Tattoos in forensics : retrieval, detection and synthesis

Xu, Xingpeng

2021

Xu, X. (2021). Tattoos in forensics : retrieval, detection and synthesis. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/151535>

<https://hdl.handle.net/10356/151535>

<https://doi.org/10.32657/10356/151535>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 20:48:29 SGT

TATTOOS IN FORENSICS: RETRIEVAL, DETECTION AND SYNTHESIS

Xu Xingpeng

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy (Ph.D)

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

17/12/2020

Xu Xingpeng

.....
Date

.....
Xu Xingpeng

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

19/12/2020

.....
Date

.....
Kong Wai Kin Adams

Authorship Attribution Statement

This thesis contains material from 2 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as X. Xu and A. W. K. Kong, "A geometric-based tattoo retrieval system," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 3019-3024.

The contributions of the co-authors are as follows:

- Assoc. Prof A. W. K. Kong provided the initial project direction.
- I wrote the drafts of the manuscript drafts. The manuscript was revised by Assoc. Prof A. W. K. Kong.
- I prepared all the images for the experiments.
- I implemented all the algorithms and performed all the experiments.

Chapter 5 is published as X. Xu, W. M. Matkowski and A. W. K. Kong, "A portrait photo-to-tattoo transform based on digital tattooing". Multimedia Tools and Applications, 2020, pp. 24367–24392.

The contributions of the co-authors are as follows:

- Assoc. Prof A. W. K. Kong provided the initial project direction.
- I wrote the drafts of the manuscript. The manuscript was revised together with Dr. Matkowski and Assoc. Prof A. W. K. Kong.
- I prepared the images for experiments. Dr. Matkowski assisted in data preparation.
- I implemented all the algorithms and performed all the experiments.

17/12/2020

Xu Xingpeng

.....
Date

.....
Xu Xingpeng

Acknowledgments

It has been my utmost privilege to be given this opportunity to pursue a Ph.D degree. I would like to express my sincere gratitude and appreciation to all the people who have contributed to my Ph.D study in many ways.

First and foremost, I would like to express my gratitude to my supervisor, Professor Kong Wai Kin Adams for his constructive advice and continual encouragements. Without his contribution of time, ideas and encouragements, this work would not have been possible. I am very thankful for his patient guidance and insightful suggestions. I would like to extend my appreciation to the members of my thesis advisory committee, Professor Hui Siu Cheung and Professor Mao Kezhi for their advice and encouragements.

I also want to thank all my friends in Cyber Security Laboratory (previously known as Forensics and Security Laboratory): Dr. Tang Chaoying, who cared me like an elder sister and gave me encouragements; Dr. Zhang Hengyi, a friend who shared happiness with me; Dr. Li Xiaojie, a diligent researcher that set an example to me; Dr. Liu Yanzhu, a talented researcher full of wonderful ideas and knowledge; Dr. Shitala Prasad, who helped a lot in my research in terms of theories and experiments. I owe a great deal of appreciation and gratitude to other members in our research team, Wojciech, Yanling and Wang Fan. I am so lucky and honored to have worked with those incredible young researchers.

I would also like to give my warm thanks to my beloved girlfriend, Dr. Wang Xueqin. She supports me all the way through my Ph.D study and she's always with me no matter the happy time or tough time. She gives me hope and strength so that I can carry on and keep fighting for my Ph.D study.

Last but not the least, I want to thank my parents for their love and support. They show me how to be a good man and live a meaningful life. Finally, I want to thank my two lovely cats who love and support me in their own special ways.

Contents

Acknowledgments	iv
List of Figures	ix
List of Tables	xvii
List of Publications	xviii
Abstract	xix
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives, Challenges and Solutions	3
1.3 Contributions	5
1.4 Organization of the Thesis	6
2 Background and Literature Review	7
2.1 Tattoos	7
2.2 Tattoo Retrieval	10
2.3 Tattoo Image Classification	17
2.4 Tattoo Detection	19
2.5 Scene Text Detection	22
2.6 Deep Object Detection	26
2.6.1 One Stage Detectors	27
2.6.2 Two Stage Detectors	29
2.7 Tattoo Image Synthesis	37
2.8 Summary	41

3	Geometry Based Tattoo Retrieval System	42
3.1	Full Body Human Images	43
3.1.1	Full Body Imaging System	43
3.1.2	Full Body Coordinate System	44
3.1.2.1	Joint Point Detection Algorithm	45
3.1.2.2	Full Body Coordinates Representation	51
3.2	GBTR System	52
3.2.1	Interface for Drawing Tattoo Sketch	53
3.2.2	Shape Extraction	54
3.2.3	Location Mapping	58
3.2.4	Tattoo Matching	59
3.3	GBTR Database	60
3.4	Experimental Results	61
3.4.1	Evaluation of Joint Points Detection	62
3.4.2	Evaluation of Patch-based Tattoo Detection	64
3.4.3	Evaluation of Tattoo Retrieval using the GBTR System	66
3.4.4	Comparisions of Tattoo Retrieval using Low-quality Tattoo Sketches and High-quality Tattoo Sketches	67
3.5	Summary	68
4	Text Tattoo Detection	70
4.1	Text Tattoo Dataset	73
4.2	Prior Knowledge-based Attention Mechanism	76
4.3	Text Tattoo Detection Network	80
4.3.1	Network Architectures	80
4.3.2	Variants of TTDN-DA	82
4.4	Implementation and Training Scheme	83
4.4.1	Training TTDN-DA	83
4.4.2	Training TTDN-DA Variants	85
4.5	Experimental Results	85
4.5.1	Evaluation Indices	86

4.5.2	Text Tattoo Detection with TTDN-DA and its Variants	86
4.5.3	Text Tattoo Detection without Attention	87
4.5.4	Text Tattoo Detection with General Tattoo Detector	90
4.5.5	Comparison with State-of-the-Art Object Detectors	91
4.5.6	Comparison with State-of-the-Art Scene Text Detectors	92
4.5.7	Evaluation of General Tattoo Detection	95
4.5.8	Evaluation of PKAM Variants	97
4.6	Summary	100
5	Portrait Tattoo Synthesis	101
5.1	Portrait Tattoos	102
5.2	A Tattoo Needle Model	103
5.3	The Digital Tattooing Algorithm	105
5.3.1	Portrait Photo Pre-processing	108
5.3.2	Shadow Simulation	111
5.3.3	Color Mapping	112
5.3.4	Digital Tattooing Based on the Tattoo Needle Model	114
5.3.5	Post-processing	117
5.4	Experimental Results	119
5.4.1	Tunable Parameters	120
5.4.2	Synthetic Portrait Tattoos	123
5.4.3	Synthetic Portrait Tattoos on 3D Skin Models	127
5.5	Synthetic Portrait Tattoo Stickers	127
5.6	Synthetic Text Tattoos	130
5.7	Summary	133
6	Conclusions and Future Work	136
6.1	Conclusions	136
6.1.1	Geometry based Tattoo Retrieval	136
6.1.2	Tattoo Dataset and Text Tattoo Dataset	137
6.1.3	Prior Knowledge-based Attention Mechanism and Text Tattoo De- tection	138

6.1.4	Tattoo Needle Model and Digital Tattooing	139
6.2	Future Work	140
	References	142

List of Figures

1.1	Sample images of biometric traits. The face image comes from CelebA [1] dataset.	2
1.2	Sample images of tattoos. From left to right: name, portrait, dates, religious, years in jail and gangster tattoo	3
1.3	Sample text tattoos show important personal information: (a, b, e and f) names, (e, f and g) dates and (h) gang.	4
1.4	Overall road map of this thesis.	6
2.1	Structure of human skin.	8
2.2	Modern tattooing procedure.	8
2.3	World oldest tattoo [2].	9
2.4	Tattoos with meaningful patterns and describable contents. (a) symbol of the Latin Kings gang, (b) five dots represents time done in prison, the four dots on the outside represent four walls, with the fifth on the inside representing the prisoner, (c) a cross, (d) face of Jesus and (e) the sun.	10
2.5	Tattoo-ID system [3].	12
2.6	Tattoo sketch samples.	15
2.7	Protocol for drawing tattoo sketches [4].	16
2.8	Tattoo detection network in [5].	22
2.9	Architecture of SegLink [6]. Each convolutional block (the gray box) predicts text on its corresponding feature layers. The predicted text was combined into whole words.	23
2.10	Scene text samples. First row: regular shape scene text. Second row: arbitrary shape scene text.	24
2.11	Pipeline of PSENet [7].	25

2.12	Two samples of Bezier curves [8].	27
2.13	Architecture of ABCNet [8].	27
2.14	Architecture of YOLO [9].	27
2.15	Architecture of SSD [10].	29
2.16	Architecture of Fast R-CNN [11].	30
2.17	Architecture of Mask R-CNN [12].	31
2.18	Architecture of R-FCN [13].	32
2.19	Architecture of RetinaNet [14].	34
2.20	The pipeline of SNIP training and inference [15].	34
2.21	Architecture of TridentNet [16].	36
2.22	Resultant images generated by LunaPic and PhotoFunia. (a) is an input real portrait photo, (b) is a synthetic tattoo portrait generated by LunaPic and (c) is a synthetic tattoo portrait generated by PhotoFunia.	38
2.23	A typical architecture of GAN [17].	39
2.24	Sample images from Kim et al.'s method [18]. (a) and (c) are raw images, (b) and (d) are corresponding converted stipple images.	40
3.1	The full-body imaging system [19] used in the GBTR system.	44
3.2	Six sequences of images taken from six poses. From (a) to (f): front, back, left front, right front, left back and right back.	45
3.3	Six full-body images generated by stitching the six sequences of images in Fig. 3.2. From (a) to (f): front, back, left front, right front, left back and right back.	46
3.4	The image coordinate system. The left-top corner of the image is the origin (0,0). x axis is the width of the image, and the y axis is the height of the image. The location of the dark pixel is (5,3).	46
3.5	(a) is an input shape on the model. (b) and (d) show the areas calculated by the image coordinates. (c) and (e) show the areas calculated by the full-body coordinate system.	47
3.6	Illustration of the preprocessing. (a) raw full-body image, (b) gray scale full-body image, (c) binarized full-body image and (d) boundary of the full-body.	48

3.7	Sample images of joint points on a full-body human image. (a) and (b) are full-body human image from two poses. (c) and (d) are the boundaries of the human body. Red dots are the detected joint points.	49
3.8	Illustration of the procedure to detect a joint point at the location with large curvature. The blue point is considered as the joint point.	50
3.9	(a)-(b) illustrate respectively the shoulder and ankle point detection in the front pose image. (c)-(g) illustrate respectively the shoulder, wrist, knee, ankle and elbow detection procedures in the side pose images. The blue dots are the joint points, the red dots are the end points of the curves around the joints and the green dot is the armpit.	52
3.10	A schematic diagram of the GBTR system.	53
3.11	The interface for the witness to draw the query tattoo boundary and the sample of points of sketch tattoo boundary.	54
3.12	A schematic diagram to illustrate the patch-based tattoo detection algorithm.	55
3.13	Illustration of the classification scheme that is used to separate tattoos and other skin characteristics, e.g., hair.	56
3.14	Samples tattoo image with boundaries (red line) extracted by the Snakes method [20].	57
3.15	Location mapping scheme.	59
3.16	(a) the query tattoo boundary. (b) the retrieved tattoos without sorting and (c) the retrieved tattoos after sorting. The images in (b) and (c) are the rank 1-7, from left to right.	61
3.17	(a) original full-body image, (b) image generated by flipping from (a), (c) image generated by stretching (a), (d) image generated by flipping from (c). Faces are covered.	62
3.18	Samples of full-body images with manually added tattoos.	63
3.19	Samples of successfully detected tattoos using patch-based tattoo detection method in the GBTR system.	65
3.20	GBTR system evaluation scheme. The solid line marks the standard low-quality sketch path and the dot line marks the high-quality sketch evaluation path.	65

3.21	Matching performance of the proposed method comparing to Hans method.	67
3.22	Comparison of the matching performance of the proposed method with guide image and without guide image.	68
4.1	Sample images in NTU Tattoo V2 dataset.	71
4.2	Sample text tattoos show important personal information: (a, b, e and f) names, (e, f and g) dates and (h) gang.	72
4.3	The 1 st and 2 nd rows: sample images of text tattoos. The 3 rd and 4 th rows: sample images of scene text.	73
4.4	Sample images of open access tattoo datasets.	74
4.5	The prior knowledge-based attention mechanism. I_t is a text tattoo image. f_t^R is the tattoo features or object features extracted by the tattoo detector or object detector, respectively. I_t' is the text tattoo image enhanced by PKAM. The highlighted block illustrates the architecture of spatial prior knowledge-based attention mechanism (spatial PKAM).	78
4.6	The architecture of channel PKAM.	79
4.7	The architecture of TTDN-DA. f_{oh} is the feature map of ObjectHumanNet, f_{tatt} is the feature map of TattooNet and f_{text} is the feature map of TextTattooNet. In training stage 1, TextTattooNet and TPKAM are disconnected from the entire network, the classification head and bounding box regression head are connected to the feature map of TattooNet. . . .	80
4.8	Details architecture of subnetworks.	80
4.9	Details architecture of ResNetUnit.	80
4.10	The architecture of TTDN-DA-V2. f_{tatt} is the feature map of TattooNet and f_{text} is the feature map of TextTattooNet. In training stage 1, TextTattooNet and TPKAM are disconnected from the entire network, the classification head and bounding box regression head are connected to the feature map of TattooNet.	82
4.11	The architecture of TTDN-DA-V3. The weights of first three blocks are loaded from a pretrained TattooNet of TTDN-DA-V2. f_{text} is the final feature map.	83

4.12	TTDN-DA text tattoo detection results. The red rectangle areas are enlarged on the right side of each image to show more details. The blue bounding boxes mark the detection results. Images are resized to fill in the page.	88
4.13	TTDN-DA text tattoo detection results. The last row shows some failed detections. The blue bounding boxes mark the detection results.	89
4.14	The architecture of non-Atten-TTDDN-V1 and non-Atten-TTDDN-V2. . . .	90
4.15	Architectures of (a) ResNet block [21], (b) ResNeXt block [22] and (c) HourGlass network [23].	92
4.16	Text tattoo detection results from close up images for comparisons. From top row to bottom row: TTDN-DA, TridentNet*, CSPNet*, DetectoRS*, Cascade R-CNN*, CenterNet*, RetinaNet* Mask R-CNN* and Faster R-CNN*. The blue bounding boxes mark the detection results.	93
4.17	Precision and recall curves of TTLN-DA, TridentNet* and ABCNet. TridentNet* and ABCNet are the best generic object detector and scene text detector, respectively on the NTU Tattoo V1 (Table 4.3).	94
4.18	Architectures of VGG-16 network [24].	96
4.19	Text tattoo detection results from close up images for comparisons. From top row to bottom row: TTDN-DA, EAST, PixelLink, SegLink, PAN, PSENet, SPCNet, TextFuseNet and ABCNet. The blue bounding boxes mark the detection results.	98
5.1	Portrait tattoo samples.	103
5.2	Tattoo needles for modern tattooing [25].	104
5.3	Digital tattoo needle masks with different configurations. First row (from left to right): RL-3-30, RL-5-30, RL-7-30, RL-9-30. Second row (from left to right): RS-3-30, RS-5-30, RS-7-30, RS-9-30. Third row (from left to right): FS-5-30, FL-7-30, M1-7-30, M2-13-30. R means the needles are organized in a round shape, F means the needles are organized in a flat shape. M means the needles are organized in a Magnum style. L means the needle is used to draw lines. S means the needle is used to draw shadows.	105

5.4	Comparison of high-quality portrait photos and portrait tattoos. (a)-(d) are high-quality portrait tattoos and (e)-(h) are high-quality portrait photos. (e) is a black and white photo of the person in (a). Note that the red in (c) is not color of tattoo ink. It is due to the skin slightly bruised from some blood leaking underneath the top layers of the skin.	106
5.5	Illustration of the noise in a portrait tattoo. (left) A high-quality tattoo and the patch from the white square in (right).	107
5.6	The schematic diagram of the proposed digital tattooing (DT) algorithm.	107
5.7	Illustration of portrait photo pre-processing. (a) an input portrait photo, (b) the grayscale image I_{gray} , (c) the sketch image I_{sketch} obtained from the XDoG method, (d) the sketch-and-edge image I_{se} , (e) the distance map D_{face} from (d), (f) the facial landmarks from (a), (g) the final distance map D_{final} from (e) and the facial landmark distance maps. (h) the final result I_{pre} from the portrait photo pre-processing. For better visualization, the distance maps in (e) and (g) are processed.	110
5.8	(a) is an input portrait photo I_{face} and (b) is its alpha-matte mask I_{α} . (c) is the corresponding $I_{\alpha w}$ with $\alpha = 0.03$ and $\beta = 1$. (d) is the pre-processed portrait photo I_{pre} of (a) and (e) is its I_{enh} with shadow effect ($\alpha = 0.03, \beta = 1$). (f) is a color map generated from (e).	113
5.9	(a) an input portrait tattoo I_{tatt} , (b) the grayscale image T_{gray} of I_{tatt} , (c) the result of histogram equalization of T_{gray} and (d) the largest component in the binary image obtained by Otsu threshold.	114
5.10	(a) and (c) synthetic portrait tattoos using 3RL and 5FL tattoo needles with different orientations, respectively (Fig. 5.3). (b) and (d) are enlarged patches from (a) and (c), respectively. (zoom in recommended)	116
5.11	A comparison of the results with and without post-processing: (a) a result without post-processing, (b) the zoom in the rectangle region in (a), (c) a result with post-processing, and (d) the zoom in the rectangle region in (c).	118

5.12	(a) The red rectangle in the image indicates the location where a user wants to tattoo a portrait on. (b) The cropped skin image and (c) its rotated version. (d) A synthetic portrait tattoo on (c) and (e) its version in the original orientation. (f) The synthetic portrait tattoo on the original image (a).	119
5.13	Grayscale image enhancement. 1 st row: grayscale portrait images. 2 nd row: synthetic portrait tattoo images.	120
5.14	Shadow around the face control. 1 st row: shadow function. 2 nd row: synthetic portrait tattoo images.	121
5.15	Needle control. (a): RL_3_35 for the whole portrait image. (b): RL_3_35 for eye, nose and mouth, RS_7_35 for the rest area on the portrait image. (c): RL_3_35 for the eyes, RL_5_40 for the nose, RS_5_40 for the mouth and RS_7_35 for the rest area on the portrait image.	122
5.16	Digital tattooing weight control. Synthetic portrait tattoo images with (a) $\delta_s = 2$ and $\delta_p = 3$, (b) $\delta_s = 5$ and $\delta_p = 5$, and (c) $\delta_s = 3$ and $\delta_p = 1.5$	122
5.17	Step size control. Synthetic portrait tattoo images with (a) $W_{step} = 3$ (default value), (b) $W_{step} = 2$, and (c) $W_{step} = 4$	123
5.18	Portrait tattoos generated by the DT algorithm. The first three columns show input portrait photos, skin images and skin images with the synthetic tattoos. The last two columns show the close up synthetic tattoos and the input reference tattoos.	124
5.19	Comparison results between the real portrait tattoos (1 st row) and the synthetic portrait tattoos generated by the DT algorithm (2 nd row). . . .	125
5.20	From left to right: input portrait photos, input reference tattoos, synthetic portrait tattoos generated by Johnson et al.'s, Huang and Belongie's, and Chen and Schmidt's methods, close-up image of synthetic portrait tattoos generated by the proposed DT algorithm, the synthetic portrait tattoos generated by the proposed DT algorithm in original images.	126
5.21	(a) a 3D skin image, (b) a 2D image unfolded from (a), (c) a synthetic portrait tattoo on the 2D image. (d)-(f) the synthetic portrait tattoo on the 3D image from three different viewpoints.	128

5.22 Synthetic portrait tattoos on 3D models.	129
5.23 Some temporary tattoos.	129
5.24 Synthetic portrait tattoos stickers.	130
5.25 Comparison of portrait tattoos (1 st row) and text tattoos (2 nd row). . . .	131
5.26 Using DT for synthesizing text tattoos.	131
5.27 Samples of curved text tattoos.	132
5.28 Samples of synthetic text images.	132
5.29 Samples of synthetic text tattoo images.	133
5.30 Samples of synthetic text tattoo on inputted skin images.	134

List of Tables

2.1	NIST tattoo classes.	11
2.2	Evaluation results of the AlexNet [26] trained and tested on different dataset.	19
2.3	Evaluation of the network performance on different patch sizes.	21
3.1	Average Euclidean distances between two types of joint points (front and back poses). “Auto” means the joint detection method, “Man.” means manual marking.	64
3.2	Average Euclidean distances between two types of joint points (side poses). “Auto” means the joint detection method, “Man.” means manual marking.	64
4.1	Comparisons of different tattoo datasets.	74
4.2	The structure of the NTU Tattoo V2 dataset	76
4.3	Experimental results on text tattoo detection. * mark means that the networks is trained on the NTU Tattoo V2 first and finetuned on the NTU Text Tattoo V1.	95
4.4	The differences between the evaluations of tatt-TridentNet-V1, tatt-TridentNet-V2, TridentNet and TridentNet*.	96
4.5	Compare with state-of-the-art scene text detectors.	97
4.6	Experimental results on the general tattoo detection. TTDN-DA and TTDN-DA-V2 use the outputs of TattooNet as the general tattoo detection results.	99
4.7	Text tattoo detection results from different PKAM modules. Experiments were conducted on TTDN-DA-V2.	99

List of Publications

International Conferences

- H. Yi, P. Yu, **X. Xu** and A. W. K. Kong, "The impact of tattoo segmentation on the performance of tattoo matching," 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Dhaka, 2015, pp. 43-46.
- **X. Xu** and A. W. K. Kong, "A geometric-based tattoo retrieval system," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 3019-3024.
- Qingyong Xu, S. Ghosh, **X. Xu**, Yi Huang and A. W. K. Kong, "Tattoo detection based on CNN and remarks on the NIST database," 2016 International Conference on Biometrics (ICB), Halmstad, 2016, pp. 1-7.

International Journals

- **X. Xu**, W. M. Matkowski and A. W. K. Kong, "A portrait photo-to-tattoo transform based on digital tattooing". Multimedia Tools and Applications (2020).
- **X. Xu**, S. Prasad, K. Cheng, A. W. K. Kong, "Using Double Attention for Text Tattoo Detection". IEEE Transaction on Biometrics, Behavior, and Identity Science (2020), submitted.

Abstract

A wide variety of biometric systems have been developed for forensic investigation based on biometric traits like fingerprint, face and iris, etc. Using those biometric traits as evidence in forensic investigation can be very challenging because they are not always visible. In the cases that those biometric traits are not available, tattoos provide us another opportunity to identify the criminals or the victims during the forensic investigation.

Tattoos have been widely used by law enforcement agencies for forensic investigation, such as criminal and victim identification. Tattoo retrieval and tattoo detection are two important use cases of tattoos in forensic investigation. Many works have been done in these areas. However, there are still some unsolved problems that limit the performance of these works. For example, most tattoo retrieval systems require a query tattoo image which is not always available; tattoo detectors suffer from the lack of a large scale tattoo dataset and their performance is limited by the architectures of the detection networks. In this research, we proposed some solutions to those problems.

First, we developed a novel tattoo retrieval system which uses geometry information of tattoos to perform tattoo searching. This geometry-based tattoo retrieval (GBTR) system only requires a simple tattoo sketch as query and avoids the requirement of a query tattoo image or a tattoo sketch with rich details. The location and shape information of the tattoo sketch are extracted first. Then, a location mapping procedure based on full body coordinates finds all the images in the searching database that have tattoos in the corresponding location. The tattoo shapes are extracted via Snake algorithm. Coherent Point Drift (CPD) algorithm is then employed to match the shape of tattoo sketch and the shapes of tattoos in the searching database. The retrieved tattoos are sorted based on the matching scores. To evaluate the GBTR system, we established a

full body human image database. All the images in the database contain at least one tattoo. The evaluation results showed that GBTR system achieves a promising tattoo retrieval accuracy.

Second, we designed a prior knowledge-based attention mechanism (PKAM) to improve the performance of text tattoo detection networks. We concentrate on text tattoo detection because text tattoos provide vital clues in forensic investigation such as names and important dates. A text tattoo detection network with double PKAMs (TTDN-DA) was proposed. The first PKAM uses the features of human instances to enhance the generic tattoo images, and the second PKAM uses the features of general tattoos to enhance the text tattoo images. Two variants of TTDN-DA, TTDN-DA-V2 and TTDN-DA-V3, were also proposed for handling different training and deploying scenarios. To train and evaluate the proposed text tattoo detection networks, we established NTU Tattoo V2 dataset and NTU Text Tattoo V1 dataset. The evaluation results show that PKAM can improve the text tattoo detection accuracy significantly.

A large tattoo dataset is essential for training robust tattoo retrieval systems and tattoo detectors, therefore, we proposed a digital tattooing (DT) approach to synthesize tattoo images. We focus on synthesizing portrait tattoo images because portrait photos contain sufficient texture details for us to evaluate the performance of the DT algorithm. DT algorithm takes a portrait photo, a real portrait tattoo image and a skin image as input. The portrait tattoo image is used as a reference. DT algorithm extracts the facial landmarks of the portrait tattoo image and enhances it by re-weighting the landmark textures and adding shadow effect, etc. Then it calculates a color mapping from reference portrait tattoo image to the enhanced portrait photo and uses a novel tattoo needle model to simulate the physical procedure of tattooing. A set of parameters are provided for controlling the visual style of the synthetic portrait tattoo images. The experimental results showed that, comparing with other image synthesizing methods, the DT algorithm generates more realistic portrait tattoos.

In this research, we provided new solutions to tattoo retrieval and tattoo detection task and showed their potential to address the unsolved problems of existing tattoo retrieval systems and tattoo detectors. Furthermore, we also proposed a tattoo synthesizing approach which can be very useful in generating large amount of tattoo images for training tattoo retrieval systems and tattoo detectors.

Chapter 1

Introduction

In this chapter, we will give an overview of the background, motivation and challenges related to the works presented in this thesis. In section 1.1, the background and motivation are presented. In section 1.2, the objectives and challenges are presented. In section 1.3, all the contributions made in this thesis are listed. Section 1.4 presents the complete structure of this thesis.

1.1 Background and Motivation

Forensic science is the application of science and scientific methods to the judicial system during the criminal investigation. One important task of criminal investigation is recognizing individuals, like victims or suspects. A wide variety of biometric systems have been developed for automatic recognition of individuals based on their anatomical or/and behavioral biometric traits like fingerprint, face, iris, retina, palmprint, voice, gait and signature [27]. Among these biometric traits, the fingerprint is one of the most popular and successful traits with a long history and it has been widely used in forensic investigations as a piece of biometric evidence. In case that fingerprints cannot be used to identify a person with sufficient confidence, the FBI developed the Next Generation Identification (NGI) system [28] to utilize additional biometric traits such as face, palmprint and iris. Besides the primary biometric traits such as fingerprint, palmprint, face and iris, soft biometric traits such as a person's height, weight, gender, sex, scars, marks and tattoos also provide important information in recognizing individuals, especially when primary

Fig. 1.1: Sample images of biometric traits. The face image comes from CelebA [1] dataset.

biometric traits are either corrupted or no longer available. Fig. 1.1 shows some sample images of biometric traits.

Scars, marks and tattoos (SMT) are imprints on the skin that provide more discriminative information than other soft biometric traits to identify a person. Scars and marks provide discriminative information only when they are available in the form of digital images or videos captured by the surveillance camera. Tattoos, on the other hand, are still able to provide descriptive information when the tattoo images are not available [29]. Furthermore, tattoos carry rich personal information, e.g., names of family members, important dates, portraits of loved ones, gangster groups, religious beliefs, years spent in jail and the appearance on different body sites with different sizes, etc (see Fig. 1.2). They are usually not too small and often have describable contents such as words, symbols, meaningful patterns with particular shapes and objects. Therefore, they can be easily observed, remembered and described by witnesses. Because of these characteristics, tattoos have been widely used by law enforcement agencies all around the world. The New York State Police used tattoos to identify the victims of the September 11 attacks [30]. The Royal Canadian Mounted Police (RCMP) collected various information including tattoos to match with data from missing people of the tsunami in 2004 [31].

The utilization of tattoo in forensic investigation includes tattoo retrieval, tattoo image classification, tattoo detection, etc. Many works have been done in these areas.

Fig. 1.2: Sample images of tattoos. From left to right: name, portrait, dates, religious, years in jail and gangster tattoo

Currently, popular tattoo retrieval systems [3, 32–34] are based on content-based image retrieval (CBIR) approaches. Most of them need query images to perform the tattoo matching. However, query tattoo images are not always available. Sketch-based tattoo matching method [4] requires a detailed tattoo sketch instead of a query image, but it is difficult for the witness to draw the tattoo sketch with rich details since most tattoos have complex patterns and multiple colors. Some works [26, 35–38] focus on tattoo image classification and tattoo detection. However, they either suffer from the lack of large scale tattoo databases or their performance is limited by the architectures of the detection network. Therefore, to make tattoo serve the forensic investigation better, those problems need to be solved.

1.2 Objectives, Challenges and Solutions

The problems in tattoo retrieval, tattoo image classification and tattoo detection have been existing for many years. With the development of computer vision and artificial intelligence, solutions to those problems become available.

To address the problems in the currently existing tattoo retrieval systems, we need to develop a new tattoo retrieval mechanism that does not need query images and releases the witness from the burden of providing detailed tattoo sketches. This is a challenging task because the performance of tattoo retrieval depends on the features extracted from tattoo images, and it is hard to extract sufficient features without providing query images or detailed tattoo sketches. Works in [3] and [34] showed that the performance of tattoo retrieval improved significantly with more features. Therefore, in the cases that query images are not available or/and witness can not provide a detailed tattoo sketch, how to obtain sufficient features from a simple tattoo sketch is essential. In this research,

Fig. 1.3: Sample text tattoos show important personal information: (a, b, e and f) names, (e, f and g) dates and (h) gang.

we investigate the potential of using the boundary of a tattoo sketch to perform tattoo retrieval and establish tattoo databases with accurate shape and location information for effective tattoo retrieval.

Current tattoo detectors [5, 36, 39] utilize convolutional features extracted from the entire tattoo image or tattoo image patches to train the detection network. These detectors all focus on general tattoo detection. Text tattoo, as a subcategory of tattoos, is especially important in forensic investigation comparing with other generic tattoo patterns, because text tattoo can provide vital clues such as names of criminals, names of their relatives, their gang groups, important dates (e.g., wedding date and date of birth) and other personal information as shown in Fig. 1.3. Therefore, detecting text tattoos can help law enforcement agencies to narrow down the suspect list quickly. So far there is no work especially focusing on text tattoo detection. To address the text tattoo detection problem and further improve the generic tattoo detection performance, new tattoo detectors need to be developed that can handle text tattoo detection and in the meanwhile achieve better general tattoo detection performance.

To achieve the optimal performance of tattoo detection and train a detector that focuses on text tattoo detection, a large scale tattoo dataset for tattoo detection and

text tattoo detection is essential. However, the existing public databases [5, 26, 40, 41] for tattoo detection are small and none of them is for text tattoo detection task. Therefore, in this research, we introduce a new version of NTU Tattoo dataset, called NTU Tattoo V2, which is the largest dataset for tattoo detection. We also introduce NTU Text Tattoo V1, the first and only dataset for text tattoo detection task.

Another approach to solving the dataset problem is generating synthetic tattoo images using image synthesizing techniques. However, tattoos are artworks composed by tattooists, and each tattooist has different understandings of tattoo design. Therefore, it is hard and challenging to find a universal method to generate a large number of synthetic tattoo images. Although there are some works [17, 42, 43] that focus on generating images or/and transferring image styles, none of them are designed for synthesizing tattoo images. Synthetic tattoo images generated by directly applying those methods lose a lot of detailed information. To generate synthetic tattoo images with better quality, we introduce Digital Tattooing (DT), a tattoo image synthesizing method that simulates the real tattooing procedure physically, and we generate synthetic portrait tattoo images using DT to evaluate the performance of our DT algorithm.

1.3 Contributions

The main contributions of this thesis can be summarized as follows:

- In section 3.1, a coordinate system based on the full body human image is designed. Full body coordinate system is the foundation component of the geometry based tattoo retrieval system.
- In section 3.2, the geometry based tattoo retrieval (GBTR) system is presented. GBTR uses full body coordinate system to locate tattoos, uses an active contour model to extract the shape information of a tattoo and uses a point set registration algorithm to calculate the matching scores.
- In section 4.1, a new tattoo dataset named NTU Tattoo V2 is presented which is the largest tattoo dataset at the moment. A text tattoo dataset named NTU Text Tattoo V1 is presented as well, which is the first and only text tattoo dataset.

Fig. 1.4: Overall road map of this thesis.

- In section 4.2, a prior knowledge-based attention mechanism (PKAM) is introduced. PKAM is a novel attention mechanism that uses the relationship among objects to enhance the detection performance.
- In section 4.3, a text tattoo detection network with double attention (TTDN-DA) is proposed. Another two variants of TTDN-DA are presented to meet the requirement of different training scenarios.
- In section 5.2, a tattoo needle model is proposed, which is a simulation of real tattoo needle configurations.
- In section 5.3, a digital tattooing (DT) algorithm is presented. DT uses tattoo needle model to simulate the physical procedure of real tattooing. DT can synthesize tattoos on both 2D skin image and 3D skin image. DT can also synthesize tattoo stickers and text tattoo images with proper modifications.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 gives a literature review on tattoo retrieval, tattoo image classification, tattoo detection and tattoo image synthesis. Chapter 3 presents the geometry based tattoo retrieval system. Chapter 4 presents a text tattoo detection network. Chapter 5 presents the Digital Tattooing (DT) algorithm. Chapter 6 concludes this thesis and offers the future works. Fig. 1.4 shows the overall road map of this thesis.

Chapter 2

Background and Literature Review

In this chapter, an overview of the background and previous works related to this thesis is provided. In section 2.1, a short introduction to tattoos is given. In section 2.2, some tattoo retrieval methods are reviewed. In section 2.3, the definitions of tattoo image classification and tattoo detection are clarified, and then some tattoo image classification methods are reviewed. In section 2.4, tattoo detection methods are reviewed, followed by some scene text detection methods in section 2.5. As most tattoo detectors and scene text detectors are based on deep object detection methods, some deep object detection methods are reviewed in section 2.6. Finally, the tattoo image synthesizing methods are reviewed in section 2.7.

2.1 Tattoos

Tattoo is a form of body modification where a design is made by inserting ink, dyes and pigments, either indelible or temporary into the dermis layer [44] (see Fig. 2.1) of the skin to change the pigment. The procedure of making tattoos is called tattooing. Traditionally, tattoos are often created by cutting designs into the skin and rubbing the ink, ashes or pigment into the wounds. Modern tattooists use a tattoo machine most of the time and often have procedures and accessories to reduce the risk to human health. Tattoo machines insert ink into the skin via a single needle or a group of needles that are attached to an oscillating unit which repeatedly drives the needles in and out of the skin, usually 80 to 150 times a second. Fig. 2.2 demonstrates the procedure of modern tattooing using a tattoo machine.

Fig. 2.1: Structure of human skin.

Fig. 2.2: Modern tattooing procedure.

Tattoos have been part of human culture for thousands of years. The first tattoo that has ever been recorded can be dated back to 3250 B.C. [2] as shown in Fig. 2.3. Since the 1970s, tattoos have become a mainstream part of Western fashion, common among both genders, to all economic classes and to age groups from the later teen years

Fig. 2.3: World oldest tattoo [2].

to middle age. Tattoos have undergone “dramatic redefinition” and have shifted from a form of deviance to an acceptable form of self-expression [45]. Today, people choose to be tattooed for artistic, cosmetic, sentimental, memorial and religious reasons, and to symbolize their belonging to or identification with particular groups, including criminal gangs. A telephone survey conducted by Pew Research [46] in September 2006 reported that 36% of Americans between 18 to 25 years old had a tattoo. This percentage changed to 40% and 10% of those between 26-40 and 41-64 years old, respectively. The ratio is even higher among prisoners. According to the statistics [47], 85% of prisoners under 35 have tattoos and 75% of the inmates who are re-incarcerated have tattoos.

Comparing to other soft-biometric traits, tattoos usually carry rich personal information, describable contents and meaningful patterns (see Fig. 2.4) which are easy for the witness to remember and describe. Based on these facts, law enforcement agencies pay more attention to tattoos for forensic investigation.

Fig. 2.4: Tattoos with meaningful patterns and describable contents. (a) symbol of the Latin Kings gang, (b) five dots represents time done in prison, the four dots on the outside represent four walls, with the fifth on the inside representing the prisoner, (c) a cross, (d) face of Jesus and (e) the sun.

2.2 Tattoo Retrieval

Tattoo retrieval is one of the most frequent use cases of tattoos in forensic investigation. Currently, most tattoo retrieval systems used in law enforcement agencies are based on text-based image retrieval (TBIR) techniques. Law enforcement agencies collect and annotate the tattoo images for identifying victims and suspects. To increase the annotation consistency, the U.S. National Institute of Standard and Technology (NIST) established a standard [29] which defines eight major class labels and around 80 sub-class labels (see Table 2.1). Note that not all police force employs this standard. Some police forces do not use any standard in the annotation. The tattoo retrieval of a typical text annotated tattoo database is performed by matching class labels of query and database tattoos. This kind of tattoo retrieval has the following limitations:

- Semantic information of tattoo images can not be captured by class labels.
- Manually annotating and maintaining millions of tattoo images are time-consuming and error-prone.
- Newly designed tattoos and tattoos with multiple objects are hard to be classified using standard class labels.
- Tattoo patterns have large intra-class variability.

Table 2.1: NIST tattoo classes.

Class	Subclasses
Human Forms and Features	Male Face, Female Face, Abstract Face, Male Body, Female Body, Abstract Body, Roles, Sports Figures, Male Body Parts, Female Body Parts, Abstract Body Parts, Skulls, Miscellaneous Human Forms
Animals and Animal Features	Cats & Cat Heads, Dogs & Dog Heads, Other Domestic Animals, Vicious Animals, Horses, Other Wild Animals, Snakes, Dragons, Birds, Insects, Abstract Animals, Animal Parts, Miscellaneous Animal Forms
Plants	Narcotics, Red Flowers, Blue Flowers, Yellow Flowers, Drawings of Flowers, Rose, Tulip, Lily, Miscellaneous Plants
Flags	American Flag, State Flag, Nazi Flag, Confederate Flag, British Flag, Miscellaneous Flags
Objects	Fire, Weapons, Airplanes, Vessels, Trains, Vehicles, Mythical, Sporting Objects, Water & Nature Scenes, Miscellaneous Objects
Abstractions	Figure, Sleeve, Bracelet, Anklet, Necklace, Shirt, Body Band, Head Band, Miscellaneous Abstract
Insignias & Symbols	National Symbols, Political Symbols, Military Symbols, Fraternal Symbols, Professional Symbols, Gang Symbols, Miscellaneous Symbols
Other Images	Wording, Freeform Drawings, Miscellaneous Images

These limitations of text-based tattoo retrieval have motivated the development of content-based image retrieval (CBIR) techniques to improve tattoo search efficiency and accuracy. Jain et al. [3] proposed a CBIR system called Tattoo-ID (see Fig. 2.5) for automatic matching and retrieval of tattoos. Tattoo-ID is known as the first content-based tattoo image retrieval system. The database used in Tattoo-ID contains 2,157 tattoo images downloaded from the internet belonging to eight major classes and around 80 sub-classes based on NIST standard [29]. For each tattoo image, 20 transformed images are generated as follows: two different intensities of blurring, additive noise and illumination, six different aspect ratios, four different rotations and four different color changes. This results in a total number of 45,297 tattoo images. Images are first segmented using edge-based operations and morphological operations in the preprocessing

Fig. 2.5: Tattoo-ID system [3].

procedure. Features of three low-level image attributes (color, shape and texture) are extracted from the segmented tattoos. A similarity score is computed for each attribute separately. An overall matching score between two images is calculated as the sum of similarity scores of individual attributes. Users select a set of tattoo images that are relevant to the query. The Tattoo-ID system computes the similarity of every image in the database to each of the relevant images and retrieves the tattoo images based on the updated similarity scores. Users can iteratively refine the retrieved images until all the matched tattoo images are found.

Although Tattoo-ID showed promising results, its performance was limited by the predefined similarity measure without appropriately weighting different features. Lee et al. [32] further improved the Tattoo-ID performance by employing rank-based distance metric learning. Rank-based distance metric learning is proposed to address the problem

when the input space is heterogeneous and the distance in must-link constraints may vary significantly from one query to another. Instead of requiring the distance of any must-link constraint to be smaller than that of a cannot-link constraint, the rank-based distance metric learning compares the distances of pairwise constraints generated by the same query. By employing rank-based distance metric learning, the rank-1 tattoo retrieval accuracy is improved by 6% comparing to the constraint-based approach.

The segmentation in Tattoo-ID is based on an edge-based approach and a morphological approach. Acton and Rossi [33] proposed an activate contour segmentation approach to capture tattoo boundaries more accurately. Skin detection is employed to remove the skin area within a tattoo pattern. The retrieval precision (average correct matches retrieved for a given query size) of query size one is improved by 17.8% comparing to Tattoo-ID. The features of three low-level image attributes (color, shape and texture) in Tattoo-ID are global features. To further improve the retrieval precision, Acton and Rossi proposed glocal image features which are local features that embed global information. A glocal feature contains edge and color information within a pair of 2D histograms that record frequency of occurrence in rings of exponentially increasing size, centered at the detected local feature. Euclidean distance is used as the measurement of matching between two features.

One problem in those approaches mentioned above is that the tattoo retrieval precision is limited by the handcraft global features. Jain et al. [34] implemented a CBIR system for tattoo images that is based on local image descriptors called SIFT keypoints [48]. SIFT, also known as scale-invariant feature transform, is a feature detector that can detect and describe local features in images. Jain et al.'s work also utilized side information of tattoo images, e.g., location of the tattoo on the body and tattoo class labels, to reduce the computational cost of retrieval. To reduce the number of false matches, a geometric constraint between the SIFT keypoints in two images is also introduced. According to their experimental results, the rank-1 accuracy with geometric constraints is 85.9% while the rank-1 accuracy without geometric constraints is only 36.4%. The location and class label information also increases the retrieval precision significantly.

The SIFT keypoints based CBIR approaches extract salient local patches from an image and present each local patch with a multi-dimensional vector. As a result, each image

is considered as a bag-of-features [49]. However, the optimal partial matching between two sets of keypoints suffers from high computational complexity. To address the computational efficiency problem, various approaches have been proposed and the bag-of-words model [50] is considered as the most popular and successful one. The bag-of-words model quantizes SIFT features to a vocabulary of visual words by a clustering algorithm. Each image feature is mapped to its closest visual word, and each image is represented by a histogram of visual words. Various bag-of-words approaches [50–53] showed promising performance in image retrieval task. However, they suffered from the following drawbacks: (1) High computational cost of clustering for a large amount of clusters, and (2) the lack of theoretical analysis. Li et al. [54] proposed a novel keypoint quantization algorithm, referred to as a random seeding approach, to address the drawbacks of the clustering-based approaches. The random seeding approach quantizes each keypoint by a set of randomly generated hyper-spheres. Each hyper-sphere is analogous to a cluster. Each keypoint is quantized by a binary bit (“1” when the keypoint is within the hyper-sphere and “0” otherwise). The bag-of-words features of each image are computed as a histogram over the hyper-spheres. Comparing to clustering-based approaches, the random seeding approach has the advantage of computational efficiency. Based on the random seeding approach, a two-stage framework for large-scale tattoo image retrieval was proposed. This tattoo image retrieval framework was evaluated on a database that consists of 101,745 images, among which 61,745 are tattoo images and 40,000 are non-tattoo images. On average, around 100 keypoints are detected from each image. This results in a total number of 10 million keypoints for the entire database. The evaluation results showed that the random seeding approach outperformed the clustering-based approaches in terms of tattoo retrieval precision, and the random seeding approach also reduced the computational cost of keypoint quantization. Li et al. [54] also presented a theoretical analysis which showed that the similarity based on the random seeding approach provides an upper bound for the optimal partial matching-based similarity.

The above CBIR approaches were designed to solve the image-to-image tattoo matching problem. They require at least one query tattoo image to perform the tattoo searching. However, the query tattoo image is not always available since the surveillance image of the crime scene is not available in many scenarios. To perform the tattoo

Fig. 2.6: Tattoo sketch samples.

matching without a query tattoo image, Han and Jain [4] proposed a method to match tattoo sketches to tattoo images using local invariant features. Tattoo sketches, like face sketches [55], are drawn based on the description provided by an eyewitness or a victim. Fig. 2.6 shows some tattoo sketches. In the cases that the face sketches cannot be drawn, a tattoo sketch may become a very important clue for identifying the suspect. There are two challenges in the tattoo sketch to tattoo image matching: (1) The texture and color of a tattoo sketch can be very different from those of a tattoo image; (2) The eyewitness can not always provide accurate descriptions of the suspect's tattoo so that a lot of details of tattoo might be missing. To create a tattoo sketch database, Han and Jain [4] proposed a protocol (see Fig. 2.7) for drawing the tattoo sketches. A tattoo image was first shown to a subject for one minute. Ten minutes later, the subject was asked to draw a tattoo sketch on a paper according to his/her memory. Then the tattoo sketch was

Fig. 2.7: Protocol for drawing tattoo sketches [4].

scanned into a digital image. Canny edge detector was employed to extract the edge map of the tattoo sketch/image. The extracted edge map is described as the tattoo shape. SIFT features are extracted from the tattoo shapes. To solve the problem that tattoo shapes are not very informative sometimes due to low image quality, the SIFT features of tattoo appearance images are also extracted. The SIFT features extracted from the tattoo shape image and tattoo appearance image are used for matching separately. The matching score is calculated by a local pattern based Sparse Representation-based Clas-

sification (SRC) approach. The two scores from tattoo shape image matching and tattoo appearance image matching are summed to calculate the final matching score between a tattoo sketch and a tattoo image. The experimental results showed that their approach outperformed the state-of-the-art image-to-image tattoo matching methods and was robust against deformations such as rotation, shear-wrap and twirl.

With the development of deep learning in computer vision tasks, the focus of CBIR approaches is shifting from handcrafted features and models to deep learning-based methods. Di and Patel [35] proposed a deep convolutional network-based method for tattoo image matching using Siamese network which is trained on Tatt-C dataset [40]. Data augmentation is used by flipping the images vertically and horizontally to enlarge the dataset. The images are grouped into genuine pairs and impostor pairs, which consist of the probe images and their verified gallery images, and probe images and their unverified gallery images, respectively. Euclidean distance is used as the training loss of the network. Comparing to MITRE and Compass reported in [56], Di and Patel's [35] method achieved the optimal accuracy at rank-1, rank-10, rank-20 and rank-30.

2.3 Tattoo Image Classification

Besides tattoo retrieval, tattoo image classification and tattoo detection are another two important use cases of tattoo in forensic investigation. There has been some confusion in the definition of tattoo image classification and tattoo detection for a long time. According to the definition of NIST tattoo challenge [40], detecting whether an image contains a tattoo or not is defined as tattoo detection, and finding the location of tattoo in images is defined as tattoo localization in some research works. However, whether an image contains a tattoo or not is a two-class classification problem, and finding the location of a tattoo is an object detection task. To eliminate this confusion, in this thesis, detecting whether an image contains a tattoo or not is defined as tattoo image classification, and localizing tattoos in images is defined as tattoo detection. Note that tattoo image classification is different from tattoo classification which focuses on classifying and assigning tattoo class labels to tattoo patterns.

In the forensic investigation, searching evidence in IT devices of suspects is a challenging task for law enforcement agencies because of the huge and increasing storage capacity

of the devices. There was a case reported that the data seized by law enforcement agencies can be over 120 TB [57]. It is impossible to check all the data manually, especially finding all the images containing tattoos. To automatically find images containing tattoos, many works have been proposed.

Di and Patel [35] proposed a deep learning-based method to perform tattoo image vs non-tattoo image classification. First, an AlexNet [58] was trained on ImageNet LSVRC-2010 dataset [59], and finetuned on the Tatt-C dataset [40] to extract deep features from tattoo images and non-tattoo images. Then, a 2-class linear SVM [60] was trained based on the deep features to determine whether an image contains tattoos or not. The performance of their method was presented by accuracy, including tattoo accuracy, non-tattoo accuracy and overall accuracy. Tattoo accuracy was defined as the number of correctly classified tattoo images divided by the total number of tattoo images. Non-tattoo accuracy was defined as the number of correctly classified non-tattoo images divided by the total number of non-tattoo images. Over accuracy is defined as the sum of correctly classified tattoo images and non-tattoo images divided by the total number of images. The evaluation of their method was conducted on a subset of the Tatt-C dataset which contains 2,349 images (1,349 tattoo images and 1,000 non-tattoo images). The evaluation result gave an overall accuracy of 99.83%.

Xu et al. [26] argued that the Tatt-C dataset, especially the non-tattoo images which contain only human faces, are not the most ideal training images for developing algorithms to find tattoo images taken from diverse backgrounds, viewpoints and poses, as those stored in IT devices of suspects. To train a robust network to classify tattoo images and non-tattoo images, Xu et al. established NTU Tattoo V1 dataset which contains 5,740 tattoo images and 4,260 non-tattoo images. All the images in the NTU Tattoo V1 dataset were collected from Flickr in which images are taken from diverse viewpoints, poses and environments with complex backgrounds, including indoor and outdoor. They trained a classic AlexNet proposed in [58] on NTU Tattoo V1 dataset and compared the evaluation results with the same AlexNet trained on Tatt-C dataset. The evaluation was conducted on both NTU Tattoo V1 dataset and Tatt-C dataset. As a result, four evaluation results were reported (Table 2.2). The evaluation results showed that NTU Tattoo V1 dataset is more challenging than Tatt-C dataset. Tatt-C dataset is not ideal

Table 2.2: Evaluation results of the AlexNet [26] trained and tested on different dataset.

Evaluations	Training Set	Testing Set	Accuracy
1	Tatt-C	Tatt-C	98.81%
2	Tatt-C	NTU Tattoo V1	65.77%
3	NTU Tattoo V1	NTU Tattoo V1	78.30%
4	NTU Tattoo V1	Tatt-C	83.31%

for developing algorithms to find tattoo images taken from diverse backgrounds, poses and viewpoints.

With the establishment of the NTU Tattoo V1 dataset, more tattoo image classification approaches were developed. Sun et al. [36] developed a tattoo detector called TATT-RBDL which was customized from Faster-RCNN [61]. TATT-RBDL was trained using 3,839 tattoo images of Tatt-C dataset [40] and 9,963 non-tattoo images of PASCAL Visual Object Classes (VOC) 2007 dataset [62], and then finetuned from the pretrained VGG_CNN_M_1024 network [63]. TATT-RBDL was evaluated on Tatt-C dataset [40], MIT-SUN dataset [64] and NTU Tattoo V1 dataset [26]. The evaluation results on Tatt-C dataset showed that the overall accuracy of tattoo image vs non-tattoo image classification improved from 96.3% reported in [56] to 98.25%. The evaluation results on NTU Tattoo V1 dataset showed that the overall accuracy of tattoo image vs non-tattoo image classification is 80.66%. A robustness test was also conducted on MIT-SUN dataset which contains images from 397 non-tattoo scene categories. The testing results showed that almost all the detection scores are below 0.5.

2.4 Tattoo Detection

Tattoo image classification can be considered as a sub-task of tattoo detection because a successful detection of a tattoo pattern in an image means this image contains a tattoo. However, tattoo detection is more challenging than tattoo image classification because it needs to give a bounding box of the tattoo pattern.

Tattoo detection is defined as localizing tattoo regions in images. The large variation of tattoos in terms of colors, shapes, and locations increases the difficulty of tattoo

detection. The differences in image quality and view angle also increase the false tattoo detection, which means that a non-tattoo pattern/object is detected as a tattoo. For advancing research and development into tattoo detection technology, the U.S. National Institute of Standards and Technology (NIST) held a challenge (Tatt-C) in 2015 [40] and released a tattoo dataset. This challenge included five tasks and one of them was named tattoo localization which is actually tattoo detection as defined in this thesis. Another competition (Tatt-E) [56] was held by NIST in 2018 and the competition results were published as an internal report. Shan et al. [65] hosted a robust tattoo detection and retrieval competition (RTDRC) in 2019. Tatt-C, Tatt-E and RTDRC pinpointed the importance of tattoo detection for forensic applications.

Some patch-based approaches were developed to detect tattoos from digital images. Huynh et al. [19] used a patch-based-decision-trees approach to detect tattoos from prisoners' full body images. The images were taken by a full-body imaging system [19], which was designed to meet the requirement of the Singapore Police Force. For a full-body image, a skin segmentation method is employed to remove the skin portion of the image. Then an edge detection method was applied on the rest portion of the image to obtain a tattoo candidate. The tattoo candidates were divided into overlapping blocks with a size of 25×25 pixels. A handcrafted feature was extracted from each block. One hundred decision trees were used to form an ensemble to classify tattoo patches and non-tattoo patches. Adaboost was used to enhance the classification performance. All the patches that were classified as tattoo patches were grouped together into a tattoo area. A bounding box was generated based on the coordinates of this tattoo area. 10,000 tattoo blocks and 10,000 non-tattoo blocks were used to train the decision trees. The evaluation results showed that their approach successfully detected 97.27% tattoos in all the tattoo images. The overall detection accuracy in terms of the number of subjects was 99.07% and in terms of the number of tattoos was 99.55%.

Hrkać et al. [39] used a deep convolutional neural network to discriminate the tattoo patches and non-tattoo patches and grouped tattoo patches into blobs. They trained three networks of different architectures with 27,616 patches (13,800 tattoo patches and 13,815 non-tattoo patches) of different patch sizes. The evaluation was conducted on 6,136 patches (3,077 tattoo patches and 3,070 non-tattoo patches) of different patch sizes.

Table 2.3: Evaluation of the network performance on different patch sizes.

Patch Size	8×8	16×16	24×24	32×32	40×40
False Negatives	691	641	887	360	404
False Positives	595	459	258	760	673
Accuracy	0.7953	0.8207	0.8134	0.8134	0.8244

The optimal performance was obtained by VGGNet [66], which achieved a classification accuracy of up to 82.44% (Table 2.3). The TATT-RBDL proposed by Sun et al. [36] can also perform tattoo detection when the information of bounding boxes is provided for training. Sun et al. evaluated TATT-RBDL on Tatt-C dataset. The evaluation result showed that when the overlap region of a detection bounding box and a ground truth bounding box being over 50% is considered correct, over 60% tattoos can be correctly detected.

The images used in [19] are taken from a controlled environment with fixed poses and viewpoints. The images used in [36, 39] are cropped close-up tattoo images. However, in forensic investigation, the collected images are more likely to be taken from an uncontrolled environment with diverse poses, viewpoints and complex backgrounds. To handle the tattoo detection in this type of scenario, Heflin et al. [37] introduced a methodology for detecting tattoos found in unconstrained images. Their method used a graph-based visual saliency model (GBVS) [67] to obtain the saliency maps of the input images. GrabCut [68] was applied to the saliency maps to segment the tattoo areas and generate the bounding boxes. Marčetić et al. [38] introduced an experimental tattoo de-identification system for privacy protection in still images. In their system, a skin detection module was developed to obtain skin-like color regions which were referred to as the region of interests (RoI). The cutout regions and holes inside the RoI were filled by the morphological operation of closing. Then, SIFT features were extracted from an RoI and matched with the template SIFT features from a tattoo dataset which consists of 1,338 SIFT features extracted from 24 tattoos. The pixels in the locations of matched SIFT features were used as seed pixels. The tattoo detection module analyzed the surrounding pixels of the seed pixels iteratively to determine whether they are tattoo pixels or not. The tattoo region was obtained by grouping all the tattoo pixels. Han et

Fig. 2.8: Tattoo detection network in [5].

al. [5] proposed an efficient tattoo detection method that is able to learn tattoo detection and representation using a single CNN via multi-task learning. Fig. 2.8 illustrates the structure of tattoo detection network in [5]. A batch of tattoo images were stitched into a single image which was processed by a ResNet backbone, a RPN and Faster R-CNN for detection and a series of fully connected layers for feature extraction. The hamming distance was used to measure the distance between the features of query tattoo image and the features of a gallery tattoo image. The minimum distance was used as the final distance between two tattoo images.

Some other works [69–72] were also proposed to perform the tattoo detection. Duangphasuk and Kurutach [69] proposed a tattoo detection algorithm based on image negative method in HSV color space and they used this method for tattoo recognition by employing a cross-correlation neural network [70]. Kim et al. [71] proposed two tattoo detection methods. The first one is a center-surround feature localization method and the second one is a graph-cut based localization method. Dong et al. [72] proposed a detection method that can segment tattoos using K-means clustering in LAB color space.

2.5 Scene Text Detection

In chapter 1, we mentioned that text tattoo is especially important in forensic investigations. Text tattoos can be also considered as a part of scene text and some scene text

Fig. 2.9: Architecture of SegLink [6]. Each convolutional block (the gray box) predicts text on its corresponding feature layers. The predicted text was combined into whole words.

detectors can be used for the text tattoo detection task. Text detection in natural scene images has been studied for decades and many image processing based [73–77] and deep learning-based [78–81] methods have been proposed.

Zhou et al. [82] proposed an efficient and accurate scene text detection pipeline called EAST, which predicts word and line-level text instances from natural scene images. EAST has two stages, the first stage utilizes a fully convolutional network (FCN) to predict the rotated bounding boxes of words and text lines directly; the second stage utilizes a non-maximum suppression (NMS) to yield final detection results. VGG16 and PVANET are used as the base model and pretrained on the ImageNet database. EAST achieved an F-score of 0.7820 on ICDAR 2015 [83] and 0.3945 on COCO-Text [84].

Shi et al. [6] proposed an oriented text detection method, named SegLink. SegLink decomposes text into two elements: segments and links. Fig. 2.9 illustrates the architecture of SegLink. A segment is an oriented box covering a part of a word or text line; A link connects two adjacent segments that belong to the same word or text line. SegLink detects text using VGG-16 as its backbone model. A convolution predictor with 3×3 kernels is connected to six of the feature layers of VGG-16 [24] to detect segments and links. Segments and links are filtered by their confidence scores, respectively. A graph is constructed using segments as nodes and links as edges. Connected components are

Fig. 2.10: Scene text samples. First row: regular shape scene text. Second row: arbitrary shape scene text.

found by a depth-first search (DFS) method. Each connected component indicates a set of segments connected by links. SegLink achieved an F-score of 0.85 on ICDAR 2013 [85] and 0.75 on ICDAR 2015 [83].

Deng et al. [86] proposed an instance segmentation-based method called PixelLink, which segments text pixels first and then generates bounding boxes from the segmentation results. PixelLink uses VGG-16 [24] as the backbone model to predict whether a pixel is a text pixel or not. All the predicted positive pixels are linked together by the predicted positive links. Bounding boxes are computed directly based on the grouped pixels without bounding box regression. A postprocessing procedure follows to filter out some noise bounding boxes. PixelLink achieved an F-score of 0.881 on ICDAR 2013 and 0.837 on ICDAR 2015.

EAST, SegLink and PixelLink detect text with regular shapes from natural scene images. However, many scene text are of arbitrary shapes as shown in Fig. 2.10. To detect text with arbitrary shapes, Wang et al. [87] proposed an efficient and accurate arbitrary-shaped text detector, called Pixel Aggregation Network (PAN). PAN uses a low computation-cost segmentation head, which consists of Feature Pyramid Enhancement Module (FPEM) and Feature Fusion Module (FFM). FPEM fuses low-level and high-level features using U-shaped convolutions. FFM fuses the features generated by FPEMs of different depths. To reconstruct the complete text instance, a learnable postprocessing method called Pixel Aggregation (PA) is proposed to guide the pixels. PAN achieved an F-score of 0.829 on ICDAR 2015.

Fig. 2.11: Pipeline of PSENet [7].

Xie et al. [88] proposed a supervised pyramid context network (SPCN) to detect scene text with arbitrary shapes. In SPCN, images are sent to feature pyramid network (FPN) [89] first and the features at different scales are extracted. Then a text context module (TCM) follows to fuse the pyramid features. A Mask R-CNN [12] branch is utilized to predict the bounding box of scene text. ResNet50 is used as the backbone of Mask R-CNN and pretrained on ImageNet. SPCN achieved an F-score of 0.872 on ICDAR 2015.

To avoid false detection when two text instances are close to each other, Wang et al. [7] proposed a Progressive Scale Expansion Network (PSENet), which can detect multiple text instances with arbitrary shapes. PSENet uses an FPN to extract features at different scales and applies an image processing based approach, named scale expansion algorithm, to calculate the region of text instances. Fig. 2.11 shows the pipeline of PSENet. PSENet achieved an F-score of 0.857 on ICDAR 2015.

In scene text detection, postprocessing of binarization is essential for segmentation-based detection. This procedure converts the probability map into the bounding box. To simplify this procedure and improve the performance of scene text detection, Liao et al. [90] proposed a differentiable binarization (DB) module. DB is fully differentiable when training along with a segmentation network. Therefore, DB can help to generate a better binary mask and produce a more accurate segmentation mask. Liao et al.'s work achieved an F-score of 0.873 on ICDAR 2015.

Ye et al. [91] proposed the TextFuseNet to exploit the use of fused richer features in scene text detection. TextFuseNet perceives text from character, word and global level of feature representation and uses a novel text representation fusion technique to achieve robust arbitrary text detection. First, TextFuseNet uses an additional semantic segmentation branch to extract global features; then character features are extracted in the detection branch and word features are extracted in the mask branch. Both detection branch and mask branch are based on Mask R-CNN. Next, all global features, character features and word features are fused via a multi-path feature fusion network. The fused features allow the TextFuseNet to learn a more discriminative representation and produce more accurate text detection results. TextFuseNet achieved an F-score of 0.943 on ICDAR 2013 and 0.921 on ICDAR 2015.

Liu et al. [8] used a parameterized Bezier curve to adaptively fit the arbitrarily-shaped text and proposed the ABCNet that uses BezierAlign layer to extract accurate convolution features of a text instance with arbitrary shapes which significantly improves the precision. ABCNet represents the oriented or curved text with parameterized Bezier curves for the first time. Bezier curve uses Bernstein Polynomials [92] to represent a parameterized curve as shown in Fig. 2.12. With the help of the Bezier curve, ABCNet achieved state-of-the-art performance in scene text spotting, showing a high level of effectiveness and efficiency. The architecture of ABCNet is shown in Fig. 2.13. A ResNet+FPN backbone extracted the features of the input image. The features were processed by a Bezier alignment layer following by a CNN classifier and a LSTM recognizer which output the final recognition results.

2.6 Deep Object Detection

Tattoo detection can be considered as a particular case of object detection. Object detection has been studied by the computer vision community for many years and its performance is improving continuously with the development of deep learning. Generally, deep learning-based object detectors can be categorized into two major branches: one-stage detectors and two-stage detectors. One-stage detectors can perform detection without the region proposal generation step, while the two-stage detectors need to generate a set of region proposals which are subsequently refined by CNN.

Fig. 2.12: Two samples of Bezier curves [8].

Fig. 2.13: Architecture of ABCNet [8].

Fig. 2.14: Architecture of YOLO [9].

2.6.1 One Stage Detectors

YOLO [9, 93, 94] and SSD [10] are two most popular one-stage detectors, which aim to be fast in object detection with a comparable accuracy. The original YOLO [9] uses

a modified GoogLeNet [95] for image classification as its backbone network. Fig. 2.14 illustrates the architecture of YOLO. The network has 24 convolutional layers followed by two fully connected layers and uses 1×1 reduction layers followed by 3×3 convolutional layers instead of the inception modules used by GoogLeNet. The training was performed in two stages. In the first stage, the first 20 convolutional layers of YOLO network followed by an average-pooling layer and a fully connected layer were pretrained on the ImageNet 1,000-class competition dataset [59]. In the second stage, four convolutional layers and two fully connected layers were added to the YOLO network for the detection training. The final layer predicts both class labels and bounding boxes. The evaluation results showed that YOLO is fast in detecting objects and makes fewer background errors compared to Fast R-CNN [11]. As a trade-off, YOLO lags behind some state-of-the-art deep detectors in terms of accuracy.

YOLO was upgraded to YOLOv2 [93], which is also known as YOLO9000 that can detect over 9,000 object classes, by using a new model named DarkNet-19. YOLOv2 was designed to improve the performance of YOLO, because YOLO has some limitations such as: (1) YOLO makes a large number of localization errors, (2) YOLO has low recall compared to other region proposal based detectors. YOLOv2 was constructed based on Darknet-19 model which has 19 convolutional layers and five maxpooling layers followed by an average pooling and softmax as output. Evaluation results showed that, with only 5.58 billion operations, Darknet-19 achieved 72.9% top 1 accuracy and 91.2% top 5 accuracy on ImageNet. YOLOv2 was trained for classification first on the standard ImageNet 1,000 classes classification dataset and it achieved a top one accuracy of 76.5% and a top five accuracy of 93.3%. YOLOv2 was modified for detection by removing the last convolutional layer and adding three 3×3 convolutional layers with 1,024 filters. Each convolutional layer was followed by a 1×1 convolutional layer with the number of outputs that are needed for detection. According to the experimental results, YOLOv2 achieved an mAP of 73.4% on PASCAL VOC2012 [62] and an mAP₅₀ of 44.0% on COCO [96] test-dev2015 dataset. YOLOv3 [94] further improved the performance of YOLOv2 to 57.9% mAP₅₀ on COCO by using a model named DarkNet-53 which increased the convolutional layers of DarkNet-19 from 19 to 53. Note that YOLO, YOLOv2 and YOLOv3 were first pretrained as image classifiers before adapted for object detection.

Fig. 2.15: Architecture of SSD [10].

SSD [10], which was introduced soon after YOLO, used a pretrained VGG-16 as its backbone network with few modifications (see Fig. 2.15). Firstly, the fully connected layers were replaced with convolutional layers. Secondly, the dropout layers were removed. Finally, the last max pooling layer was also replaced with a dilated convolutional layer. With these modifications, SSD achieved 81.6% mAP on PASCAL VOC2007, 80.0% mAP on PASCAL VOC2012 and 46.5% mAP₅₀ on COCO test-dev2015. Inspired by SSD, Fu et al. [97] proposed DSSD, which uses additional context information to improve the detection accuracy of original SSD. Some modifications were made in DSSD: (1) DSSD replaced the VGG-16 network in SSD with a Residual-101 network to improve accuracy, (2) One residual block was added to each prediction layer, (3) A series of deconvolutional layers were added after the original SSD layers. Evaluation results showed that DSSD achieved 81.5% mAP on PASCAL VOC2007, 80.0% mAP on PASCAL VOC2012 and 53.3% mAP₅₀ on COCO test-dev2015.

2.6.2 Two Stage Detectors

Two-stage detection detectors are the focus of many works. Girshick et al. [98] introduced Region-based Convolutional Neural Network (R-CNN). R-CNN consists of three modules. The first module uses the Selective Search (SS) [99] scheme to generate region proposals; the second module extracts a 4,096-dimensional feature vector from each region proposals using a classic AlexNet [58]; and the third module trains a linear SVM for each class using extracted feature vector of corresponding class and returns bounding boxes by bounding box regressors. R-CNN achieved 58.5% mAP on PASCAL VOC2007 and 53.7% mAP

Fig. 2.16: Architecture of Fast R-CNN [11].

on PASCAL VOC2012. Although the detection performance improved, R-CNN suffered from the following drawbacks:

- R-CNN is trained using a 3-stage training scheme.
- Training is time consuming and the features require a very large storage space.
- Speed of object detection is slow.

To address the disadvantages of R-CNN, Fast R-CNN [11] was proposed. Fast R-CNN takes the entire image and a set of object proposals as input. The network first extracts a feature map by processing the image with convolutional layers and max pooling layers. A RoI pooling layer is employed for each object proposal to extract a fixed-length feature vector from the feature map. Each feature vector is inputted into a sequence of fully connected layers that have two output layers. One output layer produces softmax scores over all object classes plus a “background” class; another output layer outputs four coordinates which are used as the predicted bounding box of each object. A multi-task loss is used to jointly train the classification and bounding box regression. Comparing to R-CNN, Fast R-CNN is faster and more accurate. Evaluation results showed that Fast R-CNN achieved 70.0% mAP on PASCAL VOC2007 and 68.8% map on PASCAL VOC2012.

Fig. 2.17: Architecture of Mask R-CNN [12].

Ren et al. [61] proposed Faster R-CNN to further improve the speed of Fast R-CNN. Faster R-CNN is modified by adding a region proposal network (RPN) to Fast R-CNN. RPN takes an image as input and produces a set of object proposals by sliding a small network over the convolutional feature map produced by the last shared convolutional layer of the detection network. The proposals at each sliding window location are called anchors which are centered at the sliding window. An alternating training scheme was used to train Faster R-CNN. First, the RPN is trained end-to-end by backpropagation and stochastic gradient descent (SGD). The proposals produced by RPN were used to train Fast R-CNN. Then, the network tuned by Fast R-CNN is used to initialize RPN, and the whole process is iterated. Evaluation results showed that with the help of RPN, Faster R-CNN achieved better mAP with much fewer proposals on all of PASCAL VOC2007, PASCAL VOC2012 and COCO dataset.

Another improvement of R-CNN is Mask R-CNN [12], which extends Faster R-CNN by adding a branch for producing segmentation masks on each RoI that works in parallel with existing branches for classification and bounding box regression. Fig. 2.17 illustrates the architecture of Mask R-CNN. The mask producing branch is a small fully convolutional network (FCN) applied to each RoI. Mask R-CNN changed the backbone network architecture to an architecture that combines ResNet [21] and ResNeXt [22] with

Fig. 2.18: Architecture of R-FCN [13].

a Feature Pyramid Network (FPN) [89]. Experimental results showed that Mask R-CNN achieved 60.0% mAP_{50} of instance segmentation and 62.3% mAP_{50} of object detection on COCO dataset.

Fast R-CNN and Faster R-CNN are considered to be semi-convolutional, in which one subnetwork produces proposals on the entire image and another subnetwork processes individual regions. The per-region subnetwork is applied hundreds of times in each detection. This makes detection very costly. To improve the efficiency of Faster R-CNN, Dai et al. [13] proposed Region-based Fully Convolutional Networks (R-FCN) which consists of shared, fully convolutional architecture as in the case of FCN [100] (see Fig. 2.18). To incorporate translation variance into this FCN, a set of position-sensitive score maps were constructed as the output of FCN by using a bank of specialized convolutional layers. A position-sensitive RoI pooling layer was appended on top of this FCN with no convolutional layers or fully connected layers followed. ResNet-101 was used as the backbone of R-FCN. The evaluation results showed that R-FCN achieved 83.6% mAP on PASCAL VOC2007 and 82.0% mAP on PASCAL VOC2012.

To produce region proposals, two fully connected layers are involved in Faster R-CNN. The score maps produced by R-FCN are also very large. Thus, the speed of Faster

R-CNN and R-FCN is slower than one-stage detectors like YOLO [9] and SSD [10] due to the heavy-head design. To overcome this drawback, Light-Head R-CNN [101] was proposed to make the network head lighter by using a thin feature map and a simple R-CNN subnetwork. Evaluation results showed that Light-Head R-CNN is fast, 30.7% mmAP at 102FPS on COCO dataset, and accurate, 40.8% mmAP on the COCO dataset.

In object detection, an intersection over union (IoU) threshold is required to define a positive detection and a negative detection. The detection performance tends to degrade with an increasing IoU threshold. To address the problem caused by the IoU threshold, Cai and Vasconcelos [102] proposed a Cascade R-CNN, which consists of multiple detectors trained with increasing IoU thresholds stage-by-stage. The output of one stage is used to train the next stage. This scheme is designed based on the observation that the output of a detector trained in one stage with a certain IoU threshold is a good distribution for training the detector in the next stage with a higher IoU threshold. Evaluation results showed that Cascade R-CNN achieved 62.1% mAP₅₀ on the COCO dataset, which outperformed other state-of-the-art detectors.

RPN plays an important role in two-stage object detectors. The object detectors can achieve better performance by further improving RPN. This motivated Vu et al. to propose a Cascade RPN [103]. Instead of using multiple anchors with different scales and aspect ratios as in other detectors, Cascade RPN uses a single anchor and incorporates anchor-based and anchor-free criteria in defining positive bounding boxes to improve the performance. Cascade RPN also uses the adaptive convolution to benefit from multi-stage refinement. Evaluation results showed that Fast R-CNN and Faster R-CNN, which were modified to use Cascade RPN as region proposal methods, outperformed Fast R-CNN and Faster R-CNN that used other region proposal methods.

Lin et al. [14] found that two-stage detectors that were applied on a sparse set of object locations achieved the highest accuracy, while the one-stage detectors were fast and accurate when applied on dense object locations. This was caused by the extreme foreground-background class imbalance during the training. To address this class imbalance problem, RetinaNet [14] (see Fig. 2.19) was introduced with a new type of loss named Focal Loss. This loss reshapes the standard cross-entropy loss such that it can weigh down the loss of the well-classified samples. With this new loss, RetinaNet is

Fig. 2.19: Architecture of RetinaNet [14].

Fig. 2.20: The pipeline of SNIP training and inference [15].

able to achieve the accuracy of the two-stage object detectors (61.1% mAP₅₀ on COCO dataset) in the speed of one-stage object detectors (122ms per image).

Scale variation is one of the key challenges in object detection. One common solution is multi-scale image pyramid. Image pyramid is used in methods like SNIP [15] and SNIPER [104]. SNIP [15] (see Fig. 2.20) is modified based on multi-scale training and it was proposed to handle various object instance scales in training to reduce the domain-shift for pretrained network. In multi-scale training, for each image, large objects are hard to be classified at a high resolution and small objects are hard to be classified at a low resolution. SNIP, on the other hand, performs the training only on the objects that are within the desired scale range and the remaining objects are ignored to remove the objects in extreme scales. This scheme enables the effective use of all the object instances during the training in SNIP. Experimental results showed that SNIP can improve the

object detection performance of existing detectors significantly. SNIPER [104] extends the SNIP by using chips which are context regions around the ground truth objects. SNIPER generates positive chips that cover as many ground truth boxes as possible at each scale. To avoid increasing the false positive rate, negative chips are also generated to cover the object instances in the background. The objects in the background are detected by RPN. Then, the SNIPER network is trained end-to-end on the chips. Experimental results showed that SNIPER achieved an AP_{50} of 68.5% on the COCO dataset.

However, SNIP and SNIPER still suffered from the increase of inference time. Another solution to the scale variation issue is the feature pyramid that utilizes multi-level features of different spatial resolutions. Feature pyramid is widely used in methods like HyperNet [105], ION [106], SSD [10], MS-CNN [107], TDM [108], FPN [89] and PANet [109]. HyperNet [105] obtains feature maps from different convolutional layers. The feature maps from lower layers are down-sampled by maxpooling; the feature maps from higher layers are up-sampled by deconvolutional operation. Then they are concatenated to form a single cube, which is called Hyper Feature. An RPN is applied on the Hyper Feature to generate proposals. A detection network follows to produce the final detection results. Evaluation results showed that HyperNet achieved an mAP of 76.3% on PASCAL VOC2007. Inside-Outside Net (ION) [106] uses features of the regions both inside the RoI and outside the RoI. For the regions that are inside the RoI, a skip pooling scheme is applied to the feature maps to extract multi-scale features. For the regions that are outside the RoI, features are extracted by a spatial recurrent neural network (RNN). All the inside features and outside features are concatenated, scaled and dimension reduced to form a feature descriptor for each proposal. Two fully connected layers process feature descriptors to produce the class predictions and bounding boxes. ION achieved an mAP of 76.4% on PASCAL VOC2012 dataset. Multi-scale CNN (MS-CNN) [107] is proposed to handle the fast multi-scale detection. It consists of two sub-networks for generating proposals and producing detection results. The proposal sub-network produces feature maps at different scales from different convolutional layers. Each feature map is inputted to the detection sub-network for producing the class prediction and bounding box. Top down modulation (TDM) network [108] handles the multi-scale object detection by using top-down modules and lateral connections. The feature maps produced by each convolutional layer is processed by a lateral module and concatenated to the output of a top-down

Fig. 2.21: Architecture of TridentNet [16].

module. The concatenated feature maps are inputted to the next top-down module. The first top-down module takes input from the last convolutional layer. The output of the last top-down module is used for generating class prediction and bounding box. The evaluation results showed that TDM achieved an AP_{50} of 57.8% on COCO dataset. Feature Pyramid Network (FPN) [89], similar to TDM, also has bottom-up pathway, top-down pathway and lateral connections. The bottom-up pathway is a feed-forward convolution network, which produces feature maps at multiple scales. The top-down pathway generates a higher resolution feature by upsampling feature maps from higher pyramid layers. The lateral connection merges the feature maps of the same size from both pathways. FPN can be adapted to existing detectors like Faster R-CNN [61]. Experimental results showed that Faster R-CNN that uses FPN achieved an AP_{50} of 56.9% on the COCO dataset. Path Aggregation Network (PANet) [109] adds a bottom-up path augmentation to FPN to further improve the performance of FPN. The bottom-up path augmentation downsamples each feature pyramid layer generated by FPN and adds the downsampled pyramid layer to the higher pyramid layer through lateral connection. An adaptive feature pooling is applied on all the layers produced by bottom-up path augmentation and the following branches produce the class prediction, bounding box and mask. PANet won the COCO 2016 object detection challenge by achieving an AP_{50} of 67.2% on the COCO dataset.

The feature pyramid sacrifices the feature consistency across different scales. To

get the best of those two solutions, TridentNet [16] that creates features with a uniform representational power for all scales efficiently was proposed. TridentNet is able to generate scale-specific feature maps by using a multi-branch architecture with dilated convolution [110] which are widely used in semantic segmentation [111–113]. Fig. 2.21 illustrates the architecture of TridentNet. Each branch shares the same transformation parameters but with different sizes of receptive fields. A scale-aware training scheme is adopted to specialize each branch by sampling object instances of proper scale for training. Evaluation results showed that TridentNet achieved a 69.7% mAP₅₀ on the COCO dataset, which outperformed the other state-of-the-art detectors.

Wang et al. [114] proposed Cross Stage Partial Network (CSPNet) to mitigate the problem that other works require heavy inference computations from the network architecture perspective. CSPNet reduces computations by 20% with an equivalent or even superior accuracy. Qiao et al. [115] proposed DetectoRS that improves the performance of object detection by using the recursive feature pyramid at the macro level and switchable atrous convolution at the micro level.

2.7 Tattoo Image Synthesis

Most tattoo studies considered tattoos as a soft biometric trait and focused on tattoo recognition for forensic applications [35, 116–118]. There are also a few works on tattoo visualization for augmented reality. Calmon et al. [119] proposed a method to blend tattoo image with skin image using weighted multiplication. Mauricio et al. [120] designed a mobile device application for tattoo visualization. However, in both works [119, 120], the authors assumed that a tattoo was already available and did not explicitly specify how to obtain a tattoo image. In other words, their methods do not transform an image to a tattoo but put a given tattoo image on the skin image. Outside the scientific community, two websites, PhotoFunia [121] and LunaPic [122], claim that they can convert photos to tattoos. However, their quality is far from acceptable. Fig. 2.22(a) shows a real photo and Fig. 2.22(b) and (c) show the resultant images from LunaPic and PhotoFunia, respectively. According to the observation of the resultant images, PhotoFunia [121] only smooths the contrast adjusted greyscale input image and puts it on a predefined

Fig. 2.22: Resultant images generated by LunaPic and PhotoFunia. (a) is an input real portrait photo, (b) is a synthetic tattoo portrait generated by LunaPic and (c) is a synthetic tattoo portrait generated by PhotoFunia.

forearm image. Except for the input image, there is no way to tune the resultant image. The method that LunaPic [122] uses is unknown. However, its results are nothing similar to tattoos.

Transforming portrait photos to tattoos can be considered as an image style transformation task. Recent methods employed deep neural networks that transfer one image into another with a different style. Gatys et al. [42] introduced a neural algorithm of artistic style, which is designed to perform image style transfer. This algorithm synthesizes image textures using features extracted from a CNN. New images are generated by searching and matching feature representations of example images. Although this algorithm achieved promising results, it can only process images within limited resolutions (smaller than 512×512), and its performance was affected by the low-level noise when performing photo style transfer.

Johnson et al. [43] trained transformation networks using perceptual loss functions that depend on high-level features from a pretrained loss network. The transformation network was trained using stochastic gradient descent (SGD) to minimize a weighted combination of the loss functions defined in the loss network. The training scheme requires the transform network to be retrained for different style images.

The generative adversarial networks (GAN) framework proposed by Goodfellow et al. [17] achieved promising results in image generation and representation learning. Fig. 2.23 illustrates a typical architecture of GAN. GAN estimates generative models via

Fig. 2.23: A typical architecture of GAN [17].

an adversarial process. It includes two models: a generative model G that captures data distribution, and a discriminative model D that calculates the probability of a sample coming from training data instead of generator G . D is trained to maximize the probability of correctly classifying samples from the training set and generator G , while G is trained to minimize the probability that D can correctly classify samples from the training set and generator G . Radford et al. [123] extended GAN into deep convolutional GAN (DCGAN) for image style transfer. They use a fully convolutional network as the generator and discriminator. No fully-connected layers or pooling are used in DCGAN. Isola et al. proposed the “pix2pix” framework [124] which uses conditional GAN to synthesize images from label maps or edge maps. Those methods require image pairs for training. However, in many cases, image pairs are not available. To avoid the requirement of image pairs, Zhu et al. proposed CycleGAN [125]. CycleGAN contains two mapping functions (two generators) that can convert images into another domain and then convert them back. Each mapping function has an associated adversarial discriminator that is trained to maximize the probability to classify whether images are from the same domain or not.

Huang and Belongie [126] proposed a single feed-forward neural network method that

Fig. 2.24: Sample images from Kim et al.'s method [18]. (a) and (c) are raw images, (b) and (d) are corresponding converted stipple images.

enables arbitrary style transfer in real time. The essential part of their method is a novel adaptive instance normalization (AdaIN) layer that aligns the mean and variance of the content features with those of the style features. Chen and Schmidt [127] proposed an optimization objective based on local matching that combines the content structure and style textures in a single layer of the pretrained network.

In addition to these methods, Selim et al. [128] proposed a method to transfer a portrait photo to a portrait painting and reported promising results. Their method requires a fixed head posing and face alignment between the input photo and the reference portrait painting. However, the portraits in this work have pose variations and are not limited to the head only. These methods do not provide any flexibility to tune their resultant images once input photos and style images are given. In other words, users have no control over the final outputs.

Transforming portrait photos to tattoos is also similar to synthesizing digital paintings and stippling. A data-driven technique for modeling 3D brushes to produce realistic digital paintings was developed by Baxter et al. [129]. This technique simplifies and accelerates simulation of the constrained dynamics of brushes by using a small lookup table, which can encode the range of feasible constrained states more efficiently. Another data-driven method for modeling virtual brush strokes named RealBrush was proposed by Lu et al. [130]. Wetbrush model introduced by Chen et al. [131] simulates realistic

oil paintings in real-time. Tuur et al. [132] proposed another realistic digital oil painting method for real-time applications on highly resource-constrained portable hardware. These methods enable digital artists to quickly realize their paintings. However, they are not applicable to transform photos to tattoos, because using brushes to paint on paper or cloth and using needles to tattoo on the skin are totally different physical processes and require different techniques to model. Kim et al. [18] proposed an automatic method for producing stipple images from photographs. In their method, the dot placement is guided by the feature lines extracted from photographs. Fig. 2.24 shows some samples from Kim et al.'s method. Deussen et al. [133] proposed an adaptive version of Lloyd's optimization method that distributes points based on Voronoi diagrams.

2.8 Summary

In this chapter, a short introduction of tattoo is firstly provided. Then previous works related to tattoo retrieval, tattoo image classification and tattoo detection are reviewed. Since tattoo detection can be considered as a particular case of scene text detection, some scene text detectors are also reviewed. Both tattoo detection and scene text detection utilize general object detection techniques. Therefore some deep object detectors are reviewed as well. Finally, tattoo image synthesis works are reviewed. All these works form the foundation of this thesis.

Chapter 3

Geometry Based Tattoo Retrieval System

Tattoo retrieval is one of the most frequent use cases of tattoos in forensic investigation. Currently, most tattoo retrieval systems search tattoos using image information of tattoos, such as SIFT features or some CNN features. Besides image information, tattoos also contain rich geometry information which is also very informative for forensic investigations. Geometry information of a tattoo has two components: the shape of the tattoo and the location of the tattoo. In this research, the shape of a tattoo is defined as the boundary of the tattoo pattern; the location of a tattoo means where on the human body the tattoo is. Traditional tattoo retrieval systems neglect the importance of geometry information of a tattoo. They use a query tattoo image or tattoo sketch with rich details to search tattoos. In the scenario that the query image is not available or the witness cannot provide a tattoo sketch with rich details, they are not able to perform the tattoo retrieval or their retrieval performance drops significantly. However, this scenario is common in forensic investigations, e.g., surveillance camera failed to capture the picture of the suspect's tattoo or the witness forgot the details of the tattoo, only roughly remembered where the tattoo is and the shape of the tattoo. Therefore, how to perform the tattoo retrieval with only tattoo location and shape information is essential and a blank area in tattoo retrieval.

In this chapter, a geometry based tattoo retrieval (GBTR) system¹ is proposed to search tattoos using the geometry information of a tattoo. The GBTR system is designed

¹The work in this chapter has been published in [134]

to provide an end-to-end solution for tattoo retrieval using simple tattoo sketches. Comparing to other tattoo retrieval systems [3,4], it does not need a query tattoo image or a tattoo sketch with rich details. To achieve this goal, a novel full-body coordinate system, a location mapping scheme and a shape matching scheme are designed and utilized in the GBTR system. As an end-to-end solution, a user interface and a database are also provided in the GBTR system.

3.1 Full Body Human Images

Current tattoo retrieval methods search tattoos using close-up tattoo images that contain no location information or the location information can not be mathematically represented. To use the geometry information to search tattoos, the location information must be mathematically represented. In the GBTR system, a full-body coordinate system is designed for representing the tattoo location. This full-body coordinate system uses the joint points of a human body as the reference points. The joint points are extracted from the full-body human images. Therefore, full-body human images are essential for the proposed GBTR system.

The full-body human images are used throughout all the modules in the GBTR system. Sketches are drawn on the binary full-body human model image, location information is obtained from the full-body human image and the images in the GBTR database are full-body human image with tattoos. To have a better understanding of the GBTR system, full-body human images are first introduced.

3.1.1 Full Body Imaging System

Full body human images are taken by the full-body imaging system [19] that was developed for Singapore Police Force. To collect full-body human images using the full-body imaging system, each subject is asked to make six standard poses in front of a black screen. In each posing, the DSLR camera that is mounted on an actuator (see Fig. 3.1) moves from the bottom to the top to capture seven to nine images (see Fig. 3.2) of the subject in a controlled environment. The number of images depends on the height of the subject. Each image covers a segment of the subject's body. All the images of the

Fig. 3.1: The full-body imaging system [19] used in the GBTR system.

subject in each posing are stitched together using an image stitching algorithm to form a full-body human image (see Fig. 3.3).

3.1.2 Full Body Coordinate System

In the GBTR system, the tattoo location information is obtained from a full-body image. To access the tattoo location on a full-body image, a coordinate system is needed. The traditional image coordinate system uses pixel indices. Each pixel location is represented by two integers (x, y) which range from 0 to the width/height of the image (see Fig. 3.4). However, the witness normally describes the location of the tattoo by pointing out where the tattoo is on the human body instead of the location of the tattoo in an image. Since the locations of a human body in different images are different, the image coordinate system cannot be used for representing tattoo locations.

To accurately represent tattoo locations, the GBTR system uses a full-body coordinate system instead of the image coordinate system to represent the tattoo location. Fig. 3.5 illustrates the difference between location mapping using image coordinates systems and full-body coordinates systems. The full-body coordinate system uses the joint points

Fig. 3.2: Six sequences of images taken from six poses. From (a) to (f): front, back, left front, right front, left back and right back.

of the human body to compute the coordinates of a tattoo.

3.1.2.1 Joint Point Detection Algorithm

Joint points are obtained by a joint point detection algorithm. First, each full-body image is converted into a gray-scale image for joint point detection. An adaptive threshold algorithm is applied to produce a binary image from which the boundary of the body can be extracted. Fig. 3.6 illustrates this preprocessing procedure. However, if the skin

Fig. 3.3: Six full-body images generated by stitching the six sequences of images in Fig. 3.2. From (a) to (f): front, back, left front, right front, left back and right back.

Fig. 3.4: The image coordinate system. The left-top corner of the image is the origin (0,0). x axis is the width of the image, and the y axis is the height of the image. The location of the dark pixel is (5,3).

Fig. 3.5: (a) is an input shape on the model. (b) and (d) show the areas calculated by the image coordinates. (c) and (e) show the areas calculated by the full-body coordinate system.

of the subject is too dark, the obtained boundary will not be smooth so that it might affect the results of joint point detection. Hence, a skin segmentation scheme is developed to extract the skin pixels correctly. Because different colors and materials of backgrounds were tested when designing the system, the red channels of skin images always have high values and the skin segmentation procedure becomes very simple. The red (R) and green (G) channels of the resized images are first separated. If $(t_1 < R(x, y) - G(x, y) < t_2)$, where t_1 and t_2 are two predefined thresholds, the corresponding pixel is regarded as a skin pixel; otherwise, it is considered as a non-skin pixel. The skin pixels and the non-skin pixels form a binary image. Then, a morphological filling operation is applied to provide a smooth segmentation result. Finally, the body boundary image and the skin image are combined to obtain a complete body boundary. The last two images in Fig. 3.7 show two body boundaries of the front pose and the side pose. Because prisoners will be well posed under the instructions of prison officers, their joints can be detected easily.

Second, a curve of the body boundary around a joint point is extracted. Next, a

Fig. 3.6: Illustration of the preprocessing. (a) raw full-body image, (b) gray scale full-body image, (c) binarized full-body image and (d) boundary of the full-body.

line is constructed to connect the two endpoints of the curve. The distances from each point on the curve to the line are calculated. Finally, the point that has the maximum distance from the curve to the line is regarded as the joint point. This procedure is shown in Fig. 3.8. However, if this maximum distance is less than a threshold t , the curve is considered a straight line so that the joint point cannot be detected. This can happen in the detection of knee and elbow points in the side pose images. Exception handling is performed and discussed in separate paragraphs. The detection procedures of each joint point in front and back pose images are given below.

- The shoulder points: The armpits can be detected by the algorithm given above because of their large curvature. The points on the boundary right above the armpits are regarded as the shoulder points. Fig. 3.9(a) illustrates the procedure to detect the right shoulder point.

Fig. 3.7: Sample images of joint points on a full-body human image. (a) and (b) are full-body human image from two poses. (c) and (d) are the boundaries of the human body. Red dots are the detected joint points.

- The hip points: The points on the boundary at half the distance between the head and the foot are marked as the hip points.
- The knee points: According to the anatomy study, the distance from knees to soles is a quarter of the persons height. Using this information, the knee points on the boundary of both legs are detected.
- The ankle points: Firstly, the outermost point on the right foot boundary is detected. Secondly, a point on the outer boundary of the right leg and below the knee is selected. The curve connecting these two points is obtained and the outer ankle of right leg is detected by the algorithm given above. Then, the inner ankle point of the right leg can also be detected easily. This procedure is illustrated in Fig.

Fig. 3.8: Illustration of the procedure to detect a joint point at the location with large curvature. The blue point is considered as the joint point.

3.9(b). The same procedure is applied to determinate the ankle points of the left leg.

From the side pose images, which show the full upper and lower limbs, the shoulders, the elbows, the wrists, the knees, and the ankles are detected. The detection procedures of these joints in the right side pose images are given below. The procedures of detecting joint points in the left side pose images are almost the same; thus they are not mentioned.

- The shoulder points: Firstly, a point on the outer boundary of the upper arm is selected. Next, from this point, the outer boundary of the upper arm is traveled upward and the largest x -coordinate of the point on the boundary is kept tract. If there is a decline in the value of the x -coordinate and the difference between the newest x -coordinate and the largest x -coordinate is greater than four pixels, the point with the largest x -coordinate is regarded as the neck point. Based on the starting point and the detected neck point, the shoulder point can be detected by

the algorithm given above. Fig. 3.9(c) illustrates this procedure to detect the right shoulder point in the side pose image.

- The wrist points: Firstly, the outermost point on the right hand boundary is detected. Secondly, a point on the outer boundary of the right forearm is selected. Same as other joint points, the outer wrist point can be detected by the algorithm given above. Then, the inner wrist point can also be detected easily. Fig. 3.9(d) illustrates this detection procedure.
- The knee points: The boundary curves around the knee are extracted. Then, the algorithm is used to detect the knee point in the right curve (Fig. 3.9(e)). If the joint point cannot be determined, the algorithm is applied again to the left curve (Fig. 3.9(e)). If the joint point is still not determined, a pair of points on the two curves, which has minimum distance, is selected as the knee points.
- The ankle points: The same detection procedure applied to front pose images is also employed to detect the ankle points in the side pose images. Fig. 3.9(f) illustrates it.
- The elbow points: The algorithm is not sufficient to detect the elbow points in the case that the subject keeps his/her arm straight. Thus, the ratio of the distance between the elbow and the shoulder to the distance between the wrist and the shoulder is used to perform the detection. Fig. 3.9(g) illustrates this procedure.

3.1.2.2 Full Body Coordinates Representation

Each joint point has its unique joint point ID that ranges in $[0, \dots, n]$, where n is the number of joint points. Fig. 3.7 shows joint points on a full-body human image. For each pixel p in the full-body human image, its Euclidean distances to each joint point are calculated in the image coordinate system. The three closest joint points are selected as the full-body coordinates of p . Mathematically, they are determined by

$$C = \underset{q \in E}{\operatorname{argmin}_E} \sum_p \min \|p - q\|, \quad (3.1)$$

Fig. 3.9: (a)-(b) illustrate respectively the shoulder and ankle point detection in the front pose image. (c)-(g) illustrate respectively the shoulder, wrist, knee, ankle and elbow detection procedures in the side pose images. The blue dots are the joint points, the red dots are the end points of the curves around the joints and the green dot is the armpit.

where E is a set with three non-duplicated joint points and C is the three closest joint points. Each point in the point set C is considered to be one of the three axes of the full-body coordinate system. The area on the full-body image is represented using the distance from each point on the boundary to three axes of the full-body coordinate system.

3.2 GBTR System

GBTR system consists of four modules as shown in Fig. 3.10. The first module is a user interface for the witness to draw the tattoo sketch; the second module extracts the shape information of the tattoo sketch; the third module obtains the location information of the tattoo sketch and the fourth module matches the shape and location information

Fig. 3.10: A schematic diagram of the GBTR system.

and returns the matching results. In addition to these four modules, the GBTR system also includes a full-body human image database called GBTR database from where the retrieved tattoo images are obtained.

3.2.1 Interface for Drawing Tattoo Sketch

The GBTR system uses the geometry information of a tattoo to search the corresponding tattoos from the database. The geometry information includes the shape of the tattoo and the location of the tattoo. To obtain the shape and location information of a tattoo, the GBTR system provides an interface for the witness to draw a tattoo sketch. It displays a full-body human model image on a canvas. The full-body human model image is a binary image converted from a color full-body human image. The witness uses the mouse to draw a tattoo sketch on the full-body human model as shown in Fig. 3.11. The shape of the sketch is saved in the form of points set using image coordinates. The location of the tattoo is obtained from the full body coordinate system. The interface also provides an option to let the witness draw sketch on different poses.

Fig. 3.11: (a) The interface for the witness to draw the query tattoo boundary. (b) The sample of points of sketch tattoo boundary.

3.2.2 Shape Extraction

Shape information is a key component of a tattoo's geometry information. It represents the features of the tattoo that the law enforcement agencies are searching for. In the GBTR system, the shape information of a tattoo is extracted through the following steps.

First, a patch-based tattoo detection method is applied to the full-body human image to detect the regions of candidate tattoos. Fig. 3.12 illustrates the proposed patch-based tattoo detection algorithm. It uses a multi-resolution approach to detect tattoos of different sizes. For large tattoos, an input color image collected by the consumer camera with a size of 4752×3168 pixels is resized to 297×198 pixels for edge detection. Then the edge image is resized again to 594×396 pixels for searching large tattoo candidates. For small tattoos, the same input image is resized to 832×555 pixels and is used in all the operations. The same image processing steps illustrated in Fig. 3.12 are applied to both cases except for the step of resizing the edge images.

To obtain the skin portions in an image, the skin segmentation method that was introduced in the joint detection algorithm section is employed again. This simple method

Fig. 3.12: A schematic diagram to illustrate the patch-based tattoo detection algorithm.

works well for all the color images in the GBTR database. Because the system is used in controlled environments, the images in the database do not suffer from variations of lighting conditions. The GBTR database contains images from many different races, including Chinese, Indian, Japanese, Indonesian, Malaysian, Mexican and European. Their skin colors are very diverse. To extract edges in the segmented skin, a local adaptive threshold operation is applied. Mathematically, it can be described by $K(x, y) = \sigma(M_w(x, y)R_s(x, y)t_3)$, where R_s is the red channel of the segmented skin image whose non-skin pixel values are zero; M_w is a resultant image that is obtained by applying a median filter with a size of $w \times w$ to R_s ; $t_3 \in \mathcal{R}$ is the threshold and σ is a unit step function. For large tattoos, K is resized to 594×396 pixels; for small tattoos, K is resized to 832×555 pixels. Morphological opening and thinning operators are applied to the resized K . An edge density image is obtained by counting the number of local edge pixels. Then, it is binarized and smoothed for obtaining the tattoo candidates which can be real tattoos or body sites with strong edges, e.g., androgenic hair and nipples.

Fig. 3.13: Illustration of the classification scheme that is used to separate tattoos and other skin characteristics, e.g., hair.

The tattoo candidates are passed to a classification scheme illustrated in Fig. 3.13. First, the tattoo candidates are divided into overlapping blocks with a size of 25×25 pixels. Then, features listed below are extracted from each block.

- The edge orientation histograms.
- The length of the longest edge.
- The mean of the length of the top three longest edges.
- The total number of edge pixels.
- The four 8-bin histograms from the R , G and B channels and the grayscale image.
- The difference between the number of elements in the bin having the greatest number of elements in the R (C_r) channel and those in the B (C_b) channel.
- The means of the R , G and B channels and the grayscale image.
- The difference between the standard derivations of the R and B channels.
- The difference between the standard derivations of the C_r and C_b channels from the $Y C_r C_b$ space.
- The covariance matrix of the R , G and B values.

Fig. 3.14: Samples tattoo image with boundaries (red line) extracted by the Snakes method [20].

- The mean and standard deviation of the gray scale image.

One hundred decision trees are used to form an ensemble for this classification and Ada-boost is used to enhance the classification performance. As with other classification systems, this classification scheme also makes mistakes. To correct these mistakes, some prior knowledge is used. If a block is classified as a tattoo block but none of its neighboring blocks is classified as a tattoo block, it will still be considered as a non-tattoo block. The classification results from two scales, one for large tattoo detection and the other for small tattoo detection, are combined into one image which contains the detected tattoos.

After the classification, the detected tattoo image is converted into a binary image using an adaptive threshold method [135] to remove the skin area. An active contour model called Snakes [20] is applied to the binary image to find the boundary of the tattoo. Fig. 3.14 shows some sample images with the boundaries extracted by the snake method.

3.2.3 Location Mapping

The GBTR system is designed to retrieve tattoos that match some particular shape around the location indicated by the witness. Hence, the location of the sketch needs to be mapped to the full-body human images in the GBTR database to find the corresponding location. This mapping cannot be performed in image coordinate system, because the locations of the tattoo sketches or tattoos are related to the human bodies which locations are different in different full-body human images. Therefore, the mapping is performed in the full-body coordinate system. Let S_{jpts} represent the set of joint points of the human model image and S_{bdry} represent the point set of the boundary of the query tattoo sketch that a witness draws. The Euclidean distances between each point in S_{jpts} and S_{bdry} are calculated and a set of three closest joint point S_{cljp} are selected from S_{jpts} . $S_{cljp} = \{P_i | i \in [id_1, id_2, id_3]\}$, where P_i is the joint point and id_1, id_2 and id_3 are the IDs of three selected joint points.

In the GBTR database, the joint points of each full-body human image are calculated using joint detection methods as well. Let S'_{jpts} be the joint points of a full-body human image in the GBTR database. Given the closest joint point set S_{cljp} , the IDs of the three closest joint points are also known, which are id_1, id_2 and id_3 . Thus, the three corresponding joint points in S'_{jpts} can be determined and denoted as S'_{cljp} . Two matrixes M_1 and M_2 representing S_{cljp} and S'_{cljp} respectively are constructed and given below.

$$M_1 = \begin{bmatrix} x_{id_1} & y_{id_1} \\ x_{id_2} & y_{id_2} \\ x_{id_3} & y_{id_3} \end{bmatrix}, \quad M_2 = \begin{bmatrix} x'_{id_1} & y'_{id_1} \\ x'_{id_2} & y'_{id_2} \\ x'_{id_3} & y'_{id_3} \end{bmatrix}, \quad (3.2)$$

where (x, y) and (x', y') are points in S_{cljp} and S'_{cljp} , respectively. Using M_1 and M_2 , a transform matrix T between S_{cljp} and S'_{cljp} can be calculated as

$$M_1 T = M_2, \quad (3.3)$$

$$T = (M_1^T M_1)^{-1} M_1^T M_2. \quad (3.4)$$

Using the transform matrix T , the boundary S_{bdry} that is drawn by the witness on the full-body human model can be mapped to S'_{bdry} in the full-body human images in the GBTR database. Fig. 3.15 shows the full scheme of the location mapping procedure.

Fig. 3.15: Location mapping scheme.

More clearly, S'_{bdry} is the corresponding boundary of the query tattoo boundary in the full-body human images. Let M_{bdry} and M_{bdry}' be two $2 \times n$ coordinate matrices storing the boundary points of S_{bdry} and S'_{bdry} , where n is the number points in S_{bdry} . The mapping is described by the equation,

$$M'_{bdry} = M_{bdry}T. \quad (3.5)$$

In the GBTR system, the joint points of the full human body images in the GBTR database are preprocessed and saved in files. During the location mapping procedure, the system loads the joint points file to reduce the processing time.

3.2.4 Tattoo Matching

At this stage, the GBTR system matches the shape of the query tattoo sketch with the shapes of tattoos in the GBTR database. Note that not all tattoos in the GBTR database are selected for matching. The GBTR system only searches tattoos among the

candidate tattoos. Candidate tattoos are selected based on the location mapping. Let S_{bdry} be the boundary point set of a query tattoo sketch and M_{bdry} be its coordinate matrix. For an image I in the GBTR database, the corresponding boundary point set of the query tattoo sketch on I is S'_{bdry} and the mapped coordinate matrix is M'_{bdry} . The tattoo region in image I can be detected by the patch-based tattoo detection algorithm and the coordinate matrix of the tattoo region boundary is M_{tatt} . If two areas marked by M'_{bdry} and M_{tatt} have an intersection, this area is considered to be a candidate tattoo. However, if the area enclosed by M_{tatt} is four times larger than the area of the mapped query tattoo region, the candidate tattoo is considered as an incorrect match. In addition, the aspect ratios of the boundaries of candidate tattoo and the mapped query tattoo are calculated as well. If the difference between the aspect ratios of the two boundaries is larger than 0.5, the candidate tattoo is also considered as an incorrect match. These two selection criteria filter out some incorrect candidate tattoos before the shape matching is performed, which is an effective scheme for speeding up the tattoo matching procedure.

The point sets of the boundaries of the rest candidate tattoos are formed along with the mapped query tattoo. Let S'_{bdry} be the boundary point set of the query tattoo sketch that is mapped to an image I in the GBTR database. Let S_{tatt} be the boundary point set of the tattoo area in image I . The coherent point drift algorithm (CPD) [136] is applied on S'_{bdry} and S_{tatt} to find a rigid transformation T_r . Then, T_r is applied on S_{tatt} to transform S_{tatt} to S'_{tatt} . The Euclidean distance between the corresponding points in S'_{bdry} and S'_{tatt} is calculated and used as the matching score. All the candidate tattoos are sorted in the ascent order based on the matching scores because lower matching scores means two points sets are more similar to each other. Fig. 3.16 shows the retrieved tattoo images without sorting and with sorting.

3.3 GBTR Database

The GBTR database is established for evaluating the GBTR system. It contains 2,188 full-body human images with tattoos on different parts of the human body. 547 images in the dataset are original full-body images and the other 1,641 images are generated from the original full-body images by flipping and stretching (see Fig. 3.17). As a matter of

Fig. 3.16: (a) the query tattoo boundary. (b) the retrieved tattoos without sorting and (c) the retrieved tattoos after sorting. The images in (b) and (c) are the rank 1-7, from left to right.

fact, only two subjects have tattoos on their upper arms and backs. The other subjects do not have any tattoos because they are mainly students who came from Asian countries. Therefore, tattoos collected from the Internet are manually added to the rest original full-body images using Photoshop. Fig. 3.18 shows some sample images from the GBTR database. The database will be shared with other researchers ².

3.4 Experimental Results

The locations of tattoos in the GBTR system are presented using joint points in a full-body coordinate system. The accuracy of the joint point detection will affect the tattoo

²<https://github.com/BFLTeam>

Fig. 3.17: (a) original full-body image, (b) image generated by flipping from (a), (c) image generated by stretching (a), (d) image generated by flipping from (c). Faces are covered.

retrieval performance of the GBTR system. The shape information of tattoos is extracted from detected tattoo candidates. The accuracy of tattoo detection also affects the tattoo retrieval performance of the GBTR system. Therefore, before evaluating the GBTR system in the tattoo retrieval task, the joint point detection method and tattoo detection methods are evaluated first in section 3.4.1 and section 3.4.2. The evaluation of tattoo retrieval using the GBTR system is reported in section 3.4.3. Section 3.4.4 compares the tattoo retrieval performance using low-quality tattoo sketches and high-quality tattoo sketches.

3.4.1 Evaluation of Joint Points Detection

To evaluate the performance of the joint point detection method, 5 testers were invited to manually mark the joint points on 180 full-body human images from 6 poses of 30

Fig. 3.18: Samples of full-body images with manually added tattoos.

subjects. The differences between the joint points detected by the joint point detection method and the joint points marked by the testers are measured by the average Euclidean distances between two types of joint points. The average Euclidean distances between joint points marked by different testers are also calculated. Table 3.1 and Table 3.2 reports the evaluation results of different poses. Comparing to the image size (630×240), the mean and standard deviation of the Euclidean distances between auto detection and manual marking are considered small and acceptable.

Table 3.1: Average Euclidean distances between two types of joint points (front and back poses). “Auto” means the joint detection method, “Man.” means manual marking.

Comparison type	Body parts	Front		back	
		mean	std	mean	std
Auto vs. Man.	Shoulder	7.35	4.82	9.73	3.75
	Hip	15.81	5.96	15.26	5.26
	Knee	14.25	5.84	8.50	2.84
	Ankle	9.55	3.83	8.97	3.36
Man. vs. Man.	Shoulder	6.85	3.43	7.05	2.71
	Hip	3.23	1.46	2.73	1.05
	Knee	2.77	1.34	2.91	1.04
	Ankle	2.40	0.99	2.73	1.13

Table 3.2: Average Euclidean distances between two types of joint points (side poses). “Auto” means the joint detection method, “Man.” means manual marking.

Comparison type	Body parts	Left front		Right front		Left back		Right back	
		mean	std	mean	std	mean	std	mean	std
Auto vs. Man.	Shoulder	5.39	2.22	5.89	3.26	6.75	3.51	7.34	2.50
	Elbow	7.83	3.54	7.48	2.30	8.10	3.58	6.90	3.23
	Wrist	4.67	2.48	5.00	1.53	4.63	1.06	3.75	1.16
	Knee	6.86	3.39	7.64	3.49	8.04	3.09	6.98	3.60
	Ankle	7.17	3.23	9.43	3.44	9.19	3.23	8.48	3.69
Man. vs. Man.	Shoulder	3.65	1.05	4.48	1.73	5.12	2.00	5.54	1.21
	Elbow	2.88	1.50	3.15	1.25	3.09	1.28	3.26	1.43
	Wrist	2.33	0.83	2.85	1.32	2.78	1.30	2.56	1.22
	Knee	2.38	1.06	2.57	1.17	2.62	1.21	2.39	0.86
	Ankle	2.91	1.16	2.39	0.95	2.56	1.03	2.75	1.25

3.4.2 Evaluation of Patch-based Tattoo Detection

The patch-based tattoo detection method was evaluated on 5,096 images, 1,943 of those images have tattoos and the rest are non-tattoo images. The evaluation showed that this patch-based tattoo detection method used in the GBTR system detected 97.27% of all

Fig. 3.19: Samples of successfully detected tattoos using patch-based tattoo detection method in the GBTR system.

Fig. 3.20: GBTR system evaluation scheme. The solid line marks the standard low-quality sketch path and the dot line marks the high-quality sketch evaluation path.

the tattoos in the tattoo images. Only some very small tattoos or tattoos on the skin borders can not be detected. This patch-based tattoo detection method also recognized 1.41% non-tattoo images as tattoo images. Fig. 3.19 shows some successfully detected tattoo images.

3.4.3 Evaluation of Tattoo Retrieval using the GBTR System

The experiments were conducted on the GBTR database. 547 original full-body images in the database were selected as the query images and the rest 1641 images were regarded as the suspect images. Each image in the query set has three corresponding images from the same person. To evaluate the performance of the GBTR system, a volunteer was asked to draw 182 tattoo boundaries. The evaluation scheme is as illustrated in Fig. 3.20. In each test, one image was randomly picked from the query image set and shown to the volunteer for 10-20 seconds. The volunteer was allowed to draw the shape of the tattoo on the interface after a one-minute break. Compare to the ten-minute break in [4], the one-minute break in the evaluation is reasonable since the query image was shown to the volunteers for only 10-20 seconds instead of one minute in [4]. The system used the boundary and location information to search the corresponding tattoos. Fig. 3.11 illustrates the query tattoo boundary inputted by the volunteer. The system searched full-body images in the gallery set and recorded the rank of the corresponding image of the query image in the retrieval results. The Cumulative Match Characteristic (CMC) curve is plotted to report the performance of tattoo retrieval using the GBTR system. CMC curve is a rank-based metric that represents the probability of retrieving the correct tattoos within the top K ranks. The proposed GBTR system is compared with Hans method [4], which relies on the textural details on a tattoo sketch. However, in this study, it is assumed that witnesses cannot provide these details, except for the location and shape of the tattoos. In this comparison, the query tattoo boundary was used to generate a binary image, which indicates skin and tattoo regions as an input of Hans method. The same approach was applied to gallery images for Hans method. Fig. 3.21 shows the CMC curves. The GBTR system achieves an accuracy of 45.24% at rank 30 and 52.38% at rank 50, while Hans method achieves an accuracy of 21.67% at rank 30 and 39.44% at rank 50.

The GBTR system outperformed Hans method because: (1) the GBTR system uses the location along with the shape of the query tattoo boundary to narrow down the searching range. (2) the GBTR system matches the boundary of the sketch with the boundary of the tattoos using CPD, while Hans method depends on textural details in the sketch so that SIFT descriptors can be used for matching. Once witnesses cannot provide the textural details, Hans method cannot perform well.

Fig. 3.21: Matching performance of the proposed method comparing to Hans method.

3.4.4 Comparisons of Tattoo Retrieval using Low-quality Tattoo Sketches and High-quality Tattoo Sketches

During the experiment, it was found that the quality of the sketch affects the performance of the matching. To evaluate how the sketch quality affects the matching performance, another experiment was conducted. In this experiment, the volunteer was allowed to draw the sketch with the query tattoo images being displayed. Thus, the volunteer does not need to remember the query tattoos. The matching was performed on the new boundaries, which are denoted as high-quality input sketches in Fig. 3.22. The sketches from the previous experiment are denoted as low-quality input sketches in Fig. 3.22. This result showed that the quality of the sketches is critical for the matching performance.

The comparison result proves that the performance of the GBTR system depends on the quality of the witness sketch. This means if the witness can sketch as closely as the boundary of the query tattoo, the system has a higher chance to retrieve the correct suspect image. However, as mentioned in the section 1.2, it is quite challenging for the witness to provide an accurate description of the suspects' tattoos. In most cases, the witness can only roughly provide the boundary and the location of a tattoo. The GBTR system is still able to achieve a promising searching result in such a situation.

Fig. 3.22: Comparison of the matching performance of the proposed method with guide image and without guide image.

3.5 Summary

Tattoo retrieval plays an important role in forensic investigations. Traditional tattoo retrieval systems use query tattoo images or tattoo sketches with rich details to perform tattoo searching. However, those query tattoo images or tattoo sketches are not always available. To avoid the requirement of the query tattoo images or tattoo sketches with rich details in tattoo retrieval task, a geometric based tattoo retrieval (GBTR) system is proposed and implemented.

In the GBTR system, a joint point detection method is designed to locate the joint points of a human body. Then a full-body coordinate system is designed to calculate the location of tattoos based on the human joint points. A location mapping method is designed to find the corresponding location on other images. A patch-based tattoo detection method is employed to detect the candidate tattoos. The shape information of a tattoo is extracted by an activate contour method named Snake. A point set registration method named coherent point drift (CPD) is employed to match two boundary points sets of two tattoos. The matching score is used to sort the retrieval results. To evaluate the performance of tattoo retrieval using the GBTR system, a full-body image database named GBTR database is established. All the images are collected and processed by a

full-body imaging system. An evaluation scheme is designed to examine the accuracy of tattoo retrieval using low-quality tattoo sketches and high-quality tattoo sketches. The evaluation results showed that the GBTR system achieved a promising accuracy in tattoo retrieval using low-quality tattoo sketches. The retrieval accuracy is even higher using high-quality tattoo sketches.

The GBTR system does not require witnesses to sketch the tattoo details. It only uses shape and location information. Combining with the full-body imaging system or using a high-resolution camera to take full-body images, the GBTR system can automatically locate tattoos and extract their shapes. It can significantly reduce the workload of the law enforcement officers, who perform these tasks manually. In addition to saving manpower, the proposed GBTR system extracts more accurate location and shape information of tattoos which is not accurately described by the NIST labels. In the real case operations, the manual labels and the proposed GBTR system should be used together since they collect different information from witnesses.

The tattoo detection method in the GBTR system is based on patch-based decision forests. Although the detection accuracy is promising on GBTR dataset, a better and stronger tattoo detection method is still in need. Next chapter will present a CNN-based tattoo detection method which focuses on text tattoo detection task and can be modified to handle general tattoo detection.

Chapter 4

Text Tattoo Detection

Tattoos are very informative soft biometric, which carry rich personal information in many cases. The first tattoo that has ever been recorded can be dated back to 3000B.C. [2]. Tattoos are usually not too small and often have describable contents such as text, symbols and some meaningful patterns and objects. Hence, they can be easily observed, remembered and described by witnesses. Fig. 4.1 shows some photos of persons with tattoos. Because of these characteristics, tattoos have been widely used by law enforcement agencies for forensic investigation, such as criminal and victim identification. According to the statistic [47], 85% of prisoners under 35 have tattoos and 75% of the inmates who are re-incarcerated have tattoos. Another statistic [137] reports that 36% of Americans between the age of 18 and 29 have at least one tattoo. Obviously, tattoos play an important role in criminal and victim identification.

Text tattoo, as a subcategory of tattoo, is especially important in forensic investigation comparing with other general tattoo patterns, because text tattoo can provide vital cues such as names of criminals, names of their relatives, their gang groups, important dates (e.g., wedding date and date of birth) and other personal information as shown in Fig. 4.2. Therefore, detecting text tattoos and recognizing their contents can help law enforcement agencies to narrow down the suspect list quickly. In this study, we concentrate on text tattoo detection since the current tattoo detection methods focus on general tattoo only and there is no work especially for text tattoo detection. Although text tattoo is a subcategory of tattoo and the general tattoo/object detectors should be also able to detect it, there are some important characteristics among generic objects,

Fig. 4.1: Sample images in NTU Tattoo V2 dataset.

tattoos and text tattoos, which can be exploited for text tattoo detection. Clearly, tattoos, including text tattoos can only appear on the skin but not on other objects and text tattoos usually appear as a small part of a bigger tattoo with similar color as shown

Fig. 4.2: Sample text tattoos show important personal information: (a, b, e and f) names, (e, f and g) dates and (h) gang.

in Fig. 4.2. If objects and humans can be detected or highlighted, tattoo detection will become more effective and similarly, if tattoo can be detected or highlighted, text tattoo detection will also become more effective. To make use of this prior knowledge, a novel prior knowledge-based attention mechanism (PKAM), which can be easily plugged into the backbones of standard detectors, is proposed. To use the relationship between objects and tattoos and the relationship between tattoos and text tattoos, PKAM is applied twice in the proposed Text Tattoo Detection Network Based on Double Attention (TTDN-DA)¹.

Text tattoo can be also considered as a subcategory of scene text. But the features of text tattoo and scene text are different in some ways. First, scene text is likely to be computer-generated text using a set of popular fonts, for example, SynthText [138] uses 1,200 fonts. Text tattoos, on the other hand, are designed and “written” by the tattooist and have diverse styles. Second, scene text generally appears on a flat surface of an object, while text tattoos are normally on elastic surface, i.e., human skin. Skin deformation would also affect the shape of text tattoo. Third, scene text normally has a regular shape and each character has a similar size in each word, while the shape of text tattoo is normally arbitrary and each character in the text tattoo has a different size.

¹This work is submitted to *IEEE Transactions on Biometrics, Behavior, and Identity Science*

Fig. 4.3: The 1st and 2nd rows: sample images of text tattoos. The 3rd and 4th rows: sample images of scene text.

Besides those differences, text tattoos can be inside another tattoo pattern or surrounded by other tattoos. Fig. 4.3 shows some examples.

The remainder of this chapter is structured as follows. Section 4.1 introduces the NTU Tattoo V2 and NTU Text Tattoo V1 datasets. Section 4.2 presents the proposed PKAM. Section 4.3 describes the proposed TTDN-DA its two variants. Section 4.4 gives the implementation details. Section 4.5 reports the experimental results and comparisons and Section 4.6 offers some conclusive remarks.

4.1 Text Tattoo Dataset

The existing public databases for tattoo detection are small and none of them is for text tattoo detection study. Thus, for training the proposed text tattoo detector TTDN-DA, a text tattoo dataset is essential. A general tattoo dataset is also required because TTDN-DA utilizes the relationship between general tattoo and text tattoo. To the best knowledge of the authors, there is no text tattoo dataset released for public access yet, but there are few open access tattoo datasets, including Tatt-C [40], DeMSI [41], NTU Tattoo V1 [26] and WebTattoo [5]. Table 4.1 summarizes the size of each public tattoo datasets.

Table 4.1: Comparisons of different tattoo datasets.

Dataset	Image No.	Remarks
Tatt-C	7,526	Indoor environments
DeMSI	890	Unconstrained environments
NTU Tattoo V1	5,740	No bounding box annotations
WebTattoo	5,000	Indoor and unconstrained environments

(a) Tatt-C [40]

(b) DeMSI [41]

(c) NTU Tattoo V1 [26]

Fig. 4.4: Sample images of open access tattoo datasets.

Tatt-C dataset. The NIST Tatt-C [40] is a well-known tattoo recognition and detection technology challenge. Tatt-C had several challenges, including tattoo identification, tattoo similarity, tattoo detection, etc. The Tatt-C dataset contains 7,526 images with bounding-box annotations. Those images are close-up images taken from indoor environments. Fig. 4.4 (a) shows some sample images of Tatt-C dataset.

DeMSI dataset. The DeMSI [41] dataset containing 890 tattoo images, which is a small subset of ImageNet dataset [59] was established for tattoo detection studies. All the images were taken from unconstrained environments. Fig. 4.4 (b) shows some sample images of the DeMSI dataset.

NTU Tattoo V1 dataset. The NTU Tattoo V1 dataset [26] was built to address the problem of Tatt-C dataset for the tattoo *vs* non-tattoo image classification task. It consists of 5,740 tattoo images and 4,260 non-tattoo images collected from Flickr. In Tatt-C tattoo detection dataset, all non-tattoo images are actually face images, while in the NTU Tattoo V1 dataset, the non-tattoo images were randomly collected images from Flickr. Since the dataset was established for tattoo *vs* non-tattoo image classification, no bounding-box annotations are provided. Fig. 4.4 (c) shows some sample images of the NTU Tattoo V1 dataset.

WebTattoo dataset. The WebTattoo [5] contains over 300K images, including those in Tatt-C, DeMSI and NTU Tattoo V1 datasets and images collected from the Internet. WebTattoo dataset provides bounding-box annotations of the NTU Tattoo V1. However, the dataset for tattoo detection task only contains around 5,000 images with bounding-box annotations.

For developing stronger detection algorithms and accurately evaluating them, a new dataset, NTU Tattoo V2 is established, which consists of 87,282 tattoo images collected from the Internet. The images that were taken in various environments, including indoor and outdoor have diverse quality, sizes and illumination conditions. Among them, 9,067 have text tattoos. The entire dataset has 92,838 non-text tattoo bounding boxes and 10,027 text tattoo bounding boxes. Both the non-text tattoo dataset and NTU Text Tattoo V1 dataset in NTU Tattoo V2 dataset are divided into three subsets, which are training, validation and testing. 20% of the total images are randomly selected and organized as the testing subset; 80% of the rest images are randomly selected and

Table 4.2: The structure of the NTU Tattoo V2 dataset

Dataset	Type	Training	Validation	Test	Total
Non-text tattoo dataset	Non-text tattoo images	50,058	12,514	15,643	78,215
	Non-text tattoo instances	59,498	14,874	18,466	92,838
Text tattoo dataset	Text tattoo images	5,803	1,451	1,813	9,067
	Text tattoo instances	6,453	1,565	2,009	10,027

organized as the training subset, and the remaining images are organized as the validation subset. Table 4.2 summarizes training, validation and testing sets of the NTU Tattoo V2. Fig. 4.1 shows some sample images in the NTU Tattoo V2 dataset. For the sake of convenience, the text tattoo dataset is named NTU Text Tattoo V1. The full dataset will be available for public access ².

4.2 Prior Knowledge-based Attention Mechanism

Inspired by the human visual behavior, hierarchical attention mechanisms have been proposed to improve object detection performance. Attention mechanisms are used to enhance important features and suppress unnecessary ones. There are three categories of attention mechanisms.

Spatial attention. The spatial attention mechanisms attempt to find important areas on feature maps and enhance them. Mnih et al. [139] proposed a recurrent model for visual attention. The model is able to extract information from an image and video by adaptively selecting a sequence of locations and only process those locations at high resolution. Ba et al. [140] proposed an attention-based model for detecting multiple objects in an image. Xu et al. [141] proposed an attention-based model for automatic image caption generation. The model is able to automatically focus on salient objects while generating captions. Gregor et al. [142] proposed the Deep Recurrent Attentive Writer (DRAW) for image generation. DRAW combines the spatial attention mechanism with a sequential auto-encoding framework, which is able to construct complex images. Jaderberg et al. [143] introduced a spatial transformer module that can be inserted into existing deep networks. The networks with the spatial transformer module are able to

²<https://github.com/BFLTeam>

actively spatially transform feature maps. Almahairi et al. [144] proposed the Dynamic Capacity Network (DCN). DCN is able to adaptively assign its capacity across different portions of the input data. It contains low-capacity and high-capacity subnetworks. The low-capacity subnetworks are applied to the input data and meanwhile, some RoIs are selected for high-capacity subnetworks based on an attention mechanism.

Channel attention. The channel attention mechanisms attempt to find features that are more informative and enhance them or suppress the others. Hu et al. [145] proposed the Squeeze-and-Excitation (SE) blocks. The SE blocks are able to enhance or suppress different channels in feature maps adaptively.

Hybrid/Block attention. The block attention mechanism combines the spatial attention mechanism and the channel attention mechanism. This attention mechanism, proposed by Woo et al. [146], is able to infer attention maps along the channel and spatial dimensions and then the attention maps are multiplied to the input feature map for further refinement.

These three types of attention mechanisms enhance the features based on the object saliency and the importance of the features. They neglect the relationship between objects from prior knowledge. In this research, the relationships among general objects, humans, tattoos, and text tattoos are used explicitly through prior knowledge-based attention mechanism (PKAM) to improve text tattoo and generic tattoo detection performance.

PKAM is designed to enhance the features within the portion of a particular object in the images. PKAM can be used to highlight tattoos such that text tattoos can be detected effectively. It can also be used to highlight objects and humans such that tattoos can be detected effectively. PKAM takes two inputs: one is the original image I_t , containing objects, humans, tattoos and text tattoos and the other is the features f_t^R extracted by a detector. For highlighting tattoos, the detector is trained on a tattoo dataset to extract tattoo features but for highlighting objects and humans, the detector is trained on a general image database to extract corresponding features. More clearly, f_t^R in Figs. 4.5 and 4.6 represents tattoo features and object and human features, depending on the detector. PKAM has two versions named spatial PKAM and channel PKAM. The spatial PKAM is based on element-wise operators to fuse input images with the features

Fig. 4.5: The prior knowledge-based attention mechanism. I_t is a text tattoo image. f_t^R is the tattoo features or object features extracted by the tattoo detector or object detector, respectively. I'_t is the text tattoo image enhanced by PKAM. The highlighted block illustrates the architecture of spatial prior knowledge-based attention mechanism (spatial PKAM).

from the detector and the channel PKAM is based on channel-wise operations to fuse input images with the features from the detector.

Spatial PKAM. Let the dimensions of f_t^R be $m \times n \times c$ and the dimension of I_t be $h \times w \times 3$. The spatial PKAM takes the features f_t^R and upsamples it to $h \times w \times c$ through the bilinear upsampling algorithm. Thus, its spatial resolution is the same as the input image I_t . For the spatial PKAM (see Fig. 4.5), a series of convolutional operations are applied to reduce the number of the channels of upsampled features from c to 3. After the convolution operations, normalization and ReLU are applied and the squeezed features and I_t are fused through element-wise operations such as element-wise sum and element-wise multiplication to obtain I'_t , which is submitted to the next CNN block for extracting tattoo or text tattoo features.

In the spatial PKAM, the bilinear upsampling operation can be replaced by deconvolution operation, depending upon the computational cost and the number of convolutional layers can be adjusted to achieve the optimal performance. The enhancement can be done by an element-wise sum or element-wise multiplication. The element-wise sum considers the normalized features as additional information. Thus, by summing it

Fig. 4.6: The architecture of channel PKAM.

up with the original image, it can enrich the information in the original image. Element-wise multiplication considers the features as weights. When the detector (see Fig. 4.5) is trained to detect objects and humans, these weights are used to highlight humans but suppress objects. When the detector is trained to detect tattoos, the weights are used to highlight tattoos, including text tattoos but suppress others.

Channel PKAM. Channel PKAM allows the network to learn how to select the important channels for each feature map locations. Fig. 4.6 illustrates the architecture of the channel PKAM. Inside the channel PKAM, f_t^R is first upsampled, same as the spatial PKAM. The upsampled features are denoted as f_t' . Then, f_t' is concatenated to I_t , which is the text tattoo image. The dimension of concatenated data is $h \times w \times (c + 3)$, where h is the image height, w is the image width and c is the number of channels of f_t' . A series of convolutional layers are applied to the concatenated data to enhance the relevant regions and suppress irrelevant regions and to reduce its dimension from $h \times w \times (c + 3)$ to $h \times w \times 3$. Then, normalization and ReLU operations are applied and obtained the enhanced image, I_t' which is submitted to the next CNN block for extracting tattoo or text tattoo features. The convolution operations can be considered as a channel selection scheme, where the filters weigh the importance of different channels.

Fig. 4.7: The architecture of TTDN-DA. f_{oh} is the feature map of ObjectHumanNet, f_{tatt} is the feature map of TattooNet and f_{text} is the feature map of TextTattooNet. In training stage 1, TextTattooNet and TPKAM are disconnected from the entire network, the classification head and bounding box regression head are connected to the feature map of TattooNet.

Fig. 4.8: Details architecture of subnetworks.

Fig. 4.9: Details architecture of ResNetUnit.

4.3 Text Tattoo Detection Network

4.3.1 Network Architectures

The proposed Text Tattoo Detection Network based on Double Attention (TTDN-DA) consists of three subnetworks and two spatial PKAMs. Fig. 4.7 gives the architecture of

TTDN-DA. The three subnetworks are named ObjectHumanNet, TattooNet and TextTattooNet. The ObjectHumanNet is designed to obtain the features of generic objects or human instances which are used in the PKAM to enhance the general tattoo images. The TattooNet is designed to obtain the features of generic tattoos which are used to enhance the text tattoo images. The TextTattooNet, which is the last subnetwork, is designed to extract the features of text tattoos and predict the bounding box of the text tattoos.

In this study, TridentNets [16] are used as the backbone of the subnetworks (detail architecture is as shown in Fig. 4.8). Each subnetwork consists of four ResNet [21] blocks (see Fig. 4.9). The first three ResNet blocks consist of different numbers of ResNet units. The fourth block consists of three branches with different sizes of the receptive fields and each branch with the same number of ResNet units shares weights with the other branches. All the subnetworks are connected using spatial PKAM modules. The two spatial PKAM modules in TTDN-DA are called OPKAM and TPKAM. OPKAM connects TattooNet to ObjectHumanNet, and TPKAM connects TextTattooNet to TattooNet. The classification head and bounding box regression head are connected to the feature map of TextTattooNet to output the predicted class labels and bounding boxes. A multi-task loss [11] L is employed for the network training:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda L_{loc}(t^u, v), \quad (4.1)$$

in which p is the predicted class label probability distribution, $p = (p_0, p_1, \dots, p_k)$ over $k + 1$ categories, u is the ground-truth class label, t^u is the predicted bounding box and v is the ground-truth bounding box. λ is 1 for non-background ground truth bounding boxes and 0 for background ground truth bounding boxes. L_{cls} is the classification loss and L_{loc} is the bounding box regression loss.

$$L_{cls}(p, u) = -\log p_u, \quad (4.2)$$

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^u - v_i), \quad (4.3)$$

in which p_u is the probability of the ground-truth class label in the predicted class label probability distribution and

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \quad (4.4)$$

Fig. 4.10: The architecture of TTDN-DA-V2. f_{tatt} is the feature map of TattooNet and f_{text} is the feature map of TextTattooNet. In training stage 1, TextTattooNet and TPKAM are disconnected from the entire network, the classification head and bounding box regression head are connected to the feature map of TattooNet.

4.3.2 Variants of TTDN-DA

To seek an effective architecture, another two architectures are designed to use the dependence among objects, humans, tattoos and text tattoos in different ways. The original TTDN-DA uses two PKAM and three TridenNets to achieve double attention for text tattoo detection. This approach makes the network very deep and large. As a result, training and testing time will be longer. Although it is not a serious problem in forensic investigation, because there is no real-time requirement, it is worth studying the necessity of using double attention. TTDN-DA-V2, which uses only two subnetworks and one spatial PKAM module to perform single attention based on the dependence between tattoos and text tattoos, is designed. Fig. 4.10 shows its architecture. Because of the removal of the ObjectHumanNet and the OPKAM, TTDN-DA-V2's size is around 33% smaller than TTDN-DA.

In TTDN-DA and TTDN-DA-V2, the spatial PKAMs are inserted in-between two subnetworks. In fact, it can also be inserted within a single subnetwork and use the features within the subnetwork for attention. TTDN-DA-V3 illustrated in Fig. 4.11 uses a channel PKAM within a single subnetwork. Channel PKAM is employed instead of spatial PKAM because of the dimension requirement of the last block in TTDN-DA-V3.

Fig. 4.11: The architecture of TTDN-DA-V3. The weights of first three blocks are loaded from a pretrained TattooNet of TTDN-DA-V2. f_{text} is the final feature map.

This design does not increase the depth of the network significantly. In TTDN-DA-V3, the inputs of PKAM are the feature maps f_{b3} of the third block and the original image, I_t . The size of f_{b3} is $m_3 \times n_3 \times c_3$ and the size of I_t is $h \times w \times 3$. f_{b3} is upsampled to $h \times w \times c_3$ through the bilinear upsampling algorithm and then concatenated with I_t to form a feature map with a dimension of $h \times w \times (3 + c_3)$. Next, a separated convolutional operation is applied to the feature map to enhance the features and downscale its dimension to match the dimension requirement of the last block.

4.4 Implementation and Training Scheme

The proposed TTDN-DA and its variants are implemented in MXNet based on the SimpleDet [147] framework. The code will be available for public access ³.

4.4.1 Training TTDN-DA

The proposed TTDN-DA consists of three subnetworks (ObjectHumanNet, TattooNet and TattooTextNet) and two PKAM modules (OPKAM and TPKAM). It is trained via

³<https://github.com/BFLTeam>

a three-stage training scheme. The first stage is to train the TattooNet with ObjectHumanNet fixed; the second stage is to train the TattooTextNet with both TattooNet and ObjectHumanNet fixed. Finally, in the third stage, TTDN-DA is finetuned entirely.

Stage 1. In this stage, tattoos are considered as the target object and the general objects and humans are considered as relevant objects for negative and positive attention, respectively. The training set is the entire NTU Tattoo V2 dataset and the ObjectHumanNet is pretrained on MS-COCO object detection dataset, which contains various objects and humans. In this training stage, ObjectHumanNet is fixed. When training starts, a training image, I_t is inputted into ObjectHumanNet. The outputs of ObjectHumanNet are the general object and human features extracted from I_t and denoted as f_{oh} . OPKAM takes both I_t and f_{oh} as inputs and outputs the enhanced tattoo image I'_t , as shown in Fig. 4.5, which is then inputted into TattooNet. Here, the classification loss and bounding box regression loss are both applied on the last features f_{tatt} of TattooNet. TextTattooNet is disconnected from the entire network in this stage. The training is performed for 6 epochs and only TattooNet and OPKAM are trained in this stage.

Stage 2. In this stage, TextTattooNet is connected to TattooNet via the TPKAM and the training set is the NTU Text Tattoo V1. Text tattoo is considered as the target object and tattoo is considered as a relevant object for attention. In this stage, the text tattoo image, which is a training image, is denoted as I_t . All the layers in ObjectHumanNet and TattooNet are fixed. When training starts, ObjectHumanNet outputs the general object and human features f_{oh} of I_t . OPKAM enhances I_t with its general object and human features and the enhanced image is inputted into TattooNet, which is already trained for general tattoo detection in stage 1. TattooNet outputs the tattoo features of the enhanced text tattoo image, denoted as f_{tatt} . TPKAM takes both I_t and f_{tatt} as inputs and outputs the double enhanced text tattoo image I''_t . Then, I''_t is inputted into TextTattooNet. In this stage, the classification loss and bounding box regression loss are both applied on the last features f_{text} TextTattooNet. The training is performed for 6 epochs and only TextTattooNet and TPKAM are trained in this stage.

Stage 3. In this stage, the training set is the NTU Text Tattoo V1 and entire TTDN-DA is finetuned end-to-end for additional 6 epochs.

The initial learning rate in the first and second stages is 0.01, while in the third stage, it is 0.0001 for ObjectHumanNet and TattooNet and 0.001 for TextTattooNet. The

momentum for all layers is 0.9 throughout the training period. A pretrained MS-COCO object detection TridentNet is used to initialize ObjectHumanNet. Xavier’s algorithm [148] is used to initialize all other new layers in TTDN-DA and the stochastic gradient descent (SGD) algorithm is used to train the network. The proposed TTDN-DA is trained in a batch size of 6 on 3 NVIDIA V100 GPU cards.

4.4.2 Training TTDN-DA Variants

TTDN-DA-V2 is trained again using a three-stage training scheme similar to the training scheme of TTDN-DA. In the first stage, TextTattooNet is disconnected from the entire network and TattooNet is trained on the entire NTU Tattoo V2 dataset from scratch for the first 6 epochs. The classification loss and bounding box regression loss are both applied on the last features f_{latt} of TattooNet. In the second stage, TextTattooNet and TPKAM are trained on the NTU Text Tattoo V1 for the next 6 epochs with TattooNet fixed. In the final stage, the entire TTDN-DA-V2 is finetuned on the NTU Text Tattoo V1 for the last 6 epochs. The batch size is set to 6 for training on 3 NVIDIA V100 GPU cards. The initial learning rate for the first and second stages is 0.01 and then reduced, respectively, to 0.0001 and 0.001 for TattooNet and TextTattooNet in the third stage.

Unlike the previous two versions, TTDN-DA-V3 utilizes a 2-stages training scheme. In TTDN-DA-V3, the first four blocks are loaded from a pretrained TattooNet from TTDN-DA-V2. In the first stage of training, blocks 1 to 3 are fixed, and only PKAM and block 4 are trained on the NTU Text Tattoo V1 for 6 epochs. In the second stage, the entire TTDN-DA-V3 is finetuned on the NTU Text Tattoo V1 for another 6 epochs. The initial learning rate in the first stage is 0.01 and in stage 2 is 0.0001 for blocks 1 to 3 and 0.001 for block 4. The batch size is the same as in the previous versions.

4.5 Experimental Results

To evaluate the performance of the proposed TTDN-DA and its variants and the importance of the additional information from general tattoos and objects, experiments were conducted on the NTU Text Tattoo V1 dataset. The experimental results of TTDN-DA and its variants are reported in Subsection 4.5.2. Results of ablation studies are given

in the following subsections. Subsection 4.5.5 and 4.5.6 list comparative results among TTDN-DAs, state-of-the-art object detectors and text detectors. Evaluation of general tattoo detection is reported in Subsection 4.5.7. The results of different PKAM modules are offered in Subsection 4.5.8. All the experiments were conducted on three NVIDIA V100 GPU cards with 32GB RAM each. In all the experiments, the datasets were divided into training, validation and testing sets. The distribution of each dataset is listed in Table 4.2.

4.5.1 Evaluation Indices

Precision, recall and F-score are three popular indices when evaluating object detectors or scene text detectors. Precision is the fraction of correctly detected instances (true positive) among all the detected instances; recall is the fraction of correctly detected instances (true positive) among all the ground truth instances. F-score, or F-measure is calculated from precision and recall as described in Eq. 4.5.

$$F\text{-score} = 2 * AP * AR / (AP + AR), \quad (4.5)$$

Precision indicates “how valid the detection results are”, it can be seen as a measure of quality, higher precision means the network detects more correct instances. Recall indicates “how complete the detection results are”, it can be seen as a measure of quantity, higher recall means the network misses less ground truth instances.

For text tattoo detection performance evaluation, average precision (AP) along with AP_{50}/AP_{75} , average recall (AR) and F-score are used as the performance indices. Here AP/AR is the mean value of precisions/recalls that calculated using the intersection of union (IoU) threshold from 0.50 to 0.95 with step 0.05. It means that any detected instance which has an intersection area above the IoU threshold with the ground truth instances is considered as a true positive detection. AP_{50}/AP_{75} is the precision calculated with IoU 0.50/0.75.

4.5.2 Text Tattoo Detection with TTDN-DA and its Variants

The proposed TTDN-DA and its variants, including TTDN-DA-V2 and TTDN-DA-V3, were evaluated on the NTU Text Tattoo V1 and the evaluation results are listed

in Table 4.3. The PKAMs in the proposed TTDN-DA and TTDN-DA-V2 are spatial PKAMs (see Fig. 4.5). The channel PKAM (Fig. 4.6) is used in TTDN-DA-V3 due to the dimension requirement of the last block in TTDN-DA-V3. The results show that TTDN-DA achieves the best performance with an AP_{50} of 0.740. TTDN-DA-V2 achieves an AP_{50} of 0.729. In other words, the object and human attention in ObjectHumanNet offers AP_{50} improvement of 0.011. Although TTDN-DA is larger than TTDN-DA-V2, it is still a better choice because, in forensic investigation, accuracy is much more important than network size and speed. TTDN-DA-V3, on the other hand, achieves a result with an AP_{50} of 0.711. Since TTDN-DA-V3 is constructed with a single TridentNet, it is suitable for fast training or limited GPU memory circumstance. Fig. 4.12 and Fig. 4.13 show some detection results from TTDN-DA. Fig. 4.12 shows that TTDN-DA can detect text tattoos in images taken in highly uncontrolled and uncooperative environments. Fig. 4.13 shows that TTDN-DA neither wrongly detects general text nor text like tattoo patterns as text tattoos. Some failure examples were demonstrated in the last row of Fig. 4.13. According to these failure examples, it can be observed that the network still has some limitations when dealing with text tattoos that were close to general tattoos or the color of general tattoos were close to the color of text tattoos in the same image.

4.5.3 Text Tattoo Detection without Attention

As discussed previously, PKAM plays an important role in TTDN-DA and its variants. To examine the effectiveness of PKAMs, two networks called non-Atten-TTDN-V1 and non-Atten-TTDN-V2 were constructed. Fig. 4.14 illustrates the architectures of non-Atten-TTDN-V1 and non-Atten-TTDN-V2. Non-Atten-TTDN-V1 was constructed with three TridentNets. The TridentNets in non-Atten-TTDN-V1 were connected directly without any PKAMs. This guarantees that non-Atten-TTDN-V1 and TTDN-DA have the same network depth. Non-Atten-TTDN-V2 was constructed with only two TridentNets, which were connected directly without PKAMs. This guarantees that non-Atten-TTDN-V2 and TTDN-DA-V2 have the same network depth. Both non-Atten-TTDN-V1 and non-Atten-TTDN-V2 were trained end-to-end on the NTU Tattoo V2 dataset for 6 epochs and then finetuned on the NTU Text Tattoo V1 for 12 epochs. The total training epochs were 18, which were the same as TTDN-DA and TTDN-DA-V2.

Fig. 4.12: TTDN-DA text tattoo detection results. The red rectangle areas are enlarged on the right side of each image to show more details. The blue bounding boxes mark the detection results. Images are resized to fill in the page.

The initial learning rate for non-Atten-TTDN-V1 and non-Atten-TTDN-V2 was 0.01. The results listed in Table 4.3 show that non-Atten-TTDN-V1 and non-Atten-TTDN-V2 perform worse than TTDN-DA and TTDN-DA-V2, which have the same network depths respectively. Their performance differences are very significant. Without the PKAMs, non-Atten-TTDN-V1 and non-Atten-TTDN-V2 can easily suffer from over-fitting because of their depths. TTDN-DA and TTDN-DA-V2 are able to avoid the over-fitting problem because the PKAMs highlight relevant features and suppress irrelevant features, such that learning in deep layers becomes easier.

Fig. 4.14: The architecture of non-Atten-TTDN-V1 and non-Atten-TTDN-V2.

4.5.4 Text Tattoo Detection with General Tattoo Detector

Text tattoo can be considered as a subcategory of tattoos, therefore some might argue that a well-trained general tattoo detector should be able to detect text tattoos effectively. To evaluate whether general tattoo features are sufficient for performing text tattoo detection, an experiment was conducted with a single TridentNet, which was trained on the non-text tattoo images in the NTU Tattoo V2 and tested on the NTU Text Tattoo V1. This TridentNet is called tatt-TridentNet-V1. Another experiment was conducted with a single TridentNet which was trained on the NTU Tattoo V2 but without fine-tuning on the NTU Text Tattoo V1. As with tatt-TridentNet-V1, it was tested on the NTU Text Tattoo V1. This TridentNet is called tatt-TridentNet-V2. The initial learning rate for training tatt-TridentNet-V1 and tatt-TridentNet-V2 was 0.01, and the batch size was 6 on the GPUs. The evaluation results reported in Table 4.3 show that tatt-TTDN-V1 performs worse than tatt-TTDN-V2, because tatt-TTDN-V2 was trained on the NTU Tattoo V2 containing not only non-text tattoo images but also text tattoo images. This result indicates that non-text tattoo features cannot fully represent text tattoo images. Tatt-Trident-V2 performs better than the TridentNet in Table 4.3 which

was trained only on the NTU Text Tattoo V1. This result pinpoints that non-text tattoo images can improve text tattoo detection performance. However, tatt-TridentNet-V2 still performs slightly worse than TridentNet* which was trained on the NTU Tattoo V2 and finetuned on the NTU Text Tattoo V1, because tatt-TridentNet-V2 trained on the NTU Tattoo V2 mixes the features of non-text tattoo and text tattoo and lowers down the text tattoo detection accuracy. It indicates that the non-text tattoo features and text tattoo features are not the same. Thus, text tattoos should not be regarded as general tattoos in detection. Table 4.4 summarizes the differences between the evaluations of tatt-TridentNet-V1, tatt-TridentNet-V2, TridentNet and TridentNet*.

4.5.5 Comparison with State-of-the-Art Object Detectors

TTDN-DA and its variants are compared with the state-of-the-art object detectors, including Faster R-CNN [61], Mask R-CNN [12], Cascade R-CNN [102], CenterNet [149], RetinaNet [14], CSPNet [114], DetectoRS [115] and TridentNet [16]. The backbone architectures of those object detectors include ResNet [21] (Fig. 4.15(a)), ResNeXt [22] (Fig. 4.15(b)) and Hourglass [23] (Fig. 4.15(c)). The comparison results are reported in Table 4.3. Fig. 4.17 shows the precision and the recall curves. To make a fair comparison, each method was evaluated twice. In the first evaluation, the NTU Text Tattoo V1 was used for both training and testing and all the methods were trained for 6 epochs to make sure that the training of all those methods converged. In the second evaluation, the methods were trained on the NTU Tattoo V2 dataset for 6 epochs first, then were finetuned on the NTU Text Tattoo V1 for another 6 epochs. The initial learning rate was 0.01 in all the training. The batch size was 6 on the GPUs. The comparison results show that, among the state-of-the-art object detectors, TridentNet achieves the best performance in terms of AP₅₀, which is the reason why TridentNet is used as the subnetwork in TTDN-DA and its variants. The results also show that TTDN-DA, TTDN-DA-V2 and TTDN-DA-V3 outperform TridentNet. It is because the PKAMs provide additional attention information to the detectors. Fig. 4.16 gives some detection results from TTDN-DA and the state-of-the-art object detectors. The results show that TTDN-DA is able to detect text tattoos without influenced by other tattoos. Other methods detect general tattoos as well as text tattoos. TTDN-DA can also handle different text tattoo styles and text tattoo colors (see Fig. 4.12 and Fig. 4.13).

Fig. 4.15: Architectures of (a) ResNet block [21], (b) ResNeXt block [22] and (c) Hour-Glass network [23].

4.5.6 Comparison with State-of-the-Art Scene Text Detectors

Text tattoo can be considered as subcategory of scene text. How the scene text detectors perform in text tattoo detection task need to be evaluated. Therefore, in addition to object detectors, TTDN-DA and its variants, including TTDN-DA-V2 and TTDN-DA-V3 are also compared with state-of-the-art scene text detectors, including EAST [82], SegLink [6], PixelLink [86], PAN [87], PSENet [7], SPCNet [88], TextFuseNet [91] and ABCNet [8]. The backbone architectures of those scene detectors include ResNet [21] (Fig. 4.15(a)) and VGG [24] (Fig. 4.18). All these detectors were pretrained text detection models and finetuned on the NTU Text Tattoo V1. Table 4.4 shows that

Fig. 4.16: Text tattoo detection results from close up images for comparisons. From top row to bottom row: TTDN-DA, TridentNet*, CSPNet*, DetectoRS*, Cascade R-CNN*, CenterNet*, RetinaNet*, Mask R-CNN* and Faster R-CNN*. The blue bounding boxes mark the detection results.

Fig. 4.17: Precision and recall curves of TTLN-DA, TridentNet* and ABCNet. TridentNet* and ABCNet are the best generic object detector and scene text detector, respectively on the NTU Tattoo V1 (Table 4.3).

some state-of-the-art text detectors cannot handle text tattoo detection well enough especially for images with non-text tattoos (see Fig. 4.19). These results indicate that text tattoos are different significantly from general text in images. These differences have been discussed in the introduction and illustrated in Fig. 4.3.

Table 4.3: Experimental results on text tattoo detection. * mark means that the networks is trained on the NTU Tattoo V2 first and finetuned on the NTU Text Tattoo V1.

Experiments	Architectures	AP	AP ₅₀	AP ₇₅	AR	F-score
TTDN-DA	ResNet-101	0.393	0.740	0.380	0.579	0.468
TTDN-DA-V2	ResNet-101	0.382	0.729	0.356	0.575	0.459
TTDN-DA-V3	ResNet-101	0.370	0.711	0.343	0.576	0.451
non-Atten-TTDDN-V1	ResNet-101	0.187	0.336	0.154	0.423	0.259
non-Atten-TTDDN-V2	ResNet-101	0.239	0.428	0.223	0.403	0.300
tatt-TridentNet-V1	ResNet-101	0.334	0.660	0.305	0.528	0.409
tatt-TridentNet-V2	ResNet-101	0.342	0.684	0.305	0.550	0.422
TridentNet	ResNet-101	0.323	0.666	0.279	0.521	0.399
TridentNet*	ResNet-101	0.350	0.689	0.315	0.553	0.427
CSPNet	ResNeXt	0.239	0.596	0.128	0.231	0.235
CSPNet*	ResNeXt	0.326	0.681	0.280	0.310	0.318
DetectoRS	ResNet-101	0.312	0.652	0.267	0.484	0.379
DetectoRS*	ResNet-101	0.340	0.679	0.288	0.524	0.412
Cascade R-CNN	ResNet-101	0.321	0.615	0.287	0.507	0.393
Cascade R-CNN*	ResNet-101	0.349	0.660	0.324	0.528	0.420
CenterNet	Hourglass-104	0.317	0.623	0.274	0.492	0.386
CenterNet*	Hourglass-104	0.335	0.655	0.314	0.530	0.411
RetinaNet	ResNet-101	0.298	0.591	0.259	0.476	0.367
RetinaNet*	ResNet-101	0.315	0.645	0.297	0.509	0.389
Mask R-CNN	ResNet-101	0.302	0.598	0.264	0.481	0.371
Mask R-CNN*	ResNet-101	0.324	0.643	0.273	0.511	0.397
Faster R-CNN	ResNet-101	0.284	0.587	0.244	0.457	0.350
Faster R-CNN*	ResNet-101	0.303	0.617	0.258	0.472	0.369

4.5.7 Evaluation of General Tattoo Detection

TTDN-DA and TTDN-DA-V2 utilize general tattoo features as attention to improve the text tattoo detection performance. They are also able to perform general tattoo detection by using the outputs of TattooNet. For detecting general tattoos, the classification head

Table 4.4: The differences between the evaluations of tatt-TridentNet-V1, tatt-TridentNet-V2, TridentNet and TridentNet*.

Experiments	Training sets	Finetune on NTU Text Tattoo V1	AP ₅₀
tatt-TridentNet-V1	Non-text tattoo	No	0.660
tatt-TridentNet-V2	NTU Tattoo V2	No	0.684
TridentNet	NTU Text Tattoo V1	No	0.666
TridentNet*	NTU Tattoo V2	Yes	0.689

Fig. 4.18: Architectures of VGG-16 network [24].

and bounding box regression head are connected to the feature map of TattooNet directly. The TextTattooNet in TTDN-DA and TTDN-DA-V2 are skipped. To evaluate their performance for general tattoo detection, TTDN-DA and TTDN-DA-V2 were trained on NTU Tattoo V2 and tested on NTU Tattoo V2. As in the previous experiments, the state-of-the-art detectors are employed for comparisons. These object detectors were trained on the NTU Tattoo V2 too.

In TTDN-DA, TattooNet is the second subnetwork that follows the ObjectHumanNet. The ObjectHumanNet provides the general object features, which are used as the

Table 4.5: Compare with state-of-the-art scene text detectors.

Experiments	Architectures	AP	AP ₅₀	AP ₇₅	AR	F-score
TTDN-DA	ResNet-101	0.393	0.740	0.380	0.579	0.468
TTDN-DA-V2	ResNet-101	0.382	0.729	0.356	0.575	0.459
TTDN-DA-V3	ResNet-101	0.370	0.711	0.343	0.576	0.451
EAST	ResNet-101	0.304	0.608	0.274	0.162	0.211
SegLink	VGG-16	0.188	0.462	0.127	0.140	0.160
PixelLink	VGG-16	0.148	0.321	0.117	0.201	0.170
PAN	ResNet-101	0.314	0.641	0.286	0.384	0.345
PSENet	ResNet-101	0.171	0.377	0.142	0.170	0.171
SPCNet	ResNet-101	0.043	0.118	0.023	0.591	0.080
TextFuseNet	ResNet-101	0.331	0.649	0.307	0.470	0.388
ABCNet	ResNet-101	0.327	0.680	0.403	0.076	0.123

attention in the OPKAM, and TattooNet takes the enhanced general tattoo images as training data. Therefore, TTDN-DA outperforms TTDN-DA-V2 and the state-of-the-art object detectors as reported in Table 4.5. TTDN-DA-V2 performs similarly as TridentNet because it does not use the object features as attention for tattoo detection and its tattoo detection network is the same as TridentNet.

4.5.8 Evaluation of PKAM Variants

Two types of PKAM architecture are introduced in this work to study the role of attention mechanism in text tattoo detection. One is spatial PKAM and another one is channel PKAM. The main difference between the spatial PKAM and the channel PKAM is the fusion operations. Spatial PKAM resizes the feature maps produced by the relevant object detector to the same dimension of target object images and use an element-wise operation(sum/multiplication) to fuse them together. Channel PKAM upsamples the feature maps produced by the relevant object detector to the same size of target object images but retains the channel number, then the upsampled features maps are concatenated to the target object image, a series of convolutions followed to resize the concatenated feature map to meet the dimension requirements of the following layers. To examine their

Fig. 4.19: Text tattoo detection results from close up images for comparisons. From top row to bottom row: TTDN-DA, EAST, PixelLink, SegLink, PAN, PSENet, SPCNet, TextFuseNet and ABCNet. The blue bounding boxes mark the detection results.

Table 4.6: Experimental results on the general tattoo detection. TTDN-DA and TTDN-DA-V2 use the outputs of TattooNet as the general tattoo detection results.

Experiments	AP	AP ₅₀	AP ₇₅	AR	F-score
TTDN-DA	0.445	0.807	0.433	0.587	0.506
TTDN-DA-V2	0.414	0.770	0.390	0.565	0.478
TridentNet*	0.416	0.760	0.403	0.571	0.481
CSPNet*	0.371	0.735	0.334	0.329	0.349
DetectoRS*	0.388	0.733	0.366	0.533	0.449
Cascade R-CNN*	0.396	0.738	0.375	0.533	0.454
CenterNet*	0.407	0.774	0.382	0.552	0.469
RetinaNet*	0.344	0.686	0.304	0.544	0.421
Mask R-CNN*	0.373	0.758	0.344	0.533	0.439
Faster R-CNN*	0.393	0.753	0.363	0.520	0.448

Table 4.7: Text tattoo detection results from different PKAM modules. Experiments were conducted on TTDN-DA-V2.

Fusion Types	AP	AP ₅₀	AP ₇₅	AR	F-score
Spatial-wise (Sum)	0.382	0.729	0.356	0.575	0.459
Spatial-wise (Multiplication)	0.342	0.686	0.304	0.528	0.415
Channel-wise	0.356	0.687	0.336	0.547	0.431

performance difference, experiments were conducted with TTDN-DA-V2. The evaluation results are reported in Table 4.6. The results show that the spatial PKAM with the sum operation has the best performance. It is because the sum operation works in a way similar to the mechanism of residual network [21], where tattoo features are used as the additional attention information. The multiplication operation, on the other hand, uses tattoo features as weights to modify the inputted text tattoo information. The channel PKAM selects 3 channels out of all the channels of tattoo feature maps and inputted text tattoo image. Lots of raw information in inputted text tattoo images are replaced by the tattoo features.

4.6 Summary

To utilize the content of tattoos in forensic investigation, detecting tattoos is the first and essential step. The current tattoo detection studies applied generic object detectors on small scale tattoo datasets. Text tattoos, which have rich personal information, were totally neglected. This chapter presents a solution for text tattoo detection, which is a text tattoo detection network using double attention.

For studying text tattoo detection, two new datasets, the NTU Tattoo V2 and the NTU Text Tattoo V1 are established. The NTU Tattoo V2 is the largest tattoo dataset and the NTU Text Tattoo V1 is the first and only text tattoo dataset at the moment. To use the dependence among objects, humans, tattoos and text tattoos, a prior knowledge-based attention mechanism (PKAM) is proposed. PKAM uses the features of relevant objects (humans, general tattoos) to enhance the images of target objects (general tattoos or text tattoos). For detecting text tattoos, a text tattoo detection network with double attention (TTDN-DA) architecture is designed and examined. TTDN-DA uses two PKAM to attend on human and general tattoos, respectively. The human features enhance the general tattoo instances and the general tattoo features enhance text tattoo instances. A three-stage training scheme is designed accordingly to train TTDN-DA. In order to reduce the network size and speed up the training and inference, another two variants of TTDN-DA are also proposed. TTDN-DA-V2 uses one PKAM to connect two subnetworks and attend on general tattoos only. TTDN-DA-V3 inserts PKAM into one single subnetwork and network size reduces significantly.

Precisions, recalls and F-score are used as the indices in evaluating the performances of TTDN-DA, TTDN-DA-V2 and TTDN-DA-V3. Some state-of-the-art object detectors and scene text detectors are also evaluated for comparisons. According to the evaluation results, TTDN-DA achieves the best results among all the state-of-the-art object detectors and scene text detectors for both general tattoo and text tattoo detection tasks. Extensive experiments were performed to evaluate the networks with the same subnetworks as TTDN-DA but without PKAMs. The evaluation results indicate the importance of the dependence and effectiveness of PKAM. Networks with the same architecture but trained with different dataset combinations were also evaluated, the evaluation results indicate the differences between the general tattoo instances and text tattoo instances, which highlights the importance of establishing a text tattoo dataset.

Chapter 5

Portrait Tattoo Synthesis

The size of a dataset impacts the performance of deep neural networks. The tattoo image classification results reported in [26] showed that better performance was achieved with a larger training dataset. The tattoo detection and text tattoo detection results reported in chapter 4 also pointed out that a larger training dataset can improve the detection performance. However, a large training dataset is not always available for all computer vision tasks such as classification, detection, etc. Besides, imbalanced data is also a common problem in deep learning. Tattoo detection and text tattoo detection also suffered from these issues. For example, the NTU Text Tattoo V1, although is the first and the largest text tattoo dataset, only contains 9,067 text tattoo images, hence it is only useful in fine-tuning a pretrained tattoo detector instead of training a text tattoo detector from scratch.

In addition to train tattoo detectors, the tattoo image itself is considered to be a piece of artwork. Recent studies have shown that the proportion of the US citizens, aged 18-50, having tattoos is over 20% [150, 151], aged 18-29 is more than 40% and is increasing [151, 152]. Similarly, in the UK, it is estimated that 20% of the whole population and 30% of young adults are tattooed [153]. Meanwhile, with the prevalence of tattoos, there is also an increase in regrets. In the US, in 2003, 17% of tattooed people regretted getting a tattoo. Then in 2015, the number increased to nearly a quarter (23%). One of the main reasons is that the tattoos do not fit the expectation of clients [152]. The misunderstanding between clients and tattooists can cause bitter quarrels and even legal disputes [154]. Currently, there is no effective way to see a tattoo before putting it on the skin. Because of the permanence of a tattoo, the removal is expensive, uncomfortable,

time-consuming and leaves scars [155]. Using a laser to remove a tattoo is in fact a surgery with risk. Some of these surgeries can go horribly wrong [156]. A digital tattooing algorithm would allow a person to be tattooed and his/her tattooist to see the final tattoo in an image first before putting it on skin. It can help the person make better decisions on the reference tattoo photo, size, location, etc. It also supports the tattooist's work by reducing misunderstanding on tattoo design.

Therefore, how to generate a large number of high-quality tattoo images is essential not only for training tattoo detectors but also for tattoo designing. Chapter 2 reviews some state-of-the-art image synthesizing methods. However, they either require a large scale dataset for training or their synthetic images are not visually close to real tattoo images. In this chapter, we present Digital Tattooing (DT), a tattoo image synthesizing algorithm that simulates the real tattooing procedure physically¹. We concentrate on applying the DT algorithm to generate synthetic portrait tattoo images because portrait tattoos are very meaningful to a lot of people and the portrait photos contain sufficient details to evaluate the performance of the DT algorithm. In addition to synthesizing portrait tattoos, the DT algorithm can also be used to produce tattoo stickers. Tattoo stickers are synthesized portrait tattoo images with transparent background so that they can be easily added to other background images. Since the final goal of designing DT is to generate more tattoo and text tattoo images for training tattoo retrieval systems and tattoo detectors, we also try to use DT to generate text tattoo images.

5.1 Portrait Tattoos

Portrait tattoo can be considered as art on skin made by the tattooist. Making a portrait tattoo of a loved person to pay tribute or express love has profound meaning to many people. This type of tattoos is very special because they love the person and have high expectations of the tattoos. Fig. 5.1 shows some portrait tattoo samples. From those samples, it can be observed that (1) most portrait tattoos are single color, (2) the color of hair, facial landmarks are dark (close to tattoo ink color) and the color of the skin area in the portrait tattoo is light (close to skin color), (3) some portrait tattoo has a

¹The work in this chapter has been published in [157]

Fig. 5.1: Portrait tattoo samples.

shadow outside the portrait tattoo area, this effect makes the portrait tattoo vivid and stereoscopic.

However, up until now, there is no algorithm specifically designed for synthesizing high-quality portrait tattoos. Hence, Digital Tattooing (DT) algorithm is proposed to synthesize portrait tattoos that are visually close to real portrait tattoo photos. Those observations mentioned above are used as our guidelines when we design the DT algorithm.

5.2 A Tattoo Needle Model

The proposed DT algorithm uses a series of tailor-made image processing methods and a tattoo needle model to perform digital tattooing on skin images. The advantages of this approach over the learning-based approaches are: (1) it does not rely on very large training datasets; (2) parameters with explainable meanings can provide users the control over final results; (3) it is applicable to very high-resolution images without suffering the problems from training large deep learning models, e.g., overlearning and the collapse of GAN.

Fig. 5.2: Tattoo needles for modern tattooing [25].

In general, tattooists do not use a single tattoo needle but use multiple needles (see Fig. 5.2) with different configurations to tattoo different facial components in portrait tattoos. There are two common groups of tattoo needles, liner needles and shadow needles. The liner needles are mostly utilized for outlining tattoos and tattooing thin components and the shadow needles are utilized for tattooing shadow and coloring different facial components. The needles also have different sizes. Thus, the sizes of the needles and their configurations are taken into account in the proposed digital tattoo needle model, where each digital tattoo needle is represented by a solid circle of diameter d in a binary mask. There are three common sizes of tattoo needles: 0.30mm, 0.35mm and 0.40mm. In the proposed model, one pixel of the mask represents 0.01mm. In other words, the digital tattoo needles correspond to solid circles with diameters of 30, 35 and 40 pixels in the mask. In order to model configurations with multiple needles, the size of the mask is set to 615×615 pixels. There are several common tattoo needle configurations shown in Fig. 5.3. Since in the experiments, the DT algorithm takes portrait photos with an average size of $1,000 \times 600$ pixels as inputs, the needle masks are rescaled to 63×63 pixels. As with tattooists, the DT algorithm uses needle masks with different configurations to digitally tattoo different facial components.

Fig. 5.3: Digital tattoo needle masks with different configurations. First row (from left to right): RL-3-30, RL-5-30, RL-7-30, RL-9-30. Second row (from left to right): RS-3-30, RS-5-30, RS-7-30, RS-9-30. Third row (from left to right): FS-5-30, FL-7-30, M1-7-30, M2-13-30. R means the needles are organized in a round shape, F means the needles are organized in a flat shape. M means the needles are organized in a Magnum style. L means the needle is used to draw lines. S means the needle is used to draw shadows.

5.3 The Digital Tattooing Algorithm

Before presenting the proposed DT algorithm, a visual analysis is first given to understand the essential elements in high-quality portrait tattoos. Figure 5.4 compares high-quality portrait tattoos and photos. The first and utmost importance is that the portrait tattoos and the persons are identifiable without unnecessary distortion (Figure 5.4(a) and (e)), which is different from the previous research for transferring face to emoji and face to cartoon [158–160]. The shadow effect in portrait tattoos is strong, comparing with

Fig. 5.4: Comparison of high-quality portrait photos and portrait tattoos. (a)-(d) are high-quality portrait tattoos and (e)-(h) are high-quality portrait photos. (e) is a black and white photo of the person in (a). Note that the red in (c) is not color of tattoo ink. It is due to the skin slightly bruised from some blood leaking underneath the top layers of the skin.

portrait photos. The areas close to facial features, including eyes, nose, mouth, ears and jaw, have a stronger shadow effect than the areas far away from them in general. The pixels on the facial skin of the portrait tattoos, in particular on the foreheads and cheeks, are not completely filled with ink. They in fact mix with colors of the tattoo ink and the original skin color of the persons. Furthermore, by zooming in a small region of a tattoo (see Figure 5.5), we can see that tattoo is noisy. The color is not smooth and there are many dots, similar to Gaussian noise in image processing research. This uneven color is due to the tattooing process, where it is hard for tattooists to control the exact amount of ink left in the skin. The dots are left by tattoo needles. To synthesize high-quality portrait tattoos, these properties have to be taken into consideration.

To obtain high-quality portrait tattoos with more realistic tattoo textures, tattooing

Fig. 5.5: Illustration of the noise in a portrait tattoo. (left) A high-quality tattoo and the patch from the white square in (right).

Fig. 5.6: The schematic diagram of the proposed digital tattooing (DT) algorithm.

is simulated in the proposed DT algorithm. Fig. 5.6 shows the schematic diagram of the DT algorithm. It takes a portrait photo I_{face} , a reference portrait tattoo I_{tatt} and a skin image I_{skin} as inputs. A portrait photo is a person whom a user loves and wants to put on his/her skin. The region in the portrait photo that will be transformed into a synthetic portrait tattoo is selected by the user manually. It should include face and hair and its boundary can be irregular. A fully automatic method is not employed because different users have different preferences. The input portrait tattoo is used as a reference to control the color of the final synthetic tattoo. Because single color portrait tattoos are more popular, in this study, it is assumed that the input portrait tattoo is single color. Fig. 5.4(a)-(d) show single color portrait tattoos. The skin image is used as the target image and a skin region is cropped to indicate where the client wants to tattoo. The entire DT algorithm includes the following steps: pre-processing, shadow simulation, color mapping, digital tattooing and post-processing. Tunable parameters are provided in each step to change the final outlook of synthetic portrait tattoo images.

5.3.1 Portrait Photo Pre-processing

Before using the needle models to perform digital tattooing, the input portrait photo is pre-processed. First, the input portrait photo I_{face} and the input skin image I_{skin} are converted to the LUV space and their luminance components are denoted as L_{face} and L_{skin} , respectively. Then, the intensity of L_{face} is rescaled by the equation

$$L_{face} = L_{face} \times (\max(L_{skin})/\max(L_{face})), \quad (5.1)$$

to match the intensity of the input skin image better.

The rescaled luminance component L_{face} is combined with the original U and V components and converted back to the RGB space. This luminance-adjusted image is denoted as I_{la} . A guided filter [161] is applied on I_{la} to suppress noise and smooth the image but preserve edges. The smoothed result is converted to a grayscale image [162], which is denoted as I_{gray} . Fig. 5.7(a) shows an input image and Fig. 5.7(b) shows its grayscale filtered image. Portrait sketches retain key facial features and reflection on hair, which are important elements of high-quality portrait tattoos (Fig. 5.7(c)). Thus,

the proposed algorithm uses the XDoG method [163] to generate a binary sketch image I_{sketch} (Fig. 5.7(c)) from the smoothed grayscale portrait image I_{gray} .

The XDoG method provides high-quality sketches but it removes some facial details. In addition, there are not too many edges in the face's sketch. The edges are useful to generate the shadow effect. Thus, an edge detector [164] is also applied to I_{gray} . The resultant edge image is denoted as I_{edge} . The sketch and the edge image are combined to form a new binary image, which is called a sketch-and-edge image and denoted as I_{se} (Fig. 5.7(d)). Since the two images are binary and the goal is to retain more edges for generating shadow effect, the bitwise XOR operator is applied,

$$I_{se} = I_{sketch} \oplus I_{edge}, \quad (5.2)$$

where $\oplus$ is an elementwise XOR operator.

Comparing with photo and tattoo portraits (see Fig. 5.4), the shading effect around the edges in tattoo portraits is stronger. The ink becomes darker and denser when it is closer to the edge. This property can be simulated by employing a distance map. To compute a distance map of I_{se} , a distance transform is used. Each pixel value in the distance map represents the shortest Euclidean distance between the pixel and any black pixel in I_{se} . In the implementation, the linear time method [165] is used to calculate the distance map. The distance map is denoted as D_{face} because it is computed from the whole face (Fig. 5.7(e)).

To use the facial features more precisely, a facial landmark detection method [166] is applied to the original portrait (Fig. 5.7(f)). The edges of each landmark, i.e., eyes, eyebrows, nose, mouth, and jaw, determined by the method are isolated to form binary images. Then, the distance transform is applied to each of the binary images and the values in the distance maps are rescaled to the range between zero and one because they are used as weights in the following processes. In total, five rescaled distance maps denoted as $D_1, D_2, \dots, D_5$, for respectively eyes, eyebrows, nose, mouth, and jaw are obtained. These distance maps are used to modify the distance map D_{face} and generate

$$D_{final}(p) = D_{face}(p) \times D_{min}(p), \quad (5.3)$$

where $p = [x, y]$ is a pixel location and D_{min} is defined as

$$D_{min}(p) = \min_{k \in \{1, \dots, 5\}} D_k(p). \quad (5.4)$$

Fig. 5.7: Illustration of portrait photo pre-processing. (a) an input portrait photo, (b) the grayscale image I_{gray} , (c) the sketch image I_{sketch} obtained from the XDoG method, (d) the sketch-and-edge image I_{se} , (e) the distance map D_{face} from (d), (f) the facial landmarks from (a), (g) the final distance map D_{final} from (e) and the facial landmark distance maps. (h) the final result I_{pre} from the portrait photo pre-processing. For better visualization, the distance maps in (e) and (g) are processed.

For each pixel p in the D_{min} , its distance value $D_{min}(p)$ is assigned to the minimum value of all five distance maps of facial landmarks at the location of pixel p . Note that $D_{final}(p)$, $D_{face}(p)$ and $D_{min}(p)$ are the corresponding values at the location p . The same notations will be used in the rest of the chapter. The final distance map D_{final} is applied to the grayscale image I_{gray} non-linearly to obtain a strong shadow effect. More clearly, the proposed algorithm uses the equation:

$$I_{gray}(p) = I_{gray}(p) \times (D_{final}(p) + \epsilon)^\rho, \quad (5.5)$$

where ϵ and ρ are parameters controlling the shadow effect. Then adaptive histogram

equalization is applied to I_{gray} and the final pre-processed portrait photo result is obtained and denoted as I_{pre} shown in Fig. 5.7(h).

5.3.2 Shadow Simulation

In high-quality portrait tattoos (Fig. 5.4), shadow around the face provides vital visual effects. To automatically simulate the shadow area, a shadow simulation algorithm is proposed. The details of the shadow simulation algorithm are presented as following. First, the alpha matte of the input portrait photo I_{face} , which is denoted as I_α , is computed [167]. In this calculation, the face in I_{face} is considered as a foreground and the other region is considered as a background. The alpha matte I_α is a grayscale image, whose values on the face region are 1, the values on the background are 0 and the values on the boundary between them are between 0 and 1. Then, I_α is iteratively enlarged to $(H + tK) \times (W + tK)$ pixels, where $H \times W$ is the size of I_α , t is the iteration number and K is a parameter controlling the growth of the shadow area in each iteration. In the experiments, K is set to 2. The central part of the enlarged I_α with a size of $H \times W$ pixels is cropped and denoted as $I_{\alpha t}$. Note that $I_{\alpha t}$ is a grayscale image with values from the interval $(0, 1]$ in the face and its surroundings and with values of zero in the background. In the DT algorithm, the non-linear function, $Y(t) = \exp(-\alpha t)^\beta$, $t \in \{1, \dots, s\}$, where s is used to control the size of the shadow and α and β control the shape of the function. In the implementation, s is determined automatically by the minimum distance from the portrait boundary to the image boundary, α and β are set manually. $Y(t)$ is between 0 and 1 and is used to weigh down the non-zero pixels in $I_{\alpha t}$, which is enlarged from I_α . To retain the non-zero pixels in $I_{\alpha t}$ generated from the previous iterations, where the shadow effect will be added, a maximum operator is used. More precisely, the equation below is applied

$$I_{\alpha f}(t + 1) = \max(I_{\alpha f}(t), I_{\alpha t} \times Y(t)), \quad (5.6)$$

where t is from 1 to s and $I_{\alpha f}(1)$ is set to I_α . After s iterations, $I_{\alpha f}(s + 1)$ is obtained. To simplify the notation, $I_{\alpha f}(s + 1)$ is denoted as $I_{\alpha w}$. It is used with I_α to linearly reweight I_{pre} to simulate shadow effect around the face. More clearly,

$$I_{enh} = I_{pre} \times (1 - I_{\alpha w}) + I_{pre} \times I_\alpha, \quad (5.7)$$

Algorithm 1: Shadow Simulation

```
1: procedure ShadowSimulation( $I_\alpha, I_{pre}, s, \alpha, \beta$ )
2:    $T \leftarrow 0 : 0.1 : s$ 
3:    $Y \leftarrow \exp(-\alpha T)^\beta$ 
4:    $I_{\alpha f} \leftarrow I_\alpha$ 
5:   for  $t \leftarrow 1$  to  $s$  do
6:      $I_{\alpha t} \leftarrow$  resize  $I_\alpha$  from  $H \times W$  to  $(H + 2t) \times (W + 2t)$ 
7:      $I_{\alpha t} \leftarrow$  crop the area of size  $H \times W$  from the middle of  $I_{\alpha t}$ 
8:      $I_{\alpha t} \leftarrow I_{\alpha t} \times \text{mean}(Y(10t - 5 : 10t + 5))$ 
9:      $I_{\alpha f} \leftarrow \max(I_{\alpha f}, I_{\alpha t})$ 
10:  end for
11:   $I_{\alpha w} \leftarrow I_{\alpha f}$ 
12:   $I_{enh} \leftarrow I_{pre} \times (1 - I_{\alpha w}) + I_{pre} \times I_\alpha$ 
13:  return  $I_{enh}, I_{\alpha w}$ 
14: end procedure
```

is used to perform this task. The details of shadow simulation are described in Algorithm 1. The time complexity of Algorithm 1 is $O(n)$. Fig. 5.8 (a)-(d) illustrate the shadow simulation procedure.

5.3.3 Color Mapping

In this sub-section, a tattoo color map is calculated (Fig. 5.8(f)). To perform this task, the gray level histogram of the input portrait tattoo I_{tatt} and the gray level histogram of I_{enh} , the resultant image from the processes described in Section 3.3, are constructed. To avoid the skin pixels in the portrait tattoo influencing the histogram, its tattoo pixels are segmented. It is not necessary to use a complicated segmentation method designed for generic images because the I_{tatt} has no complex background. First, I_{tatt} (Fig. 5.9(a)) is converted to a grayscale image T_{gray} (Fig. 5.9(b)) and then, histogram equalization [162] is applied (Fig. 5.9(c)). The resultant image is denoted as T_{eq} . Then, Otsu threshold [135] is computed from T_{eq} to select tattoo pixels. In this separation, the resultant binary image is noisy. Thus, its connected components are calculated and only the largest component, which is the face, is selected (Fig. 5.9(d)). The resultant image (Fig. 5.9(d)) is named tattoo mask and denoted as T_{mask} . The pixel values in T_{eq} corresponding to the black pixels in T_{mask} are selected to compute the gray level histogram of tattoo pixels with a

Fig. 5.8: (a) is an input portrait photo I_{face} and (b) is its alpha-matte mask I_{α} . (c) is the corresponding $I_{\alpha w}$ with $\alpha = 0.03$ and $\beta = 1$. (d) is the pre-processed portrait photo I_{pre} of (a) and (e) is its I_{enh} with shadow effect ($\alpha = 0.03, \beta = 1$). (f) is a color map generated from (e).

range from 0 to 255. Then, its cumulative distribution function (CDF) [162] is calculated. This CDF is denoted as f_t and its value at the intensity level a is $f_t(a)$. Given a cumulative probability α , its corresponding intensity level can be determined using the inverse of f_t i.e., $f_t^{-1}(\alpha)$. Since the pixel values are discrete, f_t^{-1} is implemented using a look-up table and each intensity level corresponds to many pixels in different locations. Similarly, the histogram of I_{enh} and its CDF are constructed. The CDF and its inverse are denoted as f_z and f_z^{-1} , respectively. Note that only pixels in the area selected by the user in the

Fig. 5.9: (a) an input portrait tattoo I_{tatt} , (b) the grayscale image T_{gray} of I_{tatt} , (c) the result of histogram equalization of T_{gray} and (d) the largest component in the binary image obtained by Otsu threshold.

input portrait photo (Fig. 5.7(a)) are used to calculate the histogram and the CDF.

With the CDFs, f_t and f_z , the RGB values of the tattoo pixels in I_{tatt} can be moved to a color map C . Algorithm 2 describes this process. The time complexity of Algorithm 2 is $O(n)$. The color map C is first initialized as the same size of I_{enh} but all the three channels are initialized with zero. For a pixel location p with an intensity value v_p in I_{enh} , its CDF value is $f_z(v_p)$. Then, the intensity value of the tattoo pixels in T_{eq} (Fig. 5.9(c)) with the same cumulative probability can be determined by $v_t = f_t^{-1}f_z(v_p)$. There are many pixels in T_{eq} with intensity of v_t . One of them is randomly selected. Let the location of the selected pixel be q . Then, a pixel location denoted as q_1 in the neighborhood of q is randomly selected. The value of color map $C(p)$ is assigned with value of $I_{tatt}(q_1)$. Fig. 5.8(f) shows a color map sample.

5.3.4 Digital Tattooing Based on the Tattoo Needle Model

The digital tattooing procedure takes the alpha-matte with shadow I_{aw} (Fig. 5.8(c)), the enhanced grayscale image I_{enh} , the skin image I_{skin} and the color map C as inputs. The DT algorithm requires the coordinates of the top-left corner of the synthetic tattoo being put in the skin image from the user. It is represented by a two-dimensional vector λ . It also needs a parameter W_{step} controlling the spatial frequency of the needle masks being applied on the skin image. Before performing digital tattooing on the skin image,

Algorithm 2: Color Mapping

```

1: procedure ColorMapping( $I_{tatt}, I_{enh}, T_{eq}, T_{mask}$ )
2:  $f_t \leftarrow$  compute CDF of  $T_{eq}$  in the tattoo region indicated by mask  $T_{mask}$ 
3:  $f_z \leftarrow$  compute CDF of  $I_{enh}$  in the tattoo region selected by the user
4:  $C \leftarrow$  same size of  $I_{enh}$  with all three channels initialized as zero
5: for each pixel  $p$  in the image  $I_{enh}$  do
6:    $v_p \leftarrow I_{enh}(p)$ 
7:    $v_t \leftarrow f_t^{-1}(f_z(v_p))$ 
8:    $q \leftarrow$  get a location of a randomly selected pixel that satisfies  $T_{eq}(q) = v_t$ 
9:    $q_1 \leftarrow$  get a location of a randomly selected pixel in the neighborhood of  $q$ 
10:   $C(p) \leftarrow I_{tatt}(q_1)$ 
11: end for
12: return  $C$ 
13: end procedure

```

the resultant image I_{DT} is set to the input skin image I_{skin} . Then, for each sample pixel p with a non-zero value in I_{aw} , digital tattooing is implemented through the equation

$$I_{DT}(p_1) = I_{DT}(p_1) \times \omega M_i(g) + C(p) \times (1 - \omega M_i(g)), \quad (5.8)$$

where $p_1 = \lambda + p + g$, $\omega = I_{enh}(p)$, M_i is a selected needle mask based on the location p , ω is the combination weight, and $g \in (1, 1), \dots, (h, w)$. h and w are the height and width of the needle mask M_i .

During the digital tattooing, the tattoo color is combined with the skin color weighted by the enhanced grayscale portrait image I_{enh} . In the experiment, two parameters δ_s and δ_p are provided to control the combination for the shadow pixels and portrait pixels, respectively. Therefore, in the experiment, the combination weight $\omega = I_{enh}(p)$ in Equation 5.8 is rescaled by

$$\omega = \begin{cases} \omega^{\delta_p}, & \text{if } p_1 \text{ is a portrait pixel} \\ \omega^{\delta_s}, & \text{if } p_1 \text{ is a shadow pixel} \end{cases} \quad (5.9)$$

The original direction of the tattoo needle masks is horizontal (Fig. 5.3). The DT algorithm rotates them to perform digital tattooing because the texture in different facial components has different directions. The angle of rotation, $M_\theta(p)$ for the pixel p in I_{enh} is determined by the directional map M_θ computed from I_{enh} . First Sobel operator is applied to I_{enh} to compute the magnitude map M_{mag} and the directional map M_{dir} . The Otsu's threshold σ is obtained from M_{mag} to remove unstable directions in M_{dir} and

Fig. 5.10: (a) and (c) synthetic portrait tattoos using 3RL and 5FL tattoo needles with different orientations, respectively (Fig. 5.3). (b) and (d) are enlarged patches from (a) and (c), respectively. (zoom in recommended)

replace them with a predefined direction. In the implementation, the predefined direction is set to zero. Since M_{dir} is computed from Sobel operator, the direction of the pixel p in M_{dir} represents the direction of the normal to the edge at p . To match the direction of the tattoo needle masks with the direction of the texture on the portrait image I_{enh} , $\pi/2$ is added to M_{dir} . The entire process determining the angle of rotation $M_{\theta}(p)$ can be described by the equation

$$M_{\theta}(p) = \begin{cases} M_{dir}(p) + \pi/2, & \text{if } M_{mag}(p) \geq \sigma \\ 0, & \text{otherwise} \end{cases} \quad (5.10)$$

In the default setting, the algorithm uses 3RL to tattoo eyes and nose, 7RL to tattoo mouth and 7RS to tattoo the rest parts (see Fig. 5.3). In addition, only the needle masks corresponding to 35mm needles were used. Tattooists can also select the rest of the needle masks in Fig. 5.3 to carry out their tattoo designs. The locations of the facial components are determined by a facial landmark detection algorithm [166]. The digital tattooing procedure described in Equation(5.8) uses $I_{enh}(p)$ to control the intensity of the color $C(p)$ and the tattoo needle mask M_i to simulate tattoo needles injecting ink into the skin. Fig. 5.10 demonstrates the difference between synthetic portrait tattoos with different tattoo needle configurations and orientations. Algorithm 3 describes the

Algorithm 3: Digital Tattooing

```

1: procedure DigitalTattooing( $C, I_{enh}, I_{\alpha w}, I_{skin}, W_{step}, \lambda, M_\theta$ )
2:    $h, w \leftarrow$  the size of tattoo needle mask
3:    $I_{DT} \leftarrow I_{skin}$ 
4:   for each sampling point  $p$  under the sample step  $W_{step}$  with non-zero value in  $I_{\alpha w}$ 
     do
5:      $C(p) \leftarrow$  pick color from the color map  $C$ 
6:      $M_i \leftarrow$  select a tattoo needle mask based on location  $p$ 
7:      $\theta \leftarrow M_\theta(p)$ 
8:      $M_i \leftarrow$  rotate  $M_i$  by  $\theta$  degrees
9:      $\omega \leftarrow I_{enh}(p)$ 
10:    for each  $g \in \{(1,1), \dots, (h,w)\}$  do
11:       $p_1 \leftarrow \lambda + p + g$ 
12:       $I_{DT}(p_1) \leftarrow I_{DT}(p_1) \times \omega M_i(g) + C(p) \times (1 - \omega M_i(g))$ 
13:    end for
14:  end for
15:  return  $I_{DT}$ 
16: end procedure

```

digital tattooing scheme. The time complexity of Algorithm 3 is $O(n^2)$. It is worth to mention that half-toning and stippling also use dots to paint photos. The proposed digital tattooing scheme attempts to simulate the physical tattooing process using tattoo needles, where different tattoo needles are used to tattoo different facial features.

5.3.5 Post-processing

The resultant image I_{DT} from the previous sub-section is noisy because it was an intention to generate the noisy effect to mimic the noise in portrait tattoos. However, some of the noise affects the skin region too much. To weaken it, a post-processing scheme is developed. Firstly, the distance transform is applied to $1 - I_\alpha$ to obtain its distance map D_α . The distance map D_α is normalized and applied to I_{DT} using

$$R(p) = I_{DT}(p) \times (1 - \sqrt{D_\alpha(p)}) + I_{skin}(p) \times \sqrt{D_\alpha(p)}, \quad (5.11)$$

where D_α is the normalized distance map. Equation(5.11) retains the pixels in I_{DT} if they are in the face region and gradually reduces their importance and enhances the importance of the skin pixels in I_{skin} . Then, a Gaussian filter [168] is applied to R to add

Fig. 5.11: A comparison of the results with and without post-processing: (a) a result without post-processing, (b) the zoom in the rectangle region in (a), (c) a result with post-processing, and (d) the zoom in the rectangle region in (c).

some blur effect on the tattoo. The blurring effect also simulates the dispersion of the tattoo pigment. Finally, Equation(5.11) is applied once more to make the skin region closer to the input skin image. More clearly, this process can be described by

$$I_{SPT}(p) = R_G(p) \times (1 - \sqrt{D_\alpha(p)}) + I_{skin}(p) \times \sqrt{D_\alpha(p)}, \quad (5.12)$$

where R_G is the resultant image R smoothed by the Gaussian filter. I_{SPT} is the generated synthetic portrait tattoo. Fig. 5.11 compares the results with and without post-processing. A blind image quality evaluation method called PIQE [169] is employed to evaluate the image quality of the resultant image with post-processing and the resultant image without post-processing. PIQE score of the resultant image without post-processing is 28.8 and the PIQE score of the resultant image after post-processing is 46.5. This means the resultant image after post-processing has better image quality.

The DT algorithm focuses on synthesizing portrait tattoos on close-up skin images. In the real application, users may require to see the synthetic tattoo on an original photo. In this case, an extra pre-processing procedure is applied before the DT algorithm. First, the user is required to specify an area where he/she wants to put the tattoo on (Fig. 5.12(a)) and the skin image from that area is cropped (Fig. 5.12(b)). Then, the skin image is rotated and resized to match the size of the input portrait photo (Fig. 5.12(c)) and the rotation and scaling parameters are stored. The resized and rotated skin image is inputted to the DT algorithm along with a portrait photo and a reference tattoo image. The resultant tattoo image (Fig. 5.12(d)) is rotated and resized based on the stored parameters (Fig. 5.12(e)) and put back to the original image (Fig. 5.12(f)).

Fig. 5.12: (a) The red rectangle in the image indicates the location where a user wants to tattoo a portrait on. (b) The cropped skin image and (c) its rotated version. (d) A synthetic portrait tattoo on (c) and (e) its version in the original orientation. (f) The synthetic portrait tattoo on the original image (a).

5.4 Experimental Results

In this section, the synthetic portrait tattoos generated by the DT algorithm are given. In Section 5.4.1, different parameters are used to synthesize tattoos with different visual effects. In Section 5.4.2, results given by the DT algorithm are compared with the results given by the state-of-the-art neural style transfer methods. In Section 5.4.3, synthetic

Fig. 5.13: Grayscale image enhancement. 1st row: grayscale portrait images. 2nd row: synthetic portrait tattoo images.

portrait tattoos on 3D skin models are presented.

5.4.1 Tunable Parameters

To synthesize portrait tattoos with different styles, a set of parameters is provided to control the output of the proposed DT algorithm. In the preprocessing procedure, the grayscale portrait image is enhanced by the distance map with the parameters ϵ and ρ . The default values of ϵ and ρ are 10 and 0.1, respectively. These values are selected based on the visual effect of the resultant images. Different ϵ and ρ would generate different grayscale portrait images. Figure 5.13 shows four synthetic portrait tattoo images using four different sets of ϵ and ρ . All the rest parameters are set as the default values. In the shadow simulation procedure, the shadow weight function is controlled by α and β . The default values of α and β are 0.07 and 1, respectively. Figure 5.14(a) shows the shadow function, $Y(t) = \exp(-\alpha t)^\beta$ with three sets of parameters and Figure 5.14(b)-(d) show the corresponding synthetic portrait tattoo images. All the rest parameters are set as the default values. In the digital tattooing procedure, the tattoo needle configuration is controlled by 3 parameters: tattoo needle type, number of needles and needle diameter.

Fig. 5.14: Shadow around the face control. 1st row: shadow function. 2nd row: synthetic portrait tattoo images.

There are four types of tattoo needles: RS, RL, FS and FL, where R and F stand for round shape and flat shape, respectively. S and L stand for tattooing shadows and lines, respectively. The needle diameter d can be 30mm, 35mm and 40mm. The number of needles is also tunable. A tattoo needle configuration can be presented as AB-C-D in which AB is one of the four types, C is the number of needles and D is the needle diameter. For example, RL-3-35 represents a round shape needle configuration using three 35mm needles for tattooing lines. In the experiments, needle configurations for

Fig. 5.15: Needle control. (a): RL_3_35 for the whole portrait image. (b): RL_3_35 for eye, nose and mouth, RS_7_35 for the rest area on the portrait image. (c): RL_3_35 for the eyes, RL_5_40 for the nose, RS_5_40 for the mouth and RS_7_35 for the rest area on the portrait image.

Fig. 5.16: Digital tattooing weight control. Synthetic portrait tattoo images with (a) $\delta_s = 2$ and $\delta_p = 3$, (b) $\delta_s = 5$ and $\delta_p = 5$, and (c) $\delta_s = 3$ and $\delta_p = 1.5$.

tattooing facial parts and the rest area of the portrait are selected manually and can be different. Figure 5.15 shows 3 synthetic portrait tattoo images using 3 different sets of tattoo needle configurations. All the rest parameters are set as the default values.

Fig. 5.17: Step size control. Synthetic portrait tattoo images with (a) $W_{step} = 3$ (default value), (b) $W_{step} = 2$, and (c) $W_{step} = 4$.

In the experiment, the default values of δ_s and δ_p are 2 and 3, respectively. Figure 5.16 shows synthetic portrait tattoo images generated using three different sets of δ_s and δ_p . To highlight the difference in the shadow, the shadow parameters α and β are set to 0.07 and 5, respectively and the default values are used for all the other parameters.

In real tattooing, sometimes there are gaps between tattoo ink injections because the injections of the tattoo needles are not continuous. To simulate it, the parameter W_{step} is provided to control the distance between two digital injections. In the experiments, the default value of W_{step} is 3. A smaller W_{step} generates tattoos with higher tattoo color density (Figure 5.17(b)) and a larger W_{step} generates tattoos with lower tattoo color density (Figure 5.17(c)). Figure 5.17 shows synthetic portrait tattoo images generated by using 3 different W_{step} values.

5.4.2 Synthetic Portrait Tattoos

To evaluate the performance of the DT algorithm, 44 portrait tattoo images, 43 high-resolution portrait photos with an average size of 1000×600 pixels, and 29 high-resolution skin images were collected from Google Image with the "free to use or share" searching option. The portrait tattoo images are used as the reference tattoo images in the experiments. The skin images are taken from underwear models and athletes of different

Fig. 5.18: Portrait tattoos generated by the DT algorithm. The first three columns show input portrait photos, skin images and skin images with the synthetic tattoos. The last two columns show the close up synthetic tattoos and the input reference tattoos.

Fig. 5.19: Comparison results between the real portrait tattoos (1st row) and the synthetic portrait tattoos generated by the DT algorithm (2nd row).

genders and skin colors. For each of the 43 portrait photos, 3 portrait tattoos and 3 skin images were randomly selected as the inputs of the DT algorithm. Thus, 9 synthetic portrait tattoos for each of the portrait photos and in total, 387 ($43 \times 3 \times 3$) synthetic portrait tattoos were produced by the DT algorithm.

Fig. 5.18 shows the results from the DT algorithm. The synthetic portrait tattoos keep the real portrait tattoos color with rich facial details. The shadows around the face region are close to the shadow in the real portrait tattoos. The shadows inside the face region are also enhanced which make the synthetic portrait tattoos look natural and close to the real portrait tattoos. Fig. 5.19 shows the real portrait tattoos of the same persons. It shows that the quality of real and synthetic portrait tattoos is comparable. The synthetic portrait tattoos generated from the DT algorithm are also compared with three state-of-the-art neural style transfer methods [43, 126, 127] (Fig. 5.20). Although some methods reviewed in chapter 2.7 achieved promising results for image style transferring problems, they have some limitations on transferring portrait photos to tattoos. Most of these deep learning-based style transfer methods require huge training databases. Obtaining a large portrait photo database is not difficult but a large real portrait tattoo image database is not publicly available. Thus, except for

Fig. 5.20: From left to right: input portrait photos, input reference tattoos, synthetic portrait tattoos generated by Johnson et al.'s, Huang and Belongie's, and Chen and Schmidt's methods, close-up image of synthetic portrait tattoos generated by the proposed DT algorithm, the synthetic portrait tattoos generated by the proposed DT algorithm in original images.

Johnson et al.’s method [43], Huang and Belongie’s method [126] and Chen and Schmidt’s method [127], which can be trained on a single style image, i.e., a reference portrait tattoo image, other methods are not applicable in this study. For a fair comparison, the portrait photos were put on the skin images before inputting to the neural style transfer methods. The comparison results show that the DT algorithm synthesizes higher quality portrait tattoos with an enhanced shadow effect.

The DT algorithm is implemented in Matlab. All the experiments are conducted on a workstation with one Intel Xeon E5 CPU and one GeForce 1080ti GPU. The total time to synthesize one tattoo portrait image (end-to-end) is around 260 seconds. In addition, an iOS app based on the proposed DT algorithm is developed and published in iOS AppStore (Singapore). The processing time on the iOS app is around 90 seconds.

5.4.3 Synthetic Portrait Tattoos on 3D Skin Models

Synthetic portrait tattoos can be generated on 3D skin models. First, a 3D scanner is used to scan a body part of a subject to build a 3D skin model (Figure 5.21(a)). Then, the 3D skin model is unfolded to a 2D image (Figure 5.21(b)), which is used as an input skin image of the DT algorithm. A synthetic portrait tattoo (Figure 5.21(c)), which is generated on the 2D image, is converted back to a 3D model. Figure 5.21(d)-(f) show a synthetic portrait tattoo on a 3D model from different viewpoints. Figure 5.22 shows more synthetic portrait tattoos on 3D models.

5.5 Synthetic Portrait Tattoo Stickers

Screen printing is a method to print a tattoo design on a sticker, which can be put on the skin as a temporary tattoo (see Fig. 5.23). Currently, tattoo stickers are some common objects, creatures, logos and animals. There is no tailor-made portrait tattoo sticker according to the best knowledge of the authors. The proposed DT algorithm can be slightly modified to produce portrait tattoo stickers. Almost all the steps described in the DT algorithm for processing the input portrait photo with reference portrait tattoo and combining it with the input skin image are the same. The only difference is that the skin image is replaced with a white image. The resultant image is denoted as I_{white} . Note

Fig. 5.21: (a) a 3D skin image, (b) a 2D image unfolded from (a), (c) a synthetic portrait tattoo on the 2D image. (d)-(f) the synthetic portrait tattoo on the 3D image from three different viewpoints.

that I_{white} is an RGB image. To generate a tattoo sticker, a transparency map M_{trans} that contains the transparency value of each pixel is required. In a transparency map, 0 indicates 100% transparency and 1 indicates 0% transparency. In our method, M_{trans} is determined by

$$M_{trans}(p) = \sqrt[\kappa]{1 - I_{wg}(p)}, \quad (5.13)$$

where $0 \leq \kappa \leq 1$ is a parameter controlling the percentage of transparency, I_{wg} is the gray scale image converted from I_{white} . When $I_{wg}(p)$ is high indicating that p is a skin pixel, $M_{trans}(p)$ is low representing high transparency. When $I_{wg}(p)$ is low indicating that p is a tattoo pixel, $M_{trans}(p)$ is high representing lower transparency. Thus, $1 - I_{wg}(p)$ is used in Equation(5.13). I_{white} is saved with M_{trans} in the Portable Network Graphics (PNG)

Fig. 5.22: Synthetic portrait tattoos on 3D models.

Fig. 5.23: Some temporary tattoos.

format to form the portrait tattoo sticker. Figure 5.24 shows synthetic portrait tattoo stickers (with $\kappa = 1$) on a skin image without any additional processing. With the PNG image, the sticker can be easily printed on different materials.

Fig. 5.24: Synthetic portrait tattoos stickers.

5.6 Synthetic Text Tattoos

There is a significant difference between text tattoo and portrait tattoo as shown in Fig. 5.25. For example, text tattoo is more likely to be solid color while portrait tattoo has different textures and colors; text tattoo is oriented (horizontal or vertical) while the portrait tattoo is vertical in most of the cases; the shape of text tattoo is more likely to be regular while the portrait tattoo is of an arbitrary shape. Based on those differences, DT algorithms needs to be modified accordingly to synthesize text tattoo images. Fig. 5.26 illustrates the modified DT algorithm for synthesizing text tattoos.

First, the preprocessing procedure is simplified and the facial landmark detection is removed. Second, the color mapping procedure is simplified. Since the color of the text in the inputted real text image and the color of the text tattoo in the reference text tattoo image are fixed color and there is no color distribution function available. Therefore, the color for digital tattooing is set as a fixed color, which is same as the color of text tattoo in the reference text tattoo image. Third, a curve text procedure is added into the DT algorithm for synthesizing text tattoo images. In synthesizing portrait tattoo, the

Fig. 5.25: Comparison of portrait tattoos (1st row) and text tattoos (2nd row).

Fig. 5.26: Using DT for synthesizing text tattoos.

portrait only needs to be resized and rotated to fit into the target area on the skin image. However, in synthesizing text tattoo, the text image needs to be bent and rotated because

Fig. 5.27: Samples of curved text tattoos.

Without music, life would be a mistake. Unrest of spirit is a mark of life.
Love is a smoke made with the fume of sighs. Character develops itself in the stream of life.
Politics is the art of the possible. I solemnly swear that I am up to no good.
We accept the love we think we deserve. *The art of simplicity is a puzzle of complexity.*

Fig. 5.28: Samples of synthetic text images.

some text tattoos are curved (see Fig. 5.27). The curve text procedure is implemented using the coherent point drift (CPD) algorithm [136]. Two points sets S_{text} and S_{skin} are constructed from inputted text image and skin mask image. The CPD algorithm takes those two points sets as input and calculates a transform matrix T . Then text points set S_{text} is transformed into curve text using transform matrix T .

To evaluate the performance of synthesizing text tattoos using the DT algorithm, we collected 152 skin images and generated 217 skin masks. One skin image can have multiple skin masks so that the text tattoo can be synthesized on different body parts. 15 real text tattoos were collected and used as the reference text tattoo images. One real text tattoo image will be randomly selected in each text tattoo synthesizing procedure. As for the inputted text images, we collected 37,491 popular quotes from the internet and generated 487,383 text images using 13 different fonts. Fig. 5.28 shows some samples of synthetic text images.

Synthetic text tattoos generated by the DT algorithm are shown in Fig. 5.29 and the images that synthetic text tattoos were put on the inputted skin images are shown in Fig. 5.30. Note that some synthetic text tattoos are curved or even in a circle shape and

Fig. 5.29: Samples of synthetic text tattoo images.

the shape of text tattoo also matches the body part's shape and orientation. From those synthetic images, it can be concluded that with proper modification, DT algorithm can be enabled to synthesize text tattoo images with various shapes and font styles.

5.7 Summary

Training an optimal tattoo retrieval system or tattoo detector requires a large tattoo dataset. The current tattoo dataset is either small or imbalanced. The NTU Tattoo V2 and NTU Text Tattoo V1 that were introduced in chapter 4, although are the largest at the moment, still suffer from the imbalance issue and still not enough for text tattoo detection and spotting. Therefore, generating tattoo images automatically seems to be the only solution.

In this chapter, digital tattooing (DT) method is proposed for synthesizing tattoo images. It uses a tattoo needle model to simulate the real tattoo needles and uses a digital tattooing algorithm to simulate the real tattooing procedure. DT method uses a

Fig. 5.30: Samples of synthetic text tattoo on inputted skin images.

series of tailor-made image processing methods to pre-process the inputted portrait photo first, then the DT algorithm performs digital tattooing based on the needle models with different configurations and has an additional scheme to simulate shadow around the face area. Tunable parameters were provided in each procedure. DT is applied to synthesize portrait tattoo images because the portrait photo contains rich texture details for us to

evaluate the performance of the DT method.

Since there is no existing method designed for this image rendering problem, the proposed DT algorithm is compared with three state-of-the-art neural style transfer methods. According to the observation of the synthetic tattoo images, the DT algorithm outperforms those state-of-the-art transfer methods and offers more realistic portrait tattoos. With the help of a 3D scanner, DT is also able to synthesize portrait tattoos on 3D skin images. Tattoo stickers can be synthesized by the DT method too with a proper set of parameters. We also investigated the potential of using DT to synthesize text tattoo images with some proper modifications to the DT algorithm. In the future, we will bring in some deep image transferring technique to enhance the performance of the DT algorithm and use it to generate more realistic text tattoo and general tattoo images. Tattoo retrieval and detection experiments will be conducted too using synthetic tattoo images in the future work.

Chapter 6

Conclusions and Future Work

In this chapter, the works presented in this thesis are summarized. The limitations of the proposed methods are discussed. Some possible solutions to the major issues that need further study are suggested.

6.1 Conclusions

Tattoo, as a soft biometric trait, has been used widely by the law enforcement agencies. Many works have been done for tattoo retrieval, tattoo detection which are two major tasks in forensic investigations. Traditional tattoo retrieval methods require query tattoo images or tattoo sketches with rich details. In this research, a novel tattoo retrieval method based on the geometry information of tattoos is proposed. For training the tattoo detection network, a new tattoo dataset named NTU Tattoo V2 is established which is the largest tattoo dataset at the moment. A new text tattoo dataset named NTU Text Tattoo V1 is established as well for text tattoo detection. To further improve the tattoo detection performance, a prior knowledge-based attention mechanism (PKAM) is introduced and a text tattoo detection network using double attention is designed. Finally, a digital tattooing algorithm is proposed for generating synthetic tattoo images.

6.1.1 Geometry based Tattoo Retrieval

Tattoo retrieval is an important task in forensic investigations. Current tattoo retrieval systems require a tattoo image as a query to search tattoos from their database. However, in real cases, the query tattoo images are not always available. Some tattoo retrieval

systems use tattoo sketches to search tattoos. However, the richness of tattoo sketches affects the performance of searching significantly. To handle tattoo retrieval without query tattoo images or tattoo sketches with rich details, a geometry based tattoo retrieval (GBTR) system is introduced in chapter 3.

GBTR system allows the user to search tattoos by drawing a simple tattoo sketch based on his/her memory on a full-body human model image. The full-body images used in the GBTR system are captured by a full-body imaging system. A full-body image database named GBTR database is established by manually putting tattoo patterns on the full-body images. The GBTR database that contains 2,188 full-body images with tattoos on different parts of the human body are used for evaluating the geometry based tattoo retrieval system. The shape and location information of the tattoo sketch is recorded and matched with the records in the GBTR database, where all tattoos are on the hull body human images taken in the controlled environment. The shape information is extracted from the candidate tattoos using Snake [20] algorithm. A patch-based tattoo detection method is employed to detect all the candidate tattoos. The location information is obtained in a full-body coordinate system which is built based on the human joint points. The human joint points are detected automatically by a novel joint point detection method. A full-body coordinates mapping method is proposed to find the corresponding location of the inputted tattoo sketch. Shape matching is performed by a coherent drift point (CPD) algorithm [136]. The matching results are displayed based on the sorted shape matching scores.

Two evaluation schemes are designed to evaluate the tattoo retrieval performance using low-quality tattoo sketches and high-quality tattoo sketches. Evaluation results showed that the system can perform the tattoo retrieval with a simple tattoo sketch as input. Comparing to other sketch-to-image searching methods, the GBTR system achieved higher accuracy. It was also found that the tattoo sketch quality affects the tattoo retrieval performance. Tattoo sketches with higher quality will give a more accurate retrieval result.

6.1.2 Tattoo Dataset and Text Tattoo Dataset

A large tattoo dataset is essential for training optimal tattoo retrieval systems and tattoo detectors. However, existing public tattoo datasets are too small to train an accurate

tattoo detector and none of them is established for text tattoo detection. As mentioned in previous chapters, text tattoo contains important personal information that can be very helpful for the law enforcement agencies in forensic investigations. Therefore, establishing a large tattoo dataset and a text tattoo dataset is very necessary.

In chapter 4, NTU Tattoo V2 dataset is introduced, which is the largest tattoo detection dataset at the moment. NTU Text Tattoo V1 dataset is also introduced, which is the first and only text tattoo dataset designed for text tattoo detection. All images in NTU Tattoo V2 and NTU Text Tattoo V1 are uncontrolled images that were collected from the internet and annotated manually. Each tattoo/text tattoo instance is annotated by a bounding box. As a result, there are totally 87,282 tattoo images with 102,865 tattoo instances in NTU Tattoo V2 dataset. In NTU Text Tattoo V1 dataset, there are 9,067 text tattoo images with 10,027 text tattoo instances.

Comparing to other public available tattoo datasets such as NIST Tatt-C [40], DeMSI [41], NTU Tattoo V1 [26] and WebTattoo [5], NTU Tattoo V2 contains much more images that are taken in various environments which have diverse quality, sizes and illumination conditions.

6.1.3 Prior Knowledge-based Attention Mechanism and Text Tattoo Detection

Attention mechanisms have been proved to be very effective in improving the detection performance of a deep neural network. Traditional attention mechanisms such as spatial attention, channel attention and block attention use the object saliency and the importance of feature maps to enhance a portion of the feature maps extracted by the convolutional layers. None of them use the relationship among objects to enhance the features of one particular object.

The relationship of different objects is an important prior knowledge that worth paying attention to. The prior knowledge-based attention mechanism (PKAM) introduced in chapter 4 is designed to utilize the relationship between the target objects (tattoo or text tattoo instance) and the relevant objects (person or generic tattoo instance) to enhance the detection performance. It uses the convolutional features of relevant objects

to enhance the corresponding portion of the inputted images and therefore enhance the target instances.

Text tattoos are always beside or around general tattoo patterns. Similarly, tattoos are always within the areas of human skins. There are geometry relationships among text tattoos, general tattoos and human instances. Therefore, PKAMs are used in text tattoo detection. A text tattoo detection network using double attention (TTDN-DA) architecture is designed and examined. TTDN-DA consists of three sub-networks and two PKAMs. The first sub-network extracts human instance features; the second sub-network extracts general tattoo features and the last sub-subnetwork extracts the text tattoo features. The first PKAM uses human instance features to enhance general tattoo images and the second PKAM uses general tattoo features to enhance the text tattoo images. A three-stage training scheme is applied to train TTDN-DA using both NTU Tattoo V2 and NTU Text Tattoo V1. Another two text tattoo detection network TTDN-DA-V2 and TTDN-DA-V3 are designed to meet the requirement of fast training and small network size. TTDN-DA-V2 consists of two sub-networks that are connected to each other via one PKAM. In TTDN-DA-V3, PKAM was inserted between blocks within one single sub-network.

The evaluation results showed that, comparing with other state-of-the-art object detectors and scene text detectors, TTDN-DA achieved an optimal accuracy in text tattoo detection. Although TTDN-DA-V2 and TTDN-DA-V3 are designed for fast training and small storage space, their text tattoo detection performance is still better than other state-of-the-art deep object detectors and scene text detectors. TTDN-DA and TTDN-DA-V2 were also evaluated for general tattoo detection task, the evaluation results showed that both networks outperformed other state-of-the-art deep object detectors.

6.1.4 Tattoo Needle Model and Digital Tattooing

Training optimal tattoo retrieval systems or tattoo detectors require a large tattoo dataset. Although NTU Tattoo V2 is the largest tattoo dataset at the moment, the imbalanced data issue still remains. Therefore, a larger tattoo dataset with more balanced data is still important. One way to obtain a large tattoo dataset with balanced data is by automatically generating synthetic tattoo images.

To synthesis tattoo images, a digital tattooing (DT) algorithm is introduced in chapter 5. DT algorithm performs digital tattooing using the tattoo needle models which are designed to simulate different configurations of the real tattoo needles. DT contains preprocessing, shadow simulation, color mapping, digital tattooing and postprocessing procedures. Each procedure provides tunable parameters to control the appearance of the generated synthetic tattoo images.

We concentrate on using the DT algorithm to synthesis portrait tattoo images which are very meaningful to a lot of people and contain sufficient details to evaluate the performance of the DT algorithm. Comparing with three state-of-the-art neural style transfer methods, the DT algorithm outperformed them and generated more realistic portrait tattoos. DT can generate synthetic tattoos not only on 2D skin images but also on 3D skin images with the help of a 3D skin scanner. When the inputted skin image is replaced by a white background image, DT can generate tattoo stickers with a transparent background that can be put on any images. With some proper modifications, DT can be enabled to generate synthetic text tattoo images as well.

6.2 Future Work

Text tattoo provides vital clues for forensic investigations. Therefore, detecting text tattoos and recognizing their contents can help law enforcement agencies to narrow down the suspect list quickly. In this thesis, the text tattoo detector TTDN-DA uses a prior knowledge-based attention mechanism (PKAM) to improve the detection accuracy. However, TTDN-DA only predicts the bounding boxes of text tattoos, it cannot “read” the content of the text tattoos. In the forensic investigation, knowing the content of the text tattoos is more important than just knowing the location of the text tattoos. An accurate text tattoo detection is only the beginning. More effort needs to be putted on an accurate text tattoo recognizer. According to some preliminary studies, the ABCNet [8] shows some potentials in text tattoo recognition task. We will try to combine the recognition head of ABCNet with the feature representation layers of TTDN-DA to investigate the performance of text tattoo recognition. In the future work, NTU Text Tattoo V1 dataset will be annotated in the word level and character level; a text recognition technique will

be added into the text tattoo detection network to “read” the content of a text tattoo. Then, we will have an end-to-end text tattoo spotting network.

To train a text spotting network for the text tattoo spotting task, a large text tattoo dataset is essential. Although NTU Text Tattoo V1 dataset is established which is the first public text tattoo dataset, there are only 9,067 text tattoo images with 10,027 text tattoo instances in NTU Text Tattoo V1 dataset. The digital tattooing (DT) algorithm offers an option to synthesize a large number of tattoo images. Currently, DT algorithm is tailor-made for synthesizing portrait tattoo images because portrait photos contain rich details to evaluate the performance of the DT algorithm. Although it can be modified to synthesize text tattoo images, the quality of synthetic text tattoos still needs to be improved. In future work, we will expand the capability of the DT algorithm to enable it to synthesize more realistic text tattoo images and general tattoo images. We will also investigate some deep image generation methods such as GANs to enhance the performance of the DT algorithm. Tattoo retrieval and detection experiments will also be conducted using the synthetic tattoo images to evaluate whether the synthetic tattoo images can help to improve the performance of the tattoo retrieval systems and tattoo detectors.

Overall, tattoo retrieval, tattoo detection, text tattoo spotting are only a few major use cases in forensic investigations. Other use cases such as tattoo recognition, tattoo image captioning and tattoo image restoring are also useful and important for law enforcement agencies. More works need to be done in order to utilize tattoos for forensic investigations.

References

- [1] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *ICCV*, 2015, pp. 3730–3738.
- [2] A. Deter-Wolf, B. Robitaille, L. F. Krutak, and S. Galliot, “The world’s oldest tattoos,” in *Journal of Archaeological Science: Reports*, 2016, pp. 19–24.
- [3] A. K. Jain, J.-E. Lee, and R. Jin, “Tattoo-id: Automatic tattoo image retrieval for suspect and victim identification,” in *PCM*, 2007, pp. 256–265.
- [4] H. Han and A. K. Jain, “Tattoo based identification: Sketch to image matching,” in *ICB*, 2013, pp. 1–8.
- [5] H. Han, J. Li, A. K. Jain, S. Shan, and X. Chen, “Tattoo image search at scale: Joint detection and compact representation learning,” in *PAMI*, vol. 41, 2018, pp. 2333–2348.
- [6] B. Shi, X. Bai, and S. J. Belongie, “Detecting oriented text in natural images by linking segments,” in *CVPR*, 2017, pp. 3482–3490.
- [7] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, “Shape robust text detection with progressive scale expansion network,” in *CVPR*, 2019, pp. 9336–9345.
- [8] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, “Abcnet: Real-time scene text spotting with adaptive bezier-curve network,” in *CVPR*, 2020, pp. 9806–9815.
- [9] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *CVPR*, 2015, pp. 779–788.
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *ECCV*, 2016, pp. 21–37.
- [11] R. B. Girshick, “Fast r-cnn,” in *ICCV*, 2015, pp. 1440–1448.
- [12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask r-cnn,” in *ICCV*, 2017, pp. 2980–2988.
- [13] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in *NIPS*, 2016, pp. 379–387.
- [14] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *ICCV*, 2017, pp. 2999–3007.
- [15] B. Singh and L. S. Davis, “An analysis of scale invariance in object detection - snip,” in *CVPR*, 2018, pp. 3578–3587.
- [16] Y. Li, Y. Chen, N. Wang, and Z. Zhang, “Scale-aware trident networks for object detection,” in *ICCV*, 2019, pp. 6053–6062.
- [17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014, pp. 2672–2680.
- [18] D. Kim, M. Son, Y. Lee, H. Kang, and S. Lee, “Feature-guided image stippling,” in *Comput. Graph. Forum*, vol. 27, 2008, pp. 1209–1216.

REFERENCES

- [19] N. Q. Huynh, X. Xu, A. W.-K. Kong, and S. Subbiah, “A preliminary report on a full-body imaging system for effectively collecting and processing biometric traits of prisoners,” in *CIBIM*, 2014, pp. 167–174.
- [20] M. Kass, A. P. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” in *IJCV*, vol. 1, 2004, pp. 321–331.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2015, pp. 770–778.
- [22] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *CVPR*, 2017, pp. 5987–5995.
- [23] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *ECCV*, 2016, pp. 483–499.
- [24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015, pp. 1–14.
- [25] Laura, “Understanding tattoo needles,” 2014. [Online]. Available: <https://info.painfulpleasures.com/blogs/understanding-tattoo-needles>
- [26] Q. Xu, S. Ghosh, X. Xu, Y. Huang, and A. W.-K. Kong, “Tattoo detection based on cnn and remarks on the nist database,” in *ICB*, 2016, pp. 1–7.
- [27] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Springer, 2008.
- [28] FBI, “Next generation identification (ngi).” [Online]. Available: <https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/ngi>
- [29] NIST, “American national standard for information systems - data format for the interchange of fingerprint, facial, scar mark and tattoo (smt) information,” in *NIST Special Publication 500-245*, 2000.
- [30] E. Lipton and J. Glanz, “Limits of dna research pushed to identify the dead of sept. 11,” <https://www.nytimes.com/2002/04/22/nyregion/limits-of-dna-research-pushed-to-identify-the-dead-of-sept-11.html>.
- [31] G. York, “Decay challenges forensic skills,” <https://www.theglobeandmail.com/news/world/decay-challenges-forensic-skills/article18181157/>.
- [32] J.-E. Lee, R. Jin, and A. K. Jain, “Rank-based distance metric learning: An application to image retrieval,” in *CVPR*, 2008, pp. 1–8.
- [33] S. T. Acton and A. Rossi, “Matching and retrieval of tattoo images: Active contour cbir and glocal image features,” in *SSIAI*, 2008, pp. 21–24.
- [34] A. K. Jain, J.-E. Lee, R. Jin, and N. Gregg, “Content-based image retrieval: An application to tattoo images,” in *ICIP*, 2009, pp. 2745–2748.
- [35] X. Di and V. M. Patel, “Deep tattoo recognition,” in *CVPRW*, 2016, pp. 119–126.
- [36] Z. Sun, J. Baumes, P. Tunison, M. W. Turek, and A. Hoogs, “Tattoo detection and localization using region-based deep learning,” in *ICPR*, 2016, pp. 3055–3060.
- [37] B. Heflin, W. J. Scheirer, and T. E. Boulton, “Detecting and classifying scars, marks, and tattoos found in the wild,” in *BTAS*, 2012, pp. 31–38.
- [38] D. Marčetić, S. Ribarić, V. Štruc, and N. Pavešić, “An experimental tattoo de-identification system for privacy protection in still images,” in *MIPRO*, 2014, pp. 1288–1293.
- [39] T. Hrkać, K. Brkić, S. Ribarić, and D. Marčetić, “Deep learning architectures for tattoo detection and de-identification,” in *SPLINE*, 2016, pp. 1–5.
- [40] M. Ngan and P. Grother, “Tattoo recognition technology - challenge (tatt-c): an open tattoo database for developing tattoo recognition research,” in *ISBA*, 2015, pp. 1–6.
- [41] T. Hrkac, K. Brkic, and Z. Kalafatic, “Tattoo detection for soft biometric de-identification based on convolutional neural networks,” in *OeAGM/AAPR*, 2016.

REFERENCES

- [42] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *CVPR*, 2016, pp. 2414–2423.
- [43] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *ECCV*, 2016, pp. 694–711.
- [44] T. Igarashi, K. Nishino, and S. K. Nayar, “The appearance of human skin: A survey,” in *Found. Trends Comput. Graph. Vis*, vol. 3, 2007, pp. 1–95.
- [45] D. J. Roberts, “Secret ink: Tattoo’s place in contemporary american culture,” in *J. Am. Cult.*, vol. 35, 2012, pp. 153–165.
- [46] P. Research, “Tattooed gen nexters,” <https://www.pewresearch.org/fact-tank/2008/12/09/tattooed-gen-nexters/>.
- [47] T. Economist, “A statistical analysis of the art on convicts bodies,” accessed: 2020-03-01. [Online]. Available: <https://www.economist.com/christmas-specials/2016/12/24/a-statistical-analysis-of-the-art-on-convicts-bodies>
- [48] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” in *IJCV*, vol. 60, 2004, pp. 91–110.
- [49] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in *ECCV*, 2004, pp. 1–22.
- [50] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” in *ICCV*, 2003, pp. 1470–1477.
- [51] D. Nister and H. Stewénus, “Scalable recognition with a vocabulary tree,” in *CVPR*, 2006, pp. 2161–2168.
- [52] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” in *CVPR*, 2007, pp. 1–8.
- [53] G. Shakhnarovich, T. Darrell, and P. Indyk, “Nearest-neighbor methods in learning and vision: Theory and practice (neural information processing),” in *Pattern Anal. Appl.*, vol. 11, 2008, pp. 221–222.
- [54] F. Li, W. Tong, R. Jin, A. K. Jain, and J.-E. Lee, “An efficient key point quantization algorithm for large scale image retrieval,” in *LS-MMRM@ACM Multimedia*, 2009, pp. 89–96.
- [55] B. Klare, Z. Li, and A. K. Jain, “Matching forensic sketches to mug shot photos,” in *PAMI*, vol. 33, 2011, pp. 639–646.
- [56] M. Ngan, G. W. Quinn, and P. J. Grother, “Tattoo recognition technology - challenge (tatt-c): Outcomes and recommendations,” in *NISTIR*, 2015.
- [57] T. Frieden, “72 charged in online global child porn,” 2011. [Online]. Available: <http://edition.cnn.com/2011/CRIME/08/03/us.child.porn.ring/>
- [58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012, pp. 1106–1114.
- [59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F.-F. Li, “Imagenet large scale visual recognition challenge,” in *IJCV*, vol. 115, 2014, pp. 211–252.
- [60] C. Cortes and V. Vapnik, “Support-vector networks,” in *Mach. Learn.*, vol. 20, 2004, pp. 273–297.
- [61] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *PAMI*, vol. 39, 2015, pp. 1137–1149.
- [62] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” in *IJCV*, vol. 88, 2010, pp. 303–338.
- [63] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *BMVC*, 2014, pp. 1–11.
- [64] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *CVPR*, 2010, pp. 3485–3492.

REFERENCES

- [65] S. Shan, H. Han, A. Das, and A. Dantcheva, “Robust tattoo detection and retrieval competition (rtdrc 2019),” <https://sites.google.com/site/rtdrc2019/>.
- [66] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015, pp. 1–14.
- [67] B. Schölkopf, J. Platt, and T. Hofmann, *Graph-Based Visual Saliency*, 2006, pp. 545–552.
- [68] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut -interactive foreground extraction using iterated graph cuts,” in *ACM Trans. Graph.*, vol. 23, 2004, pp. 309–314.
- [69] P. Duangphasuk and W. Kurutach, “Tattoo skin detection and segmentation using image negative method,” in *ISCIT*, 2013, pp. 354–359.
- [70] —, “Tattoo skin cross - correlation neural network,” in *ISCIT*, 2014, pp. 489–493.
- [71] J. Kim, H. Li, J. Yue, J. Ribera, E. J. Delp, and L. Huffman, “Automatic and manual tattoo localization,” in *HST*, 2016, pp. 1–6.
- [72] J. Dong, X. Qu, and H. Li, “Color tattoo segmentation based on skin color space and k-mean clustering,” in *ICSSS*, 2017, pp. 53–56.
- [73] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *CVPR*, 2010, pp. 2963–2970.
- [74] L. Neumann and J. Matas, “Real-time scene text localization and recognition,” in *CVPR*, 2012, pp. 3538–3545.
- [75] M. Anthimopoulos, B. Gatos, and I. Pratikakis, “Detection of artificial and scene text in images and video frames,” in *Pattern Anal Applic*, vol. 16, 2013, pp. 431–446.
- [76] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, “Robust text detection in natural images with edge-enhanced maximally stable extremal regions,” in *ICIP*, 2011, pp. 2609–2612.
- [77] I. Posner, P. I. Corke, and P. Newman, “Using text-spotting to query the world,” in *IROS*, 2010, pp. 3181–3186.
- [78] K. Wang, B. Babenko, and S. J. Belongie, “End-to-end scene text recognition,” in *ICCV*, 2011, pp. 1457–1464.
- [79] X. Rong, C. Yi, and Y. Tian, “Unambiguous text localization and retrieval for cluttered scenes,” in *CVPR*, 2017, pp. 3279–3287.
- [80] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, “Deeptext: A unified framework for text proposal generation and text detection in natural images,” in *ArXiv*, vol. abs/1605.07314, 2016.
- [81] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, “Textboxes: A fast text detector with a single deep neural network,” in *AAAI*, 2017, pp. 4161–4167.
- [82] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, “East: An efficient and accurate scene text detector,” in *CVPR*, 2017, pp. 2642–2651.
- [83] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, “Icdar 2015 competition on robust reading,” in *ICDAR*, 2015, pp. 1156–1160.
- [84] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, “Coco-text: Dataset and benchmark for text detection and recognition in natural images,” in *arXiv*, vol. abs/1601.07140, 2016. [Online]. Available: <http://vision.cornell.edu/se3/wp-content/uploads/2016/01/1601.07140v1.pdf>
- [85] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazn, and L. P. de las Heras, “Icdar 2013 robust reading competition,” in *ICDAR*, 2013, pp. 1484–1493.
- [86] D. Deng, H. Liu, X. Li, and D. Cai, “Pixellink: Detecting scene text via instance segmentation,” in *AAAI*, 2018, pp. 6773–6780.

REFERENCES

- [87] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, “Efficient and accurate arbitrary-shaped text detection with pixel aggregation network,” in *ICCV*, 2019, pp. 8439–8448.
- [88] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, “Scene text detection with supervised pyramid context network,” in *AAAI*, 2019, pp. 9038–9045.
- [89] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *CVPR*, 2017, pp. 936–944.
- [90] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, “Real-time scene text detection with differentiable binarization,” in *AAAI*, 2020, pp. 11 474–11 481.
- [91] J. Ye, Z. Chen, J. Liu, and B. Du, “Textfusenet: Scene text detection with richer fused features,” in *IJCAI*, 2020, pp. 516–522.
- [92] G. G. Lorentz, “Bernstein polynomials,” in *American Mathematical Soc.*, 2013.
- [93] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *CVPR*, 2016, pp. 6517–6525.
- [94] —, “Yolov3: An incremental improvement,” in *ArXiv*, vol. abs/1804.02767, 2018.
- [95] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *CVPR*, 2015, pp. 1–9.
- [96] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014, pp. 740–755.
- [97] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “Dssd : Deconvolutional single shot detector,” in *ArXiv*, vol. abs/1701.06659, 2017.
- [98] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *CVPR*, 2013, pp. 580–587.
- [99] J. R. R. Uijlings, K. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” in *IJCV*, vol. 104, 2013, pp. 154–171.
- [100] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.
- [101] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, “Light-head r-cnn: In defense of two-stage object detector,” in *ArXiv*, vol. abs/1711.07264, 2017.
- [102] Z. Cai and N. Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” in *CVPR*, 2018, pp. 6154–6162.
- [103] T. X. Vu, H. Jang, T. X. Pham, and C. D. Yoo, “Cascade rpn: Delving into high-quality region proposal network with adaptive convolution,” in *ArXiv*, vol. abs/1909.06720, 2019.
- [104] B. Singh, M. Najibi, and L. S. Davis, “Sniper: Efficient multi-scale training,” in *NeurIPS*, 2018, pp. 9333–9343.
- [105] T. Kong, A. Yao, Y. Chen, and F. Sun, “Hypernet: Towards accurate region proposal generation and joint object detection,” in *CVPR*, 2016, pp. 845–853.
- [106] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *CVPR*, 2016, pp. 2874–2883.
- [107] Z. Cai, Q. Fan, R. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” in *ECCV*, 2016, pp. 354–370.
- [108] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, “Beyond skip connections: Top-down modulation for object detection,” in *ArXiv*, vol. abs/1612.06851, 2016.
- [109] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *CVPR*, 2018, pp. 8759–8768.

REFERENCES

- [110] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *ICLR*, 2016, pp. 1–13.
- [111] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” in *PAMI*, vol. 40, 2018, pp. 834–848.
- [112] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017, pp. 6230–6239.
- [113] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” in *ArXiv*, vol. abs/1706.05587, 2017.
- [114] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “Cspnet: A new backbone that can enhance learning capability of cnn,” 2020, pp. 1571–1580.
- [115] S. Qiao, L.-C. Chen, and A. Yuille, “Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution,” in *ArXiv*, vol. abs/2006.02334, 2020.
- [116] J.-E. Lee, A. K. Jain, and R. Jin, “Scars, marks and tattoos (smt): Soft biometric for suspect and victim identification,” in *2008 Biometrics Symposium*, Sept 2008, pp. 1–8.
- [117] Y. Huang, P. Yu, X. Xu, and A. W. K. Kong, “The impact of tattoo segmentation on the performance of tattoo matching,” in *WIECON-ECE*, 2015, pp. 43–46.
- [118] J.-E. Lee, R. Jin, A. Jain, and W. Tong, “Image retrieval in forensics: Tattoo image database application,” in *IEEE MultiMedia*, vol. 19, 2012, pp. 40–49.
- [119] J. Calmon, J. ao Queiroz, C. Goes, and A. Loula, “Augmented tattoo: Evaluation of an augmented reality system for tattoo visualization,” in *SIBGRAPI*, 2015, pp. 265–272.
- [120] G. S. Mauricio, J. ao de Sá Bonelli, and M. das Graças Chagas, “Tattooar: Augmented reality interactive tattoos,” in *HCI*, 2015, pp. 667–674.
- [121] PhotoFunia, “Making tattoo,” 2018. [Online]. Available: https://photofunia.com/effects/making_tattoo/
- [122] LunaPic, “Use lunapic.com to tattoo an image instantly,” 2018. [Online]. Available: <https://www194.lunapic.com/editor/?action=tattoo/>
- [123] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *ICLR*, 2016, pp. 1–16.
- [124] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *CVPR*, 2017, pp. 5967–5976.
- [125] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017, pp. 2242–2251.
- [126] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *ICCV*, 2017, pp. 1510–1519.
- [127] T. Q. Chen and M. Schmidt, “Fast patch-based style transfer of arbitrary style,” in *ArXiv*, vol. abs/1612.04337, 2016.
- [128] A. Selim, M. Elgharib, and L. Doyle, “Painting style transfer for head portraits using convolutional neural networks,” in *ACM Trans. Graph.*, 2016, pp. 129:1–129:18.
- [129] W. Baxter, “Physically-based modeling techniques for interactive digital painting,” Ph.D. dissertation, Chapel Hill, NC, USA, 2004, aAI3156118.
- [130] J. Lu, C. Barnes, S. DiVerdi, and A. Finkelstein, “Realbrush: Painting with examples of physical media,” in *ACM Trans. Graph.*, vol. 32, 2013, pp. 117:1–117:12.
- [131] Z. Chen, B. Kim, D. Ito, and H. Wang, “Wetbrush: Gpu-based 3d painting simulation at the bristle level,” vol. 34, 2015, pp. 200:1–200:11.

REFERENCES

- [132] T. Stuyck, F. Da, S. Hadap, and P. Dutré, “Real-time oil painting on mobile hardware,” in *Comput. Graph. Forum*, vol. 36, 2017, pp. 69–79.
- [133] O. Deussen, M. Spicker, and Q. Zheng, “Weighted linde-buzo-gray stippling,” in *ACM Trans. Graph.*, vol. 36, 2017, pp. 233:1–233:12.
- [134] X. Xu and A. W. K. Kong, “A geometric-based tattoo retrieval system,” in *ICPR*, 2016, pp. 3019–3024.
- [135] N. Otsu, “A threshold selection method from gray-level histograms,” in *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 9, 1979, pp. 62–66.
- [136] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” in *PAMI*, vol. 32, 2010, pp. 2262–2275.
- [137] H. of Tattoos, “Tattoo statistics - how many people have tattoos?” accessed: 2020-03-01. [Online]. Available: <http://www.historyoftattoos.net/tattoo-facts/tattoo-statistics/>
- [138] A. Gupta, A. Vedaldi, and A. Zisserman, “Synthetic data for text localisation in natural images,” in *CVPR*, 2016, pp. 2315–2324.
- [139] V. Mnih, N. M. O. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” in *NIPS*, 2014, pp. 2204–2212.
- [140] J. Ba, V. Mnih, and K. Kavukcuoglu, “Multiple object recognition with visual attention,” in *ICLR*, 2015, pp. 1–10.
- [141] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *ICML*, 2015, pp. 2048–2057.
- [142] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, “Draw: A recurrent neural network for image generation,” in *ICML*, 2015, pp. 1462–1471.
- [143] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in *ArXiv*, vol. abs/1506.02025, 2015.
- [144] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle, and A. C. Courville, “Dynamic capacity networks,” in *ICML*, 2016, pp. 2549–2558.
- [145] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *CVPR*, 2017, pp. 7132–7141.
- [146] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, “Cbam: Convolutional block attention module,” in *ECCV*, 2018, pp. 3–19.
- [147] Y. Chen, C. Han, Y. Li, Z. Huang, Y. Jiang, N. Wang, and Z. Zhang, “Simpledet: A simple and versatile distributed framework for object detection and instance recognition,” in *JMLR*, vol. 20, no. 156, 2019, pp. 1–8.
- [148] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *AISTATS*, 2010, pp. 249–256.
- [149] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: Keypoint triplets for object detection,” in *ICCV*, 2019, pp. 6569–6578.
- [150] A. E. Laumann and A. J. Derick, “Tattoos and body piercings in the united states: a national data set,” vol. 55, pp. 413–421, 2006.
- [151] Statista, “Share of americans with one or more tattoos from 2003 to 2015 (by age group),” 2016. [Online]. Available: <https://www.statista.com/statistics/259601/share-of-americans-with-at-least-one-tattoo-by-age>
- [152] L. Shannon-Missal, “Health & life - tattoo takeover: Three in ten americans have tattoos, and most don’t stop at just one,” 2016. [Online]. Available: http://www.theharrispoll.com/health-and-life/Tattoo_Takeover.html
- [153] A. Proud, “We’re not going to reach peak ‘tattoo’ until 2025 - telegraph,” 2015. [Online]. Available: <http://www.telegraph.co.uk/men/fashion-and-style/11545649/Were-not-going-to-reach-peak-tattoo-until-2025.html>
- [154] L. Edmonds, “Tattooist dave stewart fined £300 for botched inking of marilyn monroe,” Website, 2016, <http://www.dailymail.co.uk/news/article-2443933/Tattooist-Dave-Stewart-fined-300-botched-inking-Marilyn-Monroe.html>.

REFERENCES

- [155] A. Smith, “Most frequently asked questions from tattoo removal patients,” 2017. [Online]. Available: <http://info.astanzalaser.com/blog/7-most-frequently-asked-questions-from-tattoo-removal-patients>
- [156] R. Sullivan, “Tattoo laser removal: Queensland man suffers serious burns after trying to get his tattoos removed,” 2017. [Online]. Available: <https://www.news.com.au/lifestyle/beauty/face-body/tattoo-laser-removal-queensland-man-suffers-serious-burns-after-trying-to-get-his-tattoos-removed/news-story/183200c8ca225d4df1d2f82dc8a40696>
- [157] X. Xu, W. M. Matkowski, and A. W. K. Kong, “A portrait photo-to-tattoo transform based on digital tattooing,” in *Multimedia Tools and Applications*, vol. 79, 2020, pp. 24 367–24 392.
- [158] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” in *ICLR*, 2017, pp. 1–14.
- [159] Y. Zhang, W. Dong, C. Ma, X. Mei, K. Li, F. Huang, B.-G. Hu, and O. Deussen, “Data-driven synthesis of cartoon faces using different styles,” in *TIP*, vol. 26, 2017, pp. 464–478.
- [160] R.-L. Hsu and A. K. Jain, “Generating discriminating cartoon faces using interacting snakes,” in *PAMI*, vol. 25, 2003, pp. 1388–1398.
- [161] K. He, J. Sun, and X. Tang, “Guided image filtering,” in *PAMI*, vol. 35, 2013, pp. 1397–1409.
- [162] W. Burger and M. J. Burge, *Principles of Digital Image Processing*. Springer, 2010.
- [163] “Xdog: An extended difference-of-gaussians compendium including advanced image stylization,” in *Comput. Graph.*, vol. 36, 2012, pp. 740 – 753.
- [164] J. R. Parker, *Algorithms for Image Processing and Computer Vision, 2nd Edition*. John Wiley & Sons, 1997.
- [165] C. R. Maurer, R. Qi, and V. Raghavan, “A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions,” vol. 25, pp. 265–270, 2003.
- [166] Y. Nirkin, I. Masi, A. T. Tran, T. Hassner, and G. G. Medioni, “On face segmentation, face swapping, and face perception,” in *FG*, 2018, pp. 98–105.
- [167] E. Shahrian, D. Rajan, B. Price, and S. Cohen, “Improving image matting using comprehensive sampling sets,” in *CVPR*, 2013, pp. 636–643.
- [168] R. A. Haddad and A. N. Akansu, “A class of fast gaussian binomial filters for speech and image processing,” in *IEEE Trans. Signal Process.*, vol. 39, 1991, pp. 723–727.
- [169] V. N. P. D., M. C. Bh., S. S. Channappayya, and S. S. Medasani, “Blind image quality evaluation using perception based features,” in *NCC*, 2015, pp. 1–6.