty, A. Asthana, and R. Goecke, "Linear Facial Expression Transfer with Active Appearance Models," in *2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey: IEEE, 2010, pp. 3789–3792. doi: 10.1109/ICPR.2010.923.

- [115] B.-J. Theobald, I. Matthews, M. Mangini, J. R. Spies, T. R. Brick, J. F. Cohn, and S. M. Boker, "Mapping and manipulating facial expression.," *Language and speech*, vol. 52, no. Pt 2-3, pp. 369–386, 2009.
- [116] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobald, "Real-time expression transfer for facial reenactment," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–14, 2015. doi: 10.1145/2816795.2818056.
- [117] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, "The IMM Face Database: An Annotated Dataset of 240 Face Images," Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Kgs. Lyngby, Tech. Rep., 2004.
- [118] W. K. Pratt, *Digital Image Processing*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007. doi: 10.1002/0470097434.
- [119] Z. Sun, L. Meng, and A. Ariyaeinia, "Distinguishable de-identified faces," in *Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 04, Ljubljana, Slovenia: IEEE, 2015, pp. 1–6. doi: 10.1109/FG.2015.7285019.
- [120] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: automatically replacing faces in photographs," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 39:1–39:8, 2008. doi: 10.1145/1360612.1360638.
- [121] L. Impett, P. Robinson, and T. Baltrusaitis, "A facial affect mapping engine," in *Proceedings of the 19th international conference on Intelligent User Interfaces*, New York, USA: ACM Press, 2014, pp. 33–36. doi: 10.1145/2559184.2559203.
- [122] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, 2003. doi: 10.1145/882262.882269.
- [123] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 533–540, 2006. doi: 10.1145/1141911.1141920.
- [124] P. J. Phillips and A. J. O'Toole, "Comparison of human and computer performance across face recognition experiments," *Image and Vision Computing*, vol. 32, no. 1, pp. 74–85, 2014. doi: 10.1016/j.imavis.2013.12.002.
- [125] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable Visual Attributes for Face Verification and Image Search.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 10, pp. 1962–77, 2011. doi: 10.1109/TPAMI.2011.48.
- [126] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward Open Set Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013. doi: 10.1109/TPAMI.2012.256.
- [127] P. Ekman, "Facial expression and emotion.," *American Psychologist*, vol. 48, no. 4, pp. 384–392, 1993. doi: 10.1037/0003-066X.48.4.384.
- [128] J. Cohn, Z. Ambadar, and P. Ekman, "Observer-based measurement of facial expression with the Facial Action Coding System," in *The handbook of emotion elicitation and assessment*, J. A. Coan and J. J. B. Allen, Eds., Oxford University Press, 2006, pp. 203–221.

- [129] P. Lucey, J. F. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K. M. Prkachin, "Automatically Detecting Pain in Video Through Facial Action Units," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 664–674, 2011. doi: 10.1109/TSMCB.2010.2082525.
- [130] F. De la Torre, Wen-Sheng Chu, Xuehan Xiong, F. Vicente, Xiaoyu Ding, and J. Cohn, "IntraFace," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana: IEEE, 2015, pp. 1–8. doi: 10.1109/FG.2015.7163082.
- [131] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "OpenFace: An open source facial behavior analysis toolkit," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, USA: IEEE, 2016, pp. 1–10. doi: 10.1109/WACV.2016.7477553.
- [132] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, "Painful data: The UNBC-McMaster shoulder pain expression archive database," in *Proceedings of IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, Santa Barbara, USA: IEEE, 2011, pp. 57–64. doi: 10.1109/FG.2011.5771462.
- [133] X. Huang, J. Gao, S.-c. S. Cheung, and R. Yang, "Manifold Estimation in View-Based Feature Space for Face Synthesis across Poses," in *Computer Vision – ACCV 2009: 9th Asian Conference on Computer Vision, Xi'an, September 23-27, 2009, Revised Selected Papers, Part I*, H. Zha, R.-i. Taniguchi, and S. Maybank, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 37–47. doi: 10.1007/978-3-642-12307-8\_4.
- [134] C. Sagonas, Y. Panagakis, S. Zafeiriou, and M. Pantic, "Robust Statistical Face Frontalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 3871–3879. doi: 10.1109/ICCV.2015.441.
- [135] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," in *Advances in Neural Information Processing Systems 27*, 2014. arXiv: 1406.2661.
- [136] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *ICLR 2016*, 2015. arXiv: 1511.06434.
- [137] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," 2016. arXiv: 1606.03498.
- [138] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," 2014. arXiv: 1411.1784.
- [139] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis With Auxiliary Classifier GANs," 2016. arXiv: 1610.09585.
- [140] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," 2015. arXiv: 1506.05751.
- [141] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based Generative Adversarial Network," in *ICLR 2017*, 2016. arXiv: 1609.03126.

- [142] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, "Least Squares Generative Adversarial Networks," in *Proceeding of the 2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. arXiv: 1611.04076.
- [143] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. arXiv: 1701.07875.
- [144] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," in *NIPS*, 2017. arXiv: 1704.00028.
- [145] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary Equilibrium Generative Adversarial Networks," 2017. arXiv: 1703.10717.
- [146] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in *International Conference on Learning Representations*, 2017. arXiv: 1710.10196.
- [147] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *ICML 2016*, 2015. arXiv: 1512.09300.
- [148] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible Conditional GANs for image editing," in *NIPS Workshop on Adversarial Training*, 2016. arXiv: 1611.06355.
- [149] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face Aging With Conditional Generative Adversarial Networks," 2017. arXiv: 1702.01983.
- [150] R. Huang, S. Zhang, T. Li, and R. He, "Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 2458–2467. doi: 10.1109/ICCV.2017.267. arXiv: 1704.04086.
- [151] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 3730–3738. doi: 10.1109/ICCV.2015.425. arXiv: 1411.7766.
- [152] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007.
- [153] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th International Conference on Computer Vision*, IEEE, 2009, pp. 365–372. doi: 10.1109/ICCV.2009.5459250.
- [154] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," 2016. arXiv: 1602.07261.
- [155] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition," 2016. arXiv: 1607.08221.