



UNIVERSITY OF  
LIVERPOOL

# **Discriminative and Generative Learning with Style Information**

A thesis submitted in accordance with the requirements of the  
University of Liverpool  
for the degree of Doctor in Philosophy  
by

Haochuan JIANG

Department of Electrical Engineering and Electronics  
School of Electrical Engineering and Electronics and  
Computer Science  
University of Liverpool

January, 2019



# Abstract

Conventional machine learning approaches usually assume that the patterns follow the identical and independent distribution (*i.i.d.*). However, in many empirical cases, such condition might be violated when data are equipped with diverse and inconsistent style information. The effectiveness of those traditional predictors may be limited due to the violation of the *i.i.d.* assumption brought by the existence of the style inconsistency. In this thesis, we investigate how the style information can be appropriately utilized for further lifting up the performance of machine learning models. It is fulfilled by not only introducing the style information into some state-of-the-art models, some new architectures, frameworks are also designed and implemented with specific purposes to make proper use of the style information. The main work is listed as the following summaries:

First, the idea of the style averaging is initially introduced by an example of an image process based sunglasses recovery algorithm to perform robust one-shot facial expression recognition task. It is named as Style Elimination Transformation (SET). By recovering the pixels corrupted by the dark colors of the sunglasses brought by the proposed algorithm, the classification performance is promoted on several state-of-the-art machine learning classifiers even in a one-shot training setting.

Then the investigation of the style normalization and style neutralization is investigated with both discriminative and generative machine learning approaches respectively. In discriminative learning models with style information, the style normalization transformation (SNT) is integrated into the support vector machines (SVM) for both classification and regression, named as the field support vector classification (F-SVC) and field support vector regression (F-SVR) respectively. The SNT can be represented with the nonlinearity by mapping the sufficiently complicated style information to the high-dimensional reproducing kernel Hilbert space. The learned SNT would normalize the inconsistent style information, producing *i.i.d.* examples, on which the SVM will be applied. Furthermore, a self-training based transductive framework will be introduced to incorporate with the unseen styles during training. The transductive SNT (T-SNT) is learned by transferring the trained styles to the unknown ones.

Besides, in generative learning with style information, the style neutralization generative adversarial classifier (SN-GAC) is investigated to incorporate with the style information when performing the classification. As a neural network based framework, the SN-GAC enables the nonlinear mapping due to the nature of the nonlinearity of the neural network transformation with the generative manner. As a generalized and novel classification framework, it is capable of synthesizing style-neutralized high-quality human-understandable patterns given any style-inconsistent ones. Being learned with the adversarial training strategy in the first step, the final classification performance will be further promoted by fine-tuning the classifier when those style-neutralized examples can be well

generated.

Finally, the reversed task of the upon-mentioned style neutralization in the SN-GAC model, namely, the generation of arbitrary-style patterns, is also investigated in this thesis. By introducing the W-Net, a deep architecture upgraded from the famous U-Net model for image-to-image translation tasks, the few-shot (even the one-shot) arbitrary-style Chinese character generation task will be fulfilled. Same as the SN-GAC model, the W-Net is also trained with the adversarial training strategy proposed by the generative adversarial network. Such W-Net architecture is capable of generating any Chinese characters with the similar style as those given a few, or even one single, stylized examples.

For all the proposed algorithms, frameworks, and models mentioned above for both the prediction and generation tasks, the inconsistent style information is taken into appropriate consideration. Inconsistent sunglasses information is eliminated by an image processing based sunglasses recovery algorithm in the SET, producing style-consistent patterns. The facial expression recognition is performed based on those transformed *i.i.d.* examples. The SNT is integrated into the SVM model, normalizing the inconsistent style information nonlinearly with the kernelized mapping. The T-SNT further enables the field prediction on those unseen styles during training. In the SN-GAC model, the style neutralization is performed by the neural network based upgraded U-Net architecture. Trained with separated steps with the adversarial optimization strategy included, it produces the high-quality style-neutralized *i.i.d.* patterns. The following classification is learned to produce superior performance with no additional computation involved. The W-Net architecture enables the free manipulation of the style data generation task with only a few, or even one single, style reference(s) available. It makes the Few-shot, or even the One-shot, Chinese Character Generation with the Arbitrary-style information task to be realized. Such appealing property is hardly seen in the literature.

**Key Words:** identical and independent distribution, inconsistent style information, style normalization, style neutralization, arbitrary-style generation

# Acknowledgement

I believe that the achievement of my academic research and studies during my Ph.D. candidate period can hardly be fulfilled without the help and the support of my supervisors, my colleagues, my friends, and my family.

I will send the most preferable gratitude to my primary supervisor and one of my close friends, Prof. Kaizhu HUANG, for his intensive guidance and patient suggestions not only for my research studies but also on my off-work life. No matter these supervising advice are supportive or critical; they inevitably pose a crucial and essential factor in my pursuit to be a qualified Doctor of Philosophy in the machine learning industry. Besides, the grateful gratitude will also be sent to two of my consecutive secondary supervisors, Dr. Tingting MU, and Dr. John Yanis GOULERMAS, for their support to present ideas and future directions on my research activities, and providing me with suggestions to get familiar with the academic writing style. All these have been carried out in my published papers and chapters during my Ph.D. studies, and the writing of this Ph.D. thesis.

Furthermore, I would also express my thanks to my colleagues in the Suzhou Municipal Key Laboratory of Cognitive Computation and Applied Technology. They are Dr. Fei CHENG, Dr. Qiufeng WANG, Dr. Yuyao YAN, Dr. Xi YANG, Mr. Shufei ZHANG, Mr. Guanyu YANG, Mr. Samer JAMMAL, Mr. Zhuang QIAN and others. They offered me intensive and continuous support both academically and in my Ph.D. life. Those friends also come from the other branches of the Xi'an Jiaotong-Liverpool University. They are Prof. Tamman TILLO, Dr. Rui LIN, Dr. Shaofeng LU, Dr. Jieming MA, Dr. Mark LEACH, Ms. Jingchen WANG, Ms. Yujie LIU, Ms. Xiaoyi WU, Mr. Bing HAN, Mr. Yanchun XIE, and Mr. Wenfei ZHU. Additional thanks will also be sent to Ms. Zijun CUI and Mr. Hengyang LUO for the help of designing several figures in this thesis, as well as Prof. Charlie C. L. WANG and Mr. Hongbo ZHOU, who had inspired me to artificial intelligence years ago.

I am also grateful to my family, particularly for my grandparents' teaching and guidance in my childhood, laying the foundation of my current studies and research work. Besides, I would also thank my younger cousin, Ms. Kexuan JIANG. Without her escort and enthusiasm, I will never stand a chance to get to these final days of my Ph.D. studies.



# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgement</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Algorithms</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Motivation . . . . .	2
1.2 Major Contributions . . . . .	4
1.3 Brief Summary of the Remaining Chapters . . . . .	6
<b>2 Research Background</b>	<b>11</b>
2.1 Non- <i>i.i.d.</i> Prediction Approaches . . . . .	13
2.2 MTL-based Approaches . . . . .	14
2.3 FPM-based Models . . . . .	15
2.3.1 Field Modeling Generative Approaches . . . . .	15
The Gaussian Mixture Model . . . . .	15
The Bilinear Model . . . . .	16
2.3.2 Image Processing Based Style Elimination . . . . .	16
2.3.3 FAM-based Approaches . . . . .	17
Discriminative Approaches for Style Normalization . . . . .	17
2.4 Generative Approaches with Style Information . . . . .	19
2.4.1 Generative Adversarial Network . . . . .	19
Image-to-Image (Img2Img) Translation . . . . .	20
2.4.2 GAN-based Models for Classification Performance Promotion . .	22
2.4.3 Few-shot Style Data Generation with Free Manipulation . . . . .	23
Few-shot Arbitrary-style Chinese Character Generation . . . . .	24
W-shaped Architecture . . . . .	26

<b>3</b>	<b>Style Elimination</b>	<b>29</b>
3.1	Research Background: Facial Expression Recognition . . . . .	30
3.2	Robust Sunglasses Detection and Region Recovery . . . . .	31
3.2.1	Correction for Roll Rotation . . . . .	31
3.2.2	Sunglasses Region Detection . . . . .	32
3.2.3	Grayscale Value Histogram Shifting . . . . .	33
3.2.4	Histogram Matching for Sunglasses Recovery . . . . .	33
3.3	Sunglasses Recovery Experiments . . . . .	36
3.3.1	Data Preparation . . . . .	36
3.3.2	Experimental Setting . . . . .	38
	Five-fold Scheme for Individual Training and Testing . . . . .	38
	Five-fold Scheme for Full Training and Testing . . . . .	38
	One-shot Scheme for Individual Training and Testing . . . . .	39
	One-shot Scheme for Full Training and Testing . . . . .	39
3.3.3	Experimental Results . . . . .	39
	Individual Training and Testing with Five-fold Scheme . . . . .	39
	Individual Training and Testing with One-shot Scheme . . . . .	42
	Full Training and Testing with both Schemes . . . . .	43
	Improvement difference between different experimental settings . . . . .	44
3.4	Summaries and Future Work . . . . .	45
3.4.1	Future Work . . . . .	46
<b>4</b>	<b>Discriminative Approaches with Style Information</b>	<b>47</b>
4.1	Problem Statement . . . . .	48
4.1.1	Typical Non- <i>i.i.d.</i> Prediction Scenarios . . . . .	48
4.1.2	F-SVM Basic Framework for Classification and Regression . . . . .	50
4.2	F-SVM Model Specification . . . . .	52
4.2.1	Basic Notation Involved . . . . .	52
4.2.2	Linear F-SVM Model . . . . .	52
4.2.3	Alternative Optimization . . . . .	54
	Predictor Learning . . . . .	54
	SNT Learning . . . . .	55
	Convergence Property . . . . .	56
4.2.4	Relationship with the MTL model . . . . .	57
4.2.5	Kernelized F-SVM Representation . . . . .	59
	Kernelized Update for the F-SVC Model . . . . .	59
	Kernelized Update for the F-SVR Model . . . . .	60
	Kernelized Objective Function . . . . .	62
	Kernelized Algorithms . . . . .	63
4.3	Prediction Rules for Future Patterns . . . . .	64
4.3.1	Singlet Prediction . . . . .	64
	Traditional Prediction Rule . . . . .	64
	Voted Prediction Rule (VPR, for F-SVC Only) . . . . .	64
	Averaged Decision Rule (APR, for F-SVR Only) . . . . .	65
4.3.2	Field Prediction Rule (FPR) . . . . .	65
4.4	Self-training based Transductive Learning . . . . .	65



4.4.1	Transductive F-SVC Formulation . . . . .	66
4.4.2	Transductive F-SVR Formulation . . . . .	67
4.5	Statistical Performance Evaluation . . . . .	68
4.5.1	Performance on the F-SVC Model . . . . .	69
	Face Classification across Head Poses . . . . .	71
	Speech Classification across Speakers . . . . .	72
	Chinese Handwriting Character Recognition across Writers . . . . .	73
4.5.2	Performance on the F-SVR Model . . . . .	74
	Synthetic Data . . . . .	74
	School Effectiveness Data . . . . .	77
	Computer Survey Data . . . . .	78
4.6	Visualized Evaluation of Field Normalization . . . . .	78
4.6.1	Style Normalization with the Linear Kernel . . . . .	79
	Original Task of the Face Data . . . . .	80
	Reversed Task of the Face Data . . . . .	80
	Original Task of the Facial Expression Data . . . . .	80
	Reversed Task of the Facial Expression Data . . . . .	80
4.7	Model Properties: Further Studies . . . . .	85
4.7.1	Class Separability Improvement . . . . .	85
4.7.2	Convergence Property . . . . .	88
4.7.3	Parameter Sensitivity . . . . .	88
4.8	Summary and Future Work . . . . .	95
4.8.1	Future Work . . . . .	95
<b>5</b>	<b>Generative Approaches with Style Information</b>	<b>97</b>
5.1	Style Neutralization Generative Adversarial Classifier based on the Upgraded U-Net Architecture . . . . .	98
	Research Background . . . . .	99
5.1.1	SN-GAC Model Specification . . . . .	100
	Preliminaries . . . . .	101
	Upgraded U-Net based Generator . . . . .	103
	Optimization Details of the $G$ network . . . . .	104
	Optimization Losses of $D - C$ Network . . . . .	105
	Two-Phase Training Strategy with Joint Losses . . . . .	105
5.1.2	SN-GAC Experiments . . . . .	106
5.1.3	Summary and Future Work . . . . .	112
	Future Work . . . . .	113
5.2	W-Net for Few-shot Multi-content Arbitrary-style Chinese Character Generation . . . . .	114
5.2.1	Research Background . . . . .	114
5.2.2	Model Definition . . . . .	115
	Preliminaries . . . . .	115
	W-Net Architecture . . . . .	117
	Optimization Strategy and Losses . . . . .	119
5.2.3	W-Net Experiment . . . . .	120
	Experiment Setting . . . . .	121

Model Reasonableness Evaluation . . . . .	121
Model Effectiveness Evaluation . . . . .	122
Generation Performance Variation due to Different Numbers of Style References . . . . .	124
Analysis on Failure Examples . . . . .	124
5.2.4 Possible Statistical Evaluation Procedures . . . . .	126
5.2.5 Further Studies for Other Eastern Asian Characters . . . . .	129
5.2.6 Summary and Future Work . . . . .	129
Future Work . . . . .	130
<b>6 Conclusion</b>	<b>135</b>
6.1 Review of the Thesis . . . . .	135
6.2 Future Work . . . . .	137
<b>Appendix: A list of Publications</b>	<b>139</b>
<b>Reference</b>	<b>141</b>

# List of Figures

1.1	Digits written by different writers . . . . .	2
1.2	Conventional prediction and prediction with style averaging . . . . .	3
1.3	Manipulation of style data generation . . . . .	4
2.1	MTL model framework . . . . .	12
2.2	Original U-Net Architecture . . . . .	21
2.3	Original TripleGAN Architecture . . . . .	22
2.4	Original MC-GAN Architecture . . . . .	23
2.5	Original Zi2Zi Architecture . . . . .	25
2.6	W-shaped Architecture . . . . .	26
3.1	Portrait examples in the modified Japanese Female Facial Expression database with multiple sunglasses . . . . .	30
3.2	Roll rotation correction of facial images . . . . .	32
3.3	Histogram comparison among sunglasses with different grayscale value dropping rate (GVDR) . . . . .	33
3.4	Histogram equalization . . . . .	34
3.5	The 77 Stasm landmarks and their numbers. . . . .	36
3.6	Examples of five-fold scheme for full training and testing . . . . .	42
3.7	Exapmles one-shot scheme for full training and testing . . . . .	43
3.8	Improvement on final recognition rate (FRR) with SVM Classifier brought by the proposed sunglasses recovery algorithm . . . . .	44
3.9	Improvement on FRR with LDA classifier . . . . .	44
3.10	Improvement on FRR with KNN classifier . . . . .	45
3.11	Overall performance improvement on FRR with consistent improvement of SVM, LDA, and KNN . . . . .	45
4.1	Non-identical data generation process from style-inconsistent data sources for classification tasks . . . . .	48
4.2	Non-identical data mapping during data generation process for regression tasks . . . . .	49
4.3	Architecture of the Field Support Vector Machines . . . . .	51
4.4	Traditional classification and field classification . . . . .	53
4.5	F-SVC testing sample repeated concatenation scheme . . . . .	66
4.6	F-SVR testing sample concatenation scheme . . . . .	69
4.7	F-SVC performance on the Point' 04 data (the Face Data) . . . . .	72
4.8	F-SVC performance on the Connectionist Bench database (the Speech Data) . . . . .	73

4.9	F-SVC performance on the CASIA-OLHWDB database (the HW Data)	74
4.10	F-SVR performance on the synthetic linear data	76
4.11	F-SVR performance on the synthetic nonlinear data	76
4.12	F-SVR performance on the School Effectiveness data	78
4.13	F-SVR performance on the Computer Survey data	79
4.14	F-SVC performance visualization of the Point' 04 data (original task): original images	81
4.15	Style information (head poses) of Fig. 4.14	82
4.16	Style-normalized images of Fig. 4.14	83
4.17	Class and field information alternated of Fig. 4.14 (reversed task): origi- nal images	84
4.18	Style information (individual identities) of Fig. 4.17	85
4.19	Style-normalized images of Fig. 4.17	86
4.20	The JAFFE Database (original task): original images	87
4.21	Style information (individual identities) of Fig. 4.20	87
4.22	Style-normalized images of Fig. 4.20	88
4.23	Class and field information alternated of Fig. 4.20 (reversed task): origi- nal images	89
4.24	Style information (facial expressions) of Fig. 4.23	90
4.25	Style-normalized images of Fig. 4.23	91
4.26	The t-SNE embedding comparison brought by the F-SVC model	92
4.27	F-SVC convergence performance visualization	93
4.28	F-SVC parameter sensitivity analysis on the Face Data	93
4.29	F-SVC parameter sensitivity analysis on the Sspeech Data	94
4.30	F-SVC parameter sensitivity analysis on the HW Data	94
5.1	Traditional classifier and the proposed SN-GAC classifier	101
5.2	SN-GAC architecture	102
5.3	Upgraded U-Net structure	103
5.4	Examples of performance visualization on the Point' 04 data with the SN- GAC model	109
5.5	Performance evaluation of the F-SVC model as the identical example given in Fig. 5.4	110
5.6	An example of a handwriting Chinese character with standard, isolated, and cursive writing styles	110
5.7	Performance Visualizaton on the HW data produced by the SN-GAC model	111
5.8	Some incorrectly classified examples on the HW data produced by the SN-GAC model	112
5.9	Generated Chinese characters synthesized by the W-Net model	116
5.10	W-Net Architecture	118
5.11	Several examples of generated data of unseen printing and handwriting styles	122
5.12	Several examples of generated characters of seen styles	123
5.13	Several examples of generated characters of unseen styles	124
5.14	Unsatisfied generated examples	126
5.15	Example printed font characters for Metric Comparison	127

5.16	Evaluation of metrics comparison . . . . .	128
5.17	The input few brush-written examples of actual Chinese characters . . . . .	129
5.18	Some Korean characters with <i>circular radicals</i> . . . . .	130
5.19	The W-Net (trained with only simplified Chinese characters) generated essay of traditional Chinese characters from the one selected style refer- ence shown in Fig. 5.17 . . . . .	131
5.20	The W-Net (trained with only simplified Chinese characters) generated essay of Korean characters from the one selected style reference shown in Fig. 5.17 . . . . .	132
5.21	The W-Net (trained with only simplified Chinese characters) generated essay of Japanese characters from the one selected style reference shown in Fig. 5.17 . . . . .	133



# List of Tables

3.1	Glass region detection . . . . .	32
3.2	Examples of histogram matching performance of sunglasses . . . . .	35
3.3	Examples of manually-added sunglasses . . . . .	37
3.4	Sunglasses recovery performance on individual training and testing with five-fold scheme . . . . .	40
3.5	Sunglasses recovery performance on individual training and testing with one-shot scheme . . . . .	41
3.6	Sunglasses recovery performance on full training and testing with both five-fold and one-shot schemes . . . . .	42
4.1	F-SVC performance summary on the Face, Speech and the HW Data. . .	71
4.2	F-SVR performance on synthetic linear datasets . . . . .	75
4.3	F-SVR performance on synthetic nonlinear datasets . . . . .	75
4.4	F-SVR performance on the School Effectiveness data . . . . .	77
4.5	F-SVR performance on the Computer Survey dataset . . . . .	79
5.1	Performance evaluation of the SN-GAC model and other relevant baselines involved . . . . .	108
5.2	Generated characters with different numbers of style references . . . . .	125





# List of Algorithms

1	F-SVC SNT alternative learning with the linear kernel . . . . .	57
2	F-SVR SNT alternative learning with the linear kernel . . . . .	58
3	Kernelized F-SVC SNT alternative learning . . . . .	63
4	Kernelized F-SVR SNT alternative learning . . . . .	64
5	Kernelized F-SVC T-SNT alternative learning . . . . .	68
6	Kernelized F-SVR T-SNT alternative learning . . . . .	70



# Chapter 1

## Introduction

In the recent decade, fantastic breakthroughs and magnificent achievements have been witnessed due to the ever-changing technology of Artificial Intelligence (AI). Quite a lot aspects of peoples' daily lives have been fundamentally transformed into being more convenient, comfortable, and efficient than ever before.

As the core components of the AI applications, the Machine Learning (ML) theories and framework have been heavily studied and effectively applied. Theoretical promotions and empirical applications are extensively and intensively proposed to both the relevant research communities and common peoples' daily lives. Particularly, it has encouraged or even forced a great amount of existing modern technologies and/or operational regulations (that people have got used to in old times) to be transformed into completely new shapes. Some of the novel and innovative occupations, professions, and industries also appear.

Upon that, the machine learning technologies have been developed and implemented into quite diverse categories with various upgradation of the existing approaches to solve many of the practical problems. In particular, deep learning approaches are becoming one of the major streams in the machine learning technologies and practical applications. Typical deep frameworks include the AlexNet [1], a famous deep Convolutional Neural Network (CNN), proposed in 2012 for the large-scale classification task, and the Generative Adversarial Networks (GAN) [2], *the most interesting idea in the last ten years* (by Yann Lecun in an interview reported in [3]).

However, there are still problems and issues awaiting to be analyzed, clarified, and tackled in the current machine learning models both theoretically and empirically. One typical example is the assumption of the *identical and independent distribution (i.i.d.)* assumption) of the data involved in most machine learning algorithms and frameworks. Sometimes, such an assumption may not hold. In some cases, the inconsistency existed in the data distribution may pose severe damage to the final prediction performance. A simple idea is to develop an algorithm to average the style inconsistency to produce corresponding patterns with the same style information. With diverse proposals, the *i.i.d.* assumption will be satisfied, benefiting the following prediction tasks including both the classification and regression. This is one of the key issues to be investigated in the research work of this thesis.

In the meanwhile, the generation of such inconsistent style data is also an interesting research topic. Seen as the reversed task of the averaging the style inconsistency, the generation process for arbitrary-style data with few, or even one single, available stylistic

example(s) will also be investigated in the thesis.

## 1.1 Research Motivation

Most of the conventional machine learning algorithms hold the assumption that data involved in the learning models shall follow the *identical and independent distribution*, namely, the *i.i.d.* assumption. It means that each random pattern is equipped with the same distribution as the others. Moreover, all of them are mutually independent [4] at the same time. However, such a hypothesis might not be true in some empirical scenarios. In particular, violations can be found when involved patterns are equipped with inconsistent style information. They are named as **style-discriminative**, or **style-inconsistent** patterns in the following chapters and sections of this thesis.

Cases of such style inconsistent input patterns can be found in some of the empirical cases. One typical scenario is the character recognition across multiple writers. In fact, peoples' writing styles vary hugely among different individuals, as illustrated in Fig. 1.1 [5]. In the example, two individual digits with identical written strokes are classi-

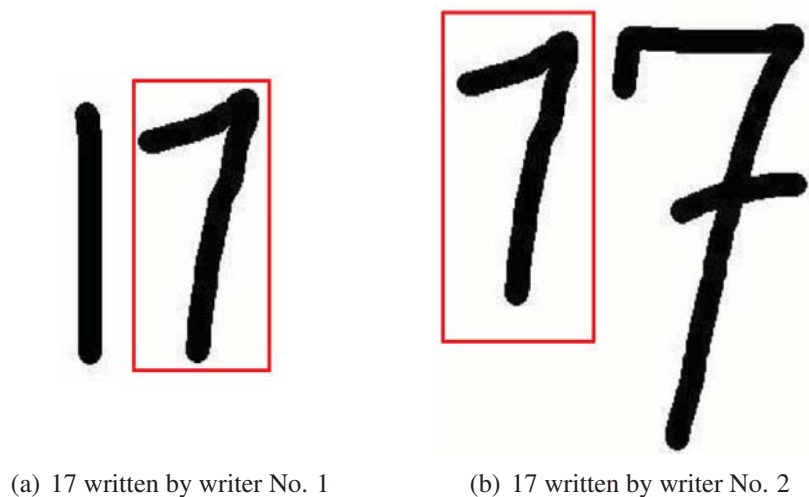


Fig. 1.1: Digits written by different writers [5]. The digit with red rectangle is '7' in (a), while it seems to be '1' in (b). However, they are actually identical in strokes.

fied into different categories since they are equipped with inconsistent style information. In this sense, the *i.i.d.* assumption is obviously violated.

Conventional machine learning approaches, holding the *i.i.d.* assumption, do not take the inconsistent style information into consideration. It can be seen in Fig. 1.2(a). These models include the state-of-the-art Support Vector Machine (SVM) [6], as well as the recently evolved and extensively developed deep learning models [1, 7, 8]. Style-inconsistent samples are simply put into the algorithms to be trained and predicted without any effort to model the potential relationship among them.

Nevertheless, with a sufficiently larger number of the training parameters, the representative power of the modern deep neural network models is significantly greater than previous frameworks including the SVM model [9]. Deep learning models have achieved

great success especially when they are fed in a large size of training samples. However, when confronted with the example given in Fig. 1.1, these deep models may be unable to assign the correct class labels. The classification performance may even be degraded severely when such style-inconsistent cases exist in the dataset, but are not taken into proper consideration.

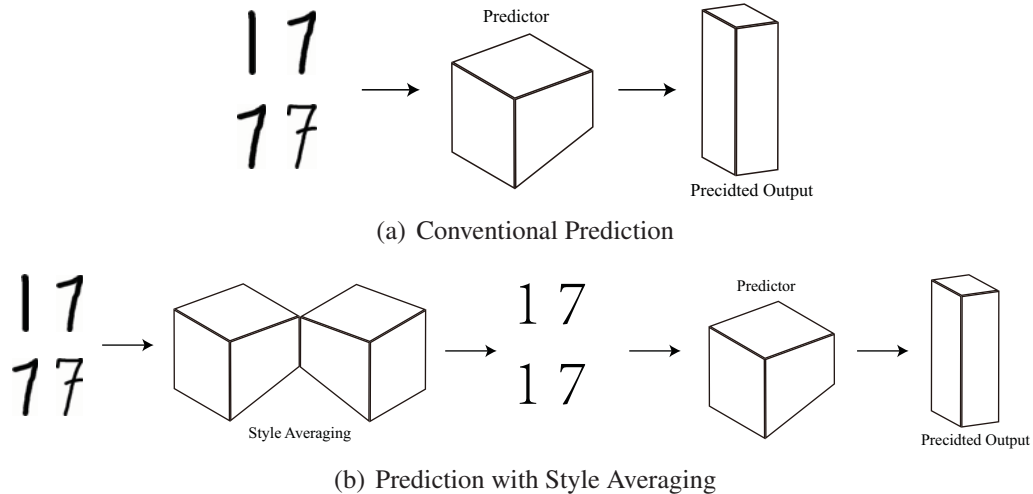


Fig. 1.2: Conventional prediction and prediction with style averaging.

A reasonable solution for the non-*i.i.d.* classification task is to find an effective manner to drive out, or to average, the style inconsistency. It will, in this way, produce patterns without style information, or equipped with the identical stylized tendency. These style-averaged *i.i.d.* examples will be helpful to improve the final prediction accuracy when fed into the predictor. Fig. 1.2(b) illustrates the basic framework of the prediction scheme with the style averaging idea.

There are various methods to average the inconsistent style information introduced in this thesis. In Chapter 3, an example to illustrate briefly the style averaging transformation is studied. The improved classification is achieved by an image processing based algorithm.<sup>1</sup> In Chapter 4, an SVM-based discriminative framework to accomplish the non-*i.i.d.* prediction task is designed in detail. Named as the Field-SVM (F-SVM), it consists of the Field Support Vector Classification (F-SVC) and the Field Support Vector Regression (F-SVR). The transformation to perform the style normalization<sup>2</sup> is also learned in an alternative manner along with the SVM training itself. Particularly, in Section 4.2.5, the kernelized version of the F-SVM framework is specified in detail. It enables the SNT to be represented by the nonlinear kernel mapping. Meanwhile, in Section 4.4, a self-training strategy is proposed in order to perform prediction on data of unknown styles. The training style information will be transferred to the unseen styles thanks to the introduction of the Transductive-SNT (T-SNT).

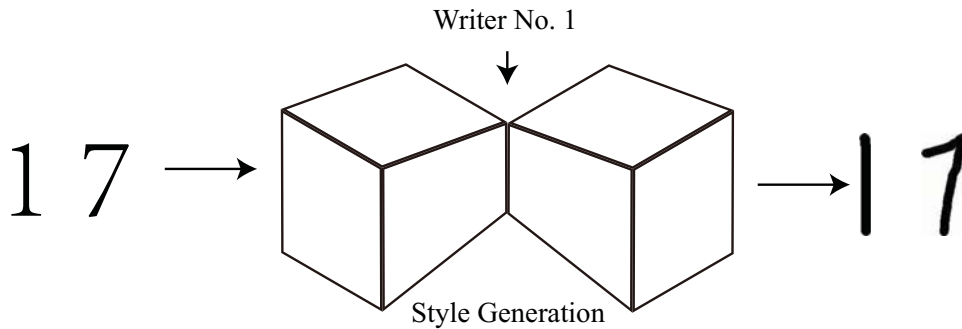
A generative framework named Style Neutralization Generative Adversarial Classifier (SN-GAC) will be introduced in Section 5.1. In that model, the style neutralization<sup>3</sup> is

<sup>1</sup>The style averaging is called as the style elimination in this image processing based algorithm.

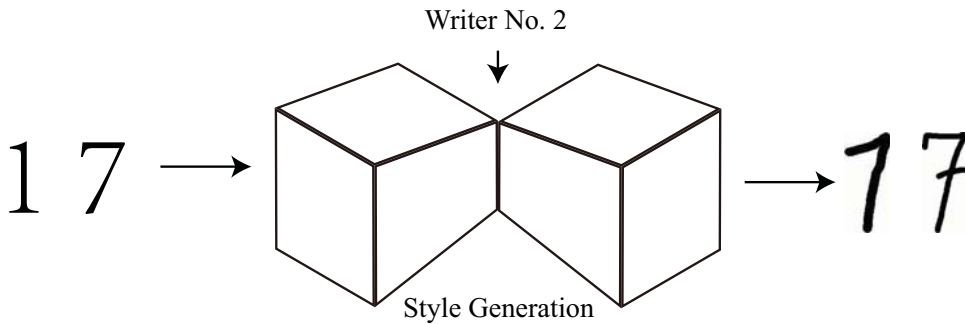
<sup>2</sup>It is known as the Style Normalization Transformation (SNT) in the F-SVM models.

<sup>3</sup>It is known as the Style Neutralization in the SN-GAC model.

fulfilled with an upgraded architecture of the U-Net network [10]. Thanks to the nonlinear nature of the neural network mapping, the style information is neutralized nonlinearly. Learned with the adversarial training strategy [2, 11], the SN-GAC model performs the classification with the style-neutralized patterns.



(a) Manipulated style generation for writer No. 1 in Fig 1.1(a)



(b) Manipulated style generation for writer No. 2 in Fig 1.1(b)

Fig. 1.3: Manipulation of style data generation.

In Section 5.2, the manipulation of style data generation task with few, or even one single, stylized example available will be also investigated. As the reversed task of the style neutralization mentioned in the last paragraph, the free manipulation of the styles and the contents of the generated examples is realized. It is accomplished by the W-Net, a generative model inherent from the upgraded U-Net architecture demonstrated in Section 5.1 and trained with the alternative adversarial strategy as well. A few-shot (or even one-shot) arbitrary-style Chinese character generation framework will be investigated as a brief introduction of such data generation manipulation scheme and procedures. It enables to generate any Chinese character of any handwriting style or printed font with only a few (even only single) sample(s) available. Such appealing property and function is hardly seen in the literature.

## 1.2 Major Contributions

The major contributions of the research reported in this thesis are summarized as follows:

- The idea of the style averaging is introduced with an example of an image processing based sunglasses recovery algorithm [12]. It is utilized in the facial expression recognition task when portraits involved wear diverse kinds of sunglasses with

changeable shapes and various luminousness. Named as style elimination in this chapter, the style information introduced by the sunglasses is eliminated before the facial images are put into the conventional *i.i.d.*-assumed classifiers.

- The input facial images are de-rotated to compensate the roll orientation variation, before which the possible sunglasses region will be effectively detected.
  - A histogram matching algorithm will then be performed on the detected sunglasses region. Consequently, the corrupted pixels in that region covered by sunglasses will be intelligently recovered. The style information brought by them can then be eliminated.
  - The proposed sunglasses recovery technique enables the one-shot training scheme. Namely, only one single training example is needed for each individual and each facial expression to perform the classifier training while keeping an acceptable recognition rate.
- The Style averaging is designed into a classical discriminative machine learning algorithm, the Support Vector Machine (SVM), for both the classification [13, 14] and regression scenarios [15]. The relevant transformation to perform style averaging is named as the Style Normalization Transformation (SNT), as demonstrated in the previous sections in this chapter. In this proposed framework named as the Field-SVM (F-SVM), a group of data equipped with consistent style information is called as a field.
    - In comparison with the traditional machine learning models, the F-SVM model makes no *i.i.d.* assumption by introducing the SNT. It is designed to normalize the inconsistent style information embedded in the original non-*i.i.d.* patterns, producing corresponding style-normalized examples.
    - The learning of the SNT is incorporated with the training of the SVM in an alternative fashion. In each training iteration, the SVM parameters are updated on the most-updated style-normalized patterns, while the SNT is computed with the theoretical closed-form solution.
    - The kernelized version of the F-SVM is also investigated. It enables the non-linear engagement of the SNT, more powerful to represent the sufficiently complicated style information in real scenarios.
    - Several related decision rules will be proposed to conduct the field prediction task. Particularly, a self-learning based transductive framework [16] is further introduced to perform the field prediction on those examples with unseen styles during the training. The Transductive-SNT (T-SNT) will be learned, while the predictor for the new style can be optimized simultaneously.
  - The inconsistent style information is also studied to be averaged, or neutralized, with a generative model to promote the final classification performance. It is named as the Style Neutralization Generative Adversarial Classifier (SN-GAC) model [17]. As a generative framework, a well-trained SN-GAC model is capable of producing high-quality human-understandable style-neutralized examples when given the

corresponding style-inconsistent counterparts. The following classification will be performed on these style-neutralized *i.i.d.* patterns.

- Learning with the adversarial training strategy introduced in [2, 11], the SN-GAC model includes a generative model and a discriminative model.
  - The generative model is responsible for producing high-quality human-readable style-neutralized *i.i.d.* examples when given a corresponding style-inconsistent non-*i.i.d.* patterns.
  - The discriminative model is trained adversarially to distinguish between the generated style-neutralized patterns and the real standard style-free instances. The classifier is attached in the discriminative model. It is trained along with both the generative and the discriminative models to perform the desired pattern classification tasks.
  - A two-stage training strategy is introduced. The first stage is the upon-mentioned adversarial training. When it is saturated, the generative model will be fixed. Only the classifier attached on the discriminative model will be further fine-tuned with the cross-entropy loss merely upon the final target to improve the classification performance.
- The data equipped with diverse style information also can be generated with a clear manipulation involvement. It is fulfilled by the proposed W-Net framework [18], which is inherent from the upgraded U-Net architecture. The few-shot (or even the one-shot) arbitrary-style Chinese character generation task will be achieved.
    - The proposed W-Net model includes two parallel encoders (the content prototype encoder and the style reference encoder) and one decoder. Extracted features of both content and style information from the both encoders respectively are combined together before being sent to the decoder.
    - The generated character is consistent in content with the characters input into the content prototype encoder, while the style tendency is optimized to be close to the input characters to the style reference encoder.
    - The training of the W-Net framework also follows the adversarial learning strategy with the help of the other discriminative model. A well learned W-Net enables to generate any character consistent in style with a few, or even a single style character available.

### 1.3 Brief Summary of the Remaining Chapters

**Chapter 2: Research Background.** In this chapter, a detailed literature overview of the relevant discriminative machine learning models designed for the non-*i.i.d.* prediction tasks is provided. Then those prediction models where the style inconsistency is taken into consideration are demonstrated. They are the Multi-task Learning (MTL) frameworks, and the Field Prediction Models (FPM). For the MTL based frameworks, an individual predictor is trained for each task, where a task represents a specific prediction on a group of patterns equipped with a particular kind of style information.



For the FPM models, on the other hand, only one predictor is optimized. The Field Modeling (FM) approaches as well as the Field Averaging Models (FAM) will be specified. The style mixture model [19] and the bilinear model [20] will be included in the review of the FM-based algorithms. In the FAM-based frameworks, not only the image processing based style elimination (as demonstrated in Chapter 3 with an example of sunglasses recovery algorithm), as well as the discriminative style normalization transformation (investigated in Chapter 4) are included. In these algorithms or models, the trained predictor is learned from those style-averaged *i.i.d.* patterns. They are generated by multiple methods. The identical style averaging idea is also integrated into a generative machine learning model to perform the style neutralization mapping, which will be reviewed in Section 2.4.2.

In the final parts of the chapter, a particular investigation of the one of the most advanced and emerging topic of the deep learning models, the Generative Adversarial Networks (GAN) [2], is demonstrated in both theoretical and empirical perspectives. It is the basis of the adversarial training strategy employed for both the upgraded U-Net (Section 5.1) and the W-Net (in Section 5.2). In this sense, it is of key importance to review the generative approaches with style information that will be studied in Chapter 5. Besides, a detailed review of the U-Net [10] as well as the relevant GAN-based Image-to-Image translation [21] (**Img2Img**) frameworks is also made. Additionally, previous research literature of GAN-based models on data augmentation and synthesization, promotion of the classification performance, and the manipulation of the generated style images (e.g., characters or alphabets) are also analyzed in detail.

**Chapter 3: Style Elimination.** In this chapter, an instructive example to demonstrate the idea of the style elimination transformation to transform the style-inconsistent patterns into the style-eliminated data is first provided. In the Facial Expression Recognition (FER) task, the corrupted region due to sunglasses with various luminousness and diverse colors would damage with final recognition performance, since they are not *i.i.d.* patterns. The inconsistent style information brought by those diverse sunglasses are eliminated by the proposed sunglasses recovery algorithm, including the portrait de-rotation, the sunglasses region detection, and the image recovery with the histogram equalization algorithm. It produces style-eliminated *i.i.d.* examples, before which they are put into the conventional classifier including the SVM, the Linear Discriminant Analysis (LDA), and the  $k$ -Nearest Neighbor (KNN). The benefit is found by introducing the sunglasses recovery algorithm to eliminated the original style information. This research has been reported in a journal paper [12].

**Chapter 4: Discriminative Approaches with Style Information.** In this chapter, a classical discriminative machine learning model, namely, the Support Vector Machine (SVM), is incorporated with the idea of the style normalization transformation to perform the non-*i.i.d.* pattern prediction task. It is investigated for both the classification (Field Support Vector Classification, F-SVC [13, 14]) and the regression (Field Support Vector Regression, F-SVR [15]) tasks, where a group of data with specific style information is named as a field. For both models, the SNT and the classifier (or regressor) are learned simultaneously with an alternative optimization strategy, in which way the inconsistent style can be normalized readily and effectively.

Besides, the kernelized representation of the F-SVM model will also be deducted theoretically. The style information can be formulated with the kernelized nonlinear mapping

accordingly. It is superior to the linear representation proposed in the previous research literature of the FPMs, capable of modeling the sufficiently complicated style information in real scenarios. The F-SVC model is reported in the journal paper [14], while the extended version is investigated in a conference paper [13]. The F-SVR model was published in a conference paper [15], while the theories are applied in two empirical applications, leading to two journal publications [22, 23].

Furthermore, multiple decision rules are discussed for the field prediction. Particularly, a self-learning based Transductive SNT (T-SNT) learning will be further proposed to perform the prediction on patterns with unseen style during training. It is effective in transferring the trained style information to the unknown style, capable of making use of the inconsistent style information embedded in the original data distribution. Such a self-learning proposal is reported as an academic book chapter in [16].

The proposed F-SVM model is extensively evaluated with both statistical performance and visualized illustration. Not only better prediction performance has been achieved when compared with several relevant baselines, but also the stylistic tendency can be observed from the obtained images representing the difference between the style-inconsistent and the style-normalized data achieved by the linear F-SVC. Additionally, the proposed SNT improves the class separability in the data manifold, which is also beneficial to performing the following classification task.

**Chapter 5: Generative Approaches with Style Information.** In this chapter, the style averaging is further developed for generative frameworks. In this way, an upgraded version of the classical U-Net [10] for the Img2Img [21] translation, namely, the Style Neutralization Generative Adversarial Classifier (SN-GAC), is proposed to perform the style neutralization of the input non-*i.i.d.* and the classification on the style-neutralized *i.i.d.* examples. Optimized with the adversarial training strategy originated from the Generative Adversarial Network (GAN) [2] and improved in [11, 24], the generative model is responsible for producing high-quality human-understandable style-neutralized patterns when given the style-inconsistent examples. Simultaneously, the discriminative model is to distinguish those generated examples by the generative model from those real instances.<sup>4</sup> The nonlinear nature of the neural network based generative model also enables the nonlinear representation of the style information. Additionally, the classification on the style-neutralized patterns is performed according to the proposal of the auxiliary classification generative adversarial network proposed in [26] by attaching the classification neurons at the end of the discriminative model. The training of the SN-GAC model is divided into two steps. The initial procedure is to train both the generative and the discriminative model in an alternative and adversarial fashion. The discriminator with the classifier will be further fine-tuned when the training of the generative model is saturated. The SN-GAC model is evaluated with several relevant baselines including the F-SVC model proposed in Chapter 4. The best performance is achieved without the access to the testing data by the transductive self-training strategy. Additionally, the SN-GAC model is also able to generate high-quality human-understandable style-neutralized patterns produced by the nonlinear generative model. On the contrast, the F-SVC can only produce linear style-normalized examples. The proposed SN-GAC model is published as a con-

---

<sup>4</sup>The discriminator in the SN-GAC model actually follows the Wasserstein-GAN with Gradient Penalty (W-GAN-GP) model [24], where it is fulfilled by minimizing the Wasserstein distance [25] between the generated examples and the real instances.

ference paper [17] and also invited and submitted for a journal publication [27].

Besides, in Chapter 5, it is investigated to make use of the style information in the generative models to realize the few-shot (or even one-shot) arbitrary-style Chinese character generation task. The proposed model is named as the W-Net, in which two parallel encoders, including the content prototype encoder and the style reference encoder, and a decoder, are engaged. Extracted content and style features from the two encoders are combined together with both the residual blocks [8] or simple feature skip connection [10] before being sent to the decoder. A well-optimized W-Net model based on the adversarial training strategy [2, 24] enables to synthesize a character which is consistent in content with those input patterns to the content prototype encoder, while maintains the style tendency of those characters to be sent to the style reference encoder. One appealing property of the proposed W-Net model is that only a few (or even one single) style characters is/are needed when performing the generation function with high-quality. This is quite hard to be seen in the relevant literature with only a handful of research records can be found. The W-Net framework for one-shot single-content arbitrary-style Chinese character generation task is also published as a conference paper [18].

**Chapter 6: Conclusion.** In this chapter, the final summary of this thesis will be presented, followed by the future work for the research in relevant domains.

For each of these chapters mentioned above, we have tried to make them self-contained. Therefore, some of the crucial contents, demonstrations, model definitions, and depictive illustrations might be reiterated in the following chapters when necessary.



# Chapter 2

## Research Background

A brief review of the relevant research literature is conducted on the approaches of performing the non-*i.i.d.* data prediction (classification or regression) tasks. After that, the review on the related models for few-shot/one-shot style data generation task will also be included in this chapter.

Two major methods to perform the non-*i.i.d.* prediction, including the Multi-task Learning [28] (MTL) frameworks and the Field Prediction Models (FPM) [29], are reviewed in detail. Compared with the approaches taking no style-inconsistency into consideration, better performance has been observed in both the MTL-based models and the FPM-based frameworks.

In the MTL-based algorithms, the prediction operation on a set of data equipped with a specific kind of information is called as a task. In each task, a corresponding predictor is trained. In another word, the number of predictors is the same as the number of kinds of inconsistent style information. The inter-relationship among different tasks is also taken into consideration.<sup>1</sup> The basic flowchart of the MTL-based models is depicted in Fig. 2.1.

On the contrary, in the FPM-based approaches, only one single predictor is learned with multiple kinds of methods. One of the FPM-based approaches is the Field Modeling (FM) frameworks, where the field inconsistency is formulated by pre-defined probabilistic models with the generative machine learning approaches. For example, in the Style Mixture Model proposed in [19] where the style information is assumed to be a linear combination of a fixed number of Gaussian distributions. In the Bilinear Model investigated in [20], the style and the content information is separated by introducing two independent distribution models.

The other kind of the FPM-based models is the Field Averaging Models (FAM). Distinguished from those MTL-based models and the FM-based formulations, the Style Averaging Transformation (SAT) is learned to average the inconsistent style information in the FAM-based models. It produces patterns with the identical stylistic tendency, or without any style information. In some cases, the SAT is learned along with the optimization

---

<sup>1</sup>There are multiple ways to depict the inner-relationship. E.g., in [30], it is represented by penalizing the variation of weight in each task with the averaged weight among different tasks. In [31], the sharing of model parameters describes the task relationship. In [32], task parameters within a cluster lie in a low-dimensional subspace. The overlapping information is represented by the number of latent subspace basis in common. These details will be described in detail in Section 2.2.

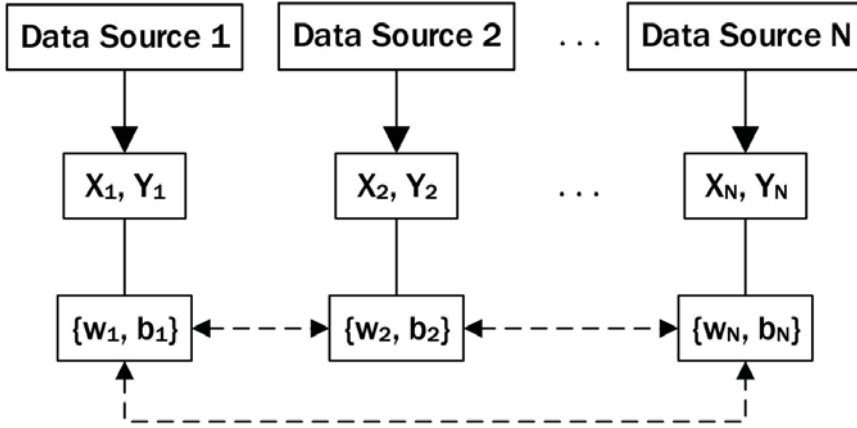


Fig. 2.1: MTL model framework:  $\{w_i, b_i\}$  represents an independent predictor trained for the  $i$ -th task dash lines represent possible relevance between different predictors during training.

of the predictor. In these models, the SAT is responsible for averaging the diverse and complicated style information embedded in the original input style-inconsistent patterns, producing style-averaged examples that satisfy the *i.i.d.* assumption. Such an idea has been briefly introduced in Chapter 1. The basic framework of the FAM models is illustrated in Fig. 4.3 by taking the Field Support Vector Classification (demonstrated in Chapter 4) as an illustrative example.

There are both discriminative and generative approaches in the machine learning research communities trying to normalize, or neutralize, the inconsistent style information to perform the field prediction task. Relevant baselines of both the discriminative and the generative models will be discussed. Additionally, as a supplementary material of the style averaging transformation in the FAM-based models, some other plain and easy image processing based techniques (named as the style elimination transformation) equivalent to SAT are also demonstrated before those machine learning based optimization approaches. It will be specified in Chapter 3.

As for the generative approaches to make use of the inconsistent style information to perform the style averaging, the Generative Adversarial Network [2] (GAN) is a hotspot and a frequently-discussed topic [3]. It engages a discriminator to distinguish the generated data synthesized by the generator in the GAN framework from those real examples enrolled in the training dataset. In the meanwhile, it is capable of approximating the real data distribution by optimizing the generator in an alternative and adversarial fashion. In the Wasserstein-GAN [24] (W-GAN) framework, the vanilla GAN is upgraded by replacing the original binary optimization based on the Jensen-Shannon divergence [33] with the minimization of the Wasserstein distance [25] between the generated samples and the corresponding real ones.<sup>2</sup>

The involvement of the discriminator in the GAN-based models adequately describes the most advanced power of the contemporary AI sciences and ML technologies since it is an automatic supervision in the approximation on real data distribution. The superi-

<sup>2</sup>The discriminator in [24] is named as *critic*, however, we will keep the name as *discriminator* in the following chapters of this thesis for the naming consistency.

ority can be even obvious when compared to other current generative models including the Gaussian Mixture Model [34] (GMM), as well as the Variational Auto-encoder [35] (VAE). The engagement of the independent discriminator plays a crucial role in this advantage.

The idea of the style averaging can also be realized by introducing the GAN-based models to perform the style neutralization transformation. Formulated to synthesize the style-neutralized examples when given the style-inconsistent counterparts by an Img2Img model [21], it can be learned in an adversarial manner.

Moreover, the reserved task of the upon-mentioned style neutralization can also be accomplished with the GAN-based adversarial training strategy. Named as the style data generation, a suitable example of the relevant research is the arbitrary-style Chinese character generation task with few, or even one single, stylized example(s) available. In the following relevant sections, several recent and instructive proposals and research will be specified. Both the advantage and drawbacks will be compared as well.

## 2.1 Non-*i.i.d.* Prediction Approaches

Most of the *i.i.d.*-assumed machine learning models deal with the non-*i.i.d.* prediction task by simply ignoring the style-inconsistency. Consequently, the prediction performance of them is in this way severely restricted. Particularly, as demonstrated in relevant research reported in [29, 15, 13, 14, 16], some of the non-deep learning models (such as SVM and NCM) fail to generalize to the style-inconsistent testing examples. Those comparison involved in this thesis without non-*i.i.d.* consideration include the ordinary linear regression, ridge regression, Support Vector Machines (SVM, for both classification [36] and regression [37]) and the Nearest Class Mean (NCM), etc.

The situation was changed due to the boosting of deep learning models. Nowadays, the science and technology in AI and the ML industries were pushed forward to a new frontier that was hardly imagined in previous times. The machine learning models are fundamentally transformed by the automatic learning of the extracted features according to the specific task, no matter it is classification, regression, image reconstruction, and/or data generation. The previous manual-designed features (e.g., Gabor wavelet feature [38], LBP features [39], and HOG features [40]) proposed in the last decades of the machine learning communities are mostly replaced by them. Some of the recently-proposed deep learning models such as the Alexnet [1], and the Vgg-Face-Net [41, 7], etc., are also engaged in this thesis in some key comparison scenarios.

It seems that the non-*i.i.d.* prediction issue can also be well handled by the deep learning models. However, as demonstrated in [42], successful training of a deep learning model requires a large amount of training data for better optimizing the huge number of parameters [43]. In real scenarios, a large amount of the training data may not always be available. As will be seen in the following chapters, the performance of the non-*i.i.d.* prediction produced by those *i.i.d.* deep learning models is less effective when compared with those MTL-based and FPM-based models.

## 2.2 MTL-based Approaches

As demonstrated in previous paragraphs of this chapter, in the MTL-based framework, the prediction of a set of data with a kind of specific style information is named as a task. For each task, there will be a single predictor optimized and trained, as seen in Fig. 2.1.

In these MTL-based models, the style correlation and consistency within each task, as well as common information between different tasks, are extracted and utilized. Improved performance has been achieved by sharing common information among relevant tasks and learning them jointly in comparison with learning each task independently [28, 31, 30, 44, 45, 32]. Nevertheless, estimating the relatedness between different tasks is the critical factor for the MTL-based models. A simple practice is to assume that all the tasks are related, as demonstrated in the proposed models investigated in [28, 44]. Inappropriate estimation or assumption will directly result in negative transfer disaster, where information is shared between unrelated tasks [32, 46]. This might severely degrade the prediction performance.

The problem of the negative transfer mentioned above can be alleviated to some degree by separating tasks into clusters or groups. In [45], the authors claimed that multiple tasks can be gathered by disjoint clusters according to their mutual relatedness. But in fact, the disjoint relationship assumption cannot sufficiently describe complex connections between tasks in real scenarios. Authors in [31] proposed a Bayesian-based approach, where model parameters are divided into two parts including those shared across all tasks and ones less tightly connected with a neural network through a joint prior mixture of the Gaussian distribution. The disjoint separation assumption is overturned in this work to formulate the partially overlapped scheme, however, the time-consuming clustering operation is required for each mixture Gaussian learning.

The overlapped task cluster situation is then considered in [32], in which task parameters within a cluster lie in a low-dimensional subspace. The overlapping information is represented by the number of latent subspace basis in common. But it only assumes a linear combination between task parameters and latent basis. In addition, regression value and input pattern are assumed linearly. Also, a similar idea can be found in [30] where an SVM-based MTL framework was proposed. The task relatedness is represented by the similarity between weight vector learned for each one ( $w_i$  for the  $i$ -th task). Although it can be also extended to represent the nonlinear relationship via the kernel trick, it is still facing the problem how to determine the relatedness between tasks correctly and appropriately.

In addition, all the MTL-based approaches can only be effective when the testing style occurs during training. It does not provide an end-to-end solution (e.g., a self-training based framework) for new style generalization to the unknown in the training process since only the predictors for known tasks are obtained from those MTL frameworks. Even worse, multiple predictors are usually optimized simultaneously for a successful MTL-based model, which consumes much storage volumes and space.

The FPM-based approaches are going to solve the issue that have been demonstrated above. Additionally, as will be fully described in Section 4.2.4, the MTL-based models can be regarded as special cases of the FPM-based frameworks when some of the pre-conditions are set.



## 2.3 FPM-based Models

In the FPM-based models, data with specific information is named as a field.<sup>3</sup> There are Field Modeling algorithms, as well as the Field Averaging Models. Some instructive and typical examples will be reviewed in this section.

### 2.3.1 Field Modeling Generative Approaches

#### The Gaussian Mixture Model

A mixture of Gaussian distribution based model named the Style Mixture Model (SMM) was proposed in [19]. It assumes a fixed number of styles across all the pattern involved. In each style, the examples are generated following the *i.i.d.* assumption. However, the distribution is different between different styles. The distribution of a specific style is represented as a mixture of  $K$  basis, where the number of  $K$  needs to be fixed beforehand. The style consistent class-conditional field-feature probability can be deduced as follows by the mixture of the Gaussian distribution [19]:

$$p(\mathbf{x}|c) = \sum_{k=1}^K \alpha_k \prod_{i=1}^L p(\mathbf{x}_i|k, c_i) \quad (2.1)$$

where  $\alpha_k$  is the probability of the occurrence of the  $k$ -th style (totally  $K$  styles), namely, the mixing proportion of Gaussian mixture model. The SMM is formulated for any style  $k$  and pattern-class  $c$  as follows with a mixture of distributions [19]:

$$p(\mathbf{x}|k, c) = \sum_{j=1}^J \pi_j(c, k) p(\mathbf{x}; \theta_j(c, k)) \quad (2.2)$$

$$0 \leq \pi_j(c, k) \leq 1, \quad \sum_{j=1}^J \pi_j(c, k) = 1, \quad \forall c, k$$

In the above equation,  $J$  is the number of mixture components in the distribution for each class and each style.  $p(\mathbf{x}; \theta_j(c, k))$  is the pattern-feature probability density conditioned on class  $c$ , style  $k$ , and component  $j$  with the parameter  $\theta_j(c, k)$ . The mixing weights are  $\pi_j(c, k)$ . The classification of the SMM model is based on the maximum-likelihood algorithm by selecting the field-class that maximizes the following objective function [19]:

$$\Psi(\mathbf{x}) = \arg \max_{c_1, c_2, \dots, c_L} \prod_{l=1}^L p(\mathbf{x}_l|c_l) \quad (2.3)$$

Nevertheless, there are still major drawbacks both theoretically and empirically in the SMM framework. On one hand, the number of mixture Gaussian components, namely, the parameter  $K$ , needs to be specified before the training of the model. On the other hand, it performs poorly on newly coming field-patterns or different style from the  $K$  styles [29]. More seriously, same as the Field Bayesian Model that will be reviewed in Section 2.3.3, the assumption of the Gaussian distribution might be too strong for empirical application. Scenarios violating such assumption could result in degraded classification performance.

<sup>3</sup>It can be referred to the task in the MTL-based models, as demonstrated in Section 2.2

## The Bilinear Model

In [20], a Bilinear Model (BM) was investigated in order to separate the content and the style from the original input style-inconsistent patterns. In that investigation, it obtains independent style and content representations through matrix decomposition operations by assuming the content and the style as a complicated combination with each one following an independent distribution. The two-factor disentanglement problem is in this way solved, as reviewed in [47].

However, there were several limitations in the BM architecture. First, it can only perform the field prediction with a group of field data. When style examples come one at a time, it would fail to assign any class labels. Additionally, the Singular Value Decomposition (SVD) of an  $N \cdot d \times M$  matrix ( $N$  represents the number of data,  $M$  denotes the number of classes,  $d$  is the dimensionality of the patterns) has to be performed, bringing computational inefficiency when  $N$  or  $M$  are relatively larger numbers. The disentanglement based on matrix decomposition limits the possibilities in the complicated situations with highly nonlinear combinations either, as analyzed in [48].

### 2.3.2 Image Processing Based Style Elimination

A basic way to perform the style averaging is to simply eliminate them, recovering the original patterns without any style information. One possible approach is to make use of the image processing based methods to eliminate the specific style information.

Style elimination on the original data is not often seen in the pure image processing literature. In the previous research records, the image processing based histogram equalization techniques was studied. In [49], it was applied in the task of hand vein image enhancement. By combining with the high-frequency emphasis, it has been verified to be effective in promoting the image contrast. In [50], the technique was engaged in order to generate better views of the curvature to extract the morphology characteristic of teeth surface. In [51], a system to detect possible and suspicious tissue region in an endoscopic procedure was invented. It is achieved by equalizing the histograms of each colour in a video sequence to make the polyp region more prominent. Such algorithm is effective both in subsequent processing and for alternative video display on a colonoscopy device being watched by the operator of the colonoscopy procedures or the relevant physician.

A novel approach is investigated in [12] and reported in Chapter 3. In the proposal, style information can be seen as the diverse kinds of sunglasses with different shapes and luminousness worn on the human frontal facial images. The histogram equalization technique is utilized to recover the corrupted region of frontal facial images, producing the style-eliminated *i.i.d.* examples. In other words, the style-inconsistency will be eliminated by the proposed sunglasses recovery algorithm. Moreover, the proposed algorithm also enables the one-shot robust facial expression recognition across diverse sunglasses. It means that a classifier can be trained with a dataset, where only one single image is available for each facial expression of each individual no matter what kind of sunglasses he/she is wearing.

Although consistent performance promotion on the final recognition rate can be found with several state-of-the-art models, it is actually a case-by-case effort. A completely new algorithm is needed for the elimination of other kinds of inconsistent-style information.

Such undesirable factor severely affects the potential utilization of the image processing based style averaging frameworks to generate the style-consistent *i.i.d.* patterns.

Fortunately, there are multiple ways to perform the style averaging transformation with the machine learning based models. For both the discriminative and the generative approaches by utilizing the style information to perform the non-*i.i.d.* prediction, the priority and flexibility have been undoubtedly shown compared to those pure image processing based algorithms. They will be reviewed in the following sections.

### 2.3.3 FAM-based Approaches

Same as the image processing bases style elimination, the basic idea of the FAM-based models is to find a transformation capable of averaging the inconsistent style information embedded in the original style-inconsistent patterns. Differently, it is based on the machine learning and the current deep learning technologies, rather than a case-by-case image processing model as demonstrated in Section 2.3.2.

Such transformation will produce corresponding examples with identical style information. They satisfy the *i.i.d.* assumption that most of the conventional machine learning models require. Then these prediction models can be readily implemented without the concerns of violation of the *i.i.d.* conditions. The basic flowchart of the FPM-based algorithms are summarized in Fig. 4.3 by taking the Field Support Vector Classification (to be detailed in Chapter 4) as an instructive example.

According to the means and models to perform the style averaging, three main categories are briefly reviewed as follows. If the style averaging transformation is fulfilled by the discriminative machine learning based models, then they are named as the **Style Normalization**, which will be described in Chapter 4 in detail. If it is completed by generative machine learning frameworks, they are in this way titled as the **Style Neutralization**. It will be included in Section 5.1. The relevant literature review will be made in Section 2.3.2, Section 2.3.3, and Section 2.4.2 respectively for Style Elimination, Style Normalization, and Style Neutralization.

#### Discriminative Approaches for Style Normalization

As a discriminative machine learning model, it calculates directly the hyperspace to separate patterns of different classes. Such optimization is fulfilled by constructing the dependence of targeted variables  $y$  on the input patterns  $x$  by modeling the conditional probability distribution  $p(y|x)$ . It can be then used for future prediction from  $x$  to  $y$  [52].

One of the state-of-the-art discriminative models with the style normalization idea for the field prediction is the Field Bayesian Model (F-BM) proposed in [29], which was then used in some down-stream application including the writer adaptation with the style transfer learning [53], as well as learning the specific style transfer matrix [54].

In the model definition of the F-BM framework proposed in [29], a Style Normalization Transformation (SNT), represented as  $\{A_i, b_i\}$ , will be optimized for data of each field. It is then applied to the original style-inconsistent pattern, producing the corresponding style-normalized ones. The transformation formulation is given as  $\tilde{\mathbf{x}}_j^i = A_i^T \mathbf{x}_j^i + b_i$ , where  $\tilde{\mathbf{x}}_j^i$  represents the style-normalized pattern of the given  $\mathbf{x}_j^i$ ,  $i$  represents the field index, and  $j$  specifies the index within the corresponding field. During the training, the

regularization term  $R_i(A_i, b_i) = \beta \|A_i^T - I\|_F^2 + \gamma \|b_i\|_2^2$  will be engaged to restrict and constrain the flexibility of the SNT variation. The overall objective formulation for minimization is given as follows [29]:

$$\mathbb{L} = - \sum_{i,j}^{N,L_i} \log p(A_i^T \mathbf{x}_j^i + b_i | y_j^i) + \sum_i^N R_i(A_i, b_i) \quad (2.4)$$

where  $\{A_i^j, b_i\}$  is designed to normalize the inconsistent style information in  $x_j^i$ , producing the corresponding style-normalized pattern.

The training of the F-BM model follows the alternative optimization strategy, same as the learning procedures of the F-SVM model that will be discussed in Chapter 4. Particularly, the SNT is learned by the following objective for a specific field (e.g., the  $i$ -th field) [29]:

$$\min_{A_i, b_i} - \sum_{j=1}^n \log p(A_i^T \mathbf{x}_j^i + b_i | y_j^i) + R(A_i, b_i) \quad (2.5)$$

Additionally, a transductive self-learning framework is also studied to perform the field prediction on the patterns with the unknown style during training by the following [29]:

$$\min_{A, b} - \sum_{j=1}^n \log p(A^T \mathbf{x}_j + b | y_j) + R(A, b) \quad (2.6)$$

When the probability distribution follows the multivariate Gaussian class-conditional probability distribution, the SNT learning formulated as Eq. (2.5) will be degraded into [29]:

$$\min_{A_i, b_i} R(A_i, b_i) + \frac{1}{2} \sum_j^n d_m(A_i^T \mathbf{x}_j^i + b_i, \mu_{y_j^i}, \Sigma_{y_j^i}) \quad (2.7)$$

In the above equation, the Mahalanobis distance [55] is calculated as  $d_m(\mathbf{x}, \mu, \Sigma) = (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)$ . Obviously, the formulation deduced in Eq. (2.7) is a convex quadratic programming (QP) optimization with a closed-form solution. There are two special cases of the  $\Sigma$  can be obtained, which lead to two different sub models as follows.

- **The Field-NCM (F-NCM) model:**  $\Sigma = I$  results in the NCM classifier in the conventional Bayesian framework. The minimization defined in Eq. (2.7) can be deduced as [29]:

$$\min_{A_i, b_i} R(A_i, b_i) + \frac{1}{2} \sum_j^n \left\| A_i^T \mathbf{x}_j^i + b_i - \mu_{y_j^i} \right\|_2^2 \quad (2.8)$$

The formulation given above is named as Field-NCM (F-NCM).

- **The Field Quadratic Discriminant Function (F-QDF):** When  $\Sigma = \Phi \Lambda \Phi^T$ , where  $\Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_d]$  with  $\lambda_t, t = 1, 2, \dots, d$  being the eigenvalues of  $\Sigma$  with the descending order, and  $\Phi = [\phi_1, \phi_2, \dots, \phi_d]$  with  $\phi_t, t = 1, 2, \dots, d$  being the

corresponding eigenvectors. As specifically demonstrated in [29], it leads to the approximation given as follows [29]:

$$\min_{A_i, b_i} R(A_i, b_i) + \frac{1}{2} \sum_j^n \left\| A_i^T \mathbf{x}_j^i + b_i - P(\mathbf{x}_j^i, \Sigma_{y_j^i}, \mu_{y_j^i}) \right\|_2^2 \quad (2.9)$$

In the above formulation,  $P(x, \Sigma, \mu) = \sum_{t=1}^T \alpha_t \phi_t + \mu$ ,  $\alpha_t = \min\{\delta\sqrt{\lambda_t}, \max\{\phi_t^T(\mathbf{x} - \mu), -\delta\sqrt{\lambda_t}\}\}$ , and  $\lambda \geq 0$ . Under this situation, the F-BM model is formulated as the Field Quadratic Discriminant Function (F-QDF).

However, there are major problems in the F-BM model as well as the two special cases demonstrated above. First, it only enables the linear representation of the style normalization. It might not be enough to represent the sufficiently complicated style information in real scenarios, as demonstrated in [13, 14, 15, 16, 17]. Furthermore, the Gaussian distribution assumption needs to be specified for the two special cases including the F-NCM and the F-QDF models. Such assumption may restrict the potential applications on the field prediction scenarios for non-Gaussian distributed examples.

## 2.4 Generative Approaches with Style Information

The generative model, seen as a statistical framework, calculates the joint probability distribution  $p(x, y)$  when an input pattern  $x$  and a target answer  $y$  are specified [56]. The predictors based on this framework are designed to learn such joint probability, while the prediction is performed based on the Bayes rule  $p(y|x) = p(x, y)/p(x)$  to calculate the conditional probability distribution  $p(y|x)$ . Then, same as those discriminative models, the label will be assigned based on the obtained probabilities.

Furthermore, such joint distribution can be used for new data generation task by sampling from the conditional probability distribution of the joint distribution. The major difference between the generative models demonstrated in this section and those discriminative ones described in Section 2.3.3 is that there will be a joint distribution to be approximated here, enabling to perform sampling based on that to accomplish the data generation task. However, in discriminative-based approaches, only the simple hyper-space for classification will be calculated and specified.

### 2.4.1 Generative Adversarial Network

The Generative Adversarial Network (GAN) is an emerging deep learning architecture introduced in [2] by optimizing the generative model with the adversarial training strategy. It trains a generative model  $G$  to synthesize a data distribution by feeding a random noise vector. A discriminative model  $D$  is jointly optimized by distinguishing between generated examples from  $G$  or real ones from the targeted distribution. The learning is terminated when  $D$  cannot be further improved, indicating that samples generated are indistinguishable from real ones.

Many extensions have been made for further development recently. The Conditional-GAN in [57] enables manipulating the generated sample with the clear class label input

by adding the it on the generator. The Info-Gan proposed in [58] decouples the input noise into the pure noise and the meaningful latent codes by maximizing the mutual information between the latent codes and the generative distribution. It enables the manipulation of some properties of the generated samples by adjusting the values of those latent codes. The Auxiliary Classifier GAN (AC-GAN) investigated in [59] is introduced by attaching an auxiliary classifier on the discriminator. The training is performed by not only the adversarial loss but also minimizing classification errors for both generated and real samples. The learning of the additional classification task is further supervising the learning of the generative model. [60] firstly applied the GAN framework to the Convolutional Neural Network (CNN). Named as DC-GAN, it develops a standard model architecture for GAN-based utilization in the computer vision domain. Several training tricks in [61] are recommended for stabilizing training and avoiding model collapse. They include the feature matching, the minibatch discrimination, the histogram averaging, the one-sided label smoothing, as well as virtual batch normalization.

The Wasserstein GAN (W-GAN) introduced in [11] solves the unstable training issue by replacing the Jensen-Shannon (JS) divergence [33] with the Wasserstein distance[25]. The restriction of the discriminator modeling capabilities is fulfilled by the weight clipping technique to meet the Lipschitz continuity condition required by W-GAN. It was further replaced by the gradient penalty proposed in [24].

In more recent literature, a Least-square GAN is proposed in [62]. In the work, the Pearson  $\chi^2$  divergence [63] between the real data distribution and the distribution of the generated samples represented as the Least-square (LS) loss, are exploited. The benefit brought is that the LS loss is more smooth and it saturates slower than the sigmoid cross-entropy loss of the JS formulation [64]. In [65, 66], a novel spectral normalization to divide each weight matrix including those convolutional kernels is engaged in the discriminator. The most appealing factor of the proposed spectral normalization is that it results in the discriminator to be equipped with a higher rank.

The major advantage of the GAN-based models (compared with those Variational Autoencoder [67] (VAE) based deep generative frameworks) is that it offers more flexibility. The reason is that when a VAE model is well trained, the posterior probability distribution of the latent vector is somehow fixed [68]. However, in the GAN-based architectures, no such distribution assumption needs to be specified beforehand. Furthermore, the adversarial training strategy in the GAN-based networks provides additional training supervision to the approximation of the real data distribution, resulting in synthesizing more sharp and clear images. On the contrary, the VAE models tend to produce blurred and confusing examples due to the involvement of the  $L_2$  reconstruction loss [69]. Moreover, the discriminator in the GAN-based models will pose a role to provide the learning on a rich similarity metric to discriminate images from non-images. Such comparison conclusions have been demonstrated and specified in multiple technical reports and research publications, e.g., [70], [71], [72], etc.

## **Image-to-Image (Img2Img) Translation**

The Img2Img Translation [21] is one specific model of the application with the GAN model concentrating on the image transformation between two domains [21]. As one further development of the Neutral Style Transfer framework proposed in [73], the Img2Img

model translates the input image to the corresponding output one. Typical cases can be specified including segmented labels to the street scenes, aerial images to satellite maps, segmented image labels to facades, grayscale images to colored ones, daylight images to night scenes, as well as edges to a real photos, as illustrated in [21].

One typical architecture of the Img2Img generator is the so-called U-Net architecture, which was initially proposed in [10] for biomedical image segmentation. The basic network architecture is illustrated in Fig. 2.2. Particularly, the most significant part of

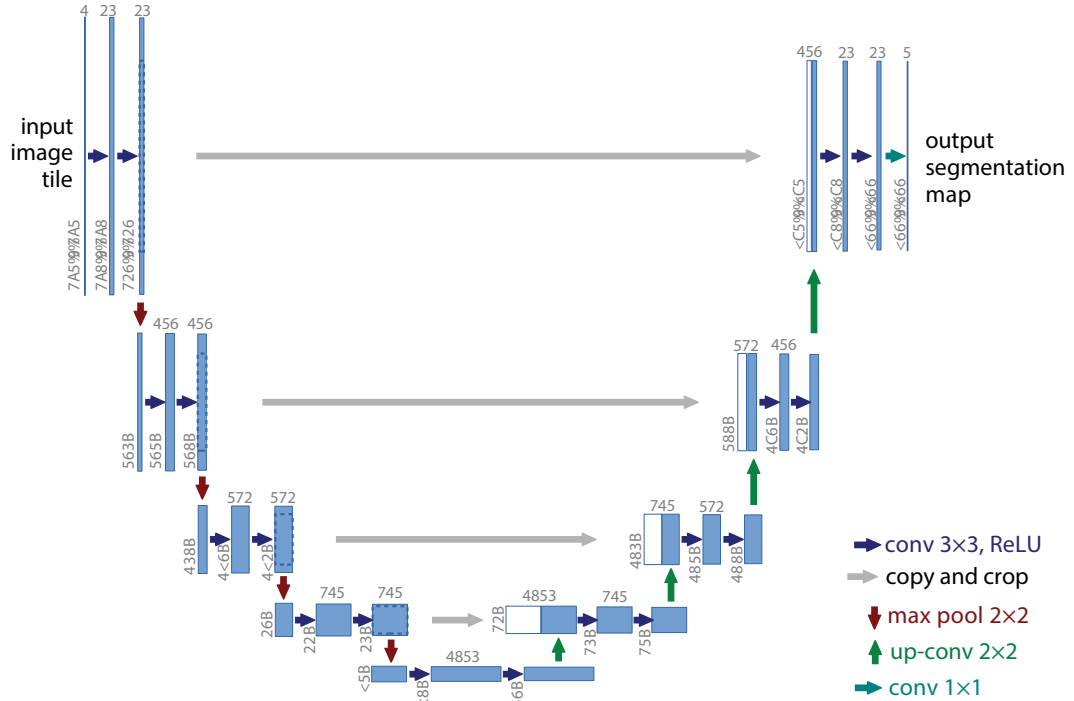


Fig. 2.2: Original U-Net Architecture [59].

the U-Net is the skipping connection between the encoding layer and the corresponding decoding layer. As a modification of the Fully Convolutional Networks (FCN) [74], the depth of an FCN model might be limited due to the vanishing gradient issue when back-propagating the signal across multiple layers [75]. The skipping connections relieve this problem by allowing gradient to flow uninterrupted in a deep architecture. In this sense, the model parameters can be effectively updated in deep models [75].

There are quite a lot of research proposals recently to perform the Img2Img translation for the style transfer task. They are high-definition image generation [76], unsupervised and unpaired binary domain transfer [77], and unpaired style transfer (CycleGAN) [78, 79, 80, 81]. Particularly, in the research named as StarGAN [82], the many-to-many domain translation can be fulfilled readily.

The Img2Img translation framework was also applied in some other empirical applications. For example, in [83], a facial image editing scheme was achieved. By simply specifying the corresponding values of several facial attributes, the generated facial images can be arbitrarily edited, e.g., age, hair color, and styles, and mouth open or not. In [84], the previously demonstrated CycleGAN was further developed and implemented to apply and remove virtual facial makeups. By taking a source facial image and one

single makeup reference, the model is able to produce a specific facial image with the corresponding makeup reference. In [59], an enhanced residual U-Net was investigated in order to accomplish the style transfer task from the sketches of girls to the corresponding anime drawings. A *style-hint* was attached to the mid-level layers of the U-Net to represent the multi-style information.

## 2.4.2 GAN-based Models for Classification Performance Promotion

When considering the recent GAN-based models for classification performance promotion, [85, 86] can be regarded as one specific kind of examples. The basic idea behind these two work is to make use of the GAN-based model to synthesize new patterns by optimizing it on the original training data. The classifier to be trained will then be optimized based on not only the original training data, but also the new-augmented synthesized patterns. It is believed to promote the final classification performance, even enhancing the few-shot learning system such as the Matching Network [85]. However, the major problem of these data augmentation GANs lies on that the training of the data augmentation is irrelevant to the training of the classifier.

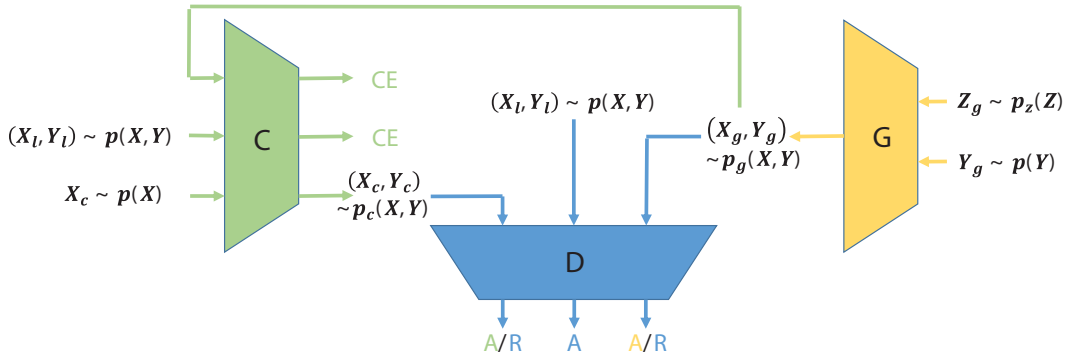


Fig. 2.3: Original TripleGAN Architecture [87].

A TripleGAN framework was proposed in [87], where the end-to-end training of both the GAN and the classifier was achieved, as depicted in Fig. 2.3. One promotion of the TripleGAN compared to the AC-GAN proposed in [59] is that the classification is disentangled from the GAN discriminator by setting an independent classifier. In the meanwhile, the training of the classifier is also helpful to the optimizing of the generative model, which is the same as the AC-GAN model.

Although state-of-the-art classification performance has been achieved, the major problem of the TripleGAN lies on the input of the class label of both the generator and the discriminator. As the Tensorflow [88] implementation of the TripleGAN model posted on [89], the one-hot coded class label was repeated multiple times in both the generator and the discriminator. The number of this repeat is the same as the size of the specific shape of the corresponding feature map to be incorporated and concatenated with. In this way, the class label information can be implemented into the architectures.

However, such repeat operations come to the drawback that too much redundant and duplicated information was copied and created. The situation can be even worse when large-scale classification was performed, e.g., the ImageNet [90] with over 1,000 classes



(an  $1 \times 1,000$  one-hot embedding), or the Chinese character classification on the GB-2312 Level 1 set [91] with 3,755 characters (an  $1 \times 3,755$  one-hot embedding). It makes the TripleGAN not suitable for such large-scale classification scenarios.

### 2.4.3 Few-shot Style Data Generation with Free Manipulation

Some of the already listed research work in this chapter can also be regarded as examples of the style data generation task. E.g., in [82], the users can manipulate the content of the generated samples by specifying the corresponding class label; in [83], the facial attributes of the generated facial images can be controlled by the corresponding property values; in [84], the style of the facial makeup is referred to by the input reference image.

However, [82, 83] take no consideration of the few-shot, or even the one-shot generation scenarios. It cannot be generalized to new-coming styles that are absent from the training set. [84] enables the one-shot style data generation with one single reference image. However, it only enables the facial image transformation where the difference of the overall shape and image appearance between the source and the target pictures remains unchanged. Nevertheless, when the style transfer between Chinese characters in different styles is considered, the variation on shapes and appearance can be much greater. One typical example is given in Fig. 5.6.

In [92], a few-shot font style transfer model, named as the Multi-Content GAN (MC-GAN), was investigated. Divided into two sub conditional GANs including the graph model (to predict the coarse glyph) and the ornamentation model (to generate color and texture of the final graph) that are stacked one and the other, the proposed model enables to transfer the style of a given glyph to the contents of unseen ones. It is capable of capturing highly stylized fonts found in the real-world such as those on movie posters or info-graphics. The basic network structure of the MC-GAN is illustrated in Fig. 2.4.<sup>4</sup>

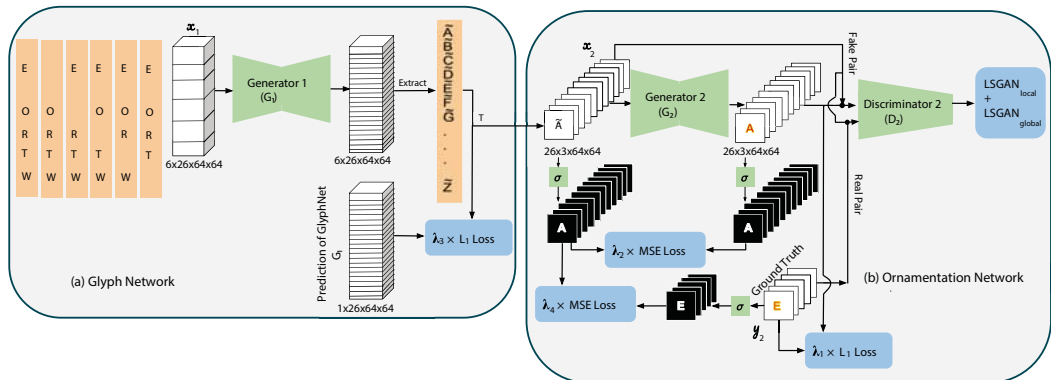


Fig. 2.4: Original MC-GAN Architecture [92], taking the English alphabet as an example with 26 letters.

Performance of the MC-GAN model seems to be breathtaking and fantastic since it enables the generation of various and diverse artistic fonts and styles with only a few

<sup>4</sup>The similar idea to divide the original style transfer task into two independent procedures can also be found in other GAN-based models, e.g., the CariGAN for unpaired photo-to-caricature translation [93].

available samples. However, the major issue is that the content to be generated needs to be fixed before the whole model is trained by specifying the relevant input of both the Generator 1 and the Generator 2, as seen in Fig. 2.4 with the feature size of  $6 \times 26 \times 64 \times 64$  (where 6 is the batch size, 64 is the size of the input letter image). Such well-trained MC-GAN model cannot be generalized to new coming contents. The issue can be even serious when considering the huge number of the Chinese character (3,755 characters in the GB-2312 Level 1 set [91]). The specific input size can be  $6 \times 3,755 \times 64 \times 64$ , which is too large to be implemented in the current deep learning devices.

The style data generation task can be regarded as a transferring framework of the given style information to the specified content data. It has been argued that the famous batch normalization [94] is inappropriate [95, 96] to be integrated into the style transfer framework with the GAN setting. A novel normalization technique is introduced in [95] to solve the problem. Named as the Adaptive Instance Normalization (AdaIN), it performs the normalization on the content feature ( $x$ ) by the statistics of the style features ( $y$ ). It is given as follows [95]:

$$\text{AdaIN}(x, y) = \sigma(y) \cdot \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \quad (2.10)$$

where the statistics, namely,  $\sigma(y)$ ,  $\mu(y)$ , for normalization are computed across spatial locations. They are given as Eq. (2.11) and Eq. (2.12) respectively [95] with an input batch  $x \in \mathbb{R}^{N \times C \times H \times W}$ , where  $N$ ,  $C$ ,  $H$ , and  $W$  denote the batch size, the number of channels, the spatial height, and the spatial width of a given feature map  $x$ .<sup>5</sup>

$$\mu_{nc}(x) = \frac{1}{HW} \sum_{h,w}^{H,W} x_{nchw} \quad (2.11)$$

$$\sigma_{nc}(x) = \sqrt{\frac{1}{HW} \sum_{h,w}^{H,W} (x_{nchw} - \mu_{nc}(x))^2 + \epsilon} \quad (2.12)$$

The AdaIN has been successfully utilized in a variety of models for data synthesis and style transfer. E.g., they are in [95] for arbitrary style transfer in real time, in [99] for universal style transfer, in [100] for multimodal unsupervised Img2Img translation, and in [96] for coarse to fine adjustment of the generated images on various scales of features.

## Few-shot Arbitrary-style Chinese Character Generation

For the generation of Chinese characters, it is time-consuming to create a full set of its characters embedded with a specific style, e.g., personalized hand-writing calligraphy or a stylistic printing font. As a special kind of ancient written language with both messaging functions and artistic values, the machine learning based automatic Chinese character

<sup>5</sup>The computation of the normalization statistics follows that for the Instance Normalization proposed in [97] and further investigated for style transfer in [98]. Meanwhile, the multi-summation operation is represented by only one summing sign for the sake of simplifying the expression, namely,  $\sum_{i,j}^{N,L_i} a_j^i = \sum_{i=1}^N \sum_{j=1}^{L_i} a_j^i$ .

generation is a less popularly studied topic since each character is a combination of various strokes and radicals with diverse kinds of interactive structures. It is even harder since the huge number of different characters equipped with various styles of handwriting styles and printed fonts.

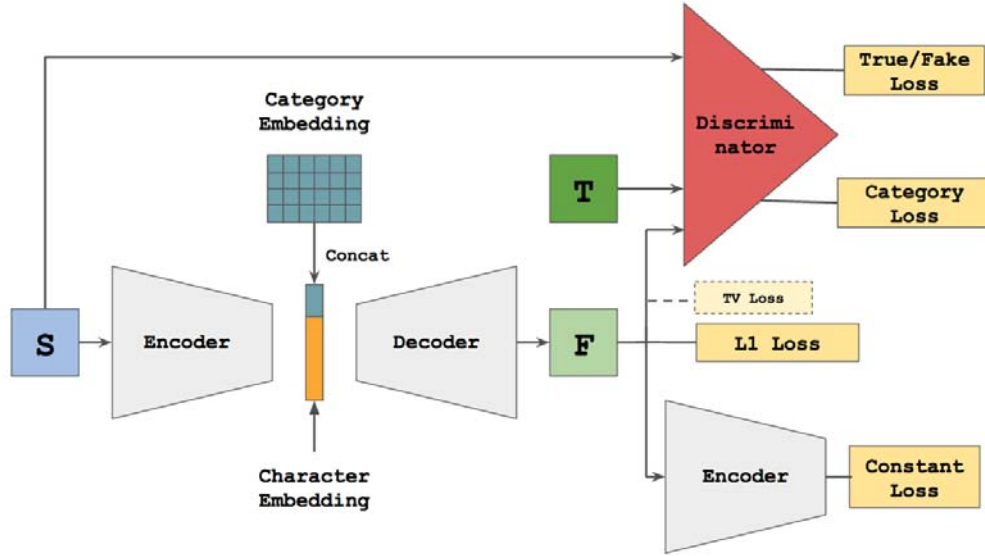


Fig. 2.5: Original Zi2Zi Architecture [101].

Several proposals in the literature are introduced to accomplish the generation task effectively, which can be divided into two major categories including the **stroke-based** (where character strokes features are computed before training) and the **image-based** approaches (where the character is seen as an image).

Recently, some non-GAN based generative models are proposed. In [102], a stroke-based architecture is investigated. In that model, the strokes are represented by time-series writing evenly-thick trajectories. Then they are sent to the Recurrent Neural Network (RNN) based generator. Regarded as the time-sequence data, the RNN-based generator may be not robust to the reversed or disordered stroke writing habits that are quite common and frequently seen in Chinese individuals. In [103] an image-based framework was introduced. The font feature reconstruction for standardized character extraction is achieved based on an additional network to assist the one-to-one *Img2Img* translation framework. However, over 700 pre-selected training images are needed as well, which is unacceptable for a few-shot Chinese character generation scenario.

In the GAN-based **Zi2Zi** [101] model, the one-to-many mapping is achieved by the fixed Gaussian-noise based categorical embedding [104]. The model architecture of the Zi2Zi framework is illustrated in Fig. 2.5. Nevertheless, a well optimized Chinese character Zi2Zi generator requires over 2,000 training examples per style. Additionally, the introduced categorical embedding is a non-trainable matrix [104], which is improper to represent the inner relationship between different but somehow related styles. Furthermore, the trained model still cannot be generalized to a new style with few available samples since the embedding has already been fixed before training.

Some modification of the Zi2Zi model to enable the generation of new style characters

have been attempted. Details will be demonstrated in Section 5.2.3 as the baselines of the relevant comparisons. However, the generation performance is still unsatisfied for a few-shot arbitrary-style Chinese character generation task.

In [48], a VAE-based generator is introduced. However, some domain knowledge of the structures, strokes, and radicals of Chinese characters are to be utilized by engaging the content hashing code from a pre-defined look-up table. The hashing code is constructed by 12 bits of common configurations in Chinese characters (*up-down*, *left-right*, *surroundings*, etc.), 101 bits of 100 frequently used radicals and the case of missing radicals, as well as the last 20 bits for additional information [48]. Nonetheless, such a look-up table is not a trainable factor. Heavy additional domain knowledge is also required, while the inappropriate setting would directly damage the synthesization performance. In this way, the specific model can only be used in the generation task of the commonly used Chinese characters in contemporary China.

### W-shaped Architecture

Recently, some research reports are noticed to fulfill the few-shot/one-shot arbitrary-style Chinese character generation task by incorporating the so-called W-shaped deep architecture. One typical example can be seen in [18]. It is fulfilled by separating the content and the style information, as depicted in Fig. 2.6.

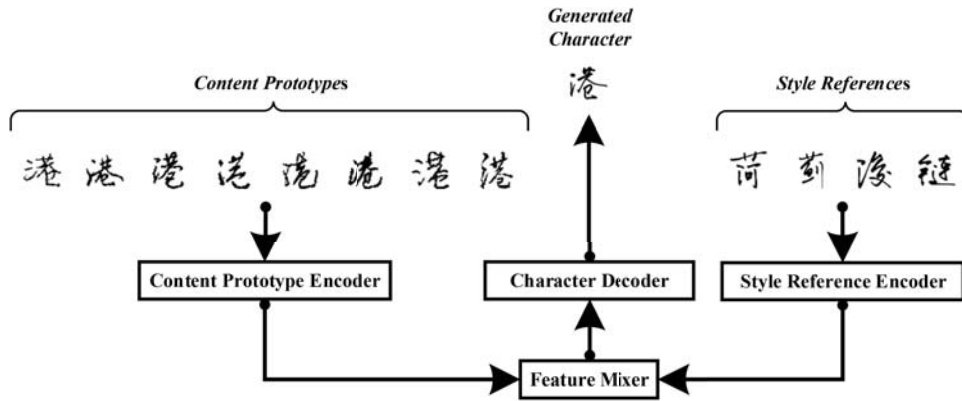


Fig. 2.6: W-shaped Architecture.

Basically, the idea of such separation is originally introduced in the BM model investigated in [20]. Relevant investigations include [18, 47], while such separation of style and content is further summarized in [105]. In them, two parallel encoders including the style reference encoder and the content reference encoder [18][105] are implemented. The input of the style reference encoder is a set of stylized characters, which are commonly different in content. They are named as the *style references*. On the contrary, a set of characters identical in content but distinguishable in styles, named as the *content prototypes*, are sent to the content prototype encoder. A feature mixer will be employed to combine the extracted features from both encoders, after which they are sent to the character decoder to produce the desired output. All of four composing sub-architectures described above form a W-shaped deep generative model

In [47], a mixer network is engaged to combine the extracted style information from the style encoder ( $S_i$ ) and the content information computed by the content encoder ( $C_j$ ).

It is given as  $F_{i,j} = S_i \cdot \mathbf{W} \cdot C_j$ . The mixture of both information is simply represented by a three-dimensional tensor  $\mathbf{W}$ . The equivalent architecture in the W-Net are feature short-cuts and multiple residual blocks, as will be seen in Fig. 5.10. In this sense, the representation power might not be enough to describe the sufficiently complicated relationship between the content and the style information in real scenarios. In addition, the number of style characters shall be given before the training of the model as part of the model parameters, which also restricts the possible applications in real scenarios.

However, the proposed W-Net is not limited to such condition. It is even possible to generate other block characters in eastern Asian languages, e.g., Korean characters and Japanese characters. Such appealing property can be seen in examples given in Section 5.2.6.



## Chapter 3

# Style Elimination

The *i.i.d.* assumption is a basic pre-condition for most of current machine learning models [29]. However, such a scenario would be violated when data are generated not by the same synthesizing source. Examples from each source would be equipped with consistent style information. Simultaneously, patterns from different synthesizing pipelines can be separated by different style information. These patterns with diverse styles are called **style-discriminative** patterns, as noted in Section 1.1.

A direct way to conquer the issue is to simply eliminate it. By removing the various and different style information existed in the original input data, patterns only equipped with the identical style information or with no style tendency will then be produced. Such a transformation is named as the style elimination transformation (SET) [29]. The classification is then performed on these **style-eliminated** transformed examples.

A simple example will be demonstrated in this chapter (published as a journal paper in [12]) in the automatic facial expression recognition (FER) task with different sunglasses. In this model, the style information occurs as the specific shape and colour of them. The SET is then fulfilled with an image processing based algorithm. Several example images of facial expressions with diverse sunglasses are given in Fig. 3.1, where images from each column represent a specific facial expression including angry, disappointment, fear, happy, nervous, sad and surprise respectively. Portraits in each row illustrate a particular kind of sunglasses. The specific sunglasses in examples given in Fig. 3.1 are Cir-65 (circular shape with luminousness at 65%), Rec-65 (rectangular shape with luminousness at 65%), Cir-35 and Rec-35 respectively.

Apart from the SET, the framework proposed in this chapter considers a practical and robust scenario different from previous FER systems. Conventional recognition system often focus on learning a classifier in a controlled environment. More specifically, traditional FER requires collecting as many as possible facial photos so that they will accurately recognize expressions no matter a particular person wears sunglasses, hats, and other accessories or not. Such requirement is however inconvenient and could impose practical difficulties for users since the style of the pictures can hardly be controlled.

To alleviate this problem, a robust one-shot FER system requiring only one single facial photo for each expression of each user is proposed in this chapter. When taking the individual picture, the user is free to choose whether to wear sunglasses or not. The sunglasses can even be in different shape and on various luminous transmittance. Such one-shot recognition improves the user-friendliness of the FER system. Importantly, a novel



Fig. 3.1: Portrait examples in the modified Japanese Female Facial Expression (JAFFE) [38, 12] database with multiple sunglasses.

and practical sunglasses detection and recovery approach is developed, which produces corresponding style-eliminated portrait images to the input style-discriminative examples with various sunglasses. The proposed model obtains a noticeable accuracy improvement of 6.09%, 5.86% and 4.33% with state-of-the-art classifiers including Support Vector Machine (SVM) [6], Linear Discriminate Analysis (LDA) [106] and  $k$ -Nearest Neighbors (KNN) [107] respectively on the modified JAFFE [38, 12] benchmark database.

### 3.1 Research Background: Facial Expression Recognition

Among various ways of human interpersonal communication, facial expression is mostly unique since it directly represents human’s emotion, idea and thought. It is both interesting and important to explore the automatic FER algorithms and applications. There have been many proposals in this field in the literature.

For example, [108] proposed an algorithm where each facial image is decomposed into an identity part and an expression part represented by their corresponding nonnegative bases. By devising graph-embedding constraints on the expression subspace, the intraclass variation issue in the expression recognition procedures can be mostly tackled and resolved. Another study is investigated in [109], which employed the stepwise linear discriminate analysis for feature extraction. The hidden conditional random field model is applied for the following recognition task. Besides, [39] introduced a geometric alignment technique to preprocess the original expression images. The Local Binary Pattern features are extracted before classifying with classifiers including SVM [6] in that algorithm.

Despite the reported good performance, previous FER approaches have often focused on learning a classifier in a controlled environment. For example, if the classifier is op-



timized with images without sunglasses, hats, or any other personal accessories, the test image is then strictly restricted. Obviously, these limitations significantly affect user-friendliness of the system and impose practical difficulties upon real application since the trained classifier cannot be adapted well to the new-coming data with unseen styles.

To conquer this problem, a robust one-shot FER system is proposed in this chapter. It requires taking one single facial photo for each expression of each user. The proposed algorithm focuses on making the system robust enough to tackle various sunglasses styles. Namely, when taking a single photo, the user is free to wear or not wear sunglasses. The sunglasses can even be in different shape and on various luminous transmittance. In contrast with previous studies, the training facial expression of one specific person may include sunglasses, while the testing one may not, and vice versa. Such a one-shot recognition system naturally promotes the user-friendliness of the FER system and will be expected to be used in various real scenarios.

To achieve this target, a novel, intelligent sunglasses recovery approach will be proposed. It produces style-eliminated patterns corresponding to the style-discriminative input data with various style information, e.g., diverse kinds of sunglasses being worn. Initially, a roll de-rotation operation based on relative positions between two eyes was applied. Then a Canny Edge Detector was implemented to locate possible sunglasses area on the Region of Interest (ROI). The histogram matching algorithm will be utilized. It transforms darker grayscale value (representing detected sunglasses region) into brighter one (representing the facial area without sunglasses). The corrupted pixels due to sunglasses are excellently recovered in this way. The style of original input patterns brought by inconsistent sunglasses is at this moment eliminated, satisfying the *i.i.d.* assumption. They are then sent to the conventional classifier for further facial expression recognition task. Besides, the proposed algorithm was tested on the modified dataset from JAFFE [38, 12], a benchmark facial expression dataset, with the proposed robust scheme. Improvement with the recognition rate of 6.09%, 5.86% and 4.33% for classifiers of SVM [6], LDA [106] and KNN [107] respectively with the proposed one-shot recognition system is achieved.

## 3.2 Robust Sunglasses Detection and Region Recovery

In this part, the proposed novel robust sunglasses detection and region recovery algorithm will be introduced. When a new image is input, the algorithm will detect if sunglasses exist in the facial region. If presents, the recovery algorithm is utilized to remove the sunglasses and recover the facial region blocked by the sunglasses, producing the corresponding style-eliminated portrait. Detail steps are described in the following sections.

### 3.2.1 Correction for Roll Rotation

As suggested in [39], a roll orientation difference (of which the axis is perpendicular to the paper) will influence the recognition performance. Hence, a derotation operation, mainly based on the difference of centres of the two eyes, is a necessary procedure. Fig. 3.2 illustrates the comparison for this operation.

To derotate the image, two landmarks representing left and right eye centres generated

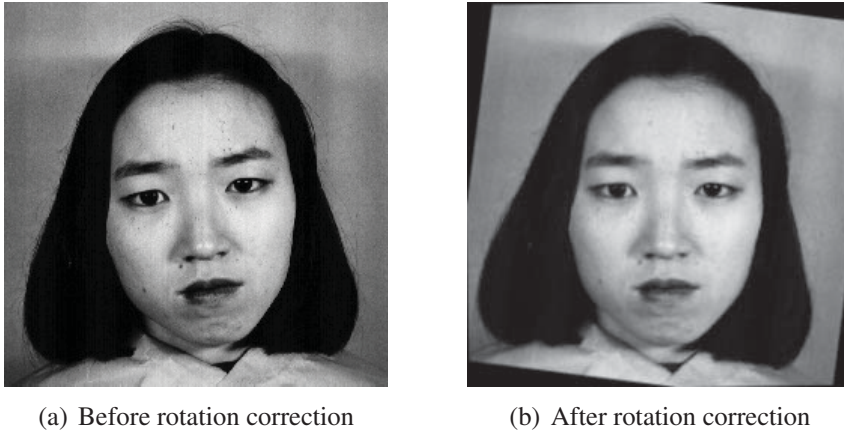


Fig. 3.2: Roll rotation correction of facial images.

from STASM [110, 111] (an Active Shape Model based facial components landmarks detector, which will be demonstrated in Section 3.3) are applied for rotation reference. It is formulated in Eq. (3.1), where  $x_r$  and  $x_l$  represent column indices for the left and right eye center points respectively, with  $y_r$  and  $y_l$  the row ones. The image is then rotated by the angle of  $-\theta$  concerning the image center.

$$\theta = \arctan\left(\frac{y_r - y_l}{x_r - x_l}\right) \quad (3.1)$$

### 3.2.2 Sunglasses Region Detection

To perform a recovery for the sunglasses region, it is essential to precisely localize the sunglasses region before the recovery algorithm is implemented. The Canny Edge Detector (CED) is then utilized [112] to meet the required function.

Table 3.1: Glass region detection.

Original Image	Original ROI	Thresholded ROI	Coarse Edge	Final Contour

As seen in Table 3.1, the ROI is generated based on STASM [110, 111] landmark positions of the corresponding eye and eyebrow. Since the CED detects sharp edges

easily, it leads to unsatisfactory performance on the eye part (as well as the eyebrow part) due to their similar colour compared with sunglasses. It is shown in the Original ROI column of Table 3.1. To tackle this defect, pixel values from both eye and eyebrow parts are transformed to the ones from their neighbouring pixels. Before this, a simple average filter will be implemented to weaken the surrounding edges further. By applying a grayscale value thresholding operation to the target region, the further enhancement can be in this way achieved.

### 3.2.3 Grayscale Value Histogram Shifting

Wearing sunglasses is regarded as an overall dropping operation on grayscale values for all pixels locating in sunglasses regions in the portrait images. It means that the grayscale value dropping rate (GVDR)<sup>1</sup> is a number that is less than one (maximum grayscale value). Additionally, the distribution of the original values is reduced gradually. From Fig. 3.3, it is found that the histogram is shifted towards the lower grayscale section as GVDR increased while the histogram distribution becomes centralized.

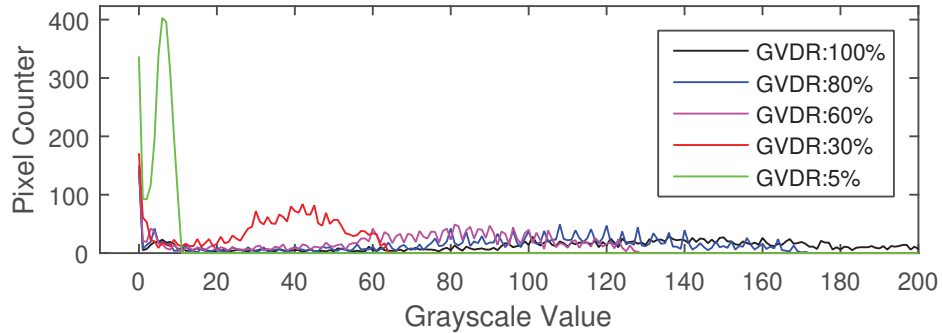


Fig. 3.3: Histogram comparison among sunglasses with different grayscale value dropping rate (GVDR).

### 3.2.4 Histogram Matching for Sunglasses Recovery

Observed from Fig. 3.3, there comes a mathematical discovery. By transforming histograms with multiple GVDR values to the one at 100% GVDR, the image region corrupted by sunglasses can be recovered. The proposed transformation, majorly based on histogram matching [113], is designed to match the histogram distribution from the source image (the image to be matched with the presence of the sunglasses with corrupted pixels) to the one from the target image (the matching target with non-corrupted pixels). In this research, the matching target is correspondence with GVDR at 100% or the facial image without wearing glasses. It is because that no grayscale change has ever been applied apart from the glass frame if GDVR is set to 100%. The target histogram model is firstly calculated from the original face images. In the meanwhile, the matching procedure will be divided into histogram equalization and matching.

<sup>1</sup>The GVDR denotes the rate of the corrupted pixel value over the original one brought by the corruption of sunglasses being worn.

**Histogram Equalization:** The histogram equalization is firstly applied on both source image (image to be matched, sunglasses corrupted pixels) and the target image (matching target, non-corrupted pixels). In Eq. (3.2),  $s_k$  and  $v_k$  represent two equalized images of source image and target image respectively.  $\frac{n_j}{n}$  and  $\frac{m_i}{m}$  represent statistical probabilities for gray value  $j$  and  $i$ .  $T(r_j)$  and  $G(z_i)$  represent the transformations which could achieve the equalized images. It is described in Fig. 3.4, where red and blue lines represent the source histogram and target one respectively.

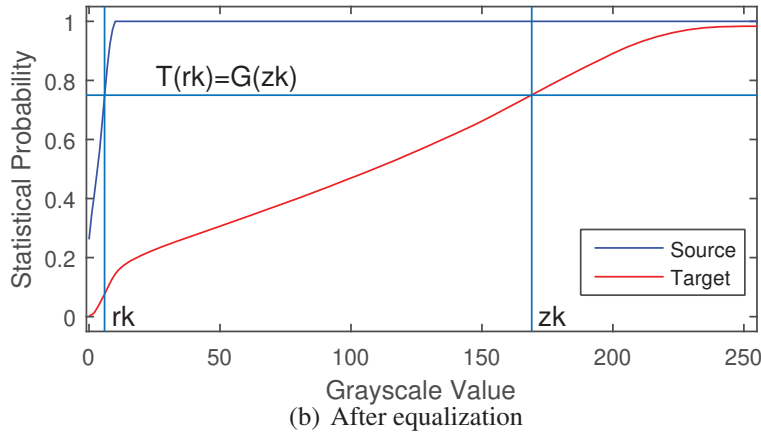
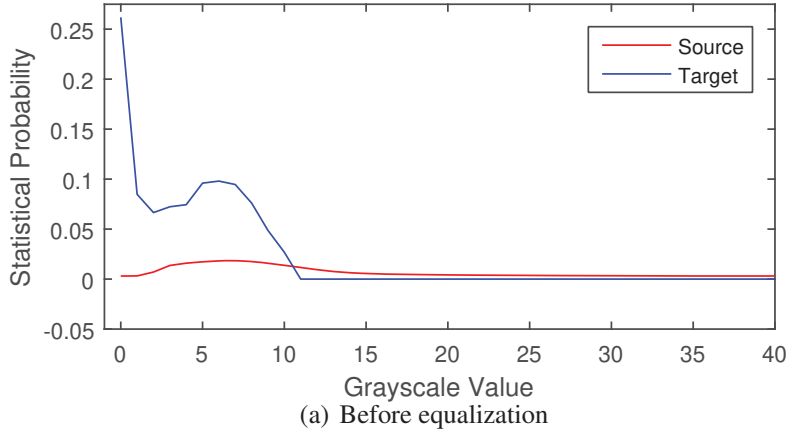


Fig. 3.4: Histogram equalization.

$$\begin{aligned}
 s_k = T(r_j) &= \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n} \\
 v_k = G(z_i) &= \sum_{i=0}^k p_z(z_i) = \sum_{i=0}^k \frac{m_i}{m}
 \end{aligned} \tag{3.2}$$

**Matching from Equalized Histograms:** After performing the equalization, histograms from both regions are stretched with the value range of  $[0, 255]$ . They have placed themselves onto the identical feature space. It means that we can simply consider  $s_k = v_k$ .

Such equivalence can be given by Eq. (3.3).

$$\sum_{j=0}^k \frac{n_j}{n} = \sum_{i=0}^k \frac{m_i}{m} \quad (3.3)$$

$$T(r_j) = G(z_i)$$




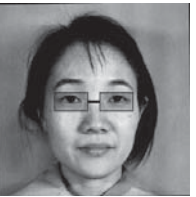

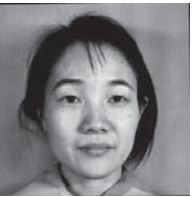
The mapping as formulated in Eq. (3.4) gives the demonstrated histogram matching.

$$z_i = G^{-1}(z_i) = G^{-1}[T(r_j)] \quad (3.4)$$

In Fig. 3.4, the smallest difference between statistical probability values for both the original and target image in equalized histogram is localized, before which an inverse transformation ( $G^{-1}(z_i)$ ) based on the target image is performed. A mapping relationship can be in this way constructed merely from the source grayscale value (style-discriminative data) of  $r_j$  to the target one  $z_i$  (style-eliminated data).

Table 3.2 depicts several examples for this matching performance. There, the style information brought by the sunglasses (as seen in images from the first row of the Table. 3.2) are mostly eliminated as seen from portraits of those recovered images listed in the second row. Compared with those from the third row, the style information is effectively eliminated in these style-eliminated portraits. Besides, most of the image details corrupted by the sunglasses are recovered correctly. These style-eliminated patterns will in this sense satisfy the *i.i.d.* assumption. Because of it, conventional classifiers including SVM [6], LDA [106] and KNN [107] are then to be optimized based on these style-eliminated data.

Table 3.2: Examples of histogram matching performance of sunglasses.

GVDR(%) Shape and ID	5 Circular KA.AN1.39	30 Rectangular KL.FE1.174	60 Circular NA.SA1.205	80 Rectangular UY.HA2.138
Sunglasses Image				
Recovered Image				
Original Correspondence				

### 3.3 Sunglasses Recovery Experiments

In this section, the results of several experiments to verify the proposed algorithm will be reported. Since there is no available facial expression dataset with sunglasses, a modified new set from the original JAFFE data will be firstly generated by automatically inserting different shapes of glasses with various luminance transmittance. The whole generation process is introduced in Section 3.3.1.

#### 3.3.1 Data Preparation

The original JAFFE dataset contains 213 images of 7 facial expressions from 10 females. For each one, there are 3 or 4 pictures involved. All portraits in the original JAFFE database are those wearing no glasses, so diverse sunglasses are added automatically around each eye section. These added sunglasses act as the different style information. Style-discriminative patterns are in this way synthesized.

By utilizing an Active Shape Model [36] based software package STASM [110, 111], in total 77 landmarks (as seen in Fig. 3.5) representing different facial component positions such as eyes, eyebrows, mouth, nose, etc., will be detected.

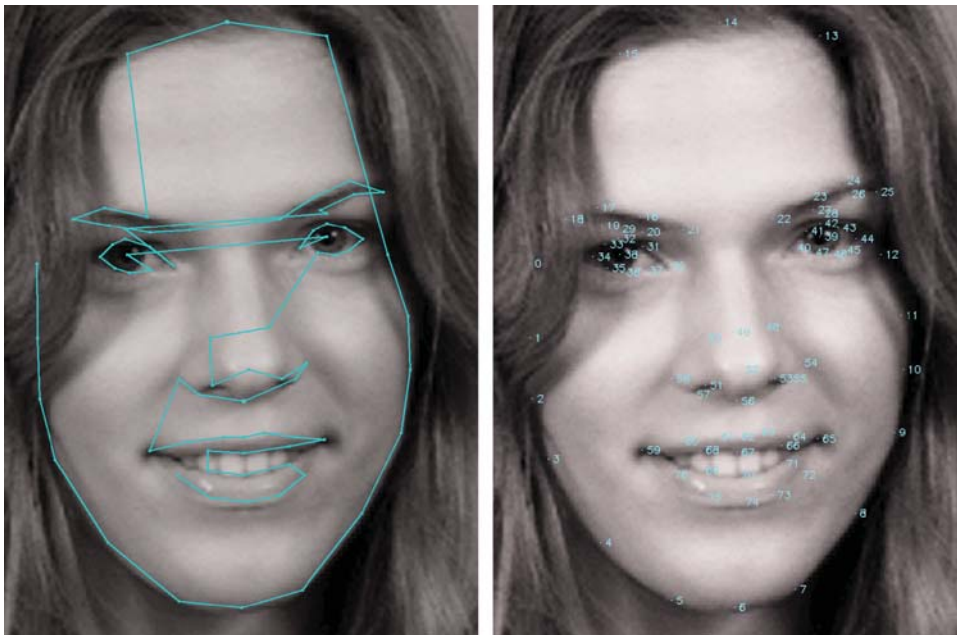


Fig. 3.5: The 77 Stasm landmarks and their numbers [110, 111]

As seen in Fig. 3.5, the corresponding regions of both eyes and eyebrows can be detected by landmarks No. 18, 17, 16, 21, 30, 37, 36,35,34 for the left eye / eyebrow and No. 25, 24, 23, 22, 40, 47, 46,45, 44 for the right eye / eyebrow. In the meanwhile, the centre of both eyes are seen as landmarks No. 38 and 39 for the left and right.

The sunglasses are manually added based on these landmarks. Specifically, each glass will be centred according to the specific eye centre. The size of the glass frame will be decided by landmark No. 18, 19, 21, 30, 36 for the left eye and No. 25, 26, 22, 40, 46 for the right eye. The exact size of the bounding box of the glass frame will be calculated as

the following two equations:

$$L_V = \frac{1}{2} \times [d_V(p_{19} - p_{36}) + d_V(p_{26} - p_{46})] \times 1.25 \quad (3.5)$$

$$L_H = \frac{1}{2} \times [d_H(p_{18} - p_{30}) + d_H(p_{25} - p_{40})] \times 1.45 \quad (3.6)$$

where  $p_i$  represents the  $i$ -th landmark,  $d_V(p_m, p_n)$  and  $d_H(p_m, p_n)$  is the calculating operator of the absolute vertical / horizontal distance between the corresponding two landmarks, namely,  $p_m$  and  $p_n$ .

Two shapes of sunglasses including rectangular and circular ones are synthesized and automatically added to the original facial image. The width and height of the rectangular glasses are  $L_V$  and  $L_H$  respectively, while the radius of the circular glasses is calculated as  $R = \min(L_V, L_H)$ . The centres of the glasses are put at the exact position of the centres of both eyes.

By changing the grayscale values in the glasses region at a specific rate, namely, Grayscale Value Dropping Rate (GVDR) no greater than 100%, different luminance transmittance is created. Thus, for each image in the original dataset, there are in total 198 correspondences with circular or rectangular sunglasses with GVDR from 5% to 100%.

Table 3.3: Examples of manually-added sunglasses.

Portrait ID	KA.AN1.39		KL.AN1.167		YM.AN1.61	
Original Image						
GVDR(%)	Circular	Rectangular	Circular	Rectangular	Circular	Rectangular
30						
60						
90						

In Table 3.3, it can be observed that lower GVDR leads to lower luminous transmittance for glasses (darker glasses), and hence more original information is erased. It is argued that if the GVDR is lower than 5%, there will be insufficient useful information

left. Accordingly, we will not consider such individual cases. This modified dataset with one-shot sunglasses portraits has been released.<sup>2</sup>

### 3.3.2 Experimental Setting

In this section, the performance is evaluated based on the recognition improvement brought by implementing the proposed recovery scheme. All images from both training and testing set will be firstly recovered according to the system demonstrated in Section 3.2. Then the recognition rate will be recorded. However, experiments with non-recovered images for both training and testing sets will also be conducted for comparison and evaluation. The style elimination performance will then be quantified according to the recognition rate improvement due to the recovery algorithm proposed in Section 3.2.

For recognition procedures, Gabor wavelet features with five scales and eight orientations are firstly extracted [114], after which Kernel PCA [115] will be performed to compress the feature data from 10240 dimensions into approximately 60-d with no less than 95% information preserved. Classifiers including SVM [6] (LibSVM [116]), LDA [106] and KNN [107], are then to be applied. The final recognition rate (FRR) was given as the average value over 50 times repeated independent experiments. The best parameters (the kernel width and balance parameter in SVM [6] and neighbor number for KNN [107]) were obtained with the grid searching strategy.<sup>3</sup>

Two major types of experiments were conducted. One was referred to as Individual Training and Testing, where images from both training and testing sets were all from the identical GVDR group. The other one was called the Full Training and Testing, where images from the training set came from the full database with various styles of sunglasses, as did the same with those in the testing set. The Full Training and Testing type was the best simulation of real application cases. When dividing the database into training and testing sets, two methods including the conventional five-fold scheme and the proposed one-shot one were utilized and compared.

#### Five-fold Scheme for Individual Training and Testing

When conducting the Individual Training and Testing with this scheme, 4/5 of images in the corresponding individual GVDR set were randomly chosen as the training set, while the remaining ones were put into the testing set.

#### Five-fold Scheme for Full Training and Testing

When performing the Full Training and Testing, the following conditions should be met: (1). A random binary switch is set to specify whether the specific image in the original JAFFE database or one of its corresponding images with various styles of sunglasses is to be put into the training set (4/5 of the original JAFFE database); (2). Another random binary switch is set to specify if the sunglasses are worn. Namely, half of the randomly selected portraits in the training set are wearing sunglasses; (3). A random integer value is set for ones wearing sunglasses from 5 to 100 to specify the GVDR of source sunglasses

---

<sup>2</sup>The Modified-JAFFE dataset is available online: <http://download.premilab.com/JAFFE-Modified.zip>

<sup>3</sup>Be noted that for each of the schemes that will be demonstrated in the following, 5% of the training data will be specified into the validation set to determine the best parameter setting.



for better simulation of real application scenario; (4). Also, a random binary switch is set to determine whether the portrait is wearing the circular sunglasses or rectangular ones.

For the last three conditions, they are only valid when the image is put into the training set. Otherwise, those pictures in the remaining 1/5 part of the original database are all put into the testing set with all corresponding images with or without various styles of sunglasses. In this sense, not only will the pictures of the training set never appear in the testing set, neither will their corresponding images do. Fig. 3.6 gives several examples for better explanation. Images with a white frame were put into the training set. In the meanwhile, those with no signs were put into the testing set. Ones with "N/A" labels were neither in the training nor testing sets.

### **One-shot Scheme for Individual Training and Testing**

As previously demonstrated in Section 3.3.1, three or four images were involved in each expression of each portrait. In this Individual Training and Testing, only one randomly selected image from three or four ones was put into the training set, while the remaining two or three were placed in the testing set.

### **One-shot Scheme for Full Training and Testing**

For the Full Training and Testing with the One-shot Scheme, similar conditions were also met as described in the five-fold section. The only difference is that the separating scheme here was one-shot scheme, rather than five-fold one. Fig. 3.7 gives several examples, in which two different expressions of two different portraits (Portrait: KA with Expression: Fear and Portrait: YM with Expression: Happy) were involved. Same as Fig. 3.6, images with "N/A" are irrelevant to both the training set and the test set. For each expression of each portrait, only one image was selected for the training set (picture with the white frame). For all the corresponding pictures of the chosen training images, all other ones were neither in the training nor the testing sets. All images apart from those mentioned above were put into the testing set. Similarly, for images from the training set, their corresponding ones will never be part of the testing set. Additionally, the five-fold training set is larger than the one of the one-shot.

## **3.3.3 Experimental Results**

### **Individual Training and Testing with Five-fold Scheme**

Table 3.4 gives the performance by separating the proposed database with a five-fold cross-validation scheme. When compared with those images without the proposed sunglasses recovery algorithm, a significant improvement on the final recognition rate is obtained for lower GVDR groups when both training set and testing set are images. However, the improvement reduces gradually as GVDR increases. When it reaches 90%, the improvement is weakened to negative values. Then there is in fact decrease rather than improvement brought due to the recovery scheme. It indicates that for higher GVDR level images, surely there is less information corrupted when sunglasses are added manually. Improvement on recognition rate due to the proposed recovery method is relatively decreased.

Table 3.4: Sunglasses recovery performance on **Individual Training and Testing with Five-fold Scheme**.

Group GVDR (%)	SVM [6]			LDA [106]			KNN [107]		
	NoRec <sup>a</sup>	Rec <sup>b</sup>	Iprm <sup>c</sup>	NoRec	Rec	Iprm	NoRec	Rec	Iprm
10 <sup>d</sup>	75.833 ±6.991	82.071 ±5.525	6.238	61.000 ±8.744	70.048 ±7.735	9.048	65.524 ±5.308	73.000 ±5.834	7.476
20	77.524 ±6.610	82.881 ±5.923	5.357	61.571 ±8.713	70.333 ±8.728	8.762	66.881 ±5.806	74.429 ±5.877	7.548
30	79.262 ±5.513	84.286 ±5.319	5.024	61.833 ±7.880	70.095 ±7.747	8.262	69.310 ±5.790	74.833 ±6.340	5.524
40	80.262 ±5.803	83.738 ±5.702	3.476	62.429 ±7.167	69.500 ±8.498	7.071	70.310 ±6.479	75.976 ±6.294	5.667
50	80.952 ±6.607	84.143 ±5.586	3.190	63.524 ±8.441	68.143 ±8.540	4.619	72.833 ±6.361	76.690 ±5.967	3.857
60	81.310 ±7.309	83.952 ±6.323	2.643	63.929 ±9.004	67.738 ±9.852	3.810	73.286 ±6.260	75.690 ±6.979	2.405
70	81.857 ±7.009	83.619 ±5.912	1.762	65.238 ±8.918	66.881 ±9.415	1.643	74.738 ±7.012	75.238 ±7.079	0.500
80	82.381 ±7.465	83.286 ±6.654	0.905	66.167 ±8.638	67.048 ±8.392	0.881	74.286 ±7.131	74.571 ±7.299	0.286
90	83.167 ±7.143	82.976 ±6.646	-0.190	66.476 ±8.259	66.214 ±7.996	-0.262	74.905 ±7.054	74.357 ±6.999	-0.548
100	83.333 ±6.263	83.310 ±6.142	-0.024	66.690 ±8.708	66.167 ±8.310	-0.524	74.619 ±7.263	74.095 ±7.356	-0.524
ORG <sup>e</sup>	86.000 ±6.911	85.714 ±6.477	-0.286	67.762 ±8.414	67.667 ±9.099	-0.095	84.286 ±7.063	84.143 ±7.920	-0.143
FULL <sup>f</sup>	74.973 ±5.853	81.485 ±5.794	6.512	60.789 ±10.006	60.760 ±7.632	6.979	65.561 ±5.270	70.382 ±6.192	4.821

<sup>a</sup> : No recovery scheme applied for both training and testing sets;

<sup>b</sup> : Recovery scheme applied for both training and testing sets;

<sup>c</sup> : Improvement achieved;

<sup>d</sup> : Individual testing with images at 10% GVDR;

<sup>e</sup> : Individual testing with images from original JAFFE database;

<sup>f</sup> : Full testing;

Table 3.5: Sunglasses recovery performance on **Individual Training and Testing with One-shot Scheme**.

Group GVDR(%)	[6]			LDA [106]			KNN [107]		
	NoRec	Rec	Iprm	NoRec	Rec	Iprm	NoRec	Rec	Iprm
10	69.804 ±3.396	77.112 ±3.683	7.308	67.874 ±6.878	73.056 ±6.229	5.182	61.888 ±3.263	70.385 ±2.337	8.497
20	71.112 ±3.246	78.643 ±3.468	7.531	68.147 ±6.474	72.154 ±7.551	4.007	63.580 ±3.438	72.399 ±2.260	8.818
30	72.762 ±3.122	79.224 ±3.249	6.462	68.580 ±6.223	73.056 ±8.167	4.476	65.203 ±3.246	72.986 ±2.531	7.783
40	74.238 ±3.561	79.986 ±3.053	5.748	69.748 ±6.398	73.280 ±7.296	3.531	67.126 ±2.949	73.706 ±2.637	6.580
50	75.804 ±3.619	80.720 ±3.391	4.916	70.692 ±6.783	74.455 ±6.923	3.762	68.951 ±2.724	74.399 ±2.147	5.488
60	77.056 ±3.340	80.538 ±3.549	3.483	71.524 ±7.365	74.294 ±6.770	2.769	70.524 ±2.276	74.371 ±2.630	3.846
70	78.427 ±3.392	80.469 ±3.227	2.024	72.406 ±7.171	74.035 ±7.685	1.629	72.399 ±2.453	74.755 ±2.425	2.357
80	79.448 ±3.049	80.594 ±3.520	1.147	73.490 ±5.928	73.622 ±7.038	0.133	73.755 ±2.417	74.552 ±2.759	0.797
90	80.007 ±3.125	80.357 ±3.281	0.350	74.371 ±6.088	73.315 ±6.428	-1.056	74.434 ±2.490	74.371 ±2.255	-0.063
100	80.483 ±3.437	79.937 ±2.845	-0.545	74.294 ±6.774	73.028 ±6.746	-1.266	74.175 ±2.278	74.427 ±2.347	0.252
ORG	88.168 ±2.747	87.245 ±2.902	-0.923	75.441 ±8.677	75.594 ±8.148	0.154	85.552 ±3.126	84.657 ±3.181	-0.895
FULL	69.090 ±2.986	75.182 ±3.145	6.192	64.302 ±6.270	70.160 ±7.258	5.858	63.855 ±2.238	68.407 ±2.638	4.552

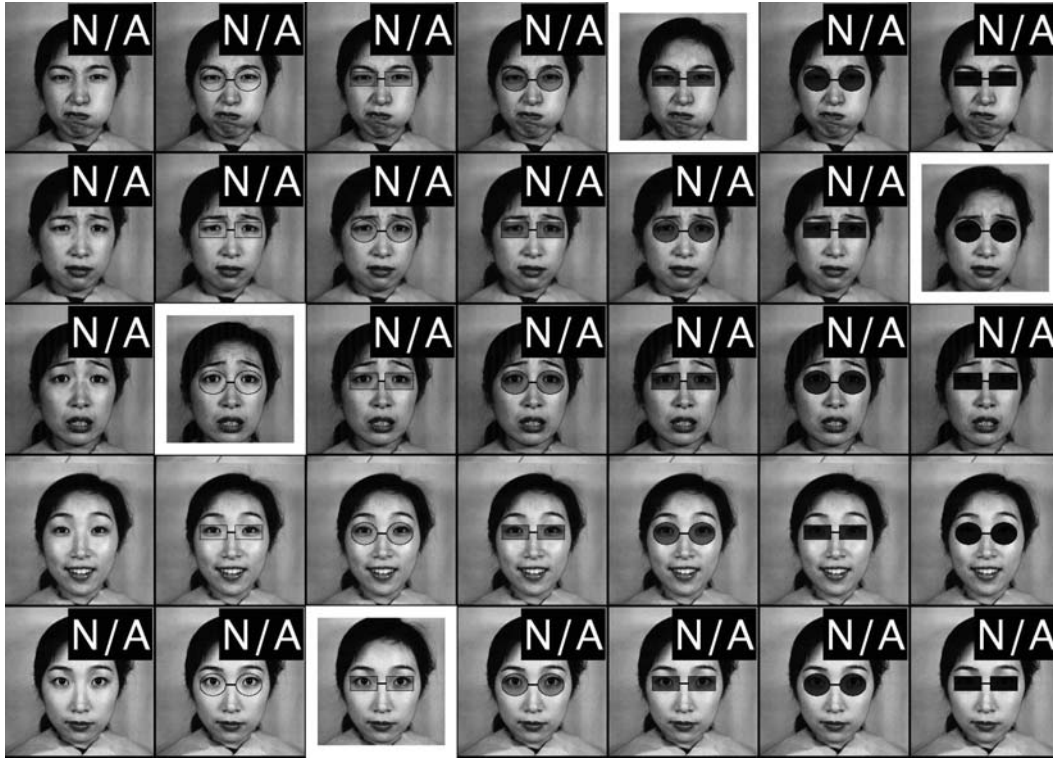


Fig. 3.6: Examples of **Five-fold Scheme for Full Training and Testing**

### Individual Training and Testing with One-shot Scheme

Table 3.5 represents recognition rate with one-shot separating scheme for the Individual Training and Testing type. Still, consistent improvement for recognition rate can be acquired for lower GVDR groups when both training set and testing set are images with recovery framework applied. The similar decrease can also be found as GVDR increases. When the GVDR is greater than 90%, the recognition performance is then weakened with the recovery scheme.

The performance for both separating schemes is given in Fig. 3.8, Fig. 3.9 and Fig. 3.10 for SVM [6], LDA [106] and KNN [107] respectively with the individual training and testing type. The improvement due to the proposed style elimination framework can be readily observed for all the three involved conventional machine learning classifiers. It concluded that the style-eliminated patterns pose a better choice when performing the traditional classification task compared with those style-discriminative ones.

Table 3.6: Sunglasses recovery performance on full training and testing with both five-fold and one-shot schemes.

Type	[6]		LDA [106]		KNN	
	Non	Rec	Non	Rec	Non	Rec
Five-Fold	74.973 ±3.240	81.485 ±3.409	60.780 ±5.927	67.760 ±7.010	65.561 ±2.298	70.382 ±2.714
One-Shot	69.090 ±2.986	75.182 ±3.145	64.302 ±6.270	70.160 ±7.258	63.855 ±2.238	68.407 ±2.638

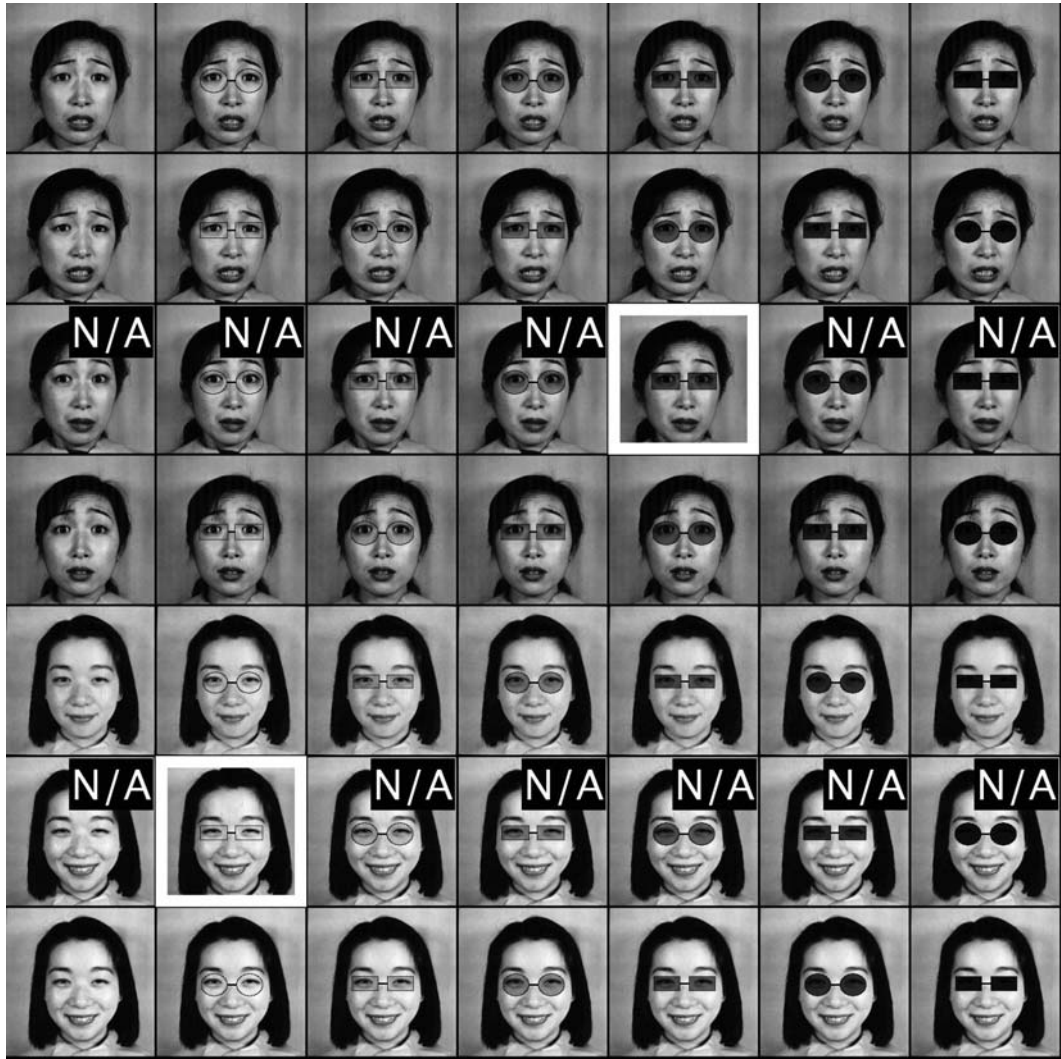


Fig. 3.7: Exapmls **One-shot Scheme for Full Training and Testing.**

### Full Training and Testing with both Schemes

Based on the results given in Table 3.6 and Fig. 3.11, a consistent promotion for SVM [6], LDA [106] and KNN [107] with both the five-fold scheme and the one-shot one can be found. When compared with results from Individual Training and Testing experiments, the benefits from the proposed recovery scheme shall never be underestimated. Although there exists decrease for higher GVDR group in individual types when considering overall performance with Full Training and Testing,<sup>4</sup> such promotion on the classification performance cannot be ignored. It can be concluded that introducing recovered images into both the training and testing set is effective in promoting the classification performance of sunglasses expression images with the One-shot Scheme. Improvements are 6.092%, 5.858% and 4.552% for SVM [6], LDA [106] and KNN [107] respectively.

Additionally, improvements are 6.512%, 6.979%, and 4.821% respectively for SVM [6], LDA [106], and KNN [107] in the Five-fold Scheme. It can also be seen that the recog-

<sup>4</sup>The Full Training and Testing is the best simulates applications for real scenarios, as demonstrated in Section 3.3.2.

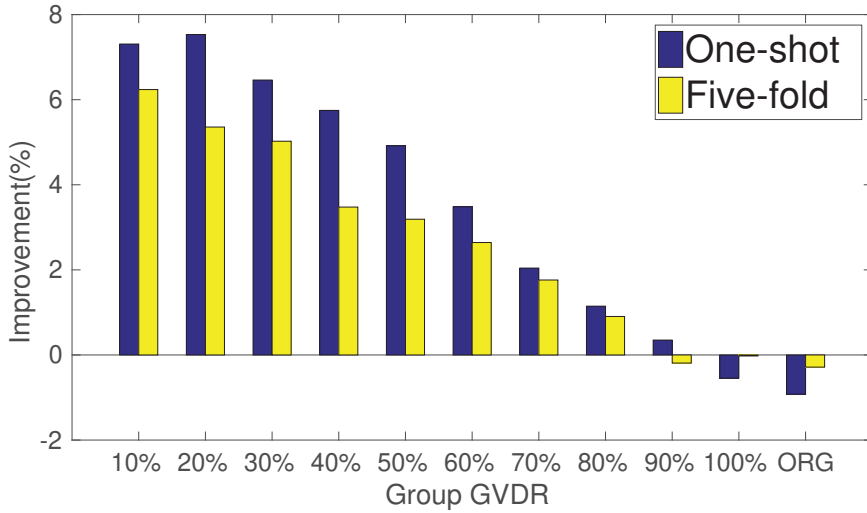


Fig. 3.8: Improvement on Final Recognition Rate (FRR) with SVM [6] Classifier brought by the proposed sunglasses recovery algorithm.

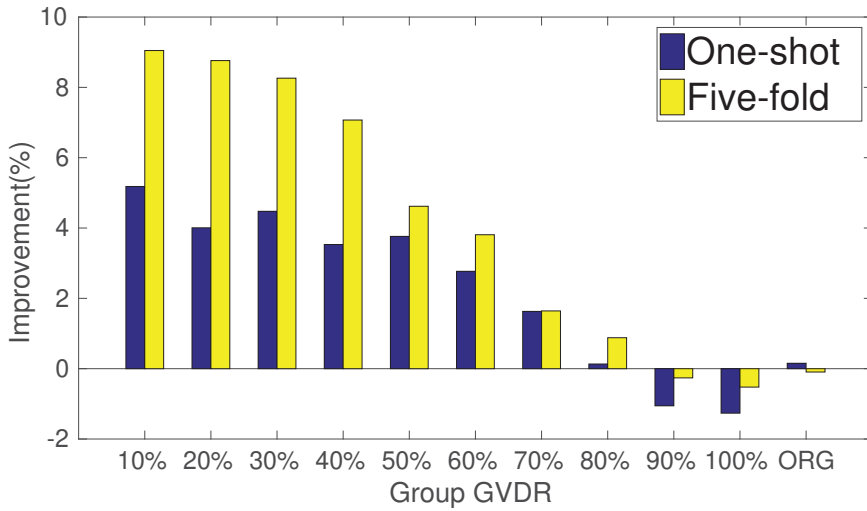


Fig. 3.9: Improvement on FRR with LDA [106] classifier.

dition performance on these style-eliminated data are consistently better those patterns equipped with diverse style information. Such conclusion is identical with the observation in the previous paragraph.

### Improvement difference between different experimental settings

There is another interesting finding raised from Fig. 3.6, Fig. 3.7, Table 3.4 and Table 3.5. It is that apart from the LDA [106] classifier, the improvement brought by the proposed sunglasses recovery algorithm to the One-shot scheme is always higher than that in the Five-fold scheme. The LDA [106] classifier is an exception, where the style elimination transformer is more effectiveness in the Five-fold case.

By looking into the data listed in Table 3.4 and Table 3.4 more carefully, one can observe that the LDA [106] classifier performs relatively worse in the Five-fold test than that in the One-shot case when there is no recovery is engaged. It means the potential im-

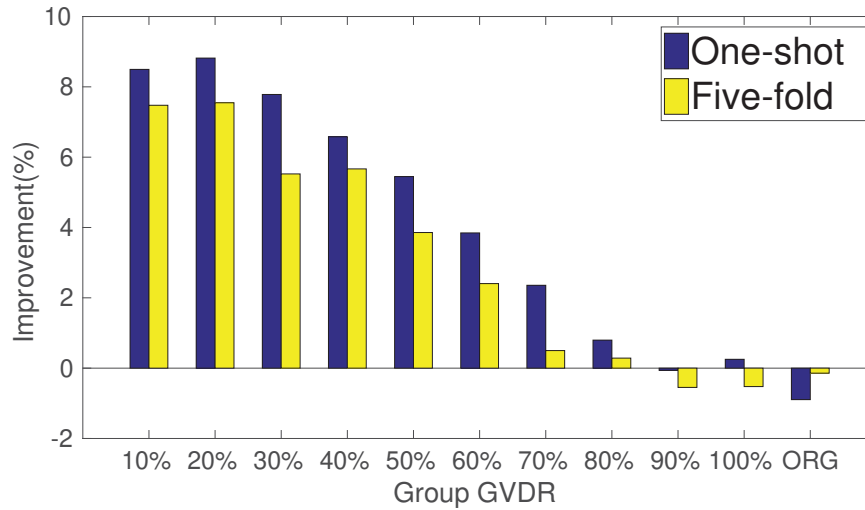


Fig. 3.10: Improvement on FRR with KNN [107] classifier.

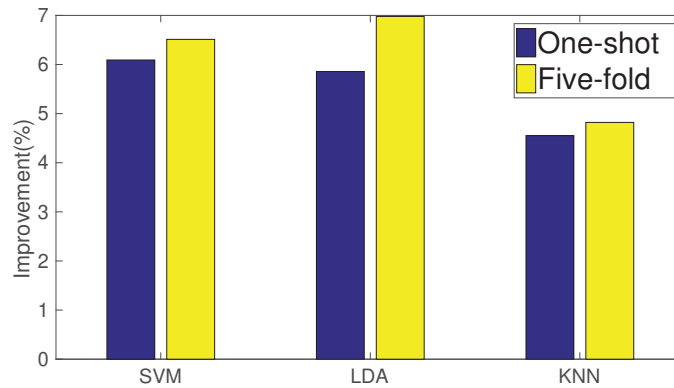


Fig. 3.11: Overall performance improvement on FRR with consistent improvement of SVM [6], LDA [106], and KNN [107].

provement of the LDA [106] classifier in the Five-fold case can be much more than other cases. However, when it comes to other cases, the classification performance has already been better, which limits the possible promotion due to the proposed style elimination transformation in this section.

### 3.4 Summaries and Future Work

A sunglasses recovery scheme based on the Canny Edge Detection and the Histogram Matching Algorithm for automatic facial expression recognition is introduced in this chapter. By randomly selecting one-shot images from a full database containing images with or without manually-added sunglasses with various luminous transmittance, the style information produced by diverse glasses are effectively eliminated. The generated style-eliminated patterns satisfy the *i.i.d.* assumption, necessary for most the conventional machine learning models. They are helpful to improve the final recognition performance on several of the conventional machine learning models for classification purposes. In comparison with experimental results without the recovery implemented, a significant improvement on the final recognition rate has been achieved with classifiers

such as SVM [6], LDA [106], and KNN [107]. An additional advantage is that it enables to utilize only one-shot image for each expression of each portrait in the training phase.

### **3.4.1 Future Work**

However, the proposed model in this chapter is an image processing based algorithm. The application scenarios are following a case-by-case manner since it is only effective in sunglasses recovery task. In the following chapter, a novel machine learning based discriminative model, namely, the Field Support Vector Machine, will be in detail described to fulfill the prediction task on non-*i.i.d.* data. It improves the flexibility when compared to those image processing based algorithms, while the state-of-the-art performance will be also achieved.



## Chapter 4

# Discriminative Approaches with Style Information

The discriminative machine learning model constructs the dependence of unobserved variables  $y$  on observed variables  $x$ . Within a probabilistic framework, this is done by modeling the conditional probability distribution  $P(y|x)$ , which can be used for predicting  $y$  from  $x$  [52]. The discriminative model enables prediction tasks including classification and regression without the necessity to estimate the joint probability distribution. It also yields superior performance with a much smaller number of parameters than those generative models, which will be demonstrated in Section 5 [52].

As demonstrated in Section 3, conventional predictors often regard input samples following the *i.i.d.* assumption, which does not always hold in many scenarios. A typical case is that patterns occur as groups, where each one shares a homogeneous style. This prediction task is named as the field prediction problem, which can be divided into the field classification (for discrete data classification) and the field regression (for continuous data regression).

By breaking the *i.i.d.* assumption, one novel framework named Field Support Vector Machine (F-SVM) will be introduced in this chapter. The F-SVM model is inherent from the state-of-the-art discriminative machine learning framework named Support Vector Machine [6, 117, 37] for both classification (Field Support Vector Classifier, F-SVC [13, 14]) and regression (Field Support Vector Regression, F-SVR [15]) purposes. The upon mentioned style averaging transformation described in Section 2 will be intelligently fulfilled with such a discriminative machine learning approach, rather than in a case-by-case image processing manner as demonstrated in Section 3. It is named as the Style Normalization in this chapter.

To be specific, the proposed F-SVM predictor is investigated by learning simultaneously both the predictor and the Style Normalization Transformation (SNT) for each group of data (named as a field). Such joint learning is even proved feasible in the high-dimensional kernel space. And efficient alternative optimization algorithm is further designed with the final convergence guaranteed theoretically and experimentally. Additionally, an intelligent self-training based optimization strategy is engaged to normalize the unseen style during training [16]. It is fulfilled by learning the transductive SNT (T-SNT) to transfer the already-trained field information to the unknown style data.

All of those relevant frameworks utilize the kernel trick [118] to deal with the sufficiently complicated style information with a proper nonlinear representation. Such an appealing factor was rarely seen in previous related models, such as the image process

based framework introduction in Section 3 [12], as well as the Field Bayesian Model, proposed in [29]. Furthermore, the kernelized style normalization requires no any data distribution assumption necessary before defining the model, which is theoretically able to approximate any kinds of mapping functions.

A series of experiments are performed to verify the effectiveness of the proposed F-SVM model with both classification and regression tasks by promoting the classification accuracy as well as declining the regression error. Empirical results demonstrate that the proposed scheme achieves in several benchmark datasets with the best performance so far. It is significantly better than those state-of-the-art predictors engaged as comparison baselines.

## 4.1 Problem Statement

Conventionally, most of the machine learning models assume that patterns shall follow the *identical and independent distribution (i.i.d.)*. However, such a preliminary condition may often be violated in some prediction scenarios in both classification and regression tasks. Particularly, when patterns occur as groups (where each group shares a homogeneous style, named as a field in this chapter), the degraded prediction performance may usually be seen on those *i.i.d.*-based models.

### 4.1.1 Typical Non-*i.i.d.* Prediction Scenarios

The phenomenon as mentioned above can be the results of the inconsistent data generation process in the most cases. For the field classification tasks, it may be due to that patterns are acquired from **non-identical data source generation procedures** with each one equipped with a specific style information [13, 14]. As seen in Fig. 4.1,  $\{X_i, Y_i\}$  represents data generated from the  $i$ -th source with the  $i$ -th specific stylistic information. Empirical examples can be found including face recognition [119] (where face images may be divided into different groups with each one sharing the same pose orientation), speech recognition [120] (where a group of speakers may share the common accent, e.g., English with a British style), and Chinese handwriting character recognition [121] (where samples written by one specific person are equipped with the same writing style).

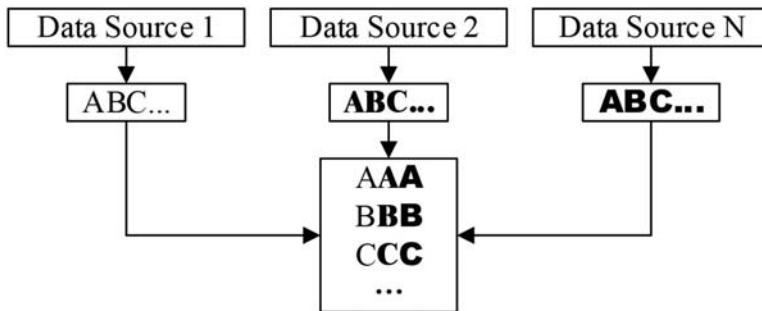


Fig. 4.1: Non-identical data generation process from style-inconsistent data sources for classification tasks.

When we consider regression situations on static patterns [15] (rather than data changing with time, e.g., stock prediction tasks [122]), the similar case can be induced by

**non-identical data mappings** when patterns are obtained. As seen with an illuminative example depicted in Fig. 4.2<sup>1</sup>, the original input data ( $X$ ) satisfy the *i.i.d.* assumption. They are placed in one identical space (represented by red and blue basis with the same orientation, which are parallel with each other among different data groups). The field regression problem is that these data  $X$  are grouped randomly with ones of each group (a field) being mapped to one other latent space ( $LS_i$ ) with the domain-specific mapping function ( $f_i$ ). In this sense,  $i$  represents the  $i$ -th group. No *i.i.d.* assumption will exist in  $LS_i$  any more.

Consequently, the basis vectors become different (those orange and black bases), representing data in different groups are placed in different spaces. It means that there exists clear and varied stylistic tendencies for latent patterns between different groups. In the meanwhile, these obtained hidden features ( $Z_i$ ) in each group are sharing one identical homogeneous style.

Then, another mapping ( $g$ ) identical for all groups will be engaged, generating target value within the same range (The additional green axes attached on the original  $LS_i$  space represent the mapping. The values on the attached green axes are noted as  $Y_i$ ). However, since there exists inconsistent mappings ( $f_i$ ), conventional regression models with the *i.i.d.* assumption would suffer declined performance severely with  $Z_i$ , or  $X$  as the input.

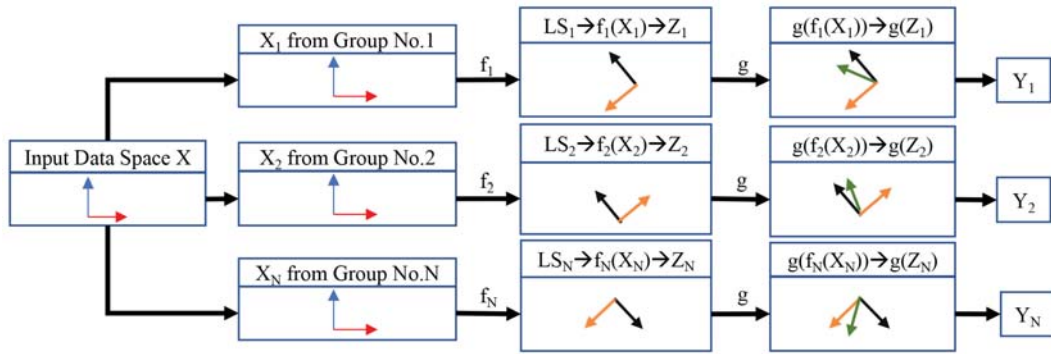


Fig. 4.2: Non-identical data mapping during data generation process for regression tasks [15].

Examples of such non-*i.i.d.* regression scenarios in real situations include scenarios as follows. For example, the same students ( $X$ ) in different schools may behave differently (latent features  $Z_i$ ) with different educational services (mapping  $f_i$ ) provided [123]. When participating in the same exam (mapping  $g$ ), their academic achievement (exam with 100 full marks represented by  $Y_i$ ) can be different as well. In this case, the index  $i$  represents the  $i$ -th school.

Similarly, different persons mostly probably have different viewpoints on the same product ( $X$ ), since each one has own thinking ( $Z_i$ ) with his or her specific background knowledge ( $f_i$ ) including education, experience or even gender and classes in a society, etc. [124]. When given the same questionnaire ( $g$ ), different opinions (ratings between 0 to 10 represented by  $Y_i$ ) would be drawn on one identical product. Here, the index  $i$  means the  $i$ -th customer.

<sup>1</sup>Fig. 4.2 only illustrates an instructive example where the data dimension is only two or three. However, in real scenarios, the data dimension can be extremely large.

However, to avoid vague concepts and unclear expressions in the following sections, the non-identical mapping situation in the field regression will be considered as a particular case of the non-identical data source generation procedures. Such scenarios have been specified in the demonstration of the field classification case in the previous paragraphs. As a summary, in each group (e.g., the  $i$ -th group) of both cases, field data are equipped with clear and consistent style information. It will be represented as  $\{X_i, Y_i\}$  for field classification, and  $\{Z_i, Y_i\}$  for field regression. Such information is inconsistent between different groups. It suggests that the field prediction is significantly different from traditional predictors since the *i.i.d.* assumption does not hold anymore.

#### 4.1.2 F-SVM Basic Framework for Classification and Regression

The idea **field prediction** is borrowed from the F-BM model proposed in [29] with the style normalization transformation (SNT). As demonstrated in that work, the SNT can only be represented as a linear transformation, which may potentially limit the performance since the style information can be possibly diverse. To tackle those problems mentioned above, we propose a field pattern prediction approach based on the kernel method [125, 118] to enable the nonlinear representation of the SNT.

One advantage brought by the kernel trick is that it provides a nonlinear mapping from the original data space to the reproduced Herbert kernel space. In that data space, initially nonlinearly distributed data can be linearly placed [125]. The kernel trick has achieved great success in the famous Support Vector Machines (SVM) models dealing with nonlinear pattern classification and regression problems [118, 125].

By taking the benefit from the nonlinear kernel mapping provided by the vanilla SVM formulations [6], we extend it with a novel Field Support Vector Machine (F-SVM) for both classification (Support Vector Classification, SVC) [6] and regression (Support Vector Regression, SVR) [117, 37] for field predictions including the field classification and the field regression tasks. They are named as the Field Support Vector Classification (F-SVC) [13, 14] and the Field Support Vector Regression (F-SVR) [15] respectively.

Specifically, the proposed F-SVM model is learned with both the predictor  $\{w, b\}$ , and the SNT simultaneously for each field to generate style-normalized (*i.i.d.*) patterns. Similar to the F-BM model [29], the proposed approach is capable of training and predicting a group of patterns with the same style of information simultaneously, which adequately considers the style consistency within each group. Such joint learning is even proved feasible in the high-dimensional kernel space, enabling the styles unnecessarily represented by linear transformation matrices only. An optimization algorithm designated to solve two convex quadratic programming problems with final convergence guaranteed is also proposed based on an efficiently and effectively iterative manner. Once the *i.i.d.* assumption among these style-normalized data is satisfied, the conventional classifier (e.g., SVM) can be after that applied properly. Fig. 4.3 depicts the basic framework of the proposed F-SVM model by taking the F-SVC as an illustrative example.

In particular, the F-SVM can be easily extended into styles unseen in training, as demonstrated in [16]. It can be noticed that a style transferring framework is proposed in the F-BM model [29] to map the known field information onto the unknown style by the linear transformation. Inspired by it, a self-training based transductive learning approach is also introduced to transfer the known style information acquired during the initial F-

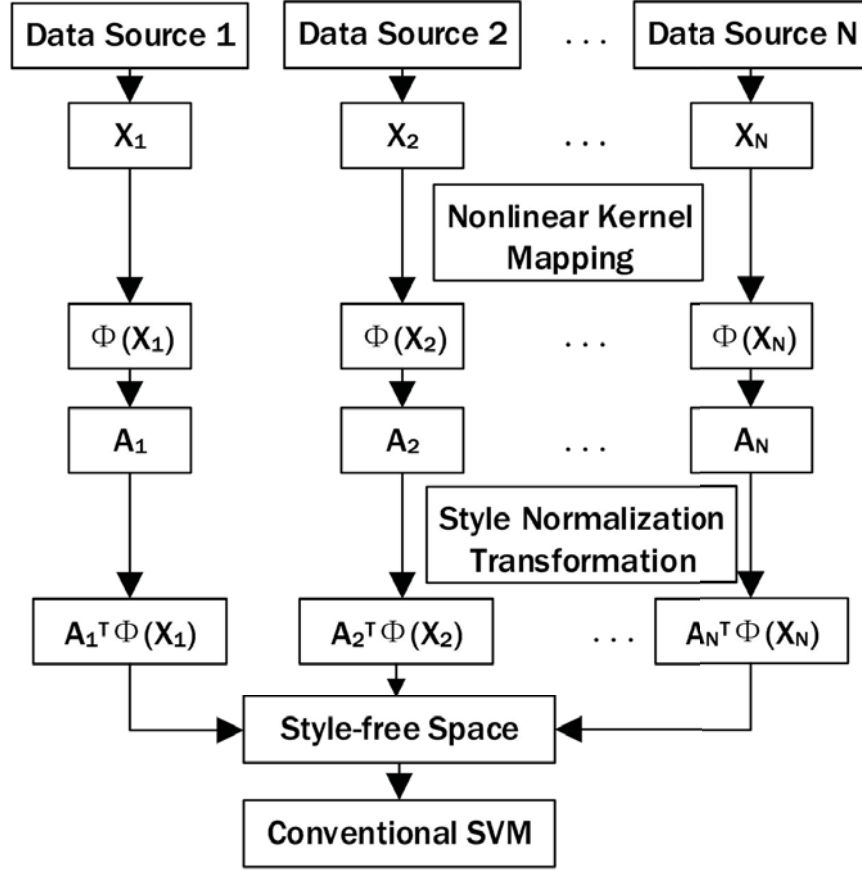


Fig. 4.3: Architecture of the Field Support Vector Machines: taking the F-SVC for field classification as an example.

SVM training procedures. It takes the advantage of the nonlinear kernel mapping provided by the kernel trick [118] to normalize the unknown style intelligently and effectively. The nonlinear transfer makes it possible to represent the complicated style information more reasonable and elegant. Details of such a style transfer framework based on the self-training strategy will be given in Section 4.4

Moreover, also distinct with previous approaches, the proposed F-SVM model is free from the before-hand data distribution assumption since the style information is simply normalized with the kernelized SNTs. Additionally, in Section 4.2.4, one of the MTL model, the MR-MTL [30] proposed in [30] is proved theoretically to be a special case of the proposed F-SVM framework. It represents that the field prediction framework provides a new perspective of the off-the-shelf MTL algorithms on the non-*i.i.d.* pattern predictions.

In this chapter, combined frameworks, notations and deduction procedures of both the F-SVC [13, 14] and the F-SVR [15] models will be given in detail, including the kernelized representation, as well as the transductive algorithm for unknown field generalization. The proposed F-SVM framework is not only justified theoretically but also proved to be able to improve the prediction accuracy in both classification and regression tasks significantly by promoting the recognition rate and reducing the regression error respectively. More particularly, F-SVM achieves so far the best performance in several

benchmark datasets in both tasks. To our best knowledge, it is the first work in kernel learning which can exploit the field information for non-*i.i.d.* data. It is also the initial step to integrate the field classification idea into a regression problem on static data.

## 4.2 F-SVM Model Specification

### 4.2.1 Basic Notation Involved

Basic notations of the proposed F-SVM model will be briefly introduced and defined in this section. These signs and concepts will be given for both the field classification as well as the field regression. The key difference will be emphasized when necessary.

**Definition 4.2.1.** Denote a group of patterns and the corresponding class labels as  $X_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \dots, \mathbf{x}_{L_i}^i\}$ ,  $Y_i = \{y_1^i, y_2^i, \dots, y_{L_i}^i\}$ , where  $\mathbf{x}_j^i \in \mathbb{R}^d$ ,  $y_j^i \in \{1, 2, \dots, M\}$  denotes the class label associated with  $\mathbf{x}_j^i$ . If all the patterns in  $X_i$  share the same style, we define  $X_i$  as a **field-pattern** with field-length  $L_i$ , and  $Y_i$  is the **field-class**. When  $L_i = 1$ , the field reduces to a single pattern, called **singlet**.

**Definition 4.2.2.** A group of latent patterns and their corresponding regression values are denoted as  $Z_i = \{\mathbf{z}_1^i, \mathbf{z}_2^i, \dots, \mathbf{z}_{L_i}^i\}$ ,  $Y_i = \{y_1^i, y_2^i, \dots, y_{L_i}^i\}$ , where  $\mathbf{z}_j^i \in \mathbb{R}^d$ ,  $y_j^i \in \mathbb{R}$  represents the association between  $\mathbf{z}_j^i$  and  $y_j^i$  (latent feature and regression value). If  $Z_i$  is obtained by the same domain-specific mapping  $f_i$  from the *i.i.d.* data group  $X_i$ , we define  $Z_i$  as **field-latent-pattern** with field-length  $L_i$ , and  $Y_i$  **field-regression-value**. The field degrades to the singlet if  $L_i = 1$  as a spacial case.

Be noted that the target values are all defined as  $Y_i$  in both Definition 4.2.1 and Definition 4.2.2. Nevertheless, they are totally different in the physical sense.  $Y_i$  in Definition 4.2.1 represents discrete class labels coming from  $\{1, 2, \dots, M\}$ . On the contrast, in Definition 4.2.2, it is a real continuous value, namely,  $y_i \in \mathbb{R}$ .

**Definition 4.2.3.** Given a training dataset  $D = \{F_1, F_2, \dots, F_N\}$ , where  $F_i = \{X_i, Y_i\}$  as defined in Definition 4.2.1 or  $F_i = \{Z_i, Y_i\}$  as defined in Definition 4.2.2, the **field prediction** is defined to train a predictor so that the field-class / field-regression-value of future field-pattern / field-latent-pattern can accurately be predicted.

Traditionally, labels or regression values are assigned one by one, as seen in Fig. 4.4(a) for a traditional classification scenario. It is quite similar to a conventional regression case. Differently, in field prediction, they are predicted **at one time** for all the patterns in a certain field (Fig. 4.4(b) for a field classification case).<sup>2</sup> In other words, the basic training or testing unit in field prediction is a group of samples with the consistent style information, or a field, while each sample is isolatedly handled traditionally.

### 4.2.2 Linear F-SVM Model

In the following section, the model definition of the F-SVM will be given for both binary field classification and regression scenarios with a linear kernel setting. Note that in the presented field classification framework, only the linear binary classification setting will be demonstrated in this section for simplicity. It is straightforward to extend the model

<sup>2</sup>A singlet can be considered as a special case of a field.

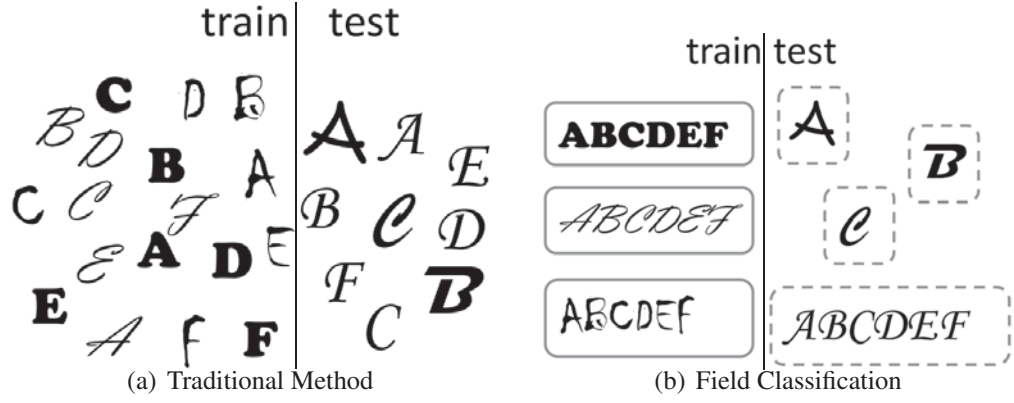


Fig. 4.4: Traditional classification and field classification [29].

into multi-class classification by one v.s. one or one v.s. others voting strategies [126]. The former one is employed to avoid the data imbalance issue brought by the other [127]. A kernelization version will be fully demonstrated after that.

Given a training dataset  $D = \{F_1, F_2, \dots, F_N\}$ , where  $F_i = \{X_i, Y_i\}$  as defined in Definition 4.2.1 for a classification case, we first present the F-SVM model for the binary field classification task as Eq. (4.1).<sup>3</sup>

$$\begin{aligned}
 \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} & \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N, L_i} \xi_j^i + ct \sum_{i=1}^N \|A_i^T - I\|_F^2 \\
 \text{s.t.} & y_j^i (\mathbf{w}^T A_i^T \mathbf{x}_j^i + b) \geq 1 - \xi_j^i, \quad \xi_j^i \geq 0, \\
 & \forall i = 1, \dots, N, \quad \forall j = 1, \dots, L_i.
 \end{aligned} \tag{4.1}$$

The difference of the F-SVM on the regression task based on the  $\epsilon$ -sensitive SVR [37] lies on the slack variable of the objective function, which is  $\xi_j^i + \xi_j^{i*}$ , rather than  $\xi_j^i$ . When given a training set as defined in Definition 4.2.2, namely,  $D = \{F_1, F_2, \dots, F_N\}$ , where  $F_i = \{Z_i, Y_i\}$ , the optimization objective function can be given by Eq. (4.2).<sup>4</sup>

$$\begin{aligned}
 \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} & \frac{1}{2} (\mathbf{w}^T \mathbf{w}) + c \sum_{i,j}^{N, L_i} (\xi_j^i + \xi_j^{i*}) + ct \sum_{i=1}^N \|A_i^T - I\|_F^2 \\
 \text{s.t.} & y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b \leq \epsilon + \xi_j^i, \quad \xi_j^i \geq 0, \\
 & \mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i \leq \epsilon + \xi_j^{i*}, \quad \xi_j^{i*} \geq 0 \\
 & \forall i = 1, \dots, N, \quad \forall j = 1, \dots, L_i
 \end{aligned} \tag{4.2}$$

Here  $\xi_j^i$  and  $\{\xi_j^i, \xi_j^{i*}\}$  are slack variables for F-SVC and F-SVR respectively.  $c$  and  $t$  (style normalization tradeoff parameter) are the positive trade-off parameters penalizing respectively slack variables and the over-flexible style transformation.<sup>5</sup> The multi-summation

<sup>3</sup>The code of the F-SVC model can be downloaded via: <https://github.com/falconjhc/FieldSVC>

<sup>4</sup>The code of the F-SVC model can be downloaded via: <https://github.com/falconjhc/FieldSVR>

<sup>5</sup> $c$  is the cost parameter identical with the vanilla SVM setting [6], while  $t$  is set particularly for the proposed F-SVM model.

operation is represented by only one summing sign for the sake of simplifying the expression, namely,  $\sum_{i,j}^{N,L_i} a_j^i = \sum_{i=1}^N \sum_{j=1}^{L_i} a_j^i$ . In following sections we will use this identical simplified representation.

There are two major novel distinctions between the proposed F-SVM model and the standard SVM framework. No.1, in addition to learning the classifier  $\{\mathbf{w}, b\}$ , the SNT matrices  $\{A_i\}$  will also be learned jointly. Each SNT is exploited to describe the style of information involved in each field. After the SNT, the style information is regarded to be normalized from samples within each field. They are hence considered to be transformed into a style-free space. No.2, a novel regularization term  $\|A_i^T - I\|_F^2$  is engaged, which is an extension of the vanilla SVM model [6]. The Frobenius norm is employed as the regularizer to constrain and restrict the SNT transformation from the identity, penalizing on over-flexible style transformation.

Be noted that there were quite a lot of kinds of other regularization terms introduced in the literature, e.g.,  $L_0$ -norm proposed in [128],  $L_1$ -norm proposed in [129]. A novel arbitrary norm is enabled in [130], but the model setting is entirely different from the proposed F-SVM model. In addition, for fixed  $c$  value, lower  $t$  values will apparently lead to greater SNT transformation, bringing more substantial field deviation from original data. Obviously, if we set  $t$  to  $+\infty$ ,  $A_i$  will be equal to  $I$ , leading that no style normalization is applied. The problem is degraded to the standard SVM optimization in this extreme case.

### 4.2.3 Alternative Optimization

The optimization defined in Eq. (4.1) and Eq. (4.2) are quadratically constrained quadratic program (QCQP) problems for F-SVC and F-SVR respectively. In particular, this objective yields a convex quadratic program in  $\mathbf{w}, b$  given  $\{A_i\}$ , and convex quadratic programs again in  $\{A_i\}$  given  $\mathbf{w}, b$ .

Although the QCQP problem is not jointly convex concerning  $\mathbf{w}, b$  and  $\{A_i\}$  at the same time, a good-enough local minimal via alternating optimization can still be obtained as long as a suitable initialization is made as  $A_i = I, \forall i$ . It is divided into two independent procedures as described in the following two subsections in each training iteration

#### Predictor Learning

When  $\{A_i\}$  ( $i = 1, 2, \dots, N$ ) are fixed, the optimization problem defined for the field classification task in Eq. (4.1) and Eq. (4.2) for F-SVC and F-SVR respectively can be degraded into Eq. (4.3) and Eq. (4.4):

$$\begin{aligned} \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} & \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N,L_i} \xi_j^i \\ \text{s.t.} & y_j^i (\mathbf{w}^T A_i^T \mathbf{x}_j^i + b) \geq 1 - \xi_j^i, \quad \xi_j^i \geq 0, \\ & \forall i = 1, \dots, N, \quad \forall j = 1, \dots, L_i. \end{aligned} \tag{4.3}$$



$$\begin{aligned}
& \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} \frac{1}{2}(\mathbf{w}^T \mathbf{w}) + c \sum_{i=1, j=1}^{N, L_1} (\xi_j^i + \xi_j^{i*}) \\
& \text{s.t.} \quad y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b \leq \epsilon + \xi_j^i, \quad \xi_j^i \geq 0, \\
& \quad \mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i \leq \epsilon + \xi_j^{i*}, \quad \xi_j^{i*} \geq 0 \\
& \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, L_i
\end{aligned} \tag{4.4}$$

As  $\{A_i\}$  are fixed, it is readily observed that the above optimization problems are standard SVM problems subject to data transformed by  $A_i^T$  for each field for both the classification and the regression scenarios. Both the linear version demonstrated in this section or the kernelized one instructed in Section 4.2.5 can all be easily optimized with the algorithm proposed in [118]. It will hence be easily solved by any SVM packages, e.g., LibSVM [131] with the option of customized input kernel matrix.

### SNT Learning

When  $\mathbf{w}, b$  are fixed, the optimization problem defined in Eq. (4.1) and Eq. (4.2) for F-SVC and F-SVR respectively can be readily transformed into  $N$  independent optimization problems with respect to  $A_i$ . They are given by Eq. (4.5) and Eq. (4.6) respectively.

$$\begin{aligned}
& \min_{A_i \in \mathbb{R}^{d \times d}} c \sum_{j=1}^{L_i} \xi_j^i + ct \|A_i^T - I\|_F^2 \\
& \text{s.t.} \quad y_j^i (\mathbf{w}^T A_i^T \mathbf{x}_j^i + b) \geq 1 - \xi_j^i, \\
& \quad \xi_j^i \geq 0, \quad \forall j = 1, \dots, L_i.
\end{aligned} \tag{4.5}$$

$$\begin{aligned}
& \min_{A_i \in \mathbb{R}^{d \times d}} c \sum_{j=1}^{L_i} (\xi_j^i + \xi_j^{i*}) + ct \|A_i^T - I\|_F^2 \\
& \text{s.t.} \quad y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b \leq \epsilon + \xi_j^i, \\
& \quad \mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i \leq \epsilon + \xi_j^{i*} \\
& \quad \xi_j^i \geq 0, \quad \xi_j^{i*} \geq 0, \quad \forall j = 1, \dots, L_i
\end{aligned} \tag{4.6}$$

Clearly, the above optimizations are convex quadratic programming problems, which can be solved by the Lagrangian multiplier  $\alpha_j^i$ . The dual problem for F-SVC and F-SVR are hereby deduced as Eq. (4.7) and Eq. (4.8) respectively:

$$\max_{0 \leq \alpha_j^i \leq 1, \forall j} \mathbb{L}_C = c \sum_{j=1}^{L_i} \xi_j^i + ct \|A_i^T - I\|_F^2 - \sum_{j=1}^{L_i} \gamma_j^i \xi_j^i - \sum_{j=1}^{L_i} \alpha_j^i [y_j^i (\mathbf{w}^T A_i^T \mathbf{x}_j^i + b) - 1 + \xi_j^i] \tag{4.7}$$

$$\begin{aligned}
\max_{0 \leq \alpha_j^i \leq 1, 0 \leq \alpha_j^{i*} \leq 1, \forall j} \mathbb{L}_R = & c \sum_{j=1}^{L_i} (\xi_j^i + \xi_j^{i*}) + ct \|A_i^T - I\|_F^2 - \sum_{j=1}^{L_i} \gamma_j^i \xi_j^i - \sum_{j=1}^{L_i} \gamma_j^{i*} \xi_j^{i*} \\
& - \sum_{j=1}^{L_i} \alpha_j^i [y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b - \epsilon - \xi_j^i] \\
& - \sum_{j=1}^{L_i} \alpha_j^{i*} [\mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i - \epsilon - \xi_j^{i*}]
\end{aligned} \tag{4.8}$$

By simply requiring  $\frac{\partial \mathbb{L}}{\partial A_i} = 0$ , the optimal solution of the SNT is obtained as Eq. (4.9).

$$A_i^T = \frac{1}{2t} \mathbf{w} \sum_{j=1}^{L_i} \alpha_j^i y_j^i \mathbf{x}_j^{iT} + I \tag{4.9}$$

Similarly, the F-SVR can be solved as Eq. (4.10).

$$A_i^T = \frac{1}{2t} \mathbf{w} \sum_{j=1}^{L_i} (\alpha_j^i - \alpha_j^{i*}) \mathbf{z}_j^{iT} + I \tag{4.10}$$

Additionally, by requiring  $\frac{\partial \mathbb{L}_C}{\partial \xi_j^i} = 0$  (for  $\xi_j^i \neq 0$  cases) in the F-SVC model,  $c = \alpha_j^i + \gamma_j^i$  will be obtained. When considering cases for  $\xi_j^i = 0$ , we will obtain Eq. (4.11).

$$\gamma_j^i \xi_j^i = (c - \alpha_j^i) \xi_j^i \tag{4.11}$$

In the same way, setting  $\frac{\partial \mathbb{L}_R}{\partial \xi_j^i} = 0$ ,  $\frac{\partial \mathbb{L}_R}{\partial \xi_j^{i*}} = 0$  simultaneously will lead to  $c = \alpha_j^i + \gamma_j^i$  and  $c = \alpha_j^{i*} + \gamma_j^{i*}$  (for  $\xi_j^i, \xi_j^{i*} \neq 0$  cases) respectively in the F-SVR model. When considering cases for  $\xi_j^i = 0$ ,  $\xi_j^{i*} = 0$ , the following equation set will be obtained.

$$\begin{cases} \gamma_j^i \xi_j^i = (c - \alpha_j^i) \xi_j^i \\ \gamma_j^{i*} \xi_j^{i*} = (c - \alpha_j^{i*}) \xi_j^{i*} \end{cases} \tag{4.12}$$

Both Eq. (4.11) and Eq. (4.12) are identical with the Karush-Kuhn-Tucker condition given in the original SVM formulation for both classification [6] and regression [37]. It can be then concluded that those Lagrangian coefficients introduced in the F-SVM formulation including  $\{\alpha_j^i, \xi_j^i\}$  in the F-SVC and  $\{\alpha_j^i, \alpha_j^{i*}, \xi_j^i, \xi_j^{i*}\}$  in the F-SVR are identical with those in the original SVM for both classification and regression tasks.

### Convergence Property

The optimization demonstrated in Section 4.2.3 is performed by minimizing the specific objective function in each step iteratively and alternatively. It includes two quadratic programmings, namely, the classifier learning and the SNT learning. It is evident that the optimization is achieved with a monotonically descending manner until it comes to a stable state generally since each quadratic programming requires the minimization on the specific optimizing status. It in this way obtains the final convergence with at least a local minimum acquired. Further proofs of this property will be demonstrated experimentally in Section 4.7.2. The overall alternative optimization for F-SVC and F-SVR with the linear kernel are summarized in the Algorithm 1 and Algorithm 2 respectively.

---

**Algorithm 1** F-SVC SNT alternative learning with the linear kernel

---

**Require:** Training field-pattern  $\{F_i\} = \{X_i, Y_i\}$ , ( $i = 1, 2, \dots, N$ ) as Definition 4.2.1;

**Ensure:** Predictor parameters:  $\{\mathbf{w}, b\}$ , SNT:  $\{A_i\}$ ,  $i = 1, 2, \dots, N$ ;

**Parameter (SVC):** cost  $c$ ;

**Parameter (Style Normalization):** style-normalization tradeoff  $t$ ;

**Initialization:**  $\{A_i\} = I$ ,  $convergence = FALSE$

1: **while**  $convergence == FALSE$  **do**

2: Field-pattern style-normalization:  $\tilde{\mathbf{x}}_j^i = A_i^T \mathbf{x}_j^i$ ,  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, L_i$ ;

3: Predictor parameters  $\{\mathbf{w}, b\}$  learning with  $\tilde{\mathbf{x}}_j^i$  (Standard SVC [6] as Eq. (4.3));

4: Calculate the objective value:  $\frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N, L_i} \xi_j^i + ct \sum_{i=1}^N \|A_i^T - I\|_F^2$  as Eq. (4.1);

5: Determine the convergence property from the objective calculations;

6: **if**  $convergence == TRUE$  **then**

7: break this while loop;

8: **else**

9: SNT learning:  $A_i^T = \frac{1}{2t} \mathbf{w} \sum_{j=1}^{L_i} \alpha_j^i y_j^i \mathbf{x}_j^i{}^T + I$ ,  $i = 1, 2, \dots, N$ , as Eq. (4.9);

10: **end if**

11: **end while**

---

#### 4.2.4 Relationship with the MTL model

The proposed F-SVM (including both the F-SVC and the F-SVR models) can be linked with a variant of Mean-regularized MTL (MR-MTL) [30]. The detailed demonstration will be fully instructed in this section.

**Theorem 4.2.1.** *The proposed F-SVC model defined in Eq. (4.1) and the F-SVR model in Eq. (4.2) is equivalent to one special case of the SVM-based MTL model.<sup>6</sup>*

Let  $\mathbf{w}_i = \mathbf{w} A_i^T$ , the F-SVC model defined in Eq. (4.1) and the F-SVR model defined in Eq. (4.2) can be rewritten and deduced as the following two proofs:

*Deduction of the duality of F-SVC in the manner of SVC-based MTL.*

$$\begin{aligned} \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} & \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N, L_i} \xi_j^i + ct \sum_{i=1}^N \|\mathbf{w}_i - \mathbf{w}\|_2^2 \\ \text{s.t.} & y_j^i (\mathbf{w}_i \mathbf{x}_j^i + b) \geq 1 - \xi_j^i, \quad \xi_j^i \geq 0 \\ & \forall i = 1, \dots, N, \quad \forall j = 1, \dots, L_i. \end{aligned} \quad (4.13)$$

The dual problem of Eq. (4.13) can be deduced by introducing the Lagrangian multipliers:

$$\begin{aligned} \max_{0 \leq \alpha_j^i \leq 1, \forall j} \mathbb{L} &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N, L_i} \xi_j^i + ct \sum_{i=1}^N \|\mathbf{w}_i - \mathbf{w}\|_2^2 \\ &\quad - \sum_{i,j}^{N, L_i} \alpha_j^i [y_j^i (\mathbf{w}_i \mathbf{x}_j^i + b) - 1 - \xi_j^i] - \gamma_j^i \xi_j^i \end{aligned} \quad (4.14)$$

□

---

<sup>6</sup>The following conclusions are deduced on eld classification task. However, it is obvious that for the field regression task, those conclusions are still valid.

---

**Algorithm 2** F-SVR SNT alternative learning with the linear kernel.

---

**Require:** Training field-pattern  $\{F_i\} = \{Z_i, Y_i\}$ , ( $i = 1, 2, \dots, N$ ) as Definition 4.2.2;

**Ensure:** Predictor parameters:  $\{\mathbf{w}, b\}$ , SNT:  $\{A_i\}$ ,  $i = 1, 2, \dots, N$ ;

**Parameter (SVR):** cost  $c$ ,  $\epsilon$ -tolerance  $\epsilon$ ,

**Parameter (Style Normalization):** style-normalization tradeoff  $t$ ;

**Initialization:**  $\{A_i\} = I$ ,  $convergence = FALSE$

```

1: while  $convergence == FALSE$  do
2:   Field-pattern style-normalization:  $\tilde{\mathbf{z}}_j^i = A_i^T \mathbf{z}_j^i$ ,  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, L_i$ ;
3:   Predictor parameters  $\{\mathbf{w}, b\}$  learning with  $\tilde{\mathbf{z}}_j^i$  (Standard SVR [37] as Eq. (4.4));
4:   Calculate the objective value:
5:    $\frac{1}{2}(\mathbf{w}^T \mathbf{w}) + c \sum_{i,j}^{N, L_i} (\xi_j^i + \xi_j^{i*}) + ct \sum_{i=1}^N \|A_i^T - I\|_F^2$  as Eq. (4.2);
6:   Determine the convergence property from the objective calculations;
7:   if  $convergence == TRUE$  then
8:     break this while loop;
9:   else
10:    SNT learning:  $A_i^T = \frac{1}{2t} \mathbf{w} \sum_{j=1}^{L_i} (\alpha_j^i - \alpha_j^{i*}) \mathbf{z}_j^i + I$ ,  $i = 1, 2, \dots, N$ , as Eq. (4.10);
11:  end if
12: end while

```

---

*Deduction of the duality of F-SVR in the manner of SVR-based MTL.*

$$\begin{aligned}
& \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N, L_i} (\xi_j^i + \xi_j^{i*}) + ct \sum_{i=1}^N \|\mathbf{w}_i - \mathbf{w}\|_2^2 \\
& \text{s.t. } y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b \leq \epsilon + \xi_j^i, \quad \xi_j^i \geq 0, \\
& \quad \mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i \leq \epsilon + \xi_j^{i*}, \quad \xi_j^{i*} \geq 0 \\
& \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, L_i
\end{aligned} \tag{4.15}$$

The dual problem of Eq. (4.15) can be deduced by introducing the Lagrangian multipliers:

$$\begin{aligned}
\max_{0 \leq \alpha_j^i \leq 1, \forall j} \mathbb{L} = & \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{j=1}^{L_i} (\xi_j^i + \xi_j^{i*}) + ct \sum_{i=1}^N \|\mathbf{w}_i - \mathbf{w}\|_2^2 - \sum_{j=1}^{L_i} \gamma_j^i \xi_j^i - \sum_{j=1}^{L_i} \gamma_j^{i*} \xi_j^{i*} \\
& - \sum_{j=1}^{L_i} \alpha_j^i [y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b - \epsilon - \xi_j^i] \\
& - \sum_{j=1}^{L_i} \alpha_j^{i*} [\mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i - \epsilon - \xi_j^{i*}]
\end{aligned} \tag{4.16}$$

□

By setting  $\frac{\partial \mathbb{L}}{\partial \mathbf{w}} = 0$  on both Eq. (4.14) and Eq. (4.16), the optimal solution for, namely  $\mathbf{w}$ , can all be obtained as  $\mathbf{w}^* = (\sum_{i=1}^N \mathbf{w}_i) / (N + \frac{1}{2ct})$ . It is an MTL formulation based on the SVM loss within total  $N$  predictors ( $\mathbf{w}_i = 1, \dots, N$ ) trained respectively for  $N$  tasks.

Additionally, if we set  $ct = +\infty$ , it can be concluded that  $\mathbf{w}^* = (\sum_{i=1}^N \mathbf{w}_i)/N$ . In this manner, the SVM-based MTL model defined in Eq. (4.13) and Eq. (4.15) for both the classification and regression tasks is very similar to the MR-MTL introduced in [30]. The only difference lies in the regularization term ( $\sum_{i=1}^N \mathbf{w}_i^2$  is applied in [30]). However, such two terms may enjoy a similar physical meaning in the sense that they are used to minimize the structure risks.

The proposed F-SVM model offers a new perspective to the previous MTL-based models [44, 45, 32]. It emphasizes the learning a single classifier and many style normalization transformations on various fields (or tasks). On the contrast, in the MTL-based models, the equal number of classifiers of tasks are learned simultaneously. Another benefit of the F-SVM model is that the field information is embedded in a  $d \times d$  matrix  $A_i$ , while task properties are described by  $d \times 1$  vector  $\mathbf{w}_i$ . The F-SVM is more flexible than the MR-MTL [30] when utilizing the style (or task) information when considering the number of parameters to be trained.

## 4.2.5 Kernelized F-SVM Representation

In [6, 37], the kernel tricks are employed to conquer the situations when data are not ideally linearly distributed. Similarly, the proposed F-SVM model including the F-SVC and the F-SVR can all be rewritten with the kernelized representation. As the fact that the kernel trick would always map the original input pattern to the high-dimensional kernel space. In some cases, e.g., for a Gaussian Kernel (GK), the dimension can be infinite [118]. In this way, formulations and algorithms demonstrated in Section 4.2.2 cannot be further utilized. In this section, detailed deduction procedures will be fully demonstrated for both the kernelized F-SVC and kernelized F-SVR representations. Modifications on the Algorithm 1 and Algorithm 2 to incorporate with the kernelized representation will also be instructed.

### Kernelized Update for the F-SVC Model

Suppose each data sample  $\mathbf{x}_j^i$  as defined in Definition 4.2.1 is projected to  $\phi(\mathbf{x}_j^i)$  in the high-dimensional space. Then the transformed data after SNT is given as  $\tilde{\phi}(\mathbf{x}_j^i) = A_i^T \phi(\mathbf{x}_j^i)$ . An update for the kernel matrix with implicit formulation is given by  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) = \phi^T(\mathbf{x}_{j_1}^{i_1}) A_{i_1} A_{i_2}^T \phi(\mathbf{x}_{j_2}^{i_2})$ , where the kernel matrix is originally represented by  $K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \phi^T(\mathbf{x}_{i_1}) \phi(\mathbf{x}_{i_2})$ . Moreover, the updating representation can also be obtained with a kernelized formulation, which makes it possible to implement with the precomputed kernel with SVM computation library such as LibSVM [131]. Hereby, the kernel is updated as the following formulation:

$$\begin{aligned} \tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) &= \phi^T(\mathbf{x}_{j_1}^{i_1}) \cdot A_{i_1} \cdot A_{i_2}^T \cdot \phi(\mathbf{x}_{j_2}^{i_2}) \\ &= \phi^T(\mathbf{x}_{j_1}^{i_1}) \cdot \left( \frac{1}{2t} \mathbf{w} \sum_{j_1=1}^{L_{i_1}} \alpha_{j_1}^{i_1} y_{j_1}^{i_1} \phi^T(\mathbf{x}_{j_1}^{i_1}) + I \right)^T \cdot \left( \frac{1}{2t} \mathbf{w} \sum_{j_2=1}^{L_{i_2}} \alpha_{j_2}^{i_2} y_{j_2}^{i_2} \phi^T(\mathbf{x}_{j_2}^{i_2}) + I \right) \cdot \phi(\mathbf{x}_{j_2}^{i_2}) \end{aligned} \quad (4.17)$$

Substitute  $\mathbf{w}^T = \sum_{i,j}^{N,L_i} \alpha_j^i \cdot y_j^i \cdot \phi^T(\mathbf{x}_j^i)$  [6] and the SNT solution Eq. (4.9) of the F-SVC model into Eq. (4.17), the following update scheme will be fulfilled readily:

$$\begin{aligned}
\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) &= \phi^T(\mathbf{x}_{j_1}^{i_1}) \cdot \left\{ \frac{1}{2t} \phi(\mathbf{x}_{j_1}^{i_1}) \alpha_{j_1}^{i_1} y_{j_1}^{i_1} \sum_{p_1, q_1}^{N, L_{p_1}} [\alpha_{q_1}^{p_1} y_{q_1}^{p_1} \phi^T(\mathbf{x}_{q_1}^{p_1})] + I \right\} \\
&\quad \cdot \left\{ \frac{1}{2t} \sum_{p_2, q_2}^{N, L_{p_2}} [\phi(\mathbf{x}_{q_2}^{p_2}) \alpha_{q_2}^{p_2} y_{q_2}^{p_2}] \alpha_{j_2}^{i_2} y_{j_2}^{i_2} \phi^T(\mathbf{x}_{j_2}^{i_2}) + I \right\} \cdot \phi(\mathbf{x}_{j_2}^{i_2}) \\
&= \left( \frac{1}{2t} \right)^2 \{ \alpha_{j_1}^{i_1} y_{j_1}^{i_1} \alpha_{j_2}^{i_2} y_{j_2}^{i_2} \phi^T(\mathbf{x}_{j_1}^{i_1}) \phi(\mathbf{x}_{j_1}^{i_1}) \phi^T(\mathbf{x}_{j_2}^{i_2}) \phi(\mathbf{x}_{j_2}^{i_2}) \} \\
&\quad \cdot \sum_{p_1, q_1}^{N, L_{p_1}} \sum_{p_2, q_2}^{N, L_{p_2}} [\alpha_{q_1}^{p_1} y_{q_1}^{p_1} \alpha_{q_2}^{p_2} y_{q_2}^{p_2} \phi^T(\mathbf{x}_{q_1}^{p_1}) \phi(\mathbf{x}_{q_2}^{p_2})] \\
&\quad + \frac{1}{2t} \sum_{p_1, q_1}^{N, L_{p_1}} [\alpha_{j_1}^{i_1} y_{j_1}^{i_1} \alpha_{q_1}^{p_1} y_{q_1}^{p_1} \phi^T(\mathbf{x}_{j_1}^{i_1}) \phi(\mathbf{x}_{j_1}^{i_1}) \phi^T(\mathbf{x}_{q_1}^{p_1}) \phi(\mathbf{x}_{j_2}^{i_2})] \\
&\quad + \frac{1}{2t} \sum_{p_2, q_2}^{N, L_{p_2}} [\alpha_{q_2}^{p_2} y_{q_2}^{p_2} \alpha_{j_2}^{i_2} y_{j_2}^{i_2} \phi^T(\mathbf{x}_{j_1}^{i_1}) \phi(\mathbf{x}_{q_2}^{p_2}) \phi^T(\mathbf{x}_{j_2}^{i_2}) \phi(\mathbf{x}_{j_2}^{i_2})] + \phi^T(\mathbf{x}_{j_1}^{i_1}) \phi(\mathbf{x}_{j_2}^{i_2})
\end{aligned} \tag{4.18}$$

Then, by substituting  $K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \phi^T(\mathbf{x}_{i_1}) \phi(\mathbf{x}_{i_2})$ , and letting  $\beta_i = \alpha_i y_i$ , we obtain the following kernelized updating scheme for the F-SVC model:

$$\begin{aligned}
\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) &= \left( \frac{1}{2t} \right)^2 \{ \beta_{j_1}^{i_1} \beta_{j_2}^{i_2} K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_1}^{i_1}) K(\mathbf{x}_{j_2}^{i_2}, \mathbf{x}_{j_2}^{i_2}) \} \cdot \sum_{p_1, q_1}^{N, L_{p_1}} \sum_{p_2, q_2}^{N, L_{p_2}} [\beta_{q_1}^{p_1} \beta_{q_2}^{p_2} K(\mathbf{x}_{q_1}^{p_1}, \mathbf{x}_{q_2}^{p_2})] \\
&\quad + \frac{1}{2t} \sum_{p_1, q_1}^{N, L_{p_1}} [\beta_{j_1}^{i_1} \beta_{q_1}^{p_1} K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_1}^{i_1}) K(\mathbf{x}_{q_1}^{p_1}, \mathbf{x}_{j_2}^{i_2})] \\
&\quad + \frac{1}{2t} \sum_{p_2, q_2}^{N, L_{p_2}} [\beta_{q_2}^{p_2} \beta_{j_2}^{i_2} K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{q_2}^{p_2}) K(\mathbf{x}_{j_2}^{i_2}, \mathbf{x}_{j_2}^{i_2})] \\
&\quad + K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})
\end{aligned} \tag{4.19}$$

### Kernelized Update for the F-SVR Model

Similar as demonstrated in Section 4.2.5, it is assumed a latent pattern  $\mathbf{z}_j^i$  as defined in Definition 4.2.2 is mapped to  $\phi(\mathbf{z}_j^i)$  with the kernel mapping. The style normalized data is in this way obtained as  $\tilde{\phi}(\mathbf{z}_j^i) = A_i^T \phi(\mathbf{z}_j^i)$ . Kernel update can be represented implicitly as  $\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2}) = \phi^T(\mathbf{z}_{j_1}^{i_1}) A_{i_1} A_{i_2}^T \phi(\mathbf{z}_{j_2}^{i_2})$ , in which the kernel matrix is originally defined as

$K(\mathbf{z}_{i_1}, \mathbf{z}_{i_2}) = \phi^T(\mathbf{z}_{i_1})\phi(\mathbf{z}_{i_2})$ . The kernelized update for F-SVR model will be deduced:

$$\begin{aligned}\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2}) &= \phi^T(\mathbf{z}_{j_1}^{i_1}) \cdot A_{i_1} \cdot A_{i_2}^T \cdot \phi(\mathbf{z}_{j_2}^{i_2}) \\ &= \phi^T(\mathbf{z}_{j_1}^{i_1}) \cdot \left( \frac{1}{2t} \mathbf{w} \sum_{j_1=1}^{L_{i_1}} (\alpha_{j_1}^{i_1} - \alpha_{j_1}^{i_1*}) \phi^T(\mathbf{z}_{j_1}^{i_1}) + I \right)^T \\ &\quad \cdot \left( \frac{1}{2t} \mathbf{w} \sum_{j_2=1}^{L_{i_2}} (\alpha_{j_2}^{i_2} - \alpha_{j_2}^{i_2*}) \phi^T(\mathbf{z}_{j_2}^{i_2}) + I \right) \cdot \phi(\mathbf{z}_{j_2}^{i_2})\end{aligned}\quad (4.20)$$

Similarly, by substituting  $\mathbf{w}^T = \sum_{i=1, j=1}^{N, L_i} (\alpha_j^i - \alpha_j^{i*}) \cdot \phi^T(\mathbf{x}_j^i)$  [37] and the SNT solution for the F-SVR model, Eq. (4.10), into Eq. (4.20), we obtain:

$$\begin{aligned}\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2}) &= \phi^T(\mathbf{z}_{j_1}^{i_1}) \cdot \left\{ \frac{1}{2t} \phi(\mathbf{z}_{j_1}^{i_1}) (\alpha_{j_1}^{i_1} - \alpha_{j_1}^{i_1*}) \sum_{p_1, q_1}^{N, L_{p_1}} [(\alpha_{q_1}^{p_1} - \alpha_{q_1}^{p_1*}) \phi^T(\mathbf{z}_{q_1}^{p_1})] + I \right\} \\ &\quad \cdot \left\{ \frac{1}{2t} \sum_{p_2, q_2}^{N, L_{p_2}} [\phi(\mathbf{z}_{q_2}^{p_2}) (\alpha_{q_2}^{p_2} - \alpha_{q_2}^{p_2*})] (\alpha_{j_2}^{i_2} - \alpha_{j_2}^{i_2*}) \phi^T(\mathbf{z}_{j_2}^{i_2}) + I \right\} \cdot \phi(\mathbf{z}_{j_2}^{i_2}) \\ &= \left( \frac{1}{2t} \right)^2 \{ (\alpha_{j_1}^{i_1} - \alpha_{j_1}^{i_1*}) (\alpha_{j_2}^{i_2} - \alpha_{j_2}^{i_2*}) \phi^T(\mathbf{z}_{j_1}^{i_1}) \phi(\mathbf{z}_{j_1}^{i_1}) \phi^T(\mathbf{z}_{j_2}^{i_2}) \phi(\mathbf{z}_{j_2}^{i_2}) \} \\ &\quad \cdot \sum_{p_1, q_1}^{N, L_{p_1}} \sum_{p_2, q_2}^{N, L_{p_2}} [(\alpha_{q_1}^{p_1} - \alpha_{q_1}^{p_1*}) (\alpha_{q_2}^{p_2} - \alpha_{q_2}^{p_2*}) \phi^T(\mathbf{z}_{q_1}^{p_1}) \phi(\mathbf{z}_{q_2}^{p_2})] \\ &\quad + \frac{1}{2t} \sum_{p_1, q_1}^{N, L_{p_1}} [(\alpha_{j_1}^{i_1} - \alpha_{j_1}^{i_1*}) (\alpha_{q_1}^{p_1} - \alpha_{q_1}^{p_1*}) \phi^T(\mathbf{z}_{j_1}^{i_1}) \phi(\mathbf{z}_{j_1}^{i_1}) \phi^T(\mathbf{z}_{q_1}^{p_1}) \phi(\mathbf{z}_{j_2}^{i_2})] \\ &\quad + \frac{1}{2t} \sum_{p_2, q_2}^{N, L_{p_2}} [(\alpha_{q_2}^{p_2} - \alpha_{q_2}^{p_2*}) (\alpha_{j_2}^{i_2} - \alpha_{j_2}^{i_2*}) \phi^T(\mathbf{z}_{j_1}^{i_1}) \phi(\mathbf{z}_{q_2}^{p_2}) \phi^T(\mathbf{z}_{j_2}^{i_2}) \phi(\mathbf{z}_{j_2}^{i_2})] \\ &\quad + \phi^T(\mathbf{z}_{j_1}^{i_1}) \phi(\mathbf{z}_{j_2}^{i_2})\end{aligned}\quad (4.21)$$

Again, by substituting  $K(\mathbf{z}_{i_1}, \mathbf{z}_{i_2}) = \phi^T(\mathbf{z}_{i_1})\phi(\mathbf{z}_{i_2})$ , and letting  $\beta_i = \alpha_i - \alpha_i^*$ , we will obtain the following kernel update representation:

$$\begin{aligned}\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2}) &= \left( \frac{1}{2t} \right)^2 \{ \beta_{j_1}^{i_1} \beta_{j_2}^{i_2} K(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_1}^{i_1}) K(\mathbf{z}_{j_2}^{i_2}, \mathbf{z}_{j_2}^{i_2}) \} \cdot \sum_{p_1, q_1}^{N, L_{p_1}} \sum_{p_2, q_2}^{N, L_{p_2}} [\beta_{q_1}^{p_1} \beta_{q_2}^{p_2} K(\mathbf{z}_{q_1}^{p_1}, \mathbf{z}_{q_2}^{p_2})] \\ &\quad + \frac{1}{2t} \sum_{p_1, q_1}^{N, L_{p_1}} [\beta_{j_1}^{i_1} \beta_{q_1}^{p_1} K(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_1}^{i_1}) K(\mathbf{z}_{q_1}^{p_1}, \mathbf{z}_{j_2}^{i_2})] \\ &\quad + \frac{1}{2t} \sum_{p_2, q_2}^{N, L_{p_2}} [\beta_{q_2}^{p_2} \beta_{j_2}^{i_2} K(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{q_2}^{p_2}) K(\mathbf{z}_{j_2}^{i_2}, \mathbf{z}_{j_2}^{i_2})] + K(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2})\end{aligned}\quad (4.22)$$

Be noted that the formulation type of Eq. (4.22) is highly consistent with the kernel update of the F-SVC model, as represented in Eq. (4.19). The only difference lies in the input pattern, where the latent feature  $\mathbf{z}$  is involved for the F-SVR framework in Eq. (4.22), while original pattern  $\mathbf{x}$  is engaged in Eq. (4.19).

### Kernelized Objective Function

For each alternative optimization iteration, the objective function of Eq. (4.3) and Eq. (4.4) shall be calculated to check the convergence for F-SVC and F-SVR respectively. These objective functions can all be divided into two parts with the kernelized representation with the similar components.

**F-SVC Kernelized Objective Function:** For the classifier objective function in the F-SVC model, as defined in Eq. (4.3), by substituting  $\mathbf{w}^T = \sum_{i,j}^{N,L_i} \alpha_j^i \cdot y_j^i \cdot \phi^T(\mathbf{x}_j^i)$ ,  $K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \phi^T(\mathbf{x}_{i_1})\phi(\mathbf{x}_{i_2})$ , and letting  $\beta_i = \alpha_i y_i$ , the maximum margin part with the kernelized representation can be deduced as follows:

$$\begin{aligned} \frac{1}{2} \mathbf{w}^T \mathbf{w} &= \frac{1}{2} \cdot \left[ \sum_{i_1, j_1}^{N, L_{i_1}} \alpha_{j_1}^{i_1} \cdot y_{j_1}^{i_1} \cdot \phi^T(\mathbf{x}_{j_1}^{i_1}) \right] \cdot \left[ \sum_{i_2, j_2}^{N, L_{i_2}} \alpha_{j_2}^{i_2} \cdot y_{j_2}^{i_2} \cdot \phi^T(\mathbf{x}_{j_2}^{i_2}) \right]^T \\ &= \frac{1}{2} \sum_{i_1, j_1}^{N, L_{i_1}} \sum_{i_2, j_2}^{N, L_{i_2}} [\beta_{j_1}^{i_1} \beta_{j_2}^{i_2} K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})] \end{aligned} \quad (4.23)$$

Second, for those support vectors (slack variables with corresponding  $\alpha$  Lagrange coefficients equal to the cost  $c$ ), the distance between these variables and their related soft margins shall be calculated with the kernelized representation as follows:

$$\begin{aligned} \xi_j^i &= \alpha_j^i \cdot \{y_j^i [(\mathbf{w}^T) \phi(\mathbf{x}_j^i) + b] - 1\} \\ &= \alpha_j^i y_j^i \left[ \sum_{m,n}^{N, L_m} \alpha_n^m \cdot y_n^m \cdot \phi^T(\mathbf{x}_n^m) \right] \phi(\mathbf{x}_j^i) + \alpha_j^i y_j^i b - \alpha_j^i \\ &= \sum_{m,n}^{N, L_m} [\beta_j^i \beta_n^m K(\mathbf{x}_n^m, \mathbf{x}_j^i)] + \beta_j^i b - \alpha_j^i \end{aligned} \quad (4.24)$$

The full objective of the SVC [6] model formulated in Eq. (4.3) are thereby deduced in a kernelized version as follows:

$$\min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} \frac{1}{2} \sum_{i_1, j_1}^{N, L_{i_1}} \sum_{i_2, j_2}^{N, L_{i_2}} [\beta_{j_1}^{i_1} \beta_{j_2}^{i_2} K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})] + c \sum_{m,n}^{N, L_m} [\beta_j^i \beta_n^m K(\mathbf{x}_n^m, \mathbf{x}_j^i)] + \beta_j^i b - \alpha_j^i \quad (4.25)$$

**Kernelized Objective Function SVR:** The SVR objective function defined in Eq. (4.4) is divided into two parts similarly, which can also be reorganized with the kernelized representation. The regularization part is deduced as follows by substituting  $\mathbf{w}^T =$



$\sum_{i=1}^N \sum_{j=1}^{L_i} (\alpha_j^i - \alpha_j^{i*}) \cdot \phi^T(\mathbf{z}_j^i)$ ,  $K(\mathbf{z}_{i_1}, \mathbf{z}_{i_2}) = \phi^T(\mathbf{z}_{i_1})\phi(\mathbf{z}_{i_2})$ , and letting  $\beta_i = \alpha_i - \alpha_i^*$ :

$$\begin{aligned} \frac{1}{2} \mathbf{w}^T \mathbf{w} &= \frac{1}{2} \cdot \left[ \sum_{i_1, j_1}^{N, L_{i_1}} (\alpha_{j_1}^{i_1} - \alpha_{j_1}^{i_1*}) \cdot \phi^T(\mathbf{z}_{j_1}^{i_1}) \right] \cdot \left[ \sum_{i_2, j_2}^{N, L_{i_2}} (\alpha_{j_2}^{i_2} - \alpha_{j_2}^{i_2*}) \cdot \phi^T(\mathbf{z}_{j_2}^{i_2}) \right]^T \\ &= \frac{1}{2} \sum_{i_1, j_1}^{N, L_{i_1}} \sum_{i_2, j_2}^{N, L_{i_2}} [\beta_{j_1}^{i_1} \beta_{j_2}^{i_2} K(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2})] \end{aligned} \quad (4.26)$$

Simultaneously, for training error part with regard to those slack variables ( $\alpha - \alpha^* = c$ ), it shall be computed with the kernelized representation [37] as follows:

$$\begin{aligned} \xi_j^i + \xi_j^{i*} &= \|(\mathbf{w}^T)\phi(\mathbf{z}_j^i) + b - y_j^i\|_1 = \left\| \left[ \sum_{m=1}^N \sum_{n=1}^{L_i} (\alpha_n^m - \alpha_n^{m*}) \cdot \phi^T(\mathbf{z}_n^m) \right] \phi(\mathbf{z}_j^i) + b - y_j^i \right\|_1 \\ &= \left\| \sum_{m=1}^N \sum_{n=1}^{L_i} [\beta_n^m K(\mathbf{z}_n^m, \mathbf{z}_j^i)] + b - y_j^i \right\|_1 \end{aligned} \quad (4.27)$$

So the full objective function of the SVR can be deduced as follows:

$$\min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} \frac{1}{2} \sum_{i_1, j_1}^{N, L_{i_1}} \sum_{i_2, j_2}^{N, L_{i_2}} [\beta_{j_1}^{i_1} \beta_{j_2}^{i_2} K(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2})] + c \left\| \sum_{m=1}^N \sum_{n=1}^{L_i} [\beta_n^m K(\mathbf{z}_n^m, \mathbf{z}_j^i)] + b - y_j^i \right\|_1 \quad (4.28)$$

## Kernelized Algorithms

The kernelized version of the training procedures for the linear F-SVC and F-SVR as demonstrated in Algorithm 1 and Algorithm 2 are summarized as Algorithm 3 and Algorithm 4 respectively.

---

### Algorithm 3 Kernelized F-SVC SNT alternative learning.

---

**Require:** Training field-pattern  $\{F_i\} = \{X_i, Y_i\}$ , ( $i = 1, 2, \dots, N$ ) as Definition 4.2.1;

**Ensure:** Final updated kernel:  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$ ;

**Parameter (SVC):** cost  $c$ , Gaussian Kernel (GK) width  $\gamma$ ;

**Parameter (Style Normalization):** style-normalization tradeoff  $t$ ;

**Initialization:**  $K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \phi^T(\mathbf{x}_{i_1})\phi(\mathbf{x}_{i_2})$  (Equivalent:  $\{A_i\} = I$ );

**Initialization:**  $convergence = FALSE$

1: **while**  $convergence == FALSE$  **do**

2:     Kernelized update:  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) = \phi^T(\mathbf{x}_{j_1}^{i_1}) A_{i_1} A_{i_2}^T \phi(\mathbf{x}_{j_2}^{i_2})$  (Eq. (4.19))

3:     Classifier learning on  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$  (kernelized SVC [6] training);

4:     Calculate the kernelized SVC [6] objective value as Eq. (4.25);

5:     Determine the convergence property from the objective calculations;

6:     **if**  $convergence == TRUE$  **then**

7:         break this while loop;

8:     **end if**

9: **end while**

---

---

**Algorithm 4** Kernelized F-SVR SNT alternative learning.

---

**Require:** Training field-pattern  $\{F_i\} = \{Z_i, Y_i\}$ , ( $i = 1, 2, \dots, N$ ) as Definition 4.2.2;

**Ensure:** Final updated kernel:  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$ ;

**Parameter (SVR):** cost  $c$ , Gaussian Kernel (GK) width  $\gamma$ ,  $\epsilon$ -tolerance  $\epsilon$ ;

**Parameter (Style Normalization):** style-normalization tradeoff  $t$ ;

**Initialization:**  $K(\mathbf{z}_{i_1}, \mathbf{z}_{i_2}) = \phi^T(\mathbf{z}_{i_1})\phi(\mathbf{z}_{i_2})$  (Equivalent:  $\{A_i\} = I$ );

**Initialization:**  $convergence = FALSE$

- 1: **while**  $convergence == FALSE$  **do**
  - 2:     Kernelized update:  $\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2}) = \phi^T(\mathbf{z}_{j_1}^{i_1})A_{i_1}A_{i_2}^T\phi(\mathbf{z}_{j_2}^{i_2})$  (Eq. (4.22))
  - 3:     Regressor learning on  $\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2})$  (kernelized SVR [37] training);
  - 4:     Calculate the kernelized SVR [37] objective value as Eq. (4.28);
  - 5:     Determine the convergence property from the objective calculations;
  - 6:     **if**  $convergence == TRUE$  **then**
  - 7:         break this while loop;
  - 8:     **end if**
  - 9: **end while**
- 

## 4.3 Prediction Rules for Future Patterns

Once the F-SVM parameters  $\{\mathbf{w}, b\}$  and  $\{A_i\}$  ( $i = 1, 2, \dots, N$ ) are learned, class labels (for the classification task) or regression values (for the regression task) of future field data can be predicted. We consider two different prediction scenarios for future patterns, including the **Singlet Prediction** and the **Field Prediction**.

In the singlet prediction, the style information of testing fields is not used. It is the conventional SVM prediction with only learned SVM parameters  $\{\mathbf{w}, b\}$  via F-SVM formulation. It will be demonstrated in Section 4.3.1. On the contrast, in the field prediction, a group of field data is normalized with the learned SNT  $\{A_i\}$  before they are being predicted. The field prediction can be also divided into two scenarios depending on whether the styles of future patterns occur during the training or not. We will introduce methods handling these cases mentioned above when the pattern is either a singlet-pattern or a field-pattern. Notably, we will specify the detailed algorithm for the case when the field information of the new data is unknown or not included in the fields of training data. A self-training based transductive approach will be demonstrated in Section 4.4 particularly.

### 4.3.1 Singlet Prediction

#### Traditional Prediction Rule

TPR is designed to exploit the same formulation as the standard SVC [6] decision function  $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b)$  for the F-SVC model, where the weight vector  $\mathbf{w}$  and the bias  $b$  are trained with the framework of F-SVC described in Section 4.2.3. For the F-SVR model, the decision function is formulated as  $f(\mathbf{z}) = \mathbf{w}^T\mathbf{z} + b$ . It evaluates the performance of the predictor trained on the style-normalized patterns with new style-discriminative data.

#### Voted Prediction Rule (VPR, for F-SVC Only)

VPR is designed to intelligently and automatically select the field index with the highest probability for the sake of utilizing those SNTs learned from the training data. It is given

by:  $y = \text{sgn}(\mathbf{w}^T A_{i^*} \mathbf{x} + b)$ , where  $i^* = \arg \max_i p(A_i^T \mathbf{x})$ . The decision of VPR is considered as the soft voting from  $N$  probabilities calculated respectively from each training field. Note that the probabilities can be easily obtained by using a softmax function over the output of SVM, as described in [132, 133]. The following equation gives an example:

$$P(y = 1|x) \approx P_{A,B}(f) \equiv \frac{1}{1 + \exp(Af + B)} \quad (4.29)$$

However, the probabilities calculation is only valid for the classification task [133]. Consequently, the VPR can only be used for the field classification task with the F-SVC model [13, 14] trained by Algorithm 1 (linear version) or Algorithm 3 (kernelized version).

### Averaged Decision Rule (APR, for F-SVR Only)

Because there is no formulation for the SVR model [37] to produce the probability estimation [37], so the previous VPR cannot be utilized for the field regression occasion. In this way, the APR is designed to make use of all the training style information. It is aimed to utilizing those SNTs ( $\{A_i\}, i = 1, 2, \dots, N$ ) as well as predictor parameters  $\{\mathbf{w}, b\}$  trained from Algorithm 2 (or the kernelized version Algorithm 4) for the field regression task, as the F-SVR model proposed in [15]. The formulation is given by  $y^i = \frac{1}{N} \sum_{i=1}^N (A_i^T \mathbf{z}^i + b)$ .

The regression of APR is calculated by averaging predictions along all the training fields. It is the representation of the averaged style-normalization of all the SNTs trained from Algorithm 2 or Algorithm 4.

### 4.3.2 Field Prediction Rule (FPR)

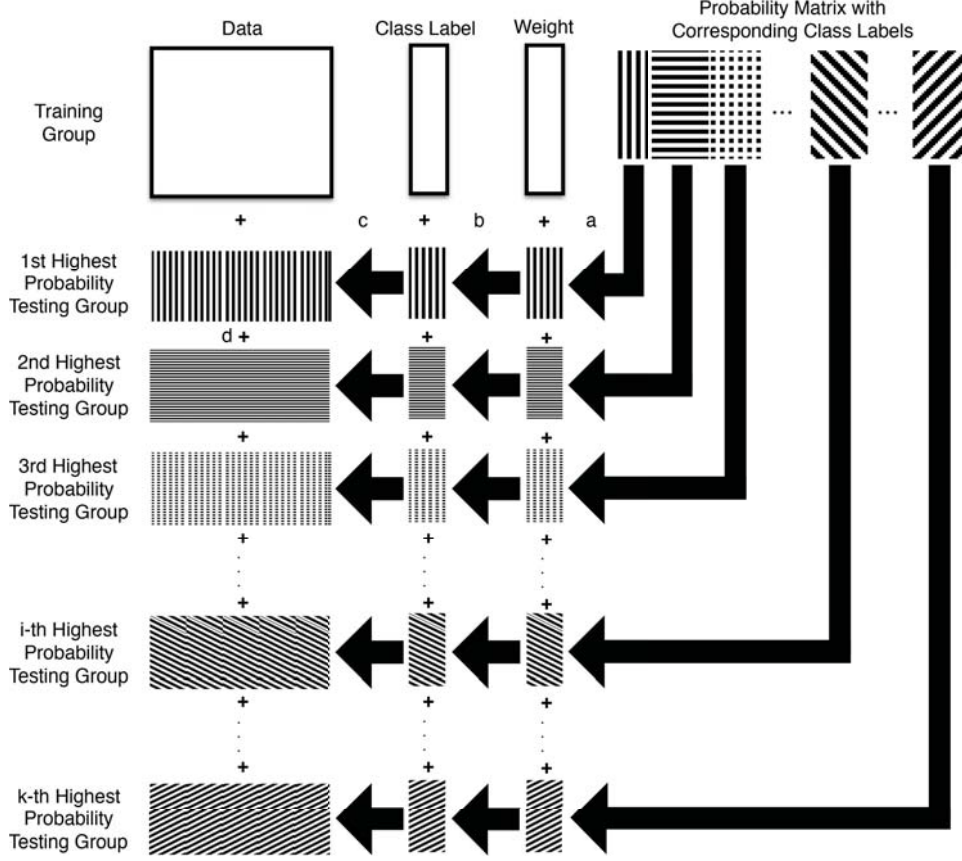
If the style of a new-coming field pattern, e.g., one specific style (say  $A_0$ ) seen in the training dataset, is known, it is straightforward and obvious to predict the future field pattern as  $y = \text{sgn}(\mathbf{w}^T A_0^T \mathbf{x} + b)$  for the F-SVC model. The field decision of the F-SVR, on the contrary, is evaluated as  $y = \mathbf{w}^T A_0^T \mathbf{z} + b$ . It is the most basic function of the proposed F-SVM model to perform the field prediction.

## 4.4 Self-training based Transductive Learning

When the case when the field information of the new data is unknown or not included in the fields of training data, a transductive algorithm with the alternative optimization strategy as described in Algorithm 5 and 6 will be utilized to perform the prediction of the F-SVC and F-SVR models respectively. At this moment, the Transductive-SNT (T-SNT) is learned, meanwhile the predictor parameter  $\{w, b\}$  particularly for those style-normalized field-unknown patterns will also be optimized. The decision rule by taking advantage of the T-SNT framework is named as Field Transfer Prediction Rule (FTPR). Supposing the new-coming style is indexed as 0, which means that the target of the self-training based transductive learning is to obtain the T-SNT matrix  $A_0$ . In this section, the self-training based procedures to transfer the known styles to unknown ones will be demonstrated in detail.

#### 4.4.1 Transductive F-SVC Formulation

For the field classification occasion of the F-SVC model, the new-coming pattern group occurring with an unknown field from the training set are represented as  $X_0 = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_n^0\}$ . Initially the class label  $Y_0 = \{y_1^0, y_2^0, \dots, y_n^0\}$  of them are estimated by the already-trained (with field-known training examples) F-SVC model with either TPR or VPR, as demonstrated in Section 4.3.1. Then, the initial prediction  $\{X_0, Y_0\}$ , seen as field-pattern with a new and known field, are put into the F-SVC model to be retrained until convergence. Specifically, the training process of T-SNT for this unseen field is mostly follow-



- a: Selecting  $k$  highest probabilities (for each test sample);
- b: Finding  $k$  class labels corresponding to  $k$  highest probabilities selected;
- c: Test data repeated concatenation for  $k$  times;
- d: + representing concatenating, not matrix addition.

Fig. 4.5: F-SVC testing sample repeated concatenation scheme.

ing the idea of the Instance-weighted SVC [134, 135] model. In their work, the weight of each pattern is taken into account. The T-SNT matrix is learned by Eq. (4.30), where  $a_j$  represents the weight for the training instance  $x_j$ :

$$\begin{aligned} \min_{A \in \mathbb{R}^{d \times d}} \quad & \sum_{j=1}^{L_0} a_j^0 \xi_j^0 + t \|A_0^T - I\|_F^2. \\ \text{s.t.} \quad & y_j^0 (\mathbf{w}^T A_0^T \mathbf{x}_j^0 + b) \geq 1 - a_j^0 \xi_j^0, \quad \xi_j^0 \geq 0, \quad \forall j = 1, \dots, L_0. \end{aligned} \quad (4.30)$$

Additionally, as noted in [134, 135], the instance weight shall also be embedded in the original SVC [6] learning formulation. It is given as:

$$\begin{aligned}
& \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i,j}^{N+1, L_i} a_j^i \xi_j^i \\
& \text{s.t. } y_j^i (\mathbf{w}^T A_i^T \mathbf{x}_j^i + b) \geq 1 - \xi_j^i, \quad \xi_j^i \geq 0, \\
& \quad \forall i = 1, \dots, N, N+1, \quad \forall j = 1, \dots, L_i.
\end{aligned} \tag{4.31}$$

One thing that is necessary to be pointed out is that the training fields in Eq. (4.31) includes the testing field, whose style information is unseen during the SNT training as demonstrated in Section 4.2.2 and Section 4.2.5. It is noted as the  $(N+1)$ -th field in the Eq. 4.31, or the 0-th field (the only field) in the Eq. (4.30). On the contrary, in Eq. (4.3) for the original SNT learning, only data from  $N$  training styles are involved.

Furthermore, in the proposed self-training framework to normalize the style information of these field-unknown patterns, samples in Eq. (4.30) consist of relevant training samples and concatenated repeated field-unknown ones (data from the  $(N+1)$ -th field in the Eq. (4.31)), as depicted briefly in Fig. 4.5. Algorithm 5 summarizes the overall T-SNT optimization procedures for the F-SVC model.

#### 4.4.2 Transductive F-SVR Formulation

For the field regression occasion of the F-SVR model,  $Z_0 = \{\mathbf{z}_1^0, \mathbf{z}_2^0, \dots, \mathbf{z}_n^0\}$  is noted as the new-coming pattern group occurring with an unknown field from the training set. Most of the transductive F-SVR learning for the unseen style examples are consistent with that of the F-SVC model, as demonstrated in Section 4.4.1. Nevertheless, the fact that there is no probability estimation in the SVR formulation [37]. There is no way to produce multiple regression fitting results with corresponding probability estimation hereby. Consequently, the transductive F-SVR is designed to be fulfilled with only the involvement of the current estimated regression value of the most updated regressor. They are the only concatenated patterns, as depicted in Fig. 4.6. Different from the Transductive F-SVC in Section 4.4.1, the T-SNT in the F-SVR model is learned by the following Eq. (4.32) without the probability involvement.

$$\begin{aligned}
& \min_{A_i \in \mathbb{R}^{d \times d}} \sum_{j=1}^{L_0} (\xi_j^0 + \xi_j^{0*}) + t \|A_0^T - I\|_F^2 \\
& \text{s.t. } y_0 - \mathbf{w}^T A_0^T \mathbf{z}_j^0 - b \leq \epsilon + \xi_j^0, \quad \mathbf{w}^T A_0^T \mathbf{z}_j^0 + b - y_0 \leq \epsilon + \xi_j^{0*} \\
& \quad \xi_j^0 \geq 0, \quad \xi_j^{0*} \geq 0, \quad \forall j = 1, \dots, L_0
\end{aligned} \tag{4.32}$$

Simultaneously, the predictor is learned by the following formulation:

$$\begin{aligned}
& \min_{\substack{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ \{A_i \in \mathbb{R}^{d \times d}\}}} \frac{1}{2} (\mathbf{w}^T \mathbf{w}) + c \sum_{i=1, j=1}^{N+1, L_1} (\xi_j^i + \xi_j^{i*}) \\
& \text{s.t. } y_i - \mathbf{w}^T A_i^T \mathbf{z}_j^i - b \leq \epsilon + \xi_j^i, \quad \xi_j^i \geq 0, \\
& \quad \mathbf{w}^T A_i^T \mathbf{z}_j^i + b - y_i \leq \epsilon + \xi_j^{i*}, \quad \xi_j^{i*} \geq 0 \\
& \quad \forall i = 1, \dots, N+1, \quad \forall j = 1, \dots, L_i
\end{aligned} \tag{4.33}$$

---

**Algorithm 5** Kernelized F-SVC T-SNT alternative learning.

---

**Require:** Training field-pattern  $\{F_i\} = \{X_i, Y_i\}$ , ( $i = 1, 2, \dots, N$ ) as Definition 4.2.1;  
**Require:** Final updated kernel  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$  from the trained F-SVC model;  
**Require:** Unknown style patterns  $X_0$ ;  
**Ensure:** Final updated kernel for the unknown style patterns:  $\tilde{K}_0(\mathbf{x}_{i_1}^0, \mathbf{x}_{i_2}^0)$   
**Parameter (SVC):** cost  $c$ , Gaussian Kernel (GK) width  $\gamma$ ;  
**Parameter (Style Normalization):** style-normalization tradeoff  $t$ ;  
**Initialization:** Unknown style label initial estimation ( $Y_0$ ) with corresponding probabilities from  $\tilde{K}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$  by TPR or VPR;  
**Initialization:** Data with  $k$  probabilities initial concatenation according to Fig. 4.5, producing  $\{F_i\}^{concat} = X_i^{concat}, Y_i^{concat}$ , ( $i = 1, 2, \dots, N, N + 1$ );  
**Initialization:**  $convergence = FALSE$   
1: **while**  $convergence == FALSE$  **do**  
2:     Concatenated kernel build:  $K_{concat}(\mathbf{x}_{i_1}^{concat}, \mathbf{x}_{i_2}^{concat}) = \phi^T(\mathbf{x}_{i_1}^{concat})\phi(\mathbf{x}_{i_2}^{concat})$   
3:     Kernel update with Eq. (4.19) on the concatenated kernel, obtaining  $\tilde{K}^{concat}(\mathbf{x}_{j_1}^{i_1^{concat}}, \mathbf{x}_{j_2}^{i_2^{concat}})$ ;  
4:     Classifier learning on  $\tilde{K}^{concat}(\mathbf{x}_{j_1}^{i_1^{concat}}, \mathbf{x}_{j_2}^{i_2^{concat}})$  (kernelized SVC [6] training with Eq. (4.31));  
5:     Calculate the kernelized SVC objective value as Eq. (4.25);  
6:     Determine the convergence property from the objective calculations;  
7:     **if**  $convergence == TRUE$  **then**  
8:         break this while loop;  
9:     **else**  
10:         Unknown style label re-estimation ( $Y_0$ ) with corresponding probabilities with the most updated kernel by TPR or VPR:  $\tilde{K}^{concat}(\mathbf{x}_{j_1}^{i_1^{concat}}, \mathbf{x}_{j_2}^{i_2^{concat}})$ ;  
11:         Data and probabilities re-concatenation according to Fig. 4.5, producing  $\{F_i\}^{concat} = X_i^{concat}, Y_i^{concat}$ , ( $i = 1, 2, \dots, N, N + 1$ );  
12:     **end if**  
13: **end while**

---

Be noted that the  $(N + 1)$ -th style represents patterns the unknown style. It follows the similar idea of the T-SNT learning strategy of the F-SVC model by attaching them at the end of the training examples. The T-SNT learning pipeline for the F-SVR model is demonstrated in Algorithm 6.

## 4.5 Statistical Performance Evaluation

The proposed F-SVM model is evaluated and accessed extensively by comparing the performance with various relevant frameworks introduced in the research literature for both the field classification (F-SVC) and the field regression (F-SVR) tasks. Multiple benchmark datasets are involved in both models. They will be detailedly demonstrated in this section. All the associated parameters (e.g.,  $c$ ,  $t$ , and  $\gamma$ , width parameter in RBF kernel (Gaussian kernel, GK), and  $\epsilon$  for the F-SVR or SVR models [37]) are searched with the grid-search strategy in the F-SVC, the F-SVR models, and other comparing baselines to get the best performance reported on the involved datasets.

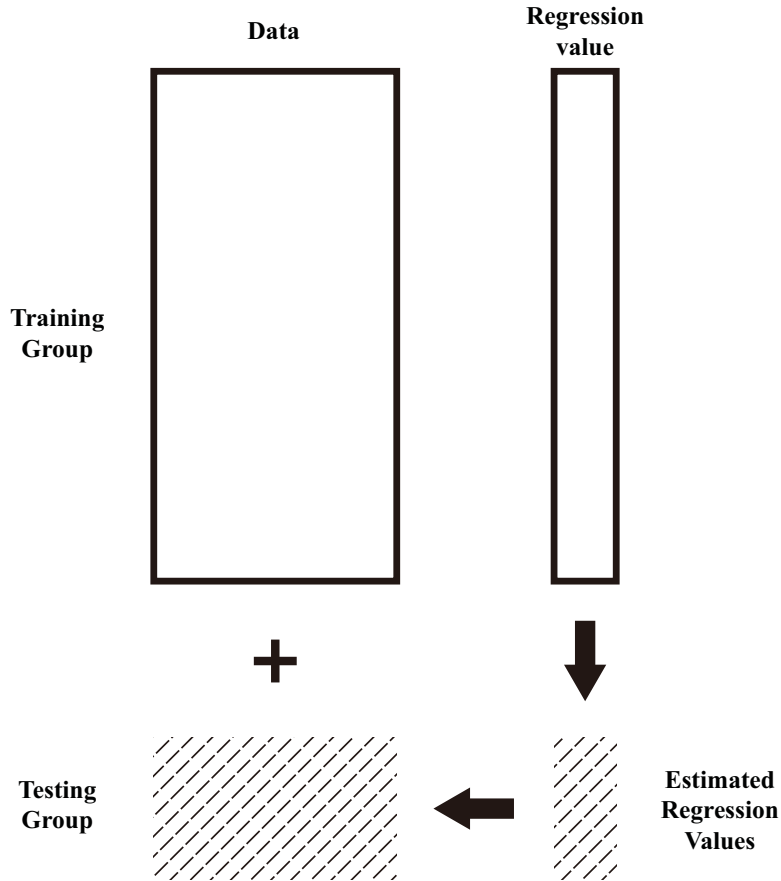


Fig. 4.6: F-SVR testing sample concatenation scheme.

#### 4.5.1 Performance on the F-SVC Model

Three benchmark datasets are introduced to evaluate the performance of the F-SVC algorithm following the experimental setting as described in [29]. Identical information is used for all the comparing methods described in following paragraphs with the proposed F-SVC model. It means that only the field (group) information as the style information is considered for both field and non-field tests. These sets include the face recognition across different head poses (yaw angles) [119] on the Point' 04 database (the Face Data), the speech recognition across multiple speakers [120] on the Connectionist Bench Database (the Speech Data), and the Chinese handwriting character recognition across diverse writers [121] on the CASIA offline database (the HW Data).

It is extensively compared with several other field classification models including two individual cases of the F-BM [29], the Field Nearest Class Mean (F-NCM) classifier [29] for the Face and the Speech Data. The Modified Quadratic Discriminant Function (MQDF) [54, 53, 139] and Field Quadratic Discriminant Function (F-QDF) [29] are involved for the HW data. Additionally, the SMM [19], the BM [20] and the MR-MTL model [30] are also involved for all the three sets as baselines as well.

Note that the proposed F-SVC model utilizes the identical information as those field and MTL models. For each set (representing one specific classification task), some of the distinct and state-of-the-art classifiers are also implemented, including the Fisher Discriminant Analysis (FDA) [106] in the Face Data, Multilayer Perceptron (MP) [136], RBF

---

**Algorithm 6** Kernelized F-SVR T-SNT alternative learning.

---

**Require:** Training field-pattern  $\{F_i\} = \{Z_i, Y_i\}$ , ( $i = 1, 2, \dots, N$ ) as Definition 4.2.2;  
**Require:** Final updated kernel  $\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2})$  from the trained F-SVR model;  
**Require:** Unknown style patterns  $Z_0$ ;  
**Ensure:** Final updated kernel for the unknown style patterns:  $\tilde{K}_0(\mathbf{z}_{i_1}^0, \mathbf{z}_{i_2}^0)$   
**Parameter (SVR):** cost  $c$ , Gaussian Kernel (GK) width  $\gamma$ ,  $\epsilon$ -tolerance  $\epsilon$ ;  
**Parameter (Style Normalization):** style-normalization tradeoff  $t$ ;  
**Initialization:** Unknown style regression value initial estimation ( $Y_0$ ) from  $\tilde{K}(\mathbf{z}_{j_1}^{i_1}, \mathbf{z}_{j_2}^{i_2})$  by TPR or APR;  
**Initialization:** Data initial concatenation according to Fig. 4.6, producing  $\{F_i\}^{concat} = Z_i^{concat}, Y_i^{concat}$ , ( $i = 1, 2, \dots, N, N + 1$ );  
**Initialization:**  $convergence = FALSE$   
1: **while**  $convergence == FALSE$  **do**  
2: Concatenated kernel build:  $K_{concat}(\mathbf{z}_{i_1}^{concat}, \mathbf{z}_{i_2}^{concat}) = \phi^T(\mathbf{z}_{i_1}^{concat})\phi(\mathbf{z}_{i_2}^{concat})$   
3: Kernel update with Eq. (4.22) on the concatenated kernel, obtaining  $\tilde{K}^{concat}(\mathbf{z}_{j_1}^{i_1^{concat}}, \mathbf{z}_{j_2}^{i_2^{concat}})$ ;  
4: Predictor learning on  $\tilde{K}^{concat}(\mathbf{z}_{j_1}^{i_1^{concat}}, \mathbf{z}_{j_2}^{i_2^{concat}})$  (kernelized SVR [37] training with Eq. (4.33));  
5: Calculate the kernelized SVR objective value as Eq. (4.28);  
6: Determine the convergence property from the objective calculations;  
7: **if**  $convergence == TRUE$  **then**  
8: break this while loop;  
9: **else**  
10: Unknown style regression value re-estimation ( $Y_0$ ) with corresponding probabilities with the most updated kernel by TPR or APR:  $\tilde{K}^{concat}(\mathbf{z}_{j_1}^{i_1^{concat}}, \mathbf{z}_{j_2}^{i_2^{concat}})$ ;  
11: Data and probabilities re-concatenation according to Fig. 4.6, producing  $\{F_i\}^{concat} = X_i^{concat}, Z_i^{concat}$ , ( $i = 1, 2, \dots, N, N + 1$ );  
12: **end if**  
13: **end while**

---

Network (RBF-N) [137], 1-nearest Neighbour (1-NN) [107], and Discriminant Adaptive Nearest Neighbor (DANN) [138] in the Speech Data.

For the sake of a better demonstration of the comparison, two deep neural network models are also involved in the face and the Speech Data. They are the VGG-FaceNet [41] and the AlexNet [1] respectively. Also, the Rectified Linear Units (ReLUs) are utilized as the nonlinear activation function in the network [141, 142, 143]. No deep learning model is involved in the Speech Data since there exists less neural network framework for this task in the literature. The drop out tricks [144, 145] are employed for both models since they are initially designed for large dataset, e.g., Imagenet [90] and the Labelled Faces in the Wild and YouTube face dataset [146]. It has been pointed out there will be the over-fitting problem occurring if such networks are trained on relatively small sets [147, 148, 149] without the drop-out trick. The batch normalization [94] is also implemented to stabilize the training process with faster and more stabilized convergence. The performance of the three datasets with F-SVC and other relevant baselines mentioned above is summarized in Table 4.1.



Table 4.1: F-SVC performance summary on the Face [119], Speech [120] and the HW Data [121] : N/A represents that the specific baseline is not compared on the given set.

Method	Recognition Rate (%)		
	Face Data [119]	Speech Data [120]	HW Data [121]
MP [136]	N/A	51.00%	N/A
RBF-N [137]	N/A	53.00% (Field)	N/A
[107]	N/A	56.00%	N/A
DANN [138]	N/A	61.70%	N/A
FDA [106]	69.33%	N/A	N/A
SMM [19]	73.33% (Field)	55.85% (Field)	N/A
BM [20]	60.00%	77.30%	N/A
CNN [54, 53, 139]	N/A	N/A	94.44%
F-QDF [29]	N/A	N/A	95.49% (FPR)
NCM [140]	60.00%	50.65%	94.44%
CNN	90.67% (Vgg-Face-Net [41, 7])	N/A	97.22% (AlexNet [1])
F-NCM [29]	78.67% (FTPR)	78.35% (FTPR)	95.49% (FPR)
SVC [6]	LK	84.00%	96.53%
	GK	85.33%	96.53%
MR-MTL [30]	LK	85.33%	96.87%
	GK	85.33%	97.22%
F-SVC (Singlet)	LK	85.33% (TPR / VPR)	96.87% (TPR / VPR)
	GK	88.00% (TPR / VPR)	96.53% (TPR / VPR)
F-SVC (Field)	LK	<b>100.00%</b> (FTPR)	<b>97.92%</b> (FPR)
	GK	<b>100.00%</b> (FTPR)	96.87% (FTPR)

### Face Classification across Head Poses

There are in total 15 people involved in the Point' 04 Database [119]<sup>7</sup> (the Face Data) with each one only the zero yaw angle images being chosen. 13 different yaw angles within the range of  $[-90^\circ, +90^\circ]$  with an interval of  $15^\circ$  are evaluated. All images are initially cropped and resized with  $48 \times 36$  pixels. The 1728-d feature (after the  $48 \times 48$  image vectorization) is further compressed to 14 with the FDA [106] model for the NCM [140], and 100 by the Principal Component Analysis (PCA) [150] for other classifiers by following the experimental setting described in [29].

Each head pose (yaw angle) is regarded as a field. The classification is conducted based on different individuals, as examples displayed in Fig. 4.14. Images from the first eight poses are involved in the training set, while the remaining five are evaluated for the generalization performance. Note that test fields are completely different from training

<sup>7</sup>The specific Face Data used in this research topic can be downloaded via the website: <http://download.premilab.com/FaceData-Reading.zip>

fields. It means that the styles of all the testing fields are unseen in training. The FTPR introduced in Section 4.4 will be at this moment utilized.

As seen in column titled *Face Data* in Table 4.1, the FDA [106] model obtains the recognition performance of 69.33%. The best performance for non-field information utilized classifier is given with the GK SVC [6] at 85.33%. The MR-MTL model [30] brings no improvement due to the assumption that all tasks are related. Because of the fixed mixture components number, the SMM [19] seems to be ineffective in transferring the trained information to the new-coming field properly. As a result of the small field-length, the performance of the BM [20] is also restricted. The F-NCM [29] generates a better style transfer performance with the improvement of 78.67%. The CNN-based VGG-Face-Net [41, 7] model claimed the best performance among these non-field approaches with the performance improved to 90.67%. In comparison, the proposed F-SVC (in the linear (LK) and GK) demonstrates the best in both singlet and field classification (with the FTPR applied), significantly superior to all the other baselines. In particular, the field classification (with the proposed self-training strategy as offered in Section 4.4.1) achieves zero error rate in both LK and GK. Such achievement is hardly produced in previous studies of the relevant literature. The performance of the F-SVC model on the face data is also depicted in Fig. 4.7 for better visualization.

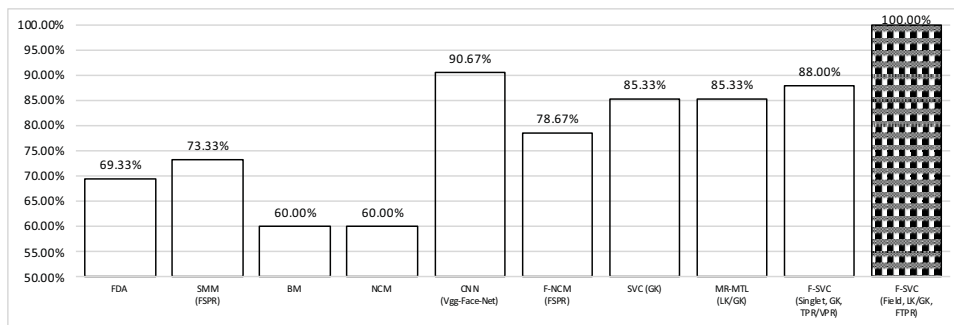


Fig. 4.7: F-SVC performance on the Point' 04 data (the Face Data) [119]: bars with pattern filled represents the model achieving the best performance.

## Speech Classification across Speakers

The Connectionist Bench Database (the Speech Data)<sup>8</sup> [120] consists of 11 different vowels uttered by 15 speakers of British English. For each vowel pronounced by each speaker, in total six samples are collected. The 10 log-area parameters are extracted from each sample. It is one kind of standard vocal tract representations computed from a linear predictive coding analysis of the digitized speech data. The identical experimental setting is followed as described in [20, 29], where data from the first 1-8 speakers are used for training, while the rest for testing. Each speaker with a specific native accent is regarded as a field. The raining fields are once again different from the testing fields in this way. It results in that the FTPR 4.4.1 will once again be assessed.

<sup>8</sup>The dataset were firstly collected by Dr. David Deterding. The data is now available at the website: <http://archive.ics.uci.edu/ml>.

The experimental result is summarized in column titled *Speech Data* in Table 4.1 and Fig. 4.8. When no field information is utilized, the GK SVC [6] model gives the best performance with the recognition rate of 72.73%. The MR-MTL model [30] only brings little improvement with the lowest error rate of 26.84%. No improvement is achieved for the singlet classification with F-SVC in the GK, while slight improvement can be observed in the LK. When field information between the training and the testing data is transferred, the GK F-SVC (FSTR) yields the best recognition evaluation result at 81.36%, which still outperforms all the other comparison methods.

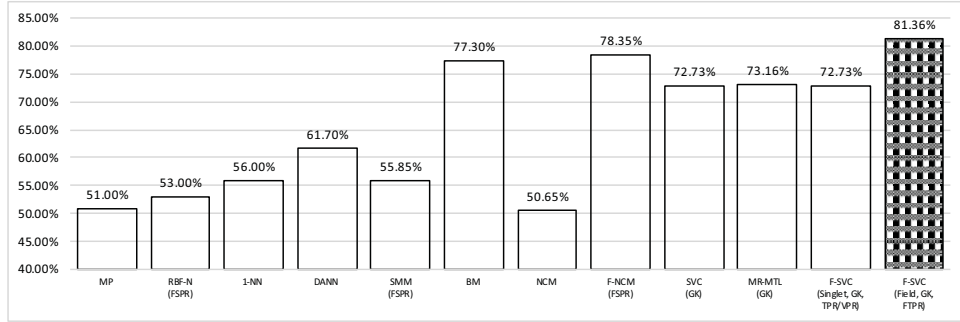


Fig. 4.8: F-SVC performance on the Connectionist Bench database (the Speech Data) [120].

### Chinese Handwriting Character Recognition across Writers

We also exploit the CASIA-OLHWDB data [121] for evaluations (the HW Data). The original database includes 3,755 categories of different Chinese handwriting characters. As described in [29], 100 writers (writer id from no.1101 to no.1200) are involved in this experiment. For simplicity, only the first 30 characters are chosen. Since people are more likely to write texts cursively than isolated characters, the isolated set is selected as the training set. The cursive text set is used for testing. The total numbers of samples are 2,995 for the training set and 288 for the testing set. It needs to be noted that field-pattern from each testing share a particular style, which can also be seen in training field patterns. Hence, the FPR introduced in Section 4.3.2 can be directly used for the prediction.

Same as [29], we extract 512-d 8-direction histogram features combined with the pseudo-2D bi-moment normalization [151] from the dataset. Features are then compressed to 160 by the FDA [106], and further reduced to 50 by the PCA. Due to the relatively large size, the BM [20] and the SMM [19] are difficult to be implemented because of the prohibitive slow optimization issue. They are hence omitted for comparison. On the contrast, the MQDF model [54, 53, 139], a state-of-the-art classifier for handwriting classification, is compared and evaluated. The performance is reported in the column titled *HW Data* in Table 4.1 and Fig. 4.9. For those non-field approaches, the best performance is fulfilled by the deep learning model with the recognition rate at 97.22%. The SVC-based multi-task method also achieves the same performance. However, it is observed that the proposed F-SVC obtains the lowest error rates. In particular, different variations of F-SVC demonstrate the best performance compared with the other remaining algorithms. The greatest rate of 97.92% is observed for the proposed linear F-SVC when field information is used.

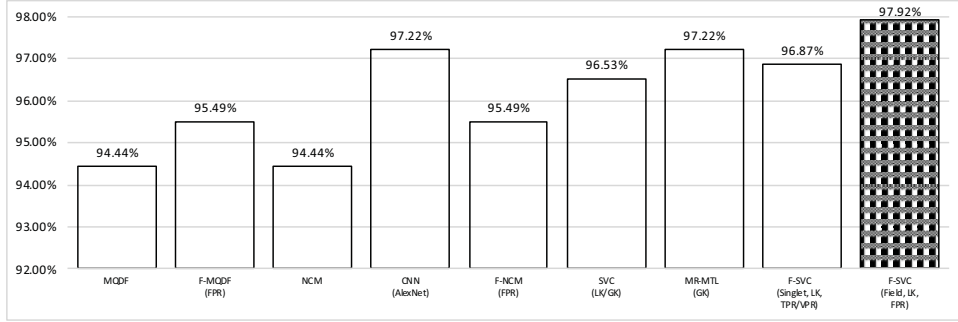


Fig. 4.9: F-SVC performance on the CASIA-OLHWDB [121] database (the HW Data).

## 4.5.2 Performance on the F-SVR Model

Similarly, there are four sets are involved to evaluate the F-SVR framework. They are two artificially generated synthetic datasets (for both linear set and the nonlinear set), the school effectiveness dataset [123] for students' academic performance across different schools, and the computer survey database [124] for different individuals' viewpoints on specific computer products. The synthetic linear data is generated according to the relevant description in [37], while the nonlinear data is created by referring to [44] with a more reasonable setting.<sup>9</sup>

Related models are compared, including the aggregation (Agrgt., where all the patterns are trained together), the Single-task Learning (STL, where each group is trained independently), and the Multi-task Learning (MTL, where groups of examples are trained together by sharing common information). The Ordinary Linear Regression (OLR) [124], the Ridge Regression (RR) [152, 153] and the SVR [37] are evaluated for models of both Agrgt., and the STL. The Multitask Feature Learning [44] (FEAT-MTL) with LK, GK, and the Variable Selection (VS), the Disjoint Group MTL [45] (DG-MTL), the Overlapped Group MTL [32] (OG-MTL), and the SVR-based Mean-regularized MTL [30] (MR-MTL) are evaluated as MTL baselines. The best method is listed for SVR-based and the FEAT-MTL [44] ones. Training and testing sets are separated via two rules, (1) *Overlapped Train / Test, (OTT) Groups*, where styles occur in testing sets are seen in the training phase. It is the standard experimental setting for most of the MTL-based work where the FPR will be used for the F-SVR; (2) *Disjoint Train / Test (DTT) Groups*, in which styles were occurring in the testing phase are unseen. The FTPR will then be implemented. Since no previous MTL-based work focuses on the DTT, only the ADR will be evaluated. The performance of all the listed approaches is measured by both the Rooted Mean Squared Error (RMSE) with the Standard Variance (SD) attached. The SD is depicted by the error bar in each summarized figures in this subsection.

### Synthetic Data

Linear and nonlinear synthetics<sup>10</sup> are evaluated with the field length ( $L_i$ ) randomly set in the range of [15, 50]. Both are given as  $X_i$  without group information, but they are mapped inconsistently. The linear set refers to [37] (Section 4.2.2: Data Generation,

<sup>9</sup>Both the linear and the nonlinear data is available at: <http://www.premilab.com/Downloads.ashx>

<sup>10</sup>Both the synthetic linear and nonlinear data can be downloaded via the website: <http://download.premilab.com/SyntheticData-FSVR.zip>

Synthetic Data  $I_b(R)$ ) with the dimension  $d_l = 10$ , and  $N = 50$  groups. Specifically, the linear data are generated within total  $N = 50$  fields (or tasks). The field length  $L_i$  is a random number in the range of  $[15, 50]$ . The linear data dimension is  $d_l = 10$ . For the  $i$ -th field, two random basis vectors are generated from the multi-variate *i.i.d.* normally distributed vector with mean and standard deviation equal to zero and one respectively. That is to say,  $\mathbf{b}_1^i, \mathbf{b}_2^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  are to be set. These two vectors are put together to span parameters of each group given by  $\mathbf{w}_i = [\mathbf{b}_1^i, \mathbf{b}_2^i] \mathbf{a}_i + \delta_i$ , where  $\mathbf{a}_i$  is a non-negative coefficient vector with entries summing to one ( $\|\mathbf{a}_i\|_1 = 1$ ) and  $\delta_i \sim \mathcal{N}(\mathbf{0}, 0.1 \times \mathbf{I})$ . The data of the  $i$ -th field  $X_i = [\mathbf{x}_1^i, \dots, \mathbf{x}_{L_i}^i]$  are generated by  $\mathbf{x}_j^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Then the regression value is calculated by  $y_j^i = \mathbf{w}_i^T \mathbf{x}_j^i + \epsilon_j^i$ , where  $\epsilon_j^i \sim \mathcal{N}(0, 0.1)$ .

A more reasonable nonlinear mapping is adopted ( $d_n = 25$ ) than that of [44]. The covariance matrix ( $X_i^T X_i$ ) is calculated for each linear group. The nonlinear mapping is  $\phi(\mathbf{x}) = (\mathbf{x}_{[r_1]} \cdot \mathbf{x}_{[c_1]}, \dots, \mathbf{x}_{[r_{d_n}]} \cdot \mathbf{x}_{[c_{d_n}]})^T$ , where  $x_{[k]}$ , selected by  $d_n$  least values in  $X_i^T X_i$ , indicates the  $\mathbf{x}$  value on the  $k$ -th dimension.

Table 4.2: F-SVR performance on synthetic linear datasets.

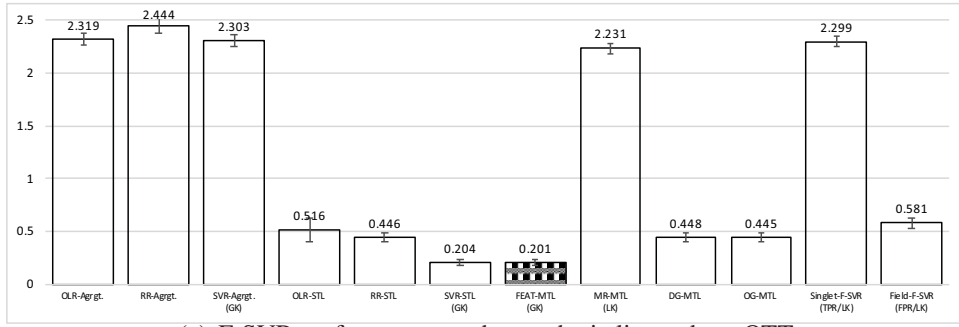
Method		OTT		DTT	
		RMSE $\pm$ SD	$R^2 \pm$ SD(%)	RMSE $\pm$ SD	$R^2 \pm$ SD (%)
Agrgt.	OLR [124]	2.319 $\pm$ 0.052	-44.165 $\pm$ 23.361	2.439 $\pm$ 0.158	-14.234 $\pm$ 8.168
	RR [152, 153]	2.444 $\pm$ 0.067	1.077 $\pm$ 0.927	2.453 $\pm$ 0.167	-0.168 $\pm$ 0.144
	SVR [37]	2.303 $\pm$ 0.055(GK)	-38.536 $\pm$ 23.942(GK)	2.359 $\pm$ 0.163(GK)	-4.118 $\pm$ 1.723(GK)
STL	OLR [124]	0.516 $\pm$ 0.112	74.832 $\pm$ 31.540	2.377 $\pm$ 0.162	-6.286 $\pm$ 4.642
	RR [152, 153]	0.446 $\pm$ 0.043	90.930 $\pm$ 2.359	2.359 $\pm$ 0.163	-4.103 $\pm$ 1.913
	SVR [37]	0.204 $\pm$ 0.025(GK)	<b>97.651 <math>\pm</math> 0.900(GK)</b>	2.360 $\pm$ 0.163(GK)	-4.176 $\pm$ 1.812(GK)
MTL	FEAT-MTL [44]	<b>0.201 <math>\pm</math> 0.027 (GK)</b>	97.546 $\pm$ 1.053(GK)	2.359 $\pm$ 0.162 (GK)	-4.090 $\pm$ 1.982(GK)
	MR-MTL [30]	2.231 $\pm$ 0.052 (LK)	-30.720 $\pm$ 22.226(LK)	2.359 $\pm$ 0.163 (GK)	-4.118 $\pm$ 1.723(GK)
	DG-MTL [45]	0.448 $\pm$ 0.042	91.105 $\pm$ 2.124	2.359 $\pm$ 0.163	-4.103 $\pm$ 1.913
	OG-MTL	0.445 $\pm$ 0.042	91.132 $\pm$ 2.195	2.359 $\pm$ 0.163	-4.103 $\pm$ 1.913
F-SVR	Singlet	2.299 $\pm$ 0.051(TPR/LK)	-36.215 $\pm$ 22.177(APR/LK)	<b>2.358 <math>\pm</math> 0.162(TPR/LK)</b>	<b>-3.992 <math>\pm</math> 1.671(TPR/LK)</b>
	Field	0.581 $\pm$ 0.052(FPR/LK)	88.246 $\pm$ 2.169(FPR/LK)	<b>2.280 <math>\pm</math> 0.149(FTPR/GK)</b>	<b>5.622 <math>\pm</math> 5.157(FTPR/GK)</b>

Table 4.3: F-SVR performance on synthetic nonlinear datasets.

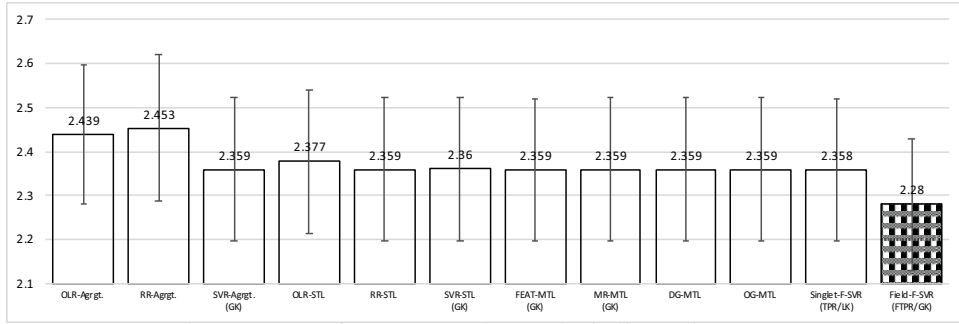
Method		OTT		DTT	
		RMSE $\pm$ SD	$R^2 \pm$ SD(%)	RMSE $\pm$ SD	$R^2 \pm$ SD (%)
Agrgt.	OLR [124]	3.630 $\pm$ 0.196	-22.908 $\pm$ 5.142	3.742 $\pm$ 0.266	-6.967 $\pm$ 2.185
	RR [152, 153]	3.897 $\pm$ 0.194	-0.257 $\pm$ 0.465	3.824 $\pm$ 0.282	-0.398 $\pm$ 0.534
	SVR [37]	3.612 $\pm$ 0.193 (GK)	-21.028 $\pm$ 4.288 (LK)	3.707 $\pm$ 0.256 (LK/GK)	-4.825 $\pm$ 1.670 (LK/GK)
STL	OLR [124]	6.927 $\pm$ 1.146	-1136.873 $\pm$ 756.096	<i>3nonlinear</i> 743 $\pm$ 0.265	-7.036 $\pm$ 1.604
	RR [152, 153]	3.612 $\pm$ 0.195	-20.841 $\pm$ 4.529	3.709 $\pm$ 0.258	-4.884 $\pm$ 1.999
	SVR [37]	3.561 $\pm$ 0.177 (GK)	23.165 $\pm$ 9.293 (GK)	3.708 $\pm$ 0.256 (GK)	-4.856 $\pm$ 1.716 (LK)
MTL	FEAT-MTL [44]	3.596 $\pm$ 0.195 (GK)	-19.753 $\pm$ 4.597 (GK)	3.709 $\pm$ 0.258 (GK)	-4.878 $\pm$ 1.996 (GK)
	MR-MTL [30]	3.609 $\pm$ 0.193 (GK)	-20.766 $\pm$ 4.272 (GK)	3.707 $\pm$ 0.256 (LK)	-4.823 $\pm$ 1.665 (LK)
	DG-MTL [45]	3.612 $\pm$ 0.195	-20.841 $\pm$ 4.529	3.709 $\pm$ 0.258	-4.884 $\pm$ 1.999
	OG-MTL [32]	3.612 $\pm$ 0.195	-20.841 $\pm$ 4.527	3.709 $\pm$ 0.258	-4.884 $\pm$ 1.999
F-SVR	Singlet	<b>3.610 <math>\pm</math> 0.193 (APR/LK)</b>	<b>-20.580 <math>\pm</math> 4.391 (APR/LK)</b>	<b>3.705 <math>\pm</math> 0.256 (TPR/LK)</b>	<b>-4.666 <math>\pm</math> 1.749 (TPR/LK)</b>
	Field	<b>3.444 <math>\pm</math> 0.196(FPR/GK)</b>	<b>7.998 <math>\pm</math> 2.848(FPR/GK)</b>	<b>3.674 <math>\pm</math> 0.263(FTPR/GK)</b>	<b>1.990 <math>\pm</math> 2.383(FTPR/GK)</b>

For the OTT case, three quarters randomly selected samples from the whole ones are put in the training set, while remains for the test in the OTT case. The field length is number that over five. Same setting is kept for the DTT case. 12 random splits are also made for both OTT and DTT. Performance is normalized over the field length, listed in Table 4.2 and Table 4.3 respectively for synthetic linear data and synthetic nonlinear data. Additionally, Fig. 4.10 and Fig. 4.11 also illustrate the performance measured by RMSE.

The proposed F-SVR is ineffective in the linear OTT case. The FEAT-MTL [44] (GK) and the STL SVR [37] (GK) achieve the lowest error. Other STL-based methods all perform comparably. MTL approaches (except MR-MTL [30]) improve the performance further. The reason is mostly due to that the linear relation between data and regression values is not complicated. With abundant training examples available (around  $15 \times 75\% \approx$

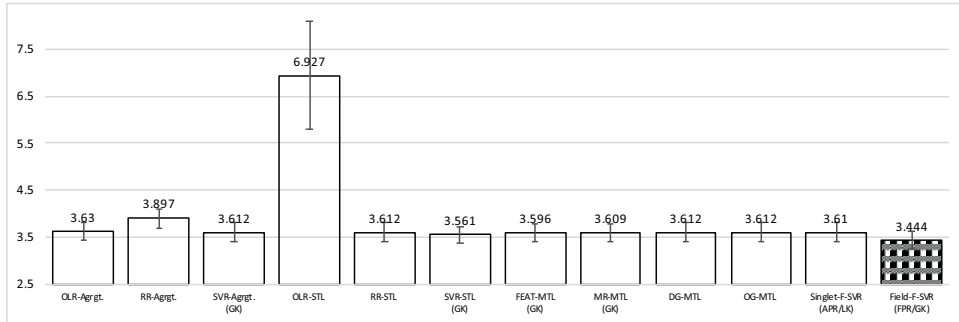


(a) F-SVR performance on the synthetic linear data: OTT

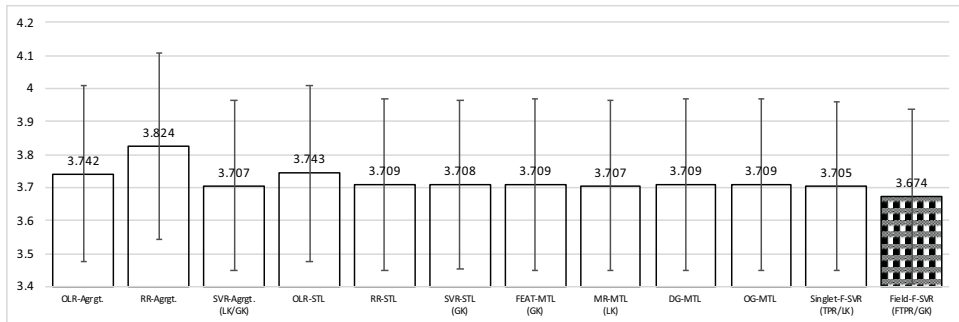


(b) F-SVR performance on the synthetic linear data: DTT

Fig. 4.10: F-SVR performance on the synthetic linear data: bars with pattern filled represents the model achieving the best performance (lowest error).



(a) F-SVR performance on the synthetic nonlinear data: OTT



(b) F-SVR performance on the synthetic nonlinear data: DTT

Fig. 4.11: F-SVR performance on the synthetic nonlinear data.

11 to  $50 \times 75\% \approx 38$ ) for each STL-based regressor, a low error can be attained. The further improvement brought by the MTL-based models are achieved by applying task-

Table 4.4: F-SVR performance on the School Effectiveness data [123].

Method	OTT		DTT		
	RMSE±SD	R <sup>2</sup> ±SD(%)	RMSE±SD	R <sup>2</sup> ±SD (%)	
Agrgt.	OLR [124]	10.203 ± 0.081	12.507 ± 2.824	10.395 ± 0.122	21.216 ± 3.237
	RR [152, 153]	10.136 ± 0.119	16.439 ± 2.037	13.839 ± 0.356	-39.909 ± 9.547
	SVR [37]	10.128 ± 0.088 (GK)	14.256 ± 2.837 (GK)	10.384 ± 0.111 (GK)	21.481 ± 2.929 (GK)
STL	OLR [124]	10.545 ± 0.136	5.710 ± 4.415	12.321 ± 0.534	-11.150 ± 8.597
	RR [152, 153]	10.368 ± 0.116	34.230 ± 1.150	10.508 ± 0.157	<b>33.014±0.965</b>
	SVR [37]	10.176 ± 0.114 (GK)	15.763 ± 2.308 (GK)	10.685 ± 0.172 (GK)	16.799 ± 4.090 (GK)
MTL	FEAT-MTL [44]	9.904 ± 0.112 (LK)	20.036 ± 2.123 (LK)	10.605 ± 0.165 (GK)	17.986 ± 2.603 (GK)
	MR-MTL [30]	10.174 ± 0.096 (LK)	14.294 ± 3.051 (LK)	<b>10.377±0.135 (GK)</b>	21.432 ± 3.307 (GK)
	DG-MTL [45]	9.902 ± 0.114	20.561 ± 1.897	10.843 ± 0.202	14.288 ± 4.805
	OG-MTL [32]	10.140 ± 0.121	16.567 ± 1.951	13.853 ± 0.357	-40.359 ± 9.661
F-SVR	Singlet	<b>10.128±0.088(TPR/GK)</b>	<b>14.255±2.837(APR/GK)</b>	10.384±0.111(TPR/GK)	21.495±2.936(APR/GK)
	Field	<b>9.743±0.107(FPR/GK)</b>	<b>39.683±1.087(FPR/GK)</b>	<b>10.372±0.111(FTPR/GK)</b>	<b>33.103±1.514(FTPR/GK)</b>

consistency. However, because of enormous inconsistency among fields, the performance of the Agrgt. SVR [37] is much restricted, limiting its inherit models including the MR-MTL [30] and F-SVR.

In the DTT case, the performance of both STL and MTL approaches significantly dropped. The proposed F-SVR model (singlet LK) generates a slight lower error than those MTL-based algorithms. When conducting field transferring procedures, the performance is further improved with the RMSE at about 2.280 (GK).

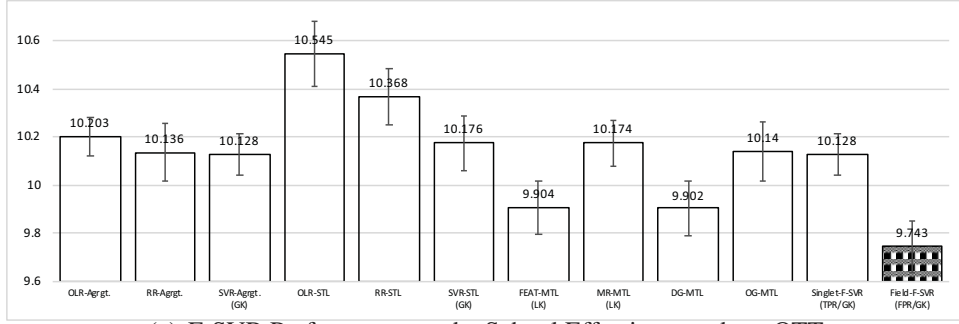
The F-SVR (Singlet) achieves the best result for the OTT case comparing with other Agrgt., STL or MTL methods on the nonlinear data. The F-SVR (Field GK) further achieves the lowest error rate of 3.444. For the DTT case, the F-SVR singlet with LK gives the best regression performance with the RMSE of 3.70. It is a bit lower than that of those MTL approaches. The field transferring scheme gives a lower error rate at 3.674.

In summary, the proposed F-SVR achieves better performance with complicated data relationship. In the OTT case, it performs better than other MTL methods. In the DTT, the F-SVR singlet scheme (discarding testing field information) only brings slight improvement. However, the style transferring plan achieves further promotion with the T-SNT learned.

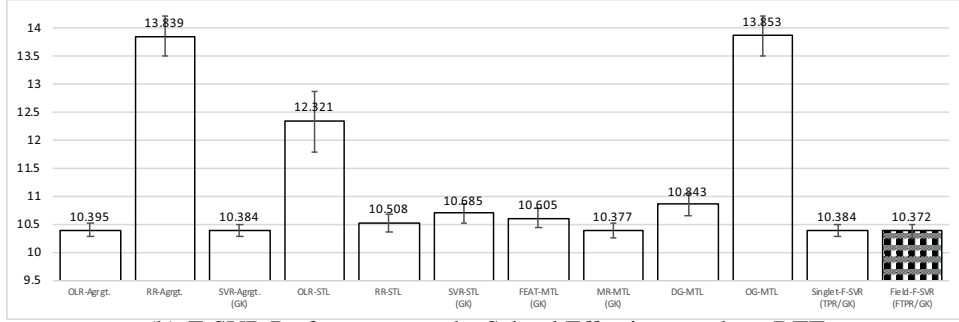
### School Effectiveness Data

This set includes 15362 students’ exam scores (100 full marks, regression values) of 139 schools (styles). Each input consists of four school and three student attributes, resulting in a 27-d data representation [44]. The group information is denoted as  $i$  (for the  $i$ -th school  $j$ ) in  $Z_i$ . Training and test data are followed with [44] for OTT. The dataset is also divided as described in Section 4.5.2 for the DTT with ten random splits. The RMSE is normalized for each group, as demonstrated in Table 4.4 and visualized in Fig. 4.12 for both the OTT and the DTT cases.

The proposed F-SVR still outperforms others in OTT case with RMSE of 9.743. However, the singlet setting underperforms several other MTL-based or STL-based approaches for the DTT one. The MR-MTL [30] model generates the lowest error rate with the RMSE of 10.377, while the STL RR explains the highest variance of 33.014%. However, the style transferring is still able to achieve the best performance with RMSE 10.372.



(a) F-SVR Performance on the School Effectiveness data: OTT



(b) F-SVR Performance on the School Effectiveness data: DTT

Fig. 4.12: F-SVR performance on the School Effectiveness data [123].

## Computer Survey Data

This set is about customers' ratings on different computers [124] from a survey rating (0 to 10 integers, regression values) of 180 people (styles) on 20 products.<sup>11</sup> We follow the 14-d data format including 13 binary attributes describing technical properties and one bias.<sup>12</sup> The input of this set is  $X_i$  for the  $i$ -th customer without style embedded. Apparently, each input would be related to multiple ratings because of customers, so we omit the Agrgt. experiment. Both OTT and DTT cases are generated as described in Section 4.5.2 with 12 randomly splits. No performance normalization is applied. Results are summarized in Table 4.5 and Fig. 4.12.

The proposed F-SVR model achieves the best performance for the OTT with both the RMSE, which are 1.807 and 58.347% respectively. The singlet F-SVR brings tiny improvement in the DTT, while the transferring scheme achieves much better result with RMSE of 2.101. The summarized experimental results demonstrate that the proposed F-SVR model is effective in improving the regression performance in both cases in this computer survey dataset, outperforming all the others.

## 4.6 Visualized Evaluation of Field Normalization

In this section, some experimental performance will be further interrogated in a visualized manner. Several detailed analysis based on the experimental performance as described in

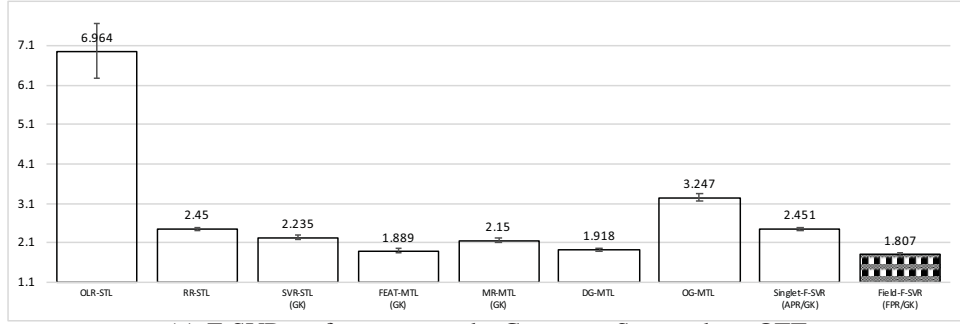
<sup>11</sup>We thank Prof. Peter LENK presented the original dataset.

<sup>12</sup>We thank Dr. Andrew MCDONALD and Dr. Jailei WANG sent us the experimental data with the appropriate 14-d format.

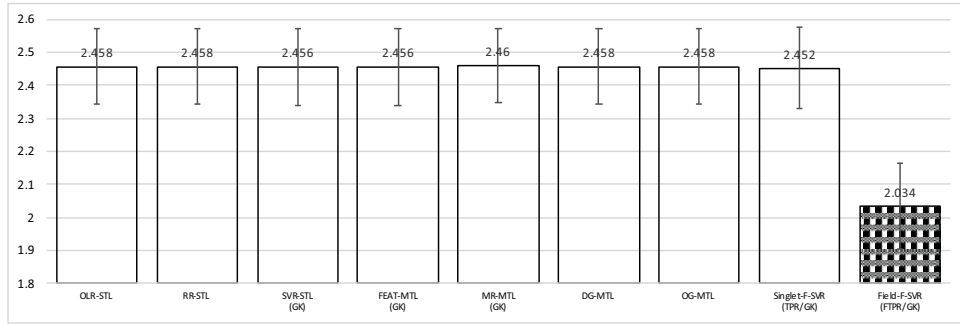


Table 4.5: F-SVR performance on the Computer Survey dataset [124].

Method	OTT		DTT		
	RMSE±SD	$R^2$ ±SD(%)	RMSE±SD	$R^2$ ±SD (%)	
STL	OLR [124]	6.964 ± 0.704	-524.985 ± 143.635	2.458 ± 0.1156	22.655 ± 5.172
	RR [152, 153]	2.450 ± 0.044	23.465 ± 2.657	2.458 ± 0.114	22.655 ± 5.062
	SVR [37]	2.235 ± 0.061(GK)	36.288 ± 3.162(GK)	2.456 ± 0.116(GK)	22.812 ± 5.115(GK)
MTL	FEAT-MTL [44]	1.889 ± 0.050(GK)	54.4671 ± 2.235(GK)	2.456 ± 0.116(GK)	22.804 ± 5.126(GK)
	MR-MTL [30]	2.150 ± 0.052(GK)	41.029 ± 2.372(GK)	2.460 ± 0.113(GK)	22.528 ± 5.147(GK)
	DG-MTL [45]	1.918 ± 0.054	53.075 ± 2.311	2.458 ± 0.114	22.655 ± 5.075
	OG-MTL [32]	3.247 ± 0.092	-34.517 ± 6.883	2.458 ± 0.115	22.656 ± 5.143
F-SVR	Singlet	<b>2.451±0.047(APR/GK)</b>	<b>23.406±2.764(TPR/LK)</b>	<b>2.452±0.123(TPR/GK)</b>	<b>23.025±5.546(TPR/GK)</b>
	Field	<b>1.807±0.042(FPR/GK)</b>	<b>58.347±1.393(FPR/GK)</b>	<b>2.034±0.131(FTPR/GK)</b>	<b>46.900±6.173(FTPR/GK)</b>



(a) F-SVR performance on the Computer Survey data: OTT



(b) F-SVR performance on the Computer Survey data: DTT

Fig. 4.13: F-SVR performance on the Computer Survey data [124]

Section 4.5 will be demonstrated to further prove the effectiveness of the proposed F-SVM model. Note that only the F-SVC model will be demonstrated in this section. It is quite apparent that the identical conclusion can be extended into the F-SVR model.

#### 4.6.1 Style Normalization with the Linear Kernel

The Point' 04 Database will be detailed demonstrated visually with the performance of the linear F-SVC model on the initial classification task, namely, face classification across yaw poses as described in Section 4.5.1. Inspired by it, another question is raised that what it would be like if the field and the class information of the face dataset are simply alternated, resulting in a reversed task of classification on poses across multiple individuals.<sup>13</sup>

<sup>13</sup>We only examine the field normalization performance visually on this setting. It is the same with the following JAFFE set as described in the following paragraphs.

### Original Task of the Face Data

As described in Section 4.5.1 and seen from Fig. 4.14, images from each column represent a field (a head pose), while ones from each row illustrate one class (one face). Images from Fig. 4.15 and Fig. 4.16 depict the linear F-SVC performance visualization for both the style-normalized patterns  $A_i^T \mathbf{x}_j^i$  and the style information  $(\mathbf{x}_j^i - A_i^T \mathbf{x}_j^i)$  respectively. Apparently, field information is correctly extracted, since there is a strong consistency in common in head pose directions in images from Fig. 4.15. Different pose styles can be observed in the exact order shown in Fig. 4.14. At the same time, for each picture in Fig. 4.16, they are free from head poses. Only the relevant properties related to perform the face classification are kept. These produced style-normalized patterns satisfy the *i.i.d.*

### Reversed Task of the Face Data

The setting of the reversed task is that each people are considered as a field (as seen in each column in Fig. 4.17), while the classification is performed due to different pose angles (as seen in different rows of Fig. 4.17).

From Fig. 4.18, it is clearly seen that images in each column share a common field tendency representing a specific individual (mostly illustrated as the outer facial shape). Simultaneously a clear difference in the class information (head poses) can also be found in pictures from different rows in Fig. 4.19, where images of each row witness less significant difference (the style information is eliminated). It once again proves the effectiveness of the SNT.

### Original Task of the Facial Expression Data

We further examine the SNT performance visually on the JAFFE database [38], a benchmark facial expression set. Initially, it is designed for the facial expression classification, so we will first check the performance when each individual is regarded as a field. With the linear kernel, the classification is performed based on different facial expressions (including anger, disappointment, fear, happiness, nervous, sad and surprise [38]). As seen in Fig. 4.20, images in each row represent a specific class (a specific facial expression), while faces from each column represent a specific field (a specific individual involved). We further evaluate the images in Fig. 4.21 and Fig. 4.22, which are all placed with the exact order as seen in Fig. 4.20. It can be concluded that style-normalized images in each row of Fig. 4.22 are almost the same, representing the field information (individual identities) is normalized. In the meanwhile, those from different rows are indeed representing different facial expressions as the class information. Simultaneously, for images of each column of Fig. 4.21, a definite style information difference can be quickly discovered (the difference on facial shapes representing different individuals, showing the difference in the style, or the field information).

### Reversed Task of the Facial Expression Data

Similarly, we further interrogate the F-SVC model by simply alternating the class and field information again. The class information now is represented as different individuals (as seen on images from different rows of Fig. 4.23). The field information is hereby demonstrated as the facial expressions (as seen on images from different columns of Fig. 4.23).



Fig. 4.14: F-SVC performance visualization of the Point' 04 data (original task) [119]: original images in each row represent a class (a specific individual involved), while those in each column depict a field (a specific head pose).

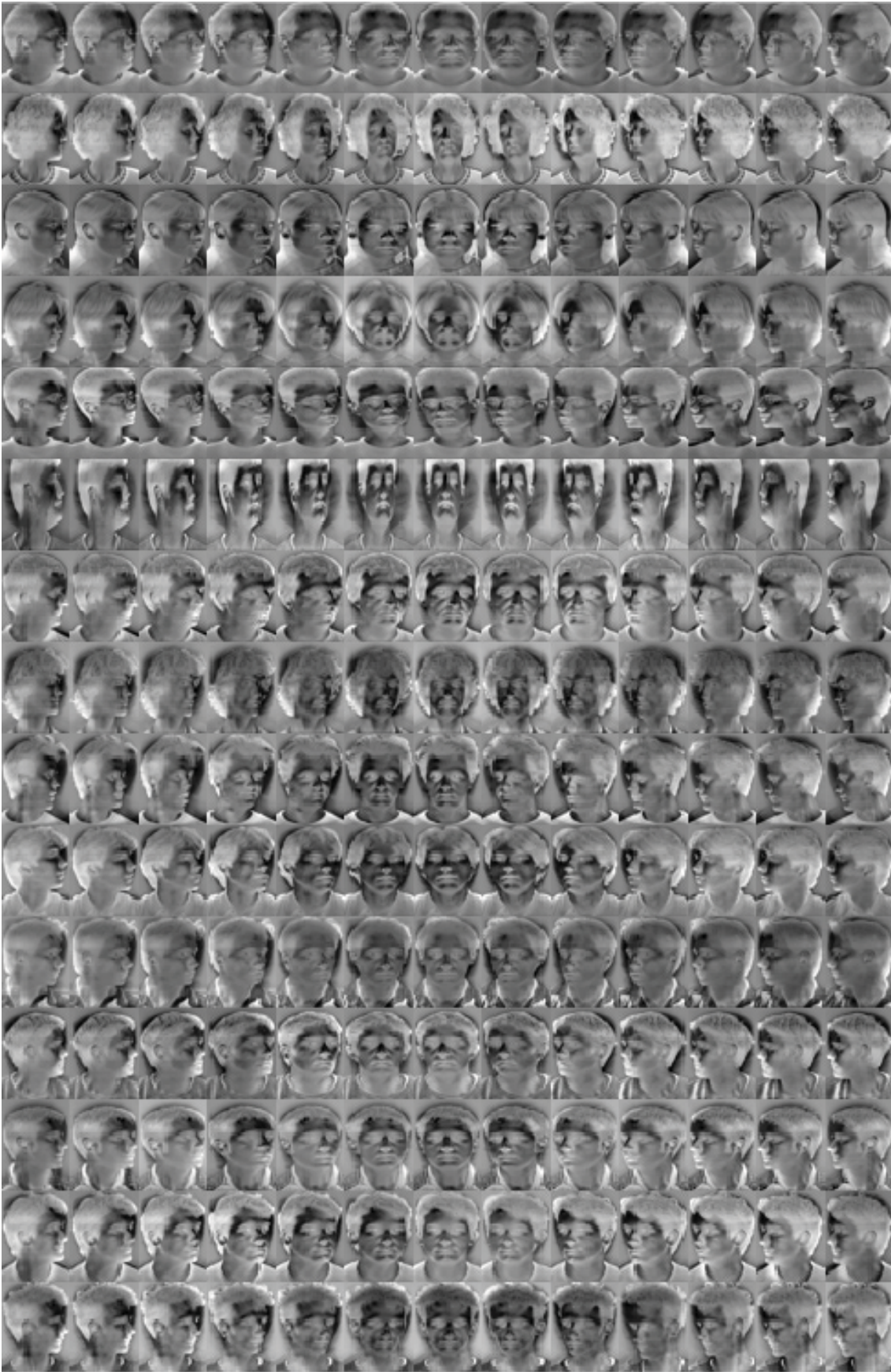


Fig. 4.15: Style information (head poses) of Fig. 4.14 extracted by the F-SVC with the linear kernel.

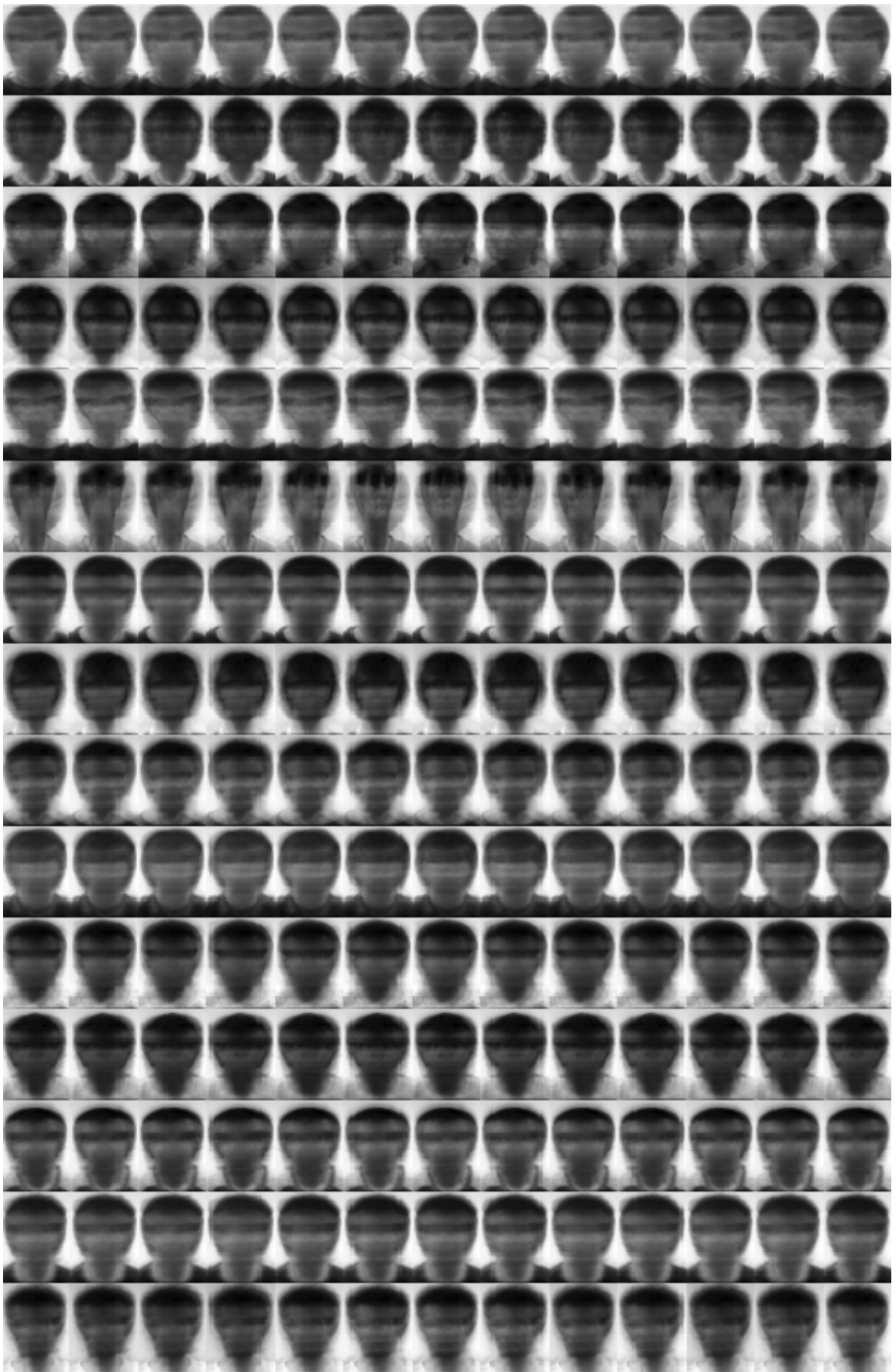


Fig. 4.16: Style-normalized images of Fig. 4.14 with head poses normalized. Only the individual properties remain.



Fig. 4.17: Class and field information alternated (reversed task) of Fig. 4.14:(reversed task): original images. Images in each row represent a class (a specific head pose), while those in each column depict a field (a specific individual involved).

Once again we can see the significant difference for images from different columns of Fig. 4.24, which specifies the different facial expressions (the field information). Additionally, for style-normalized images from each row of Fig. 4.25, no vast difference can be found. Such scenario represents the performance of the field information normalization. While for ones from different rows, significant different facial shapes representing different individuals (class information) are easily observed. It undoubtedly once again demonstrates visually the effectiveness of the SNT operation within F-SVC.

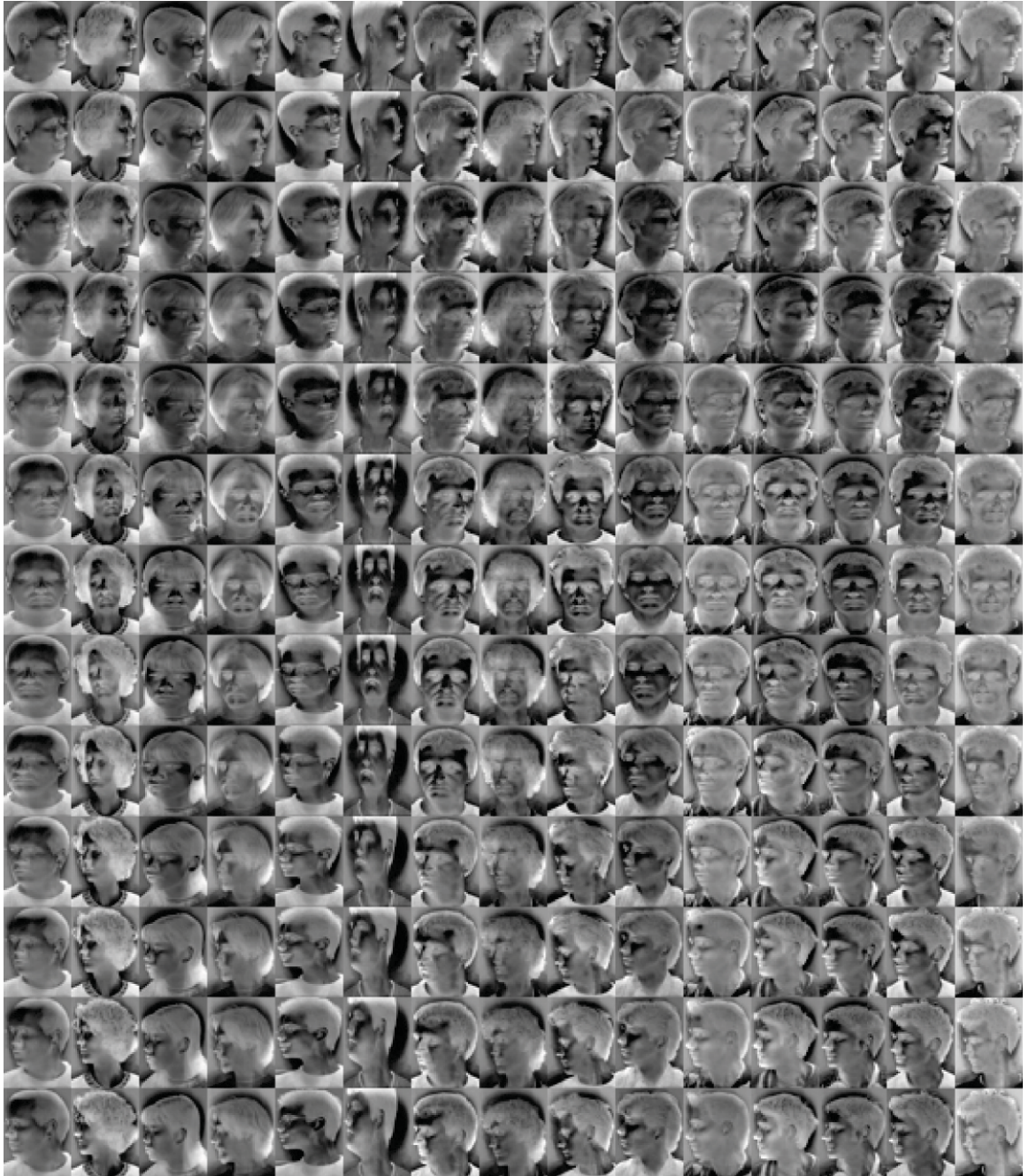


Fig. 4.18: Style information (individual identities) of Fig. 4.17 extracted by the F-SVC with the linear kernel.

## 4.7 Model Properties: Further Studies

### 4.7.1 Class Separability Improvement

In this section, the t-SNE embedding [154] for both non-*i.i.d.* patterns and style-normalized data produced by the proposed linear F-SVC model will be depicted for the better understanding of the performance of it.

The class separability improvement due to the proposed F-SVC model (linear ker-

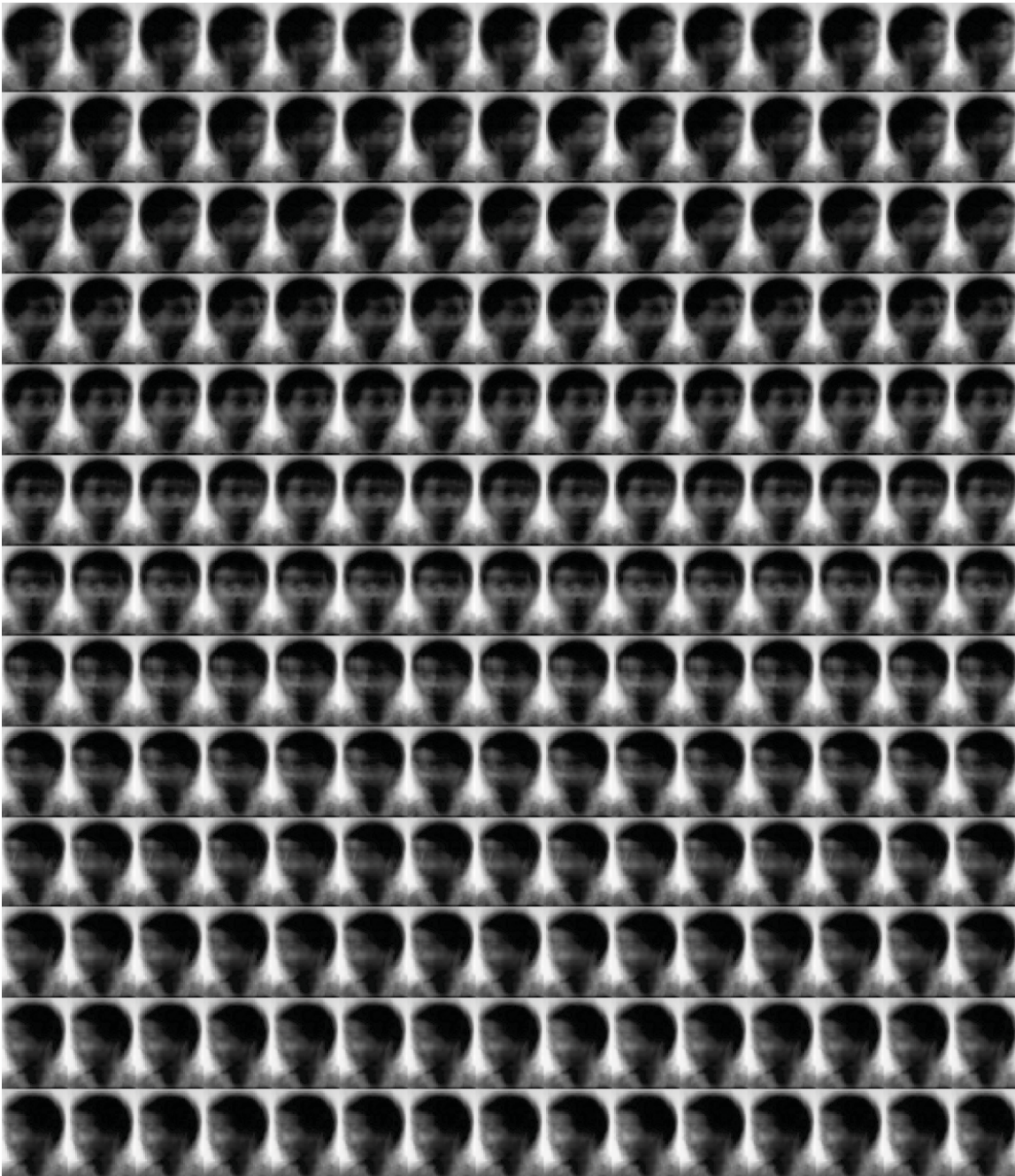


Fig. 4.19: Style-normalized images of Fig. 4.17 with individual identities normalized. Only the pose information is kept.

nel) for the Face Data can also be visualized according to the t-SNE embeddings plot in Fig. 4.26(a) and (b) for style-discriminative and style-normalized patterns. Clearly, style-normalized patterns are gathering into different groups with their corresponding class labels. However, all the style-discriminative ones place themselves in a mass without clear boundaries.

Additionally, as also seen from Fig. 4.26(a) and (b), after the SNT, data envelope of each class is equipped with more similar orientation and magnitude. As demonstrated in [155], SVC [6] trained with identical orientation and compactness is equivalent to the





Fig. 4.20: The JAFFE Database [38](original task): original images. Images in each row represent a class (a specific facial expression), while those in each column depict a field (a specific individual involved).



Fig. 4.21: Style information (individual identities) of Fig. 4.20 extracted by the F-SVC with the linear kernel.

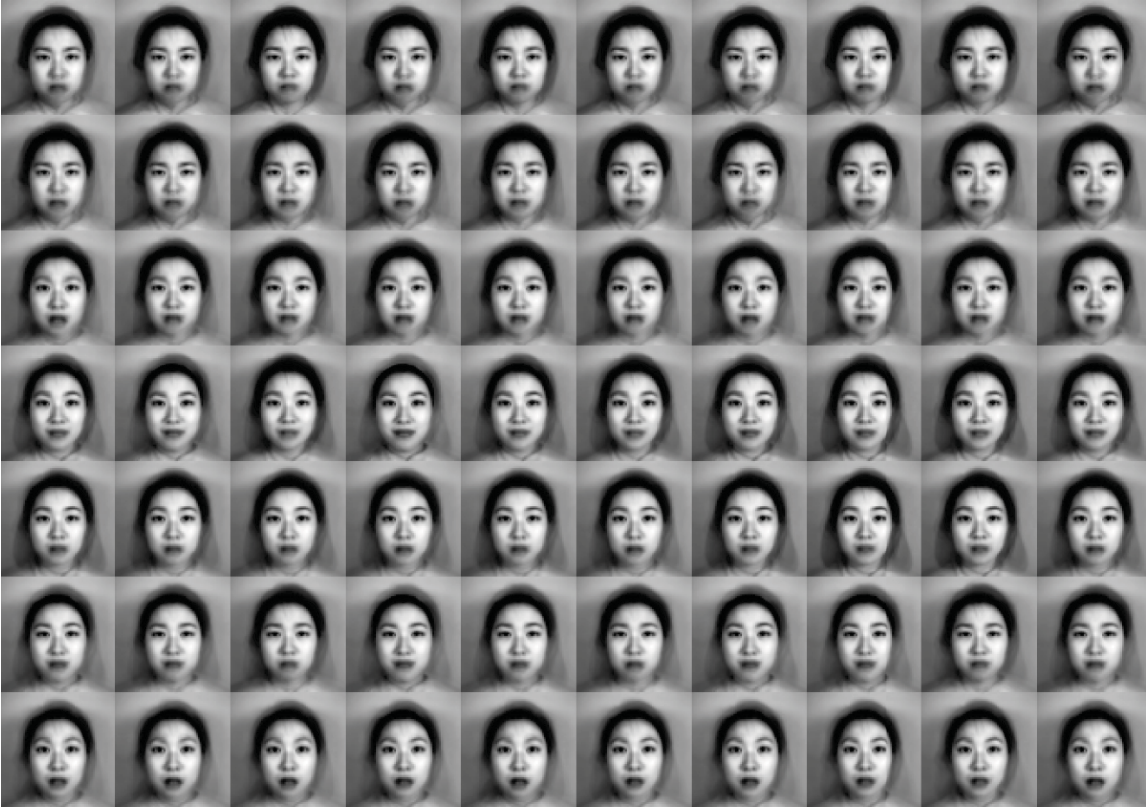


Fig. 4.22: Style-normalized images of Fig. 4.20 with individual identities normalized. Only the expression information remains.

authors' proposed Maxi-Min Margin Machine ( $M^4$ ). It is designed to calculate more reasonable decision boundaries with all the training data, rather than only part of those (named Support Vectors, SVs) in conventional SVC [6] setting. The F-SVC model is at this moment more capable of utilizing better global information in the training data. Such appealing property enables the exact optimization of decision functions.

## 4.7.2 Convergence Property

As noted in Section 4.2.3, theoretically the optimization is performed monotonically and decreasingly with at least a local minimal value achieved. Take the F-SVC as an example, as seen in Fig. 4.27, for all the involved datasets, the proposed F-SVC is optimized with the objective value descending monotonically until it finally converges in a stable manner for both the linear and the RBF kernels.

One point to be noted is that the objective values can be hugely different from different datasets or different kernelization options. For the sake of saving the space, we normalized the value range to be  $[0,1]$  in Fig. 4.27. Hence, it only depicts the descending trend rather than exact values.

## 4.7.3 Parameter Sensitivity

The parameter sensitivity is also analyzed by taking the example of the F-SVC model since it has been demonstrated that the factor brought by  $\epsilon$  in the SVR models [37] is



Fig. 4.23: Class and field information alternated of Fig. 4.20 (reversed task): original images. Images in each row represent a class (a specific individual involved), while those in each column depicts a field (a specific facial expression).



Fig. 4.24: Style information (facial expressions) of Fig. 4.23 extracted by the F-SVC with the linear kernel.



Fig. 4.25: Style-normalized images of Fig. 4.23 with facial expressions normalized. Only the individual information is kept.

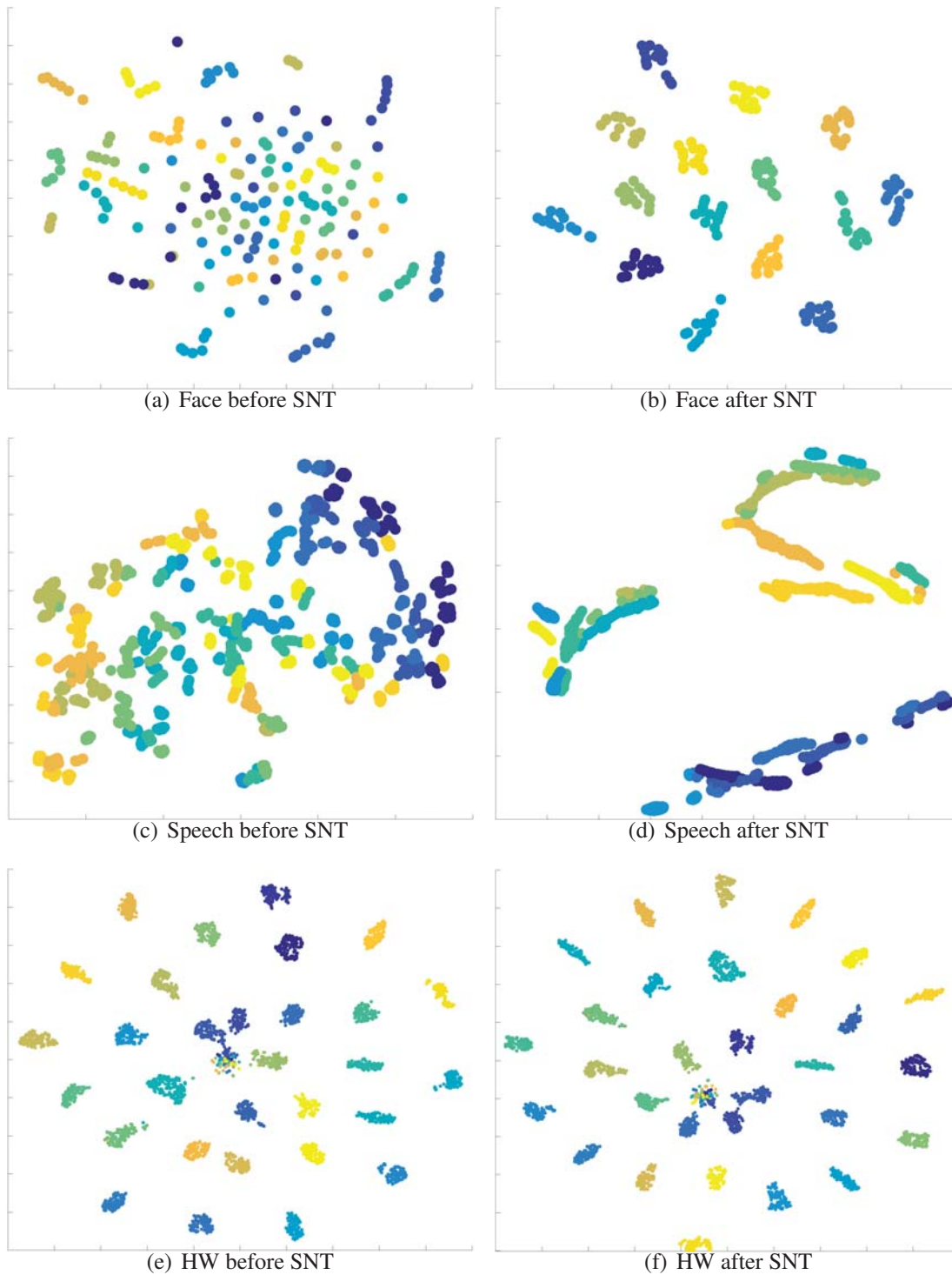


Fig. 4.26: The t-SNE embedding comparison brought by the F-SVC model: (a), (c) and (e): embeddings for the original Face, Speech, and HW data; (b), (d) and (f): respectively those after the field information filtered via SNT; Better viewed in colour.

much less than  $c$  and  $\gamma$  (with GK). Such case can also be seen in the SVC model [6, 37]. Fig. 4.28 shows the sensitivity of  $c$ ,  $t$ , and  $\gamma$  of the proposed F-SVC on the Face Data with both linear and the RBF kernel applied, where the center of the horizontal axis is the best

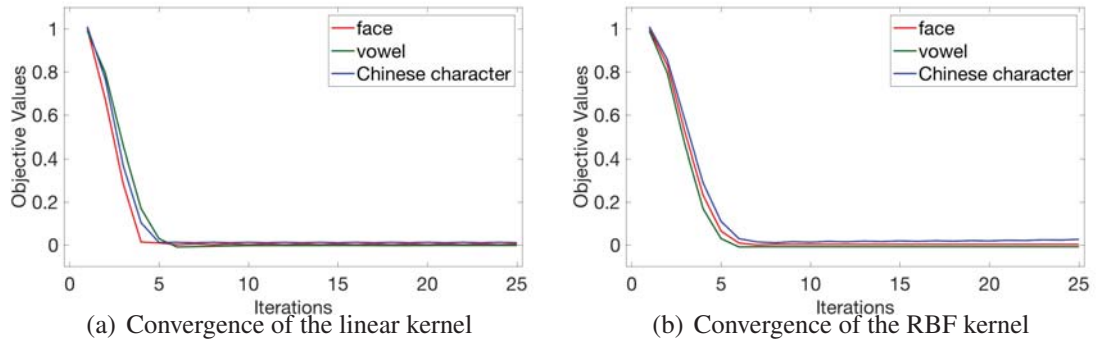


Fig. 4.27: F-SVC convergence performance visualization: (a): linear kernel; (b): RBF kernel; Better viewed in colour.

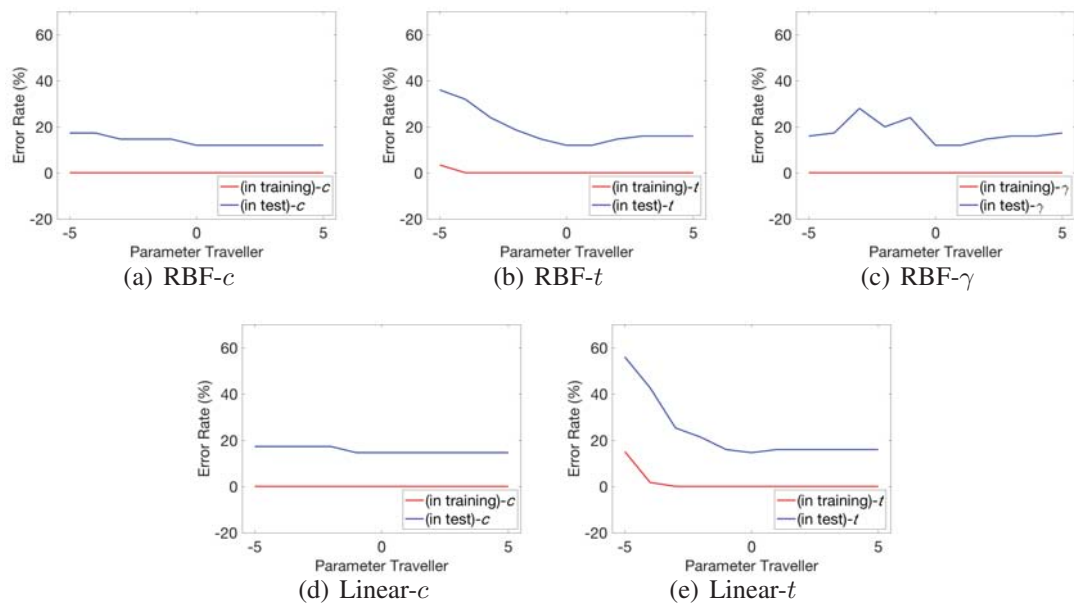


Fig. 4.28: F-SVC parameter sensitivity analysis on the Face Data: (a), (b) and (c): parameter sensitivity analysis of the RBF kernel with  $c$  (cost),  $t$  (style normalization tradeoff), and  $\gamma$  (width parameter in RBF kernel) respectively; (d) and (e): parameter sensitivity analysis of the linear kernel  $c$ ,  $t$  respectively; All the graphs are drawn by only alternating the specific parameter while keeping all the others fixed. The middle of the horizontal axis represents the specific parameter achieving the best performance. Better viewed in colours.

parameter obtained by grid-searching. The vertical one represents the error rate with the parameter changing as the horizontal one. As can be seen, the style normalization tradeoff parameter  $t$  is more sensitive than the cost parameter  $c$  on the Face Data. It affects the performance even more than  $\gamma$ , the width parameter of the RBF kernel, which is a well-known factor that hugely affects the performance of the SVM-based approaches. [156]. However, when  $t$  is approaching the best parameter, both the training as well as the test error keep being close to the peak performance.

Different from the Face Data,  $c$  seems to be more sensitive than  $t$  for the Speech Data, as represented in Fig. 4.29. Additionally, the difference in performance between

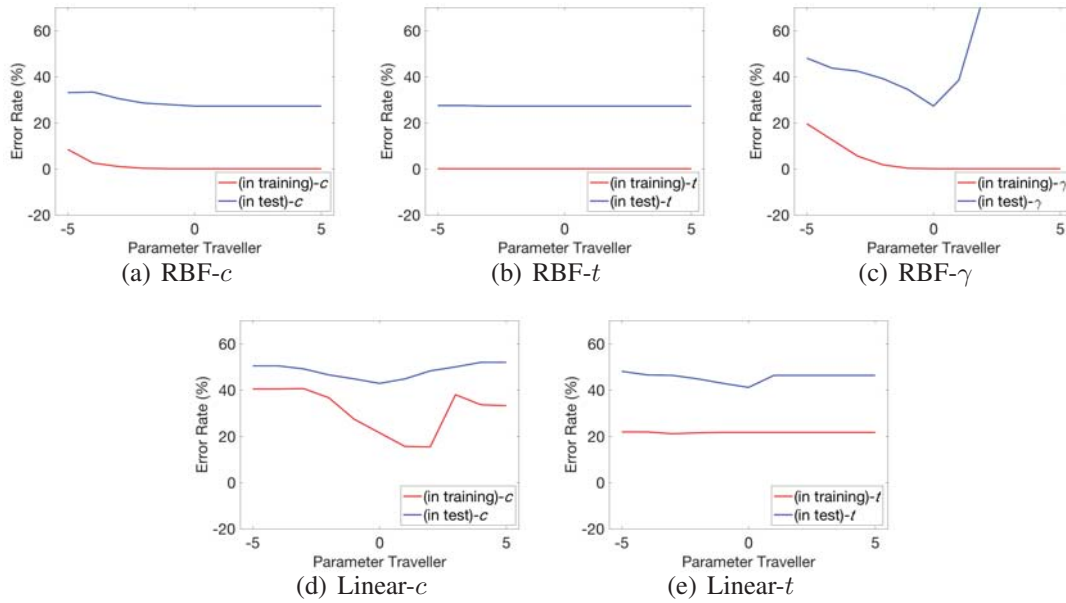


Fig. 4.29: F-SVC parameter sensitivity analysis on the Speech Data.

alternative parameter settings is more likely to be great. The reason for the finding might be that the lowest error rate of this set is much lower than that of the Face Data. The improvement of changing parameters may be hugely limited for the Face Data due to the relatively high performance. However, better performance seems to be hard to be achieved on the Speech Data. Hence significant performance difference can be observed. Finally,  $\gamma$  is the most sensitive factor that affects the performance of the Speech Data. We

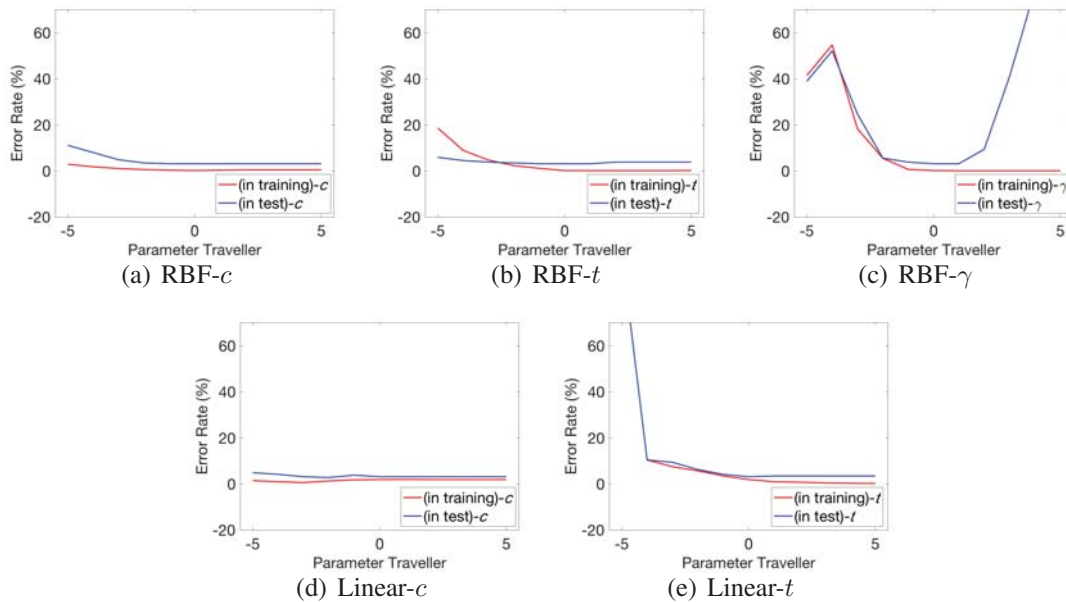


Fig. 4.30: F-SVC parameter sensitivity analysis on the HW Data.

can see from Fig. 4.30 that  $t$  is of more significant influence on the performance than  $c$  for both linear and the RBF cases. Once again  $\gamma$  is the most influential factor among all



the other parameters. However, the performance keeps steady when it closes to the best parameter setting obtained by the grid-searching strategy.

## 4.8 Summary and Future Work

In this chapter, a novel framework called Field Support Vector Machine (F-SVM) model, including both the Field Support Vector Classifier (F-SVC) and the Field Support Vector Regression (F-SVR), is proposed, where the *i.i.d.* assumption is overturned for both ones. It enables to train and predict a group of patterns (i.e., a field pattern) simultaneously.

To be specific, the proposed F-SVM model has been investigated by learning both the classifier (F-SVC), or the regressor (F-SVR), and the Style Normalization Transformation (SNT) for each field simultaneously. By taking advantage of the nonlinear kernel mapping, the sufficiently complicated field information in real case normalized by the proposed F-SVM model can also be represented with the nonlinear SNT feasibly. The alternative training strategy of the proposed F-SVM model is also evaluated theoretically and experimentally by producing the final convergence. Moreover, a self-learning based style transfer scheme is also investigated in order to predict patterns equipping with unknown styles during training for both the classification and the regression tasks. It is fulfilled by means of transferring the trained style information to the unseen field by learning the Transductive-SNT (T-SNT).

A series of experiments have been conducted to verify the effectiveness of the F-SVM model and the style transfer procedure. Empirical results showed that the proposed F-SVM significantly outperforms diverse state-of-the-art predictors in multiple artificially generated synthetic and real benchmark datasets for both the field classification and field regression occasions.

Moreover, the proposed F-SVC model is further studied visually. By producing style-normalized patterns, the F-SVC is proved effective to normalize the style information. The performance of the F-SVC model is also evaluated by the class separability improvement and the parameter sensitivity analysis. By normalizing the style information from the original input patterns, the F-SVC model is useful to improve the prediction performance by promoting centralization and compactness of each classifying class.

### 4.8.1 Future Work

Nevertheless, one potential drawback of the proposed F-SVM framework is that it can hardly be used to classification cases when the number of classes is relatively large. As demonstrated in [157], such issue is the consequence of the voting strategy of the SVC-based models.  $N \times (N - 1)$  independent classifiers are to be built for a  $N$  class classification scenarios.<sup>14</sup>

Besides, the only trained parameters of the F-SVM framework are the predictor parameter  $\{w, b\}$ , as well as the SNT (T-SNT) matrix  $A$ . The number of relevant trainable factors are much smaller than those of deep learning models [1]. As the fact that the number of trained parameters are the key factor of the representative power and the model

---

<sup>14</sup>In total  $3,755 \times (3,755 - 1) = 14,096,270$  independent classifiers shall be built for a 3,755-class classification scenario for the GB2312 Level 1 Chinese character set [91] based on one v.s. one voting strategy.

freedom of the machine learning machines [43], the proposed F-SVM framework is less flexible to represent the sufficiently complicated style information in real scenarios.

Nowadays, the deep learning models are massively developed in the machine learning communities. Those problem mentioned in the last two paragraph can be readily solved by a novel neural network based generative framework named Style Neutralization Generative Adversarial Classifier (SN-GAC) by producing style-neutralized high-quality human-understandable patterns. It will be fully demonstrated in Section 5.1.

In addition, such generative based architecture can naturally be used for new data synthesization and generation. The SN-GAC framework can be in this way extended to another model, namely, the W-Net model. It enables the reversed task of the style elimination transformation and the style normalization transformation that have been demonstrated in Chapter 3 and Chapter 4. It is called as the free manipulation of the arbitrary-style data generation task. The W-Net model will be demonstrated in Section 5.2 in detail.

## Chapter 5

# Generative Approaches with Style Information

Given an observable variable  $X$ , and a target variable  $Y$ , a generative machine learning framework is a statistical model to estimate the joint probability distribution on  $X \times Y$ , namely,  $P(X, Y)$  [56]. In other words, a classifier based on the generative machine learning framework optimizes the joint probability  $p(x, y)$  for a single data-value pair  $\{x, y\}$ . The prediction is performed by using the Bayes rule, namely,  $p(y|x) = p(x, y)/p(x)$ , to calculate conditional probability distribution after which the most likely label will be assigned. The full conditional distribution will then be obtained. It would be hereby used to get the new latent variables for generation task by the sampling algorithms. On the contrary, the discriminative models (demonstrated in Chapter 4) do not estimate the joint distribution between  $x$  and  $y$ . It is significantly different from those generative models that are going to be demonstrated in this chapter.

A direct manner to implement the generative model for non-*i.i.d.* data classification scenario is to simply produce the corresponding *i.i.d.* patterns. By taking the advantage of the capability of generating new samples, the produced patterns are optimized to satisfy the *i.i.d.* assumption. These obtained *i.i.d.* examples are then sent to other machine learning models.

One novel example employs the Generative Adversarial Network (GAN) [2] training strategy to neutralize the style information embedded in the original non-*i.i.d.* input patterns, producing the non-style data. It is named as the Style Neutralization Generative Adversarial Classifier (**SN-GAC**), which has been reported in a conference paper [17], and invited for submission in a journal paper [27]. In the model, the discriminator in the original GAN framework is attached with an auxiliary classifier in charge of assigning the correct class labels of the generated patterns. Improved classification performance was obtained with both the face and the Chinese handwriting character classification tasks. It will be explained in Section 5.1

Furthermore, based on the data synthesizing nature of those generative models, the data generation can be fulfilled readily. A novel framework, named as **W-Net**, is designed for **Few-shot Multi-content Arbitrary-style Chinese Character Generation (FMACCG)** task [18].<sup>1</sup> As a reversed task of the style-neutralization with SN-GAC, it enables to generate a full set of Chinese handwriting characters when a few (even a

---

<sup>1</sup>The work reported in [18] fulfills the task of the **One-shot Single-content Arbitrary-style Chinese Character Generation (OSACCG)**. It is a special case of the FMACCG when there is an only single content character and single style character are available.

single) samples available with a specific writing style. It is even capable of synthesizing characters in other Eastern Asian languages including traditional Chinese, Korean, and Japanese when the proposed W-Net is only trained with simplified Chinese characters. Such a framework is demonstrated in Section 5.2 and has been reported as another conference paper [18].

## 5.1 Style Neutralization Generative Adversarial Classifier based on the Upgraded U-Net Architecture

Traditional machine learning approaches always hold the assumption that data for model training and in real applications are created following the identical and independent distribution (*i.i.d.*). However, several relevant research topics have demonstrated that such condition may not always describe the real scenarios. One particular case is that the patterns are equipped with diverse and changeable style information, which is inconsistent with each other, as have been fully demonstrated in the previous chapters. Such problem has been extensively demonstrated in previous related chapters and sections.

In this section, a novel classification framework named Style Neutralization Generative Adversarial Classifier (SN-GAC) is introduced to accomplish the classification task in such disparate and inconsistent stylistic data distribution case. The SN-GAC model is based on an upgraded U-Net architecture [10, 21]. It is trained adversarially with the Generative Adversarial Network (GAN) framework [2].

To be specific, the generative model in the SN-GAC framework neutralizes inconsistent and diverse style information from the original style-discriminative patterns (*style-source*) by building the mapping function from them to their style-free counterparts (corresponding standard examples, *standard-target*). A well-learned generator in the SN-GAC framework is capable of producing the targeted style-neutralized data (*generated-target*), satisfying *i.i.d.* assumption.

Additionally, the training of the SN-GAC model follows the adversarial optimization strategy. There is an independent discriminator set to surveil and supervise the training progress of the above-mentioned generator by feeding both the standard pair (*real pair, style-source + standard-target*) and the generated pair (*fake pair, style-source + generated-targets*) into it. The discriminator is at this moment optimized by distinguishing between the real pair and the fake one.

Simultaneously, an auxiliary classifier is also embedded in the upon-mentioned discriminator to assign the correct class label for both pairs. Such setting scheme has been proved effective in aiding the generator to produce high-quality human-readable style-neutralized patterns [26]. It will then be further fine-tuned for the sake of promoting the final classification performance. Extensive experiments have adequately demonstrated the effectiveness of the proposed SN-GAC framework by outperforming several relevant state-of-the-art baselines in the literature on two empirical dataset in the non-*i.i.d.* data classification task, including a face classification dataset [119] and a Chinese handwriting classification database [121].

## Research Background

As demonstrated in Chapter 3 and Chapter 4, conventional machine learning frameworks assume that data involved shall be created following the *identical and independent distribution (i.i.d.)* assumption. Nevertheless, there are quite many relevant publications which have already pointed out that such condition may not always describe the real scenarios desirably in several of the practical situations. These research topics include [13, 14, 15, 29]. One empirical case is when patterns are generated by groups. Examples from each group are created from a specific data source with homogeneous style information, which is inconsistent with other origins.

As demonstrated in Section 4.1.1, such scenarios include face recognition with various head poses (yaw angles) [119], and handwriting character classification of multiple writers with diverse handwriting styles [121]. Traditional methods, in this case, may suffer from degraded classification performance because of the existence of multiple, diverse, and inconsistent style information embedded in the original patterns.

A novel generalized framework named Style Neutralization Generative Adversarial Classifier (SN-GAC) is investigated to accomplish such a classification task with inconsistent style information in this section. Different from the F-SVM model demonstrated in Chapter 4, the SN-GAC model takes the full advantage of the extensively-studied deep learning frameworks with better flexibility in model architecture design and more powerful on model representation capability.

Specifically, the proposed SN-GAC model is inspired by the recently-introduced and progressively-developed Generative Adversarial Network (GAN, as demonstrated in [2]) based Img2Img model investigated by [21] for high-quality image translation tasks. Simultaneously, it is modified from the U-Net (initially introduced in [10] for biomedical image segmentation) according to several of the specific purpose in the style neutralization transformation. It results in an upgrading structure of the traditional U-Net, enabling neutralizing diverse styles from original input style-inconsistent data by learning the corresponding style-free standard patterns. The final purpose of the SN-GAC framework is to improve the classification performance. In the same time, thanks to the merits brought by the GAN-based model, it is also capable of producing high-quality human-understandable style-neutralized patterns.

Particularly, the generator of the SN-GAC framework ( $G$ ) builds the neural network based nonlinear mapping between the original input patterns with various style-inconsistent information (style-discriminative *style-source* data space) to the corresponding style-free counterparts (style-consistent *standard-target* space). It is fulfilled by implicitly approximating the high-dimensional style-free data distribution. The nonlinearity introduced by the  $G$  network enables the representation of the sufficiently complicated style information in the real scenario. In the meanwhile, there is no restriction on the number of the data involved.<sup>2</sup> A well-trained generator hereby enables to neutralize diverse styles from original input data by producing high-quality human-understandable style-neutralized patterns (*generated-targets*), satisfying the *i.i.d.* assumption.

In more detail, the previous U-Net framework introduced in [21] is only effective in the Img2Img translation between binary domains. That is to say, it merely enables the

---

<sup>2</sup>Such problem can often be seen in those models and algorithms taking the merits of the kernel trick [118], e.g., the F-SVM model introduced in Chapter 4.

one-to-one mapping. Nevertheless, the style-neutralization task is a many-to-one mapping function, meaning that a single standard pattern can be corresponded to over one style-inconsistent original data. Thanks to the *style hint* idea mentioned in [59], the style information can be readily obtained by the in-depth, high-level feature (named as the *style vector*) calculated from a pre-trained classification network on the relevant task. It is then concatenated with the encoded output of the U-Net encoder before being sent to the decoder. Such a network structure enables the many-to-one mapping function in the proposed SN-GAC framework. Empirical experiment results reported in this chapter even demonstrated that it is also valid for unseen style neutralization occasions.

However, different from the traditional GAN framework in the literature that aims to synthesize realistic patterns only, the proposed SN-GAC model further excavates the potential capabilities of the GAN models by promoting the final classification performance of the generated style-neutralized examples. It is fulfilled by attaching an auxiliary classifier ( $C$ ), as proposed in [26], on the discriminative model ( $D$ ). As demonstrated in [2], the  $D$  is inherent from the vanilla GAN framework. The discriminator and the attached classifier ( $D - C$ ) is in this way not only supervising the adversarial training procedures by distinguishing the fake pair (*style-source + generated-target*) from the real one (*style-source + real-target*). It is also responsible for assigning the correct class label of the input pair. Such the classification formulation is significantly different from many existing traditional models (Fig. 5.1(a)) where all the samples are simply fed into the classifier.

Importantly, the two composing models (Fig. 5.1(b)) (the generator and the discriminator) in the SN-GAC framework are optimized simultaneously in an iterative and alternative learning fashion. Some other joint losses are also minimized together with an end-to-end scheme. The style neutralization and the data classification can be readily fulfilled. Simultaneously advantages can be obtained from both alternative procedures. For the sake of promoting the final classification performance, the classifier will be further fine-tuned with only the fake pair by minimizing just the classification loss when the generator optimization is saturated. In this stage,  $G$  is ready to produce high-quality human-understandable style-neutralized examples so that it will be fixed. After that, the classifier will be directly used to assign class labels given the generated high-quality style-neutralized instances produced by a well-optimized  $G$  network.

### 5.1.1 SN-GAC Model Specification

As an inherent machine learning framework from the GAN-based models, the proposed SN-GAC framework is composed of a generator and a discriminator. An attached classifier is also embedded in the discriminator. The full SN-GAC model with both the generator and the discriminator is illustrated in Fig. 5.2.

The generator of the proposed SN-GAC framework ( $G$  network) is based on the adversarially trained Img2Img model [21] initially designed for the image translation between binary domains. The original Img2Img framework was constructed on the U-Net architecture introduced in [10]. It is initially designed for biomedical image segmentation. However, in this chapter, multiple modifications on the original U-Net with several specific purposes are proposed. It composed of an upgraded version of the U-Net framework.

Meanwhile, the discriminative model ( $D$  network) in the proposed SN-GAC framework is attached with an auxiliary classifier ( $C$ ) to assign class labels correctly. It is known

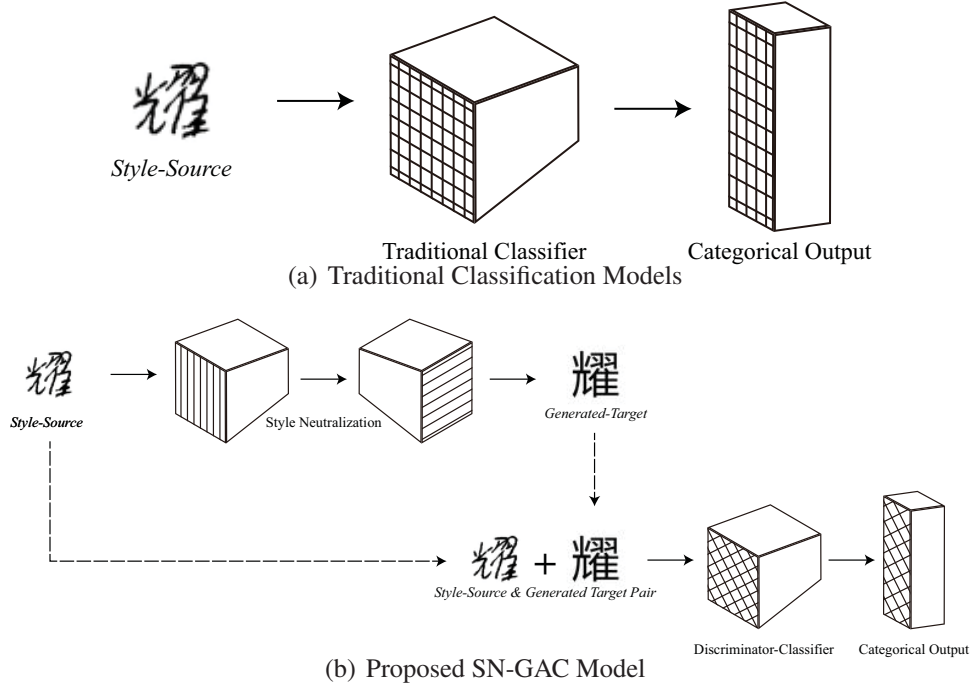


Fig. 5.1: Traditional classifier and the proposed SN-GAC classifier.

as the  $D - C$  network since they share most of the model architecture. Such architecture is initially introduced in [26]. Additionally, such auxiliary classification loss would also be helpful to supervise the adversarial training progress of the  $G$  model as well.

### Preliminaries

The notations necessary for the demonstration in this chapter will be firstly instructed here. Then the SN-GAC model structure as well as the training details will be specified.

**Definition 5.1.1.** *Style-Source*, noted as  $x$ , is specified as a pattern equipped with a specific kind of style information. It is inconsistent with others.<sup>3</sup>

**Definition 5.1.2.** Given the *style-source* as defined in Definition 5.1.1, the corresponding **class label** is given as  $y$ , specifying the content of the given pattern by labeling the ground truth for the classification task.

Both  $x$  and  $y$  as defined in Definition 5.1.1 and Definition 5.1.2 respectively form the associated relationship with each other, noted as  $\{x, y\}$ .

**Definition 5.1.3.** *Standard-Target* is defined as the corresponding pattern with the standard style (neutralized style) given the style-inconsistent *style-source*  $x$  as defined in Definition 5.1.1. It is denoted as  $x_*$ .

Both  $x_*$  and  $x$  share the identical class label  $y$ . It means that the associative relationship between data and class label should also be extended to  $x_*$ . Noted as  $\{x_*, y\}$ ,

<sup>3</sup>Such style inconsistency can be found when data are created by multiple sources, while each source is generating examples with a special kind of style information. The stylistic tendency differs from different sources. Such case has been demonstrated in previous chapters.

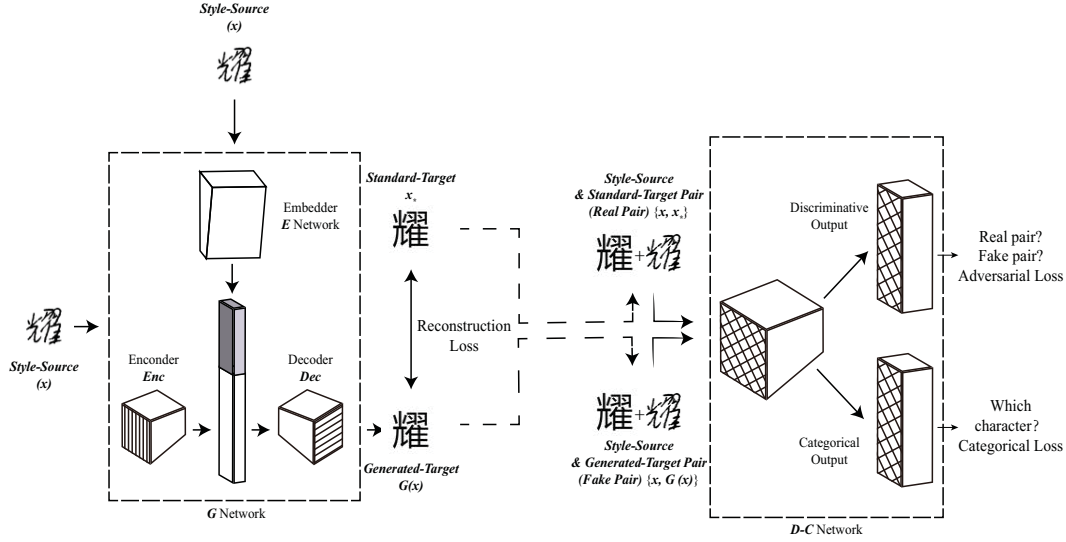


Fig. 5.2: SN-GAC architecture. The proposed SN-GAC model includes a generator ( $G$ ) and a discriminator with an auxiliary classifier ( $D - C$ ).  $G$  consists an embedder network ( $E$ ) for style vector inference, a convolutional encoder, and a deconvolutional decoder. It generates a style-neutralized *Generated-Target* ( $G(x)$ ) when given a style-inconsistent *Style-Source* ( $x$ ).  $D - C$  is a convolutional network, capable of distinguishing the input pair coming from the real or the generated data with the discriminative output. In the meanwhile, it will assign the class label of the input pair by the categorical output.

it represents that the information targeted to be neutralized shall only be related to the inconsistent stylistic details. In the same time, the pattern content ( $y$ ) shall be well kept.

Additionally, the standard target style ( $x_*$ ) needs to be specified before the model is optimized since it is part of the training data. The  $G$  network in the proposed SN-GAC model builds the nonlinear neural mapping from those multiple and diverse style-inconsistent *style-sources* to the corresponding style-free standard targets (*standard-targets*). The upon-mentioned one-to-one association embedded in  $\{x, y\}$  and  $\{x_*, y\}$  will as a result leads to another pair-wise relationship as follows:

**Definition 5.1.4.** *Style-Source & Standard-Target Pair* is denoted as  $\{x, x_*\}$ .

In fact, each style-free *standard-target* ( $x_*$ ) can be in correspondence with many style-inconsistent *style-sources* ( $x$ ). They are created by more than a single data generator. In this way, they are inconsistent and diverse in style information.

**Definition 5.1.5.** *Generated-Target* ( $G(x)$ ) is defined as the style-neutralized output of  $G$  in the proposed SN-GAC model given the style-inconsistent *style-source*  $x$ .

The SN-GAC is trained to minimize the difference between  $x_*$  and  $G(x)$ . The content of  $x_*$  (the class label  $y$ ) shall also be shared with  $G(x)$  if  $G$  is well optimized, formulating the correspondence of  $\{G(x), y\}$ . Similarly, the following data pair will be in hand:

**Definition 5.1.6.** *Style-Source & Generated-Target Pair* is denoted as  $\{x, G(x)\}$  to represent the correspondence between  $x$  and  $G(x)$ .



## Upgraded U-Net based Generator

Similar to [21], the  $G$  network in the proposed SN-GAC model is built upon the U-Net architecture with skipping connections. Given that a style-inconsistent *style-source* pattern  $x$  as defined in Definition 5.1.1, the learning target of the  $G$  network is to generate a style-neutralized *generated-target* pattern  $G(x)$  (as defined in Definition 5.1.5) with high quality. It should be human-understandable as well.

**Architecture of the  $G$  network:** The basic network architecture of the proposed upgraded U-Net structure in the proposed SN-GAC framework is briefly depicted in Fig 5.3. It consists of a convolutional encoder (the upper part in Fig. 5.3 with vertical patterns) and a deconvolutional decoder (the lower part in Fig. 5.3 with horizontal patterns). They are denoted as  $Enc$  and  $Dec$  respectively.

$Enc$  maps the style-discriminative *style-source*  $x$  to the high-level encoded feature space (denoted as  $Enc(x)$ ). It extracts structured convolutional features from the *style-source* by performing down-sampling convolutional operations. Meanwhile,  $Dec$  recovers high-level encoded features to the targeted style-neutralized *generated-target*  $G(x)$ . It reconstructs targeted data in a structured manner by engaging the down-sampling deconvolutional implementations.

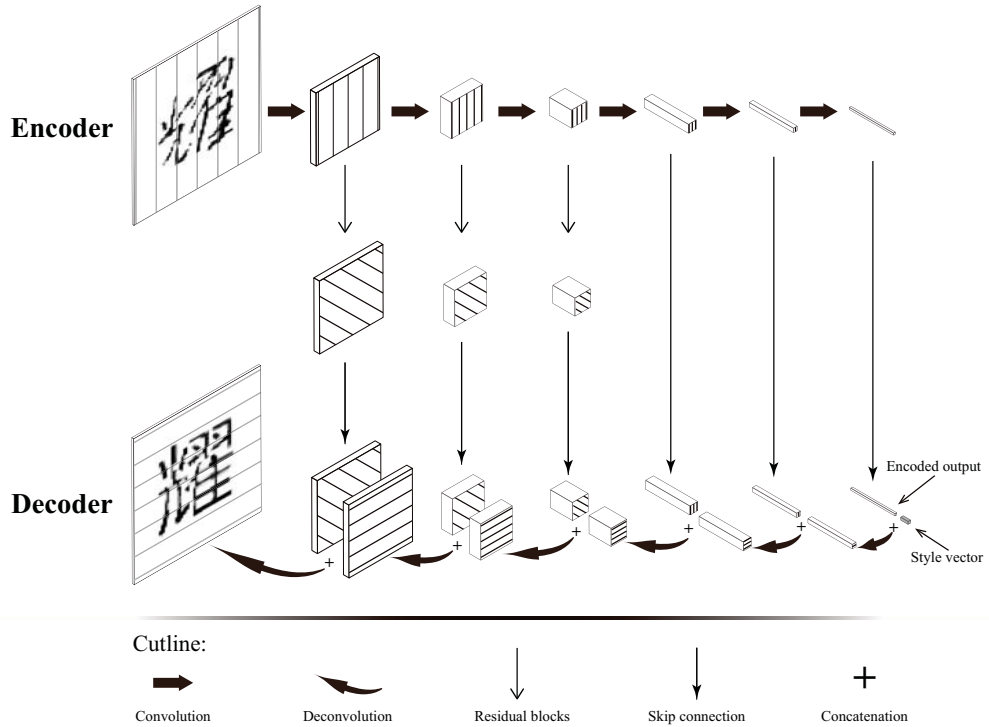


Fig. 5.3: Upgraded U-Net structure.

Similar to the description in [10], skipping connections from the encoder to the decoder are implemented to align structures and features on the equivalent level. It leads to that the input feature of each decoding layer comes not only from the previous decoding layer but also the encoding one at the same level. Specifically, features acquired from the  $i$ -th layer of the encoder is directly concatenated at the tail of the feature produced by the  $(n - i)$ -th layer of the decoder. However, in the proposed SN-GAC framework, they are

only applied in the higher levels of features, where information is more deeply extracted and centralized than those from lower levels. Differently, in the original U-Net framework introduced in [10], all the relevant features are related to the skipping connections.

Nevertheless, the extracted features in the relatively lower level of the original U-Net is much more dispersed and less compacted. In the same time, the feature map size of them is more significant than those in higher layers. Such case will result in a relatively massive amount of the extracted information. Simple skip-connection seems not to be a suitable option in this case since there are no further feature extraction operations. Thanks to the residual blocks introduced and investigated in [8], these skip-connections in the lower levels are merely replaced by them. It provides further centralization on the features obtained from the encoder. In the meanwhile, the feature dimension will be kept. They are then be concatenated with the corresponding decoded features in  $Dec$ .

**Embedder Network for Style Representation:** The vanilla U-Net proposed in [10] and utilized in [21] only enables the  $Img2Img$  translation function between binary domains. However, in the style neutralization task considered in this research topic, the mapping should be built in a many-to-one manner. In order to still make use of the upgraded U-Net structure, an extra embedder network (noted as  $E$ ) is employed as part of  $G$ . It represents the embedded style information of the input style-discriminative *style-source* pattern to incorporate with the multi-to-one mapping model.

According to the description demonstrated in [103], patterns in the same style tend to be placed themselves closer with each other in the deep feature space. In such sense,  $E$  can be realized with an extra deep network, fine-tuned from a pre-trained model optimized based on a similar or relevant classification task or trained from scratch. It is fulfilled by choosing features from the final layer (inferred logits before the sigmoid or softmax function) to compose the *style vector* ( $E(x)$ ) [59]. They are then concatenated with the output of the encoder ( $Enc(x)$ ) before being fed into the decoder together. Such concatenation of the *style vector* and the encoded output can also be seen in Fig. 5.3.

One thing to be pointed out is that the *style vector* can also be obtained by the idea introduced in [104], where a non-trainable randomly initialized style dictionary with each row representing the specific style vector of the corresponding group of data in the training phase. In the project of [101], such an idea was well implemented and effectively performed. It saves the training effort of the extra embedder, as well as the time consumed during the inference of the deep embedder network. However, it is infeasible when facing unknown styles as the fact that only known style vectors can be looked up from the dictionary. In addition, the Gaussian-based non-trainable matrix can not effectively represent the inner relationship between different but somehow related styles. On the contrary, a valid style vector can be deduced by a well-trained external embedder on optimized based on relevant classification tasks, no matter whether it is known or not.

### Optimization Details of the $G$ network

The quality of the *Generated-Target*  $G(x)$  is well maintained by penalizing the adversarial loss proposed in [2] to maximally confuse the  $D$  network during the training, as will be demonstrated in Section 5.1.1. Additionally, the L1 reconstruction error between  $x_*$  and  $G(x)$ , namely,  $\mathbb{L}_{l_1} = \|x_* - G(x)\|_1$ , is also applied. It encourages  $G$  to synthesize sharp and clear image details, as engaged in [21].

Moreover, the constant loss introduced in [77] is also engaged as an additional restriction to encourage high-quality output patterns. Specifically, it regulates with the L2 difference between encoded spaces of two input patterns. In the proposed SN-GAC model, it is given by  $\mathbb{L}_{const} = \|Enc(x) - Enc(G(x))\|^2$ .

Instead of explicit random noise fed into  $G$  in the traditional GANs, the dropout trick proposed in [144], severed as the implicit random noise ([21]), is applied to several layers in the decoder during training. It is shut down when performing network inference in the testing phase, and in real applications.

### Optimization Losses of $D - C$ Network

As part of the adversarial training strategy proposed in [2], the discriminator with an auxiliary classifier ([26], in short, the  $D - C$  network) is applied in the proposed SN-GAC model. Besides, the classifier embedded is also responsible for correct class label assignment of the generated style-neutralized patterns from the  $G$  model, as demonstrated in Section 5.1.1. In specific detail, the discriminator is implemented with the framework of the Wasserstein-GAN with the gradient penalty (W-GAN-GP as introduced in [24]). The architecture is fulfilled by a conventional convolutional neural network (CNN) structure.

As illustrated in Fig. 5.2, in each training iteration, pattern pair batches consisting of both the (*style-source* & *standard-target*, real pair) and (*style-source* & *generated-target*, fake pair) as defined in Definition 5.1.4 and Definition 5.1.6 respectively are fed into the  $D - C$  network. It minimizes the adversarial loss in  $D$  (the Wasserstein distance [25] measured based on  $D$ ) as  $\mathbb{L}_{adv-D} = D(x, x_*) - D(x, G(x))$ . Adversarially, in the training process of  $G$ ,  $\mathbb{L}_{adv-G} = D(x, G(x))$  is optimized in a reversed fashion. Meanwhile, as instructed in [24], a gradient penalty  $\mathbb{L}_{adv-GP} = \|\nabla_{\hat{x}} D(x, \hat{x})\|_2 - 1\|_2$  will also be engaged to encourage the Lipschitz continuity condition required by the Wasserstein-based adversarial training strategy. Furthermore, the training is also progressed by assigning the correct class label ( $y$ ) of the given pair ( $C$  training, as depicted in Fig. 5.2). It provides additional training supervising restrictions to the learning progress of the  $G$  model, as instructed in [26].

In addition, the input fake pair of  $D$ , namely,  $(x, G(x))$ , can be considered as an implicit regularization, penalizing over-flexible style transformation between the style-inconsistent *style-source* (with inconsistent style information) and the style-neutralized *generated-target*. Similar ideas are implemented in [13, 14, 29, 15] with explicit expressions. The Frobenius norm of the difference between the style normalization transformation and the identity matrix is engaged in those work.<sup>4</sup>

### Two-Phase Training Strategy with Joint Losses

As demonstrated in Section 5.1, the training of the proposed SN-GAC framework can be divided into two separated steps for different purposes. The initial step is the optimization in an adversarial fashion in order to produce high-quality human-understandable style-neutralized *generated-target* patterns given the style-inconsistent *style-source* counterparts. The following procedure is to fine-tune the classifier with only the fake pair of

<sup>4</sup>Paired input is not evaluated for conventional baselines in Section 5.1.2 since style-neutralization cannot be achieved with traditional approaches, e.g., the kernelized dimension can be infinite with some kernel trick [118].

*style-source* and *generated target* ( $\{x, G(x)\}$ ) to further minimize the recognition error. Both steps are all contributing to the final target to improve the classification performance.

**Initial training for both  $G$  and  $D - C$ :** Both the  $G$  and the  $D - C$  networks are updated in an iterative and alternative fashion.<sup>5</sup>  $G$  is trained by minimizing the summation of losses for both reversing the training of the Wasserstein-based  $D$  model, simultaneously, assigning correct label by  $C$ . Additionally, both the constant loss and the reconstruction loss are also jointly optimized. The final formulation for minimizing loss summation is given as follows:

$$\mathbb{L}_G = (\mathbb{L}_{C_{initial}} - \mathbb{L}_{adv-G}) + \alpha \cdot \mathbb{L}_{l_1} + \beta \cdot \mathbb{L}_{const} \quad (5.1)$$

where  $\alpha$  and  $\beta$  are hyper-parameters. The categorical losses are given as Eq. (5.2) with the conventional cross entropy calculation.  $C(x, x_*)$  and  $C(x, G(x))$  denote the classification logit output of the real and the fake pair respectively.

$$\mathbb{L}_{C_{initial}} = E_{x, x_*} [\log C(x, x_*)] + E_x [\log C(x, G(x))] \quad (5.2)$$

Alternatively, the  $D - C$  network is optimized to minimize the Wasserstein distance between the real and the fake pair. They are given as Definition 5.1.4 and Definition 5.1.6 respectively. Meanwhile, it is also trying to assign the correct class label of the given pair. The combined loss of  $D - C$  is formulated as follows:

$$\mathbb{L}_{D-C} = (\mathbb{L}_{C_{initial}} + \mathbb{L}_{adv-D}) \quad (5.3)$$

In each training iteration,  $G$  is only accessible to one batch of pairs (*Source & Standard-Target*), while two pair batches including (*Source & Standard-Target*) and (*Source & Generated-Target*) are fed into  $D - C$ .

**Fine-tuning for the Classifier  $C$ :** When  $G$  is stabilized and the relevant minimization of losses are saturated, it is believed to be capable of generating high-quality human-understandable style-neutralized *generated-target* patterns when given various *style-source* data. To further improve the final classification performance of the classifier,  $C$  will then be fine-tuned by fixing the  $G$  network while minimizing only the categorical objective  $\mathbb{L}_C$  in an iterative fashion. Be noted that only the fake pairs, as defined in Definition 5.1.6, will be fed into the  $C$  network in this fine-tuning step. Such a setting scheme will force the classifier to be trained by being adapted to the generated samples. The minimizing loss is given as  $\mathbb{L}_{C_{fine-tune}} = E_x [\log C(x, G(x))]$ .

The fine-tuning step will be terminated when  $\mathbb{L}_{C_{fine-tune}}$  is saturated. The whole trained SN-GAC model including both  $G$  and  $D - C$  is ready to be tested and utilized since it is not only able to produce high-quality human-understandable patterns, the attached classifier in  $D - C$  is also capable of assigning correct class labels on these style-neutralized *generated-targets*.

## 5.1.2 SN-GAC Experiments

Two benchmark datasets are involved in order to evaluate the performance and effectiveness of proposed SN-GAC model. One of them is the Point' 04 ([119]), named the Face

<sup>5</sup>As suggested in [24],  $G$  is trained once after  $D - C$  is learned for five times to guarantee the best Wasserstein distance estimation at the current training progress.

Data in short) for face recognition across multiple head poses (yaw angles). The other one is the CASIA handwriting offline database ([121], the HW Data in short) for Chinese handwriting character classification with different handwriting styles written by different individuals. These datasets have already been involved in the demonstration of the [13, 14] evaluation described in Section 4.5.1.

Some of the relevant state-of-the-art machine learning approaches for non-*i.i.d.* data classification tasks are first evaluated. In these models, the field inconsistency issue is not considered. These approaches include the Support Vector Machine (SVM) for classification ([6], SVC for short), as well as the Nearest Class Mean (NCM) [140]. These two classification approaches are performed on both datasets. Two deep CNN models with no *i.i.d.* assumption are also evaluated, including the Vgg-Face model proposed in [41] for the Face Data, and the Alexnet introduced in [1] for the HW Data respectively.

There are also models as baselines where the *i.i.d.* assumption is taken into consideration. However, they are formulated in different directions. The MTL framework where an individual classification model is going to be trained for data with one specific style (seen as a task) will be compared. One MTL-based example is the SVM-based Mean Regularized MTL introduced in [30]. Also, field prediction approaches, which tries to normalize the style information from original input style-inconsistent patterns, are also investigated. These methods include approaches such as one individual case of the Field Bayesian model (F-BM) proposed in [29], namely, the Field Nearest Class Mean (F-NCM), and the Field-SVC (F-SVC) model introduced in [13, 14]. Performance of the SVC-based models is evaluated with both the linear kernel (LK) and the RBF (nonlinear Gaussian kernel, GK for short) kernel.

For each dataset, several state-of-the-art classifiers in the literature are also involved. The Style Mixture Model (SMM) referred to in [19], the Bilinear Model (BM) investigated in [20], and the Fisherface Discriminant Analysis (FDA) introduced in [158] are involved for the Face Data classification task. In the meanwhile, the Modified Quadratic Discriminant Function (MQDF) [139] is conducted for the HW data since it is a state-of-the-art classifier for the character classification task, as demonstrated in [139]. Additionally, following the experimental comparing scheme demonstrated in [29], the field classification version of the QDF model [139], namely, the Field-QDF (F-QDF) model [29], is also involved in the comparison on this database. According to the demonstration in [29], the F-QDF model [29] is another particular case of the F-BM model.

The F-SVC model [13, 14] is compared with the self-learning based style-transferring algorithm introduced in [16] for the Face Data since the styles in testing are unknown during training. According to it, the specific decision rule for the style-transferring scheme is known as the Field Transfer Prediction Rule (FTPR). However, only the field prediction rule (FPR) is implemented in the HW Data as the fact that the testing fields (writers) are available during training.<sup>6</sup> There is no necessity to transfer the trained field information to the styles on the testing data.

The choice of the embedder network  $E$  depends on different sets. Meanwhile, for each set, the style-free *Standard-Target* needs to be specified before the training. They will be demonstrated in details in the following sections. The whole SN-GAC model is

---

<sup>6</sup>Although there exists a style shift between the cursive characters in testing and the isolated examples in training for a specific writer, as will be demonstrated in the following part with regard to the HW evaluation, the writing style seems to be similar since they are written by the identical individual.

built on the Google Tensorflow Deep Learning Library (r1.90) [88]. The implementation code is originated from the Github project Zi2Zi, which can be found in the description of [101].<sup>7</sup> The experimental results are briefly summarized in Table 5.1.

Table 5.1: Performance evaluation of the SN-GAC model and other relevant baselines involved: N/A represents that the specific baseline is not compared on the given set.

Method	Classification Accuracy (%)	
	Face Data ([119])	HW Data ([121])
FDA [158]	69.33%	N/A
SMM [19]	73.33% (Field)	N/A
BM [20]	60.00%	N/A
MQDF [139]	N/A	94.44%
F-QDF [29]	N/A	95.49% (FPR)
NCM [140]	60.00%	94.44%
CNN	90.67% (Vgg-Face-Net [41, 7])	97.22% (AlexNet [1])
F-NCM [29]	78.67% (FTPR)	95.49% (FPR)
SVC [6]	LK	84.00%
	GK	85.33%
MR-MTL [30]	LK	85.33%
	GK	85.33%
F-SVC [13, 14]	LK	<b>100.00% (FTPR)</b>
	GK	<b>100.00% (FTPR)</b>
SN-GAC	<b>100.00%</b>	<b>98.26%</b>

**Face Recognition with Head Yaw Poses:** There are in total 15 individuals involved in the Point’04 Database (the Face Data), as demonstrated in [119]. For each, only the zero pitch pose faces with no horizontal variations are selected. Yaw angles in the range of  $[-90^\circ, +90^\circ]$  partitioned with  $15^\circ$  from each other are chosen. Such experimental setting results in the yaw angle values as:  $[\pm 90^\circ, \pm 75^\circ, \pm 60^\circ, \pm 45^\circ, \pm 30^\circ, \pm 15^\circ, 0^\circ]$ . The identical experimental setting has been used in Section 4.5.1.

Pixel values are initially normalized to  $[-1, 1]$ . Then, they are cropped and resized with  $48 \times 36$  pixels, obtaining 1728-d feature after the vectorization. Following [29], the feature dimension is further compressed to 14 with the FDA model for the NCM [140], and 100 by the Principal Component Analysis (PCA) [150] for other baseline classifiers.<sup>8</sup>

The embedder  $E$  is chosen as a pre-trained Vgg-Face network introduced in [41]. It is fine-tuned with the last fully connected and the first convolutional layers from the original loaded model to be adapted with the Point’04 database. The images are resized to  $256 \times 256$  so that they can be easily incorporated with both the  $E$  and the  $D-C$  network of the proposed SN-GAC model. Furthermore, it is straightforward to select images with zero yaw angles (frontal faces) as style-free *Standard-Targets*.

Images taken of each yaw pose is regarded to be equipped with consistent style information (as each column in Fig. 5.4). The classification is conducted based on individual faces, as examples displayed in the second row in Fig. 5.4. Images from the first eight poses (left eight columns in Fig. 5.4) are put into the training set, while the remaining five ones are placed into the testing set.

<sup>7</sup>The code of the SN-GAC model is available online: <https://github.com/falconjhc/SN-GAC>

<sup>8</sup>The experiment setting, as well as all the experimental results except the proposed SN-GAC model, is referred to [13, 14].



Each column represents a specific head pose (a kind of inconsistent style).

1st Row: Style-inconsistent *Style-Sources* ( $x$ );

2nd Row: Style-free *Standard-Targets* ( $x_*$ );

3rd Row: Style-neutralized *Generated-Targets* ( $G(x)$ );

4th Row: *Difference between  $x_*$  and  $G(x)$* .

Fig. 5.4: Examples of performance visualization on the Point' 04 [119] data with the SN-GAC model.

It can be seen clearly from the second column of Table 5.1 that the proposed SN-GAC model and the F-SVC model [13, 14] with style transfer achieve the zero error rate. However, the F-SVC [13, 14] framework achieves the performance by taking advantage of the test data with a self-training strategy to transfer the trained style to the unseen one, as detailedly demonstrated in [13, 14]. On the contrary, such over-realistic experimental setting is not needed in the proposed SN-GAC framework proposed in this chapter.

Moreover, by looking into the images in the third row of Fig. 5.4, the nonlinearly mapped generated style-neutralized images by the  $G$  model of the proposed SN-GAC framework can be readily understood by human observers. Only tiny and insignificant defects can be found when it is carefully checked. It recovers dominantly what the corresponding real style-free standard data should look like. The difference between the style-neutralized *generated-targets* (3rd row in Fig. 5.4) and the style-free *standard-targets* (2nd row in Fig. 5.4) tends to be zero as well. It can be seen in the last row in Fig. 5.4.

In comparison, in the F-SVC model [13, 14], the obtained standard-normalized images may usually be less similar to a real style-free image, as examples given in Fig. 5.5. Additionally, it cannot be well understood by ordinary individuals. Furthermore, the F-SVC model [13, 14] can only produce style-normalized data by the linear kernel, since the fact that the nonlinear kernelized dimension can be extremely large, or even infinite, as told in [118]. However, only the linear kernels cannot represent sufficiently multiple, diverse, and complicated style information in real scenarios.

**Chinese Handwriting Classification across Writers:** The CASIA handwriting database (offline version, the HW Data in short) instructed in [121] is also exploited for evaluation of the proposed SN-GAC model while being compared with other relevant baselines. The original dataset includes 3,755 categories of different Chinese characters. It covers all the materials in the GB-2312 Level 1 set where commonly used characters in Chinese daily lives are collected [91].

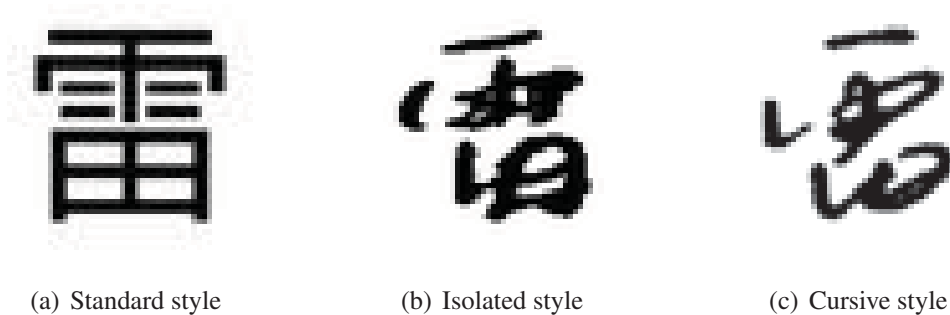
As described in [29], handwriting Chinese characters written by in total 100 writers (no.1,101 - no.1,200) are involved in this experiment for evaluation. For simplicity, only



1st Row: Style-inconsistent patterns;  
 2nd Row: Style-normalized patterns;  
 3rd Row: Style information (difference between the 2nd row and the 1st row).

Fig. 5.5: Performance evaluation of the F-SVC model ([13, 14]) as the identical example given in Fig. 5.4

the first 30 characters are chosen in this experiment. Additionally, following the experimental setting described in [29, 13, 14], the isolated character set is chosen as the training set (CASIA-HWDB-1.1), while the cursive text set (CASIA-HWDB-2.1) is used for testing. Such a setting scheme results in 2995 training examples and 288 testing characters. It is noted that each testing sample shares a particular training style. However, there is still style variation between an isolated example and a cursive character, as seen in Fig. 5.6. Strokes and radicals seem to be connected in a calligraphic fashion with each other in a cursive written character (Fig. 5.1.2 (c)). On the contrary, in an isolated written one (Fig. 5.1.2 (b)), they are much clearer and independent with each other. It leads to the overall looking of an isolated written character becoming closer to the standard printed font style (Fig. 5.1.2 (a)).<sup>9</sup>



(a) Standard style

(b) Isolated style

(c) Cursive style

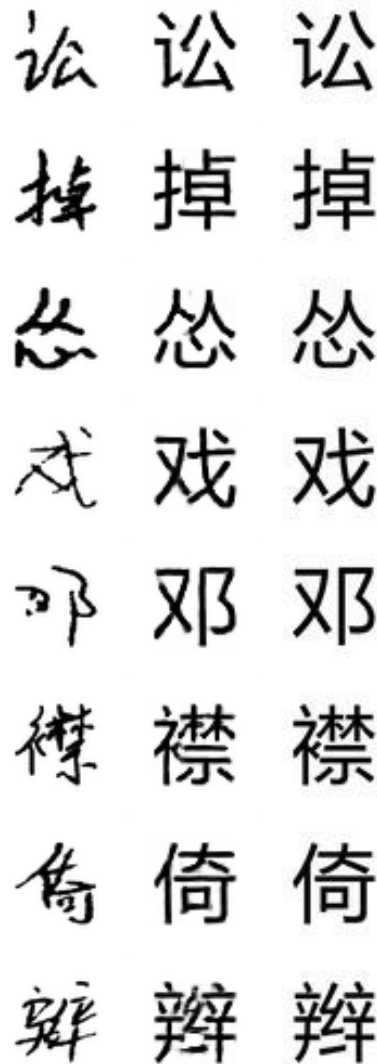
Fig. 5.6: An example of a handwriting Chinese character with standard, isolated, and cursive writing styles.

As seen from the last column of Table 5.1, the proposed SN-GAC model attains the best recognition performance. It is even higher than the state-of-the-art F-SVC framework [13, 14] for such non-*i.i.d.* classification task. The most significant achievement can be then found that the style neutralization mapping is even adapted to the style shift

<sup>9</sup>The 'Heiti' font.



from isolated to cursive handwriting variation in an intelligent fashion. Such an appealing property can hardly be seen in the previous research literature. Figure 5.7 give some examples of the correctly classified generated targets.

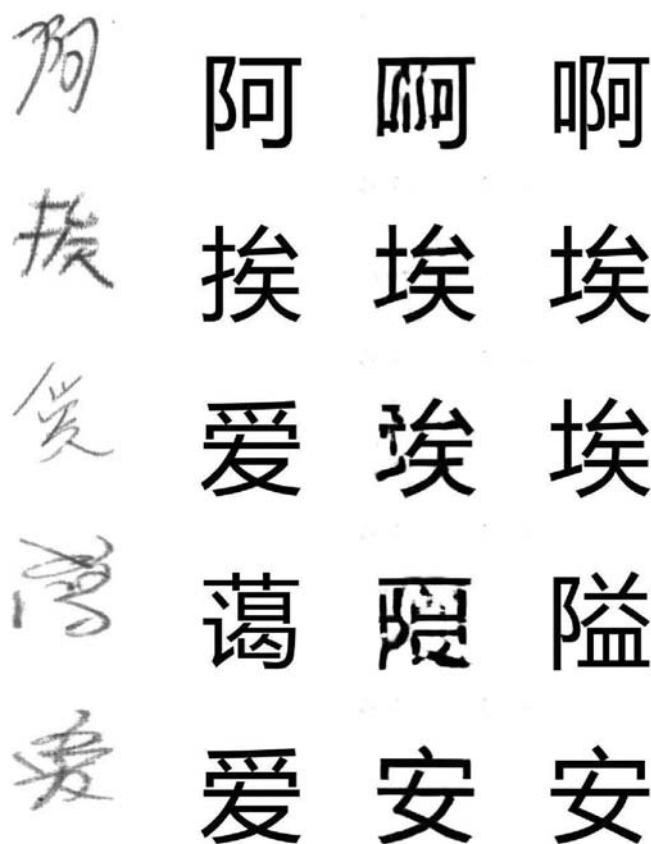


1st Column: *Source* ( $x$ );  
 2nd Column: *Generated-Target* ( $G(x)$ );  
 3rd Column: *Real-Target* ( $x$ );

Fig. 5.7: Performance Visualizatton on the HW data produced by the SN-GAC model.

However, there are still examples that are inappropriately neutralized or falsely classified. By examining those incorrect examples given in Fig. 5.8 a step further, it can be observed that most of the error comes from the confusing and cursive written style of the *style-source* character. Some of them are even too difficult to be recognized for a well educated Chinese people. In this case, the  $G$  network would generate unclear (the forth row of Fig. 5.8) or even incorrect (e.g., the first row of Fig. 5.8) *Generated-Target* examples. However, what makes it desired is that even if  $G$  does not perform well, the  $D - C$

may still able to produce a reliable classification assignment based on the generated style-neutralized samples. It proves the robustness and the reasonableness of the classification scheme of the proposed SN-GAC framework.



1st Column: *Source* ( $x$ );  
 2nd Column: *Standard-Target* ( $x_*$ );  
 3rd Column: *Generated-Target* ( $G(x)$ );  
 4th Column: class label assigned by  $D - C$  from the generated sample in the 3rd Column.

Fig. 5.8: Some incorrectly classified examples on the HW data produced by the SN-GAC model.

### 5.1.3 Summary and Future Work

A novel classification framework, named Style Neutralized Generative Adversarial Classifier (SN-GAC), based on the upgraded network structure of the U-Net framework, is proposed in this chapter. Trained adversarially according to the Generative Adversarial Network (GAN) framework, it is designed to neutralize diverse and inconsistent style information from the original style-discriminative data by mapping them to patterns with the standard style-free counterparts. The obtained style-neutralized examples are equipped with the identical style information, which satisfies the *i.i.d.* assumption that conventional machine learning models require. Aiming at promoting the final recognition accuracy, it trains no extra classification model except the SN-GAC framework itself. Em-

pirical experiments on two benchmark datasets have demonstrated that the proposed SN-GAC model not only achieves the highest classification performance so-far but taking no advantage of the test data during training with the self-training strategy. It generates high-quality human-understandable style-neutralized patterns in the meanwhile. Such appealing property can hardly be seen in the previous research literature.

### **Future Work**

The SN-GAC will be interrogated by evaluating the classification performance on the larger number of classes. The neural network based models can easily be extended for large-scale classification scenarios [1] with the huge number of classes. The proposed SN-GAC framework will be in this way tested for classification on the GB-2312 L1 set [91], as have been planned in Section 4.8.1. In that set, in total 3,755 kinds of Chinese characters are involved. Such classification task can hardly be handled by the SVM-based framework including the F-SVC [13, 14] model proposed in Chapter 4 because of the nature of the voting strategy [126, 127] to extend the binary SVC model for multi-class classification scenarios [36].

## 5.2 W-Net for Few-shot Multi-content Arbitrary-style Chinese Character Generation

Due to the huge category number, the sophisticated combinations of various strokes and radicals, and free writing or printing styles, generating Chinese characters with arbitrary styles is always considered as a difficult task. Several previous research has been made to accomplish the generation task. However, most of them are neither inefficient to be utilized, nor unlikely to produce high-quality characters. Moreover, huge numbers of training examples are always necessary to produce high-quality and verisimilar examples.

In this chapter, an efficient and generalized deep framework, namely, the W-Net, is introduced for the Few-shot Multi-content Arbitrary-style Chinese Character Generation (FMACCG) task. Specifically, given a few character (few-shot) with a specific style (e.g., a printed font or handwriting style), a well adversarially trained [2] W-Net model is capable of generating any characters sharing the style similar to the given few. It is even possible for the One-shot arbitrary-style Chinese character generation task when only one single style example is available. Such appealing property was rarely seen in the literature.

The proposed W-Net framework has been evaluated and compared against many other competitive methods extensively. Experimental results have demonstrated that the proposed method is significantly superior in the one-shot or few-shot setting. Above all, it is also a potentially interesting and useful application in this Internet era. Several of the further commercial and technical utilization for both entertainments, teaching activities and scientific research will in this way be enabled.

### 5.2.1 Research Background

Chinese is a special language with both messaging functions and artistic values. Besides, it contains thousands of different categories or over 10,000 different characters among which 3,755 characters, defined as level-1 characters [91]. These characters are commonly used ones in contemporary China, Singapore, and other relevant nations, regions in eastern Asia and all around the world.

Given a limited number of Chinese characters or even one single character with a specific style (e.g., personalized handwriting calligraphy or a stylistic printing font), it is interesting to mimic automatically many other characters with the same specific style. This topic is very difficult and less studied simply because of the large category number of different Chinese characters with various styles. This problem is even harder due to the unique nature of Chinese characters among which each is a combination of various strokes and radicals with diverse interactive structures. In this section, aiming to generate Chinese characters when given few shot samples with a specific arbitrary style (seen or unseen in training), we propose a novel deep model named W-Net. As a generalized style transformation framework, it better solves the above-mentioned drawbacks and could be easily used in empirical practice.

Particularly, inherent from the U-Net framework [10] for the one-to-one  $\text{Img2Img}$  translation task [21], the proposed W-Net employs two parallel convolution-based encoders to extract style and content information respectively. The generated image will be obtained by the deconvolution-based decoder with the encoded information. Short-cut connections [10] and multiple residual blocks [8] are set to deal with the gradient van-

ishing problem and balance information from both encoders to the decoder. The training of the W-Net follows an adversarial manner. Inspired by the recently proposed Wasserstein Generative Adversarial Network (W-GAN) [11] framework with gradient penalty (W-GAN-GP) [24], an independent discriminator ( $D$ )<sup>10</sup> are employed to assist the W-Net ( $G$ ) learning. Several examples of both one-shot and few-shot Chinese character generation task are given in Fig. 5.9. With such a proposal, the data synthesizing tasks with few samples, or even one single example available, can be fulfilled much more readily and effectively than previous approaches in the literature.

## 5.2.2 Model Definition

Some notations and preliminaries that are necessary to demonstrate the proposed W-Net architecture will be given firstly in this section. It is followed by the model structure, training strategy, and optimization details.

### Preliminaries

Denote  $X$  be a Chinese character dataset, consisting of  $J$  different characters with in total  $I$  different fonts.

**Definition 5.2.1.** Let  $x_j^i$  be a specific sample in  $X$ , regarded as the *real target*. Following [13, 14, 15], the superscript  $i \in [1, 2, \dots, I]$  represents  $i$ -th character style, while the subscript  $j \in [1, 2, \dots, J]$  denotes  $j$ -th character content.

**Definition 5.2.2.** Denote  $x_j^{c_m}, c_m \in [1, 2, \dots, I], m = 1, 2, \dots, M$  as the set of *content prototypes*. It describes a content of the  $j$ -th character, the same as the content of  $x_j^i$  defined in Definition 5.2.1.

Specifically, during the training,  $M$  different styles are to be pre-selected. Commonly, they are out of  $[1, 2, \dots, I]$  fonts for the real target.

**Definition 5.2.3.** Denote  $x_{s_n}^i, s_n \in [1, 2, \dots, J], n = 1, 2, \dots, N$  to be a set of *style references* equipping with the  $i$ -th style information, identical to the style of  $x_j^i$  defined in Definition 5.2.1.

Be noted that generally,  $i$  and  $c_m$  are different, while  $j$  differs from  $s_n$  as well. The same as  $M$ , the number of style references  $N$  should also be determined before the model is trained.<sup>11</sup> In the proposed W-Net framework, each  $x_j^i$  is assumed to be combined with  $j$ -th content information from those prototypes of  $x_j^{c_m}$  and the  $i$ -th writing style learned from references  $x_{s_n}^i$ , where  $m \in [1, 2, \dots, M]$  and  $n \in [1, 2, \dots, N]$  follow the definitions given in Definition 5.2.2 and Definition 5.2.3 respectively.

**Definition 5.2.4.** The proposed W-Net model will produce the **generated target** by taking content prototypes ( $x_j^{c_1}, x_j^{c_2}, \dots, x_j^{c_M}$ , as defined in Definition 5.2.2) and style references ( $x_{s_1}^i, x_{s_2}^i, \dots, x_{s_N}^i$ , as defined in Definition 5.2.3) simultaneously. The corresponding *generated target* is denoted as  $G(x_j^{c_1}, x_j^{c_2}, \dots, x_j^{c_M}, x_{s_1}^i, x_{s_2}^i, \dots, x_{s_N}^i)$ <sup>12</sup> by taking both

<sup>10</sup>The discriminator actually attaches an auxiliary classifier proposed in [26] to assign the style label of the input character.

<sup>11</sup>The proposed framework reported in [18] can be regarded as a special case of the few-shot (Few-shot Multi-content Arbitrary-style Chinese Character Generation, FMACCG) setting when  $M = 1$  and  $N = 1$ . It is named as the **One-shot Single-content Arbitrary-style Chinese Character Generation (OSACCG)**.

<sup>12</sup>The **generated target** will be noted as  $G(x_j^{c_m}, x_{s_n}^i)$  for simplicity in the following sections.

滾 滾 長 江 東 逝 水 浪  
 花 淘 盡 英 雄 是 非 成  
 敗 轉 頭 空 青 山 依 舊  
 在 幾 度 夕 陽 紅 白 發  
 漁 樵 江 諸 上 慣 看 秋  
 月 春 風 壺 壺 濁 酒 喜  
 相 逢 古 今 多 少 事 都  
 付 笑 談 中

(a) Printing Font No.77

滾 滾 長 江 東 逝 水 浪  
 花 淘 盡 英 雄 是 非 成  
 敗 轉 頭 空 青 山 依 舊  
 在 幾 度 夕 陽 紅 白 發  
 漁 樵 江 諸 上 慣 看 秋  
 月 春 風 壺 壺 濁 酒 喜  
 相 逢 古 今 多 少 事 都  
 付 笑 談 中

(b) Handwriting Style No.1296

滾 滾 長 江 東 逝 水 浪  
 花 淘 盡 英 雄 是 非 成  
 敗 轉 頭 空 青 山 依 舊  
 在 幾 度 夕 陽 紅 白 發  
 漁 樵 江 諸 上 慣 看 秋  
 月 春 風 壺 壺 濁 酒 喜  
 相 逢 古 今 多 少 事 都  
 付 笑 談 中

(c) Printing Font No.78

滾 滾 長 江 東 逝 水 浪  
 花 淘 盡 英 雄 是 非 成  
 敗 轉 頭 空 青 山 依 舊  
 在 幾 度 夕 陽 紅 白 發  
 漁 樵 江 諸 上 慣 看 秋  
 月 春 風 壺 壺 濁 酒 喜  
 相 逢 古 今 多 少 事 都  
 付 笑 談 中

(d) Handwriting Style No.1298

裁 勃 幸 罢

禍 從 滅 息

Fig. 5.9: Generated Chinese characters synthesized by the W-Net model: **With few or even one single sample(s) available** (the right-bottom character with red boxes). The ancient Chinese poetry shown is titled as *The Yangtse River flows to the oriental lands*. It was originally written by Shen YANG in the Chinese Ming Dynasty (Approx. 1500s A.D.).

$\{x_j^{c_m}, c_m \in [1, 2, \dots, I], m = 1, 2, \dots, M\}$  and  $\{x_{s_n}^i, s_n \in [1, 2, \dots, J], n = 1, 2, \dots, N\}$  simultaneously.

The target of the W-Net training is to make the *generated target*  $G(x_j^{c_m}, x_{s_n}^i)$  closed to the *real target*  $x_j^i$  in both the content information and the style tendency. In the FMACCG task, the given few style samples (E.g.,  $x_{p_l}^h, l = 1, 2, \dots, L$ , where  $h$  can be any arbitrary style, meanwhile  $p_l$  could be any single character and  $L$  can be a small number, e.g.,  $L = 4$ .) are seen as the **few-shot** style references. The task can be readily fulfilled by feeding a set of content prototypes ( $x_q^{c_m}, m = 1, 2, \dots, M$ ) of the desired  $q$ -th character on

condition of those relevant outputs of the style reference encoder  $Enc_r(x_{p_1}^h, x_{p_2}^h, \dots, x_{p_L}^h)$ <sup>13</sup> to produce  $G(x_q^{c_m}, x_{p_l}^h)$  with the given the few style examples  $(x_{p_l}^h, l = 1, 2, \dots, L)$ .

In such setting, alternating  $q$  will lead to synthesizing different characters. Simultaneously, all the generated examples are expected to imitate the  $h$ -th style information given by  $x_{p_l}^h, l = 1, 2, \dots, L$ . Both  $p_l$  and  $q$  could also be out of  $[1, 2, \dots, J]$ , while  $h$  can be not in the range of  $[1, 2, \dots, I]$  or  $[c_1, c_2, \dots, c_M]$ .

## W-Net Architecture

Fig. 5.10 illustrates the basic structure of the proposed W-Net model.<sup>14</sup> It consists of the content prototype encoder ( $Enc_p$ , the upper part with vertical lines), the style reference encoder ( $Enc_r$ , the lower pattern with horizontal lines), and the decoder ( $Dec$ , the middle part with both vertical and horizontal lines).

The  $Enc_p$  and  $Enc_r$  are constructed as sequences of convolutional layers, where  $5 \times 5$  filters with fixed stride 2 and ReLU function are implemented.<sup>15</sup> By this setting,  $M$   $64 \times 64$  prototypes  $x_j^{c_m}$  ( $c_m \in [1, 2, \dots, I], m = 1, 2, \dots, M$ ) and  $N$  references  $x_{s_n}^i$  ( $s_n \in [1, 2, \dots, J], n = 1, 2, \dots, N$ ) will all be mapped into two  $1 \times 512$  feature vectors, denoted as  $Enc_p(x_j^{c_1}, x_j^{c_2}, \dots, x_j^{c_M})$  and  $Enc_r(x_{s_1}^i, x_{s_2}^i, \dots, x_{s_N}^i)$  respectively.<sup>16</sup>

However, there are major difference between the  $Enc_p$  and  $Enc_r$ . The input of the  $Enc_p$  are  $M$  content prototype images. They are concatenated on the image channel before being sent into the decoder. In this way, the number of channels to input  $Enc_p$  is actually  $M$  (since characters are represented by single-channel gray-scale images), where  $M$  is a fixed hyper-parameter before the implementation.

On the contrast,  $N$  input style references would produce  $N$  features on each level. It means that the weights in the style encoder  $Enc_r$  are shared among these  $N$  references. Then the average, maximum, and minimum operations on different  $N$  features corresponding to  $N$  style references are performed. The three results are then concatenated with each other. They are then sent to the following network architecture including the residual blocks and the short-cuts. In this sense,  $N$  is a number that can be alternated flexibly according to both the training conditions and the application scenarios.

Identical to the decoder in the U-Net framework [10],  $Dec$  is designed as a deconvolutional progress layer-wisely connected with  $Enc_p$  and  $Enc_r$ . It produces a generated image, the size of which is consistent with all the input images of both encoders. Specifically, for higher-level features between the decoder and the both encoders, connections are achieved by simple feature shortcut. For lower-level layers of  $Enc_p$ , a series of residual blocks<sup>17</sup> [8] are applied and connected to the  $Dec$ . The number of blocks is controlled by a super parameter  $K$ . On the contrast, as the writing style is a kind of high-level deep feature, there is only one residual block connection (with  $K$  blocks) between  $Enc_r$  and  $Dec$ , omitting lower-level feature concatenation at the same time.

<sup>13</sup>The output of the style reference encoder  $Enc_r(x_{p_1}^h, x_{p_2}^h, \dots, x_{p_L}^h)$  is to be connected to the  $Dec$  with both shortcut or residual block connections. It will be demonstrated in details in the following paragraphs

<sup>14</sup>The code of the W-Net model can be downloaded via the Github service: <https://github.com/falconjhc/W-Net>

<sup>15</sup>The  $\tanh$  rather than the ReLU nonlinearity is implemented in the last layer of the decoder  $Dec$ .

<sup>16</sup>The two encoded output will be noted as  $Enc_p(x_j^{c_m})$  and  $Enc_r(x_{s_n}^i)$  for simplicity in the following.

<sup>17</sup>The structure of the residual block follows the setting in [103]

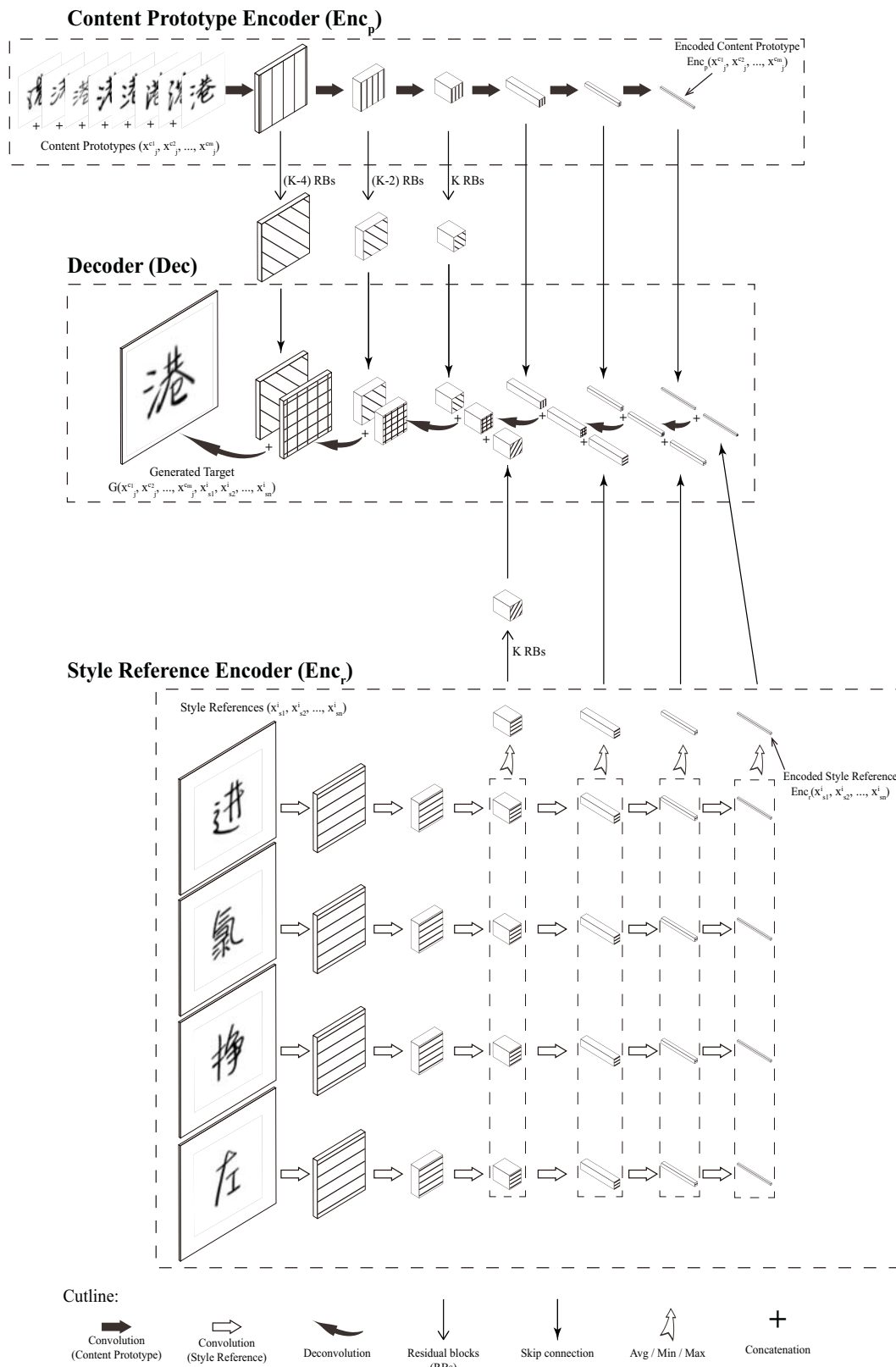


Fig. 5.10: W-Net Architecture.



## Optimization Strategy and Losses

The proposed W-Net is trained adversarially based on the Wasserstein Generative Adversarial Network with the Gradient Penalty (W-GAN-GP) framework [11], in which framework the W-Net is regarded as the generator  $G$ . Specifically, it takes the content prototypes and the style references, and returns generated target. It is formulated as  $G(x_j^{c_m}, x_{s_n}^i) = Dec(Enc_p(x_j^{c_m}), Enc_r(x_{s_n}^i))$ , which is optimized to be closed to  $x_j^i$ . The measurement of the closeness is determined by several reconstruction, perceptual, and the adversarial losses given by the discriminative model  $D$  in the WGAN-GP framework. They will be demonstrated in the following paragraphs.

**Training Strategy:** The learning of the W-Net follows the adversarial training scheme. In each learning iteration, there are two independent procedures, including the  $G$  training and the  $D$  training respectively. The  $G$  and the  $D$  are trained to optimize Eq. (5.4) and Eq. (5.5) respectively.

$$\begin{aligned} \mathbb{L}_G = & -\alpha \mathbb{L}_{adv-G} + \beta_d \mathbb{L}_{dac} + \beta_p \mathbb{L}_{enc-p-cls} + \beta_r \mathbb{L}_{enc-r-cls} \\ & + \lambda_{pixel} \mathbb{L}_{pixel} + \mathbb{L}_{\phi_{total}} + \psi_p \mathbb{L}_{Const_p} + \psi_r \mathbb{L}_{Const_r} \end{aligned} \quad (5.4)$$

$$\mathbb{L}_D = \alpha \mathbb{L}_{adv-D} + \alpha_{GP} \mathbb{L}_{adv-GP} + \beta_d \mathbb{L}_{dac} + \beta_p \mathbb{L}_{enc-p-cls} + \beta_r \mathbb{L}_{enc-r-cls} \quad (5.5)$$

**Adversarial loss:**  $G$  optimizes  $\mathbb{L}_{adv-G} = D(x_j^{c_{m'}}, G(x_j^{c_m}, x_{s_n}^i), x_{s_{n'}}^i)$ , while  $D$  minimizes  $\mathbb{L}_{adv-D} = D(x_j^{c_{m'}}, x_j^i, x_{s_{n'}}^i) - D(x_j^{c_{m'}}, G(x_j^{c_m}, x_{s_n}^i), x_{s_{n'}}^i)$ .  $m'$  and  $n'$  are randomly sampled from  $[1, 2, \dots, M]$  and  $[1, 2, \dots, N]$  respectively for each training example  $x_j^i$ . Be noted that a gradient penalty is set as  $\mathbb{L}_{adv-GP} = \|\nabla_{\hat{x}} D(x_j^{c_{m'}}, \hat{x}, x_{s_{n'}}^i) - 1\|_2$  [24] to satisfy the Lipschitz continuity condition required by the Wasserstein-based adversarial training [11], where  $\hat{x}$  is uniformly interpolated along the line between  $x_j^i$  and  $G(x_j^{c_m}, x_{s_n}^i)$ .

**Categorical loss of the discriminator auxiliary classifier:** Inspired by [26], the auxiliary classifier on the discriminator is optimized by  $\mathbb{L}_{dac} = \left[ \log C_{dac}(i|x_j^{c_{m'}}, x_j^i, x_{s_{n'}}^i) \right] + \left[ \log C_{dac}(i|x_j^{c_{m'}}, G(x_j^{c_m}, x_{s_n}^i), x_{s_{n'}}^i) \right]$ .

**Constant Losses of the encoders:** The constant losses [77] are also employed for both encoders. They are given by  $\mathbb{L}_{Const_p} = \|Enc_p(x_j^{c_m}) - Enc_p(G(x_j^{c_m}, x_{s_n}^i))\|^2$  and  $\mathbb{L}_{Const_r} = \|Enc_r(x_{s_n}^i) - Enc_r(G(x_j^{c_m}, x_{s_n}^i))\|^2$  respectively for  $Enc_p$  and  $Enc_r$ .

**Categorical Losses on both Encoders:** To ensure the specific functionalities of the two encoders, we forced the content and style features extracted by them to be equipped with the corresponding commonality separately for the same kind. It is designated by adding a fully-connecting mappings at the end of both encoders to implement the category classification task. It leads to that the both encoders will learn their own representative features, simultaneously over-fitting is avoided thereby.

$\theta_p$  and  $\theta_r$  are used to denote the fully-connecting and softmax functions together for both output feature vectors of both encoders respectively, while the classifications are noted as  $C_{encp}$  and  $C_{encr}$ . The upon-mentioned cross entropy losses of both classifications of different contents and styles are given as  $\mathbb{L}_{enc-p-cls} = [\log C_{encp}(j|\theta_p(Enc_p(x_j^0)))]$  and  $\mathbb{L}_{enc-r-cls} = [\log C_{encr}(i|\theta_r(Enc_r(x_k^i)))]$  respectively. Be noted that  $i$  and  $j$  represent the specific style and the character labels.

**Pixel Reconstruction Losses:** It represents the difference on the image pixel level. It is measured by two variations including the L1 difference and the von-Neumann divergence proposed in [159]. Both of them are penalized by the same scale. It is given by

$\mathbb{L}_{pixel} = \mathbb{L}_1 + \mathbb{L}_{vN-pixel} = \|(x_j^i - G(x_j^{c_m}, x_{s_n}^i))\|_1 + \text{tr}(A)$ , where  $A = G(x_j^{c_m}, x_{s_n}^i) \cdot \log[G(x_j^{c_m}, x_{s_n}^i)] - G(x_j^{c_m}, x_{s_n}^i) \cdot \log(x_j^i) - G(x_j^{c_m}, x_{s_n}^i) + x_j^i$ .

**Deep Perceptual Losses:** The deep perceptual loss minimizes the variation between the generated target and the corresponding input images by calculating the difference of the high-level features. There are three deep perceptual losses involved. The total perceptual loss is calculated as  $\mathbb{L}_{\phi_{total}} = \lambda_{real} \cdot \mathbb{L}_{\phi_{real}} + \lambda_{content} \cdot \mathbb{L}_{\phi_{content}} + \lambda_{style} \cdot \mathbb{L}_{\phi_{style}}$ . All the composing losses, namely,  $\mathbb{L}_{\phi_{real}}$ ,  $\mathbb{L}_{\phi_{content}}$ , and  $\mathbb{L}_{\phi_{style}}$ , are calculated following the equation:  $\mathbb{L}_{\phi} = \sqrt{\sum_{\phi} [\phi(x) - \phi(G(x_j^{c_m}, x_{s_n}^i))]^2} + \text{tr}(B)$ , where  $B = G(x_j^{c_m}, x_{s_n}^i) \cdot \log[G(x_j^{c_m}, x_{s_n}^i)] - G(x_j^{c_m}, x_{s_n}^i) \cdot \log(x) - G(x_j^{c_m}, x_{s_n}^i) + x$  with the von-Neumann divergence in [159].

$\mathbb{L}_{\phi_{real}}$  are measured by letting  $x = x_j^i$ , while  $\phi_{real}$  is a VGG-16 network [7] trained with multiple images by classifying both of characters and writing styles (or printed fonts). Five convolutional features including  $\phi_{1-2}$ ,  $\phi_{2-2}$ ,  $\phi_{3-3}$ ,  $\phi_{4-3}$ ,  $\phi_{5-3}$  are involved. Lower  $\mathbb{L}_{\phi_{real}}$  represents higher similarity between the real target  $x_j^i$  and the generated target  $G(x_j^{c_m}, x_{s_n}^i)$ .

On the contrary,  $\mathbb{L}_{\phi_{content}}$  is minimized by setting  $x = x_j^{c_{m'}}$ , where  $m'$  is a random sample from the range of  $[1, 2, \dots, M]$ . It means  $c_{m'} \neq c_m$ , resulting in that  $x_j^{c_{m'}}$  and  $G(x_j^{c_m}, x_{s_n}^i)$  are different on styles. However, they share the same  $j$ -th character content. The VGG-16 network  $\phi_{content}$  is trained by simply classifying the character contents, while  $x_j^{c_{m'}}$  and  $G(x_j^{c_m}, x_{s_n}^i)$  shall enclosure different handwriting or printed fonts. Nevertheless, they should be close to each other on the deep features of  $\phi_{content}$  thereafter. Because of the fact that the input images are completely different on the writing style, lower-level deep features might experience larger variation comparing with those of  $\phi_{real}$ . In this way, only  $\phi_{4-3}$ ,  $\phi_{5-3}$  features are calculated in order to merely minimize the high-level abstract features difference.

Similarly,  $\mathbb{L}_{\phi_{style}}$  is minimized by setting  $x = x_{s_n}^i$ , where  $n'$  is a random sample from the range of  $[1, 2, \dots, N]$ , namely,  $s_n \neq s_{n'}$ . It represents that  $x_{s_n}^i$  and  $G(x_j^{c_m}, x_{s_n}^i)$  are different on character contents, but they are the same on the  $i$ -th writing style.  $\phi_{style}$  is learned by minimizing the cross entropy to well recognize character styles. High-level extracted patterns should be similar between  $x_{s_n}^i$  and  $G(x_j^{c_m}, x_{s_n}^i)$  on  $\phi_{style}$  hereby. Similarly, only  $\phi_{4-3}$ ,  $\phi_{5-3}$  features are involved.

### 5.2.3 W-Net Experiment

A series of experiments have been conducted to verify the effectiveness of the proposed W-Net network. Both printed fonts and handwriting styles are evaluated. Several relevant baselines are also referred to for the comparison as well.

As demonstrated in previous paragraphs, only the special case of the proposed Few-shot Multi-content Arbitrary-style Chinese Character Generation (FMACCG) task is majorly instructed in this section. That is to say, the One-shot Single-content Arbitrary-style Chinese Character Generation (OSACCG) task will be evaluated. In the setting of the OSACCG task, only a single content prototype character ( $M = 1$ ) and single style reference character ( $N = 1$ ) are available. Such architecture has been demonstrated in detail in the conference paper [18] and will be introduced in the following sections.

## Experiment Setting

80 fonts are chosen in standard Chinese printed font database. 50 of them, each containing 3,755 level-1 simplified Chinese characters, are involved in the training set. The offline version of both CASIA-HWDB-1.1 (for simplified isolated characters) and the CASIA-HWDB-2.1 (for simplified cursive characters) [121] are involved as the handwriting dataset. Characters written by 50 writers (No. 1,101 to 1,150) are selected as the training set, resulting in total 249,066 samples (4,980 examples per writer averagely). For both sets, the testing data are chosen due to different evaluation purposes. *HeiTi* (boldface font) is used as the single content prototype font for both the sets. Several examples of characters used as the content prototypes are given in Fig. 5.11(a).

Baseline models include two upgraded version of the Zi2Zi [101] framework. They are modified for the one-shot or the few-shot new-coming style synthesization task. One utilizes a fine-tuning strategy (noted as **Zi2Zi-V1**), where the style information is assumed to be the linear combination of multiple known styles represented by the fixed Gaussian-noise based categorical embedding. The other (**Zi2Zi-V2**) discards the categorical embedding by introducing the final softmax output of a pre-trained VGG-16 network (embedder network), identical to the one employed to calculate the deep perceptual loss for the W-Net training. All the other network architectures and training settings of these baselines are all the same by following [101]. Characters from both databases are represented by  $64 \times 64$  gray-scale images, after which they are then binarized. One thing to be particularly noted is that both the proposed W-Net and the Zi2Zi-V2 follow the **one-shot** setting, where only a single style example ( $x_p^m$ ) is referred to during the evaluation process. However, the Zi2Zi-V1 employ the few-shot (32 references) scheme in order to obtain a valid fine-tuning performance.

The Adam optimizer with  $\beta_1 = 0.5$  and  $\beta_2 = 0.999$  is implemented, while the initial learning rate is set to be 0.0005 and decayed exponentially in each epoch. The architecture of  $D$  follows the setting of the Zi2Zi framework [101] with the W-GAN-GP framework. For the sake of speeding up and stabilizing the training progress, the batch normalization [94] is applied several layers to the  $G$  network, while the layer normalization [11, 160] is selected for  $D$ . Dropout [144] trick is also applied to both  $G$  and  $D$  to improve the generalization performance. Weight decay [161] is as well engaged as additional regularization to avoid the over-fitting issue. The proposed W-Net framework and other baselines are implemented with the Tensorflow (r1.9) deep learning framework [88].

## Model Reasonableness Evaluation

In this section, the reasonableness of the proposed W-Net model is verified by setting  $p = q$  for content  $x_q^0$  and the style  $x_p^m$ . Hereby, the reference is exactly the real target ( $x_p^m = x_q^m$ ). For each evaluation, as previously instructed, only single style reference ( $x_p^m$ , characters of 2nd rows in (b)-(e) of Fig. 5.11) is engaged. The generated image is seen to follow the style tendency of the one-shot reference if the proposed W-Net is capable of reconstructing the extracted style information in the reference image  $x_p^m$ .

Fig. 5.11 illustrates several examples of the comparison result for synthesizing unseen styles during training. It can be observed that styles of both printed and handwriting types are learned and transferred to the given prototypes by the W-Net model with the proper performance by maintaining the style consistency.

赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨

(a) Prototype

赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨  
赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨

(b) Printing font No. 62

赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨  
赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨

(c) Printing font No. 77

赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨  
赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨

(d) handwriting style No. 1293

赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨  
赵 钱 孙 李 周 吴 郑 王 冯 陈 卫 蒋 沈 韩 杨

(e) handwriting style No. 1295

Fig. 5.11: Several examples of generated data of unseen printing and handwriting styles: (a): The input content prototypes; (b)-(e): 1st row: generated characters; 2nd row: corresponding style references (ground truth characters). The written characters are several of the most commonly-used family names in modern China.

### Model Effectiveness Evaluation

The effectiveness of W-Net is tested by generating commonly-used Chinese characters (simplified and traditional) with alternative styles. In this setting,  $x_p^m$  are randomly selected one-shot character with the  $m$ -th style information to imitate the real application scenario, while  $q$  are referred to the desired content prototypes to be generated. In the most common scenarios,  $p \neq q$ .

Fig. 5.12 and Fig. 5.13 list several examples of the generated images by W-Net and two baselines for seen and unseen styles during training respectively. Be noted that only the simplified Chinese characters are accessible during training. Characters in the left four columns of each subfigure in Fig. 5.12 are those simplified ones. Traditional ones are shown in the remaining four columns with no ground truth in the specified database.<sup>18</sup>

When generating characters with a specific seen style during training, it can be intuitively observed in Fig. 5.12 that even given one-shot style reference, the generated

<sup>18</sup>They are the same in Fig. 5.13



Fig. 5.12: Several examples of generated characters of seen styles: (a), (c) and (e) are printing fonts; (b), (d) and (f) are handwriting styles. In each figure, 1st row: ground truth characters (with blue boxes) and the one-shot style reference (with red boxes); 2nd: W-Net generated characters; 3rd row: Zi2Zi-V1 performance; 4th row: Zi2Zi-V2 performance.

fonts by W-Net look very similar to the corresponding real targets. Differently, under the few-shot setting, the Zi2Zi-V1 still produces blurred images, while Zi2Zi-V2 seems to synthesize characters with the averaged style. The proposed W-Net outperforms others by producing characters with both desired contents and consistent styles with only one-shot style reference available.

Simultaneously, acceptable generations can still be obtained from Fig. 5.13 by the proposed scheme when constructing unseen styles with one-shot style reference as well. Though the generated samples are not similar enough as that in previous examples, a clear stylistic tendency can still be clearly observed. On the contrast, Zi2Zi-V1 failed to produce high-quality images even 32 references are given for the fine-tuning due to the over-fitting issue. At the same time, the Zi2Zi-V2 failed to generate distinguishable styles. A possible reason is that it is only capable of learning styles from the original basis provided by the embedder network (VGG-16).



Fig. 5.13: Several examples of generated characters of unseen styles.

### Generation Performance Variation due to Different Numbers of Style References

As a further investigation of the character synthesization performance proposed W-Net model, several additional evaluations have been fulfilled by alternating the numbers of style references that are sent to the style reference encoder. Examples are given in Table 5.2, where the specific font is Printing Font No. 21.

From the table, it can be clearly seen that the generated performance will not be affected hugely due to inputting different style references. Compared to the real characters in the last row, one can also conclude that the generated characters are mostly following the writing styles of the specific printing font. Such finding also proves the effectiveness of the synthesizing performance of the proposed W-Net.

### Analysis on Failure Examples

The proposed model would sometimes fail to capture the style information when it is over far away from the standard font of the prototype prototype. For example, some cursive writing may play a negative role in the generation process since input contents are all isolated characters. Some failed generated characters are given in Fig. 5.14, of which the 2nd row lists the corresponding one-shot style references.

Table 5.2: Generated characters with different numbers of style references.

# of Style References	Generated Characters
1	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
2	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
3	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
4	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
5	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
6	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
7	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
8	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
9	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
10	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
11	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
12	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
13	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
14	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
15	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
16	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
17	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
18	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
19	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
20	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
21	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
22	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
23	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
24	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
25	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨
*Real	赵 钱 孙 李 周 吴 郑 王 冯 陈 储 卫 蒋 枕 韩 杨

Upon the proposed W-Net, each target is regarded as a non-linear style transformation from a reference to a prototype. However, when the style is too illegible and different from the single content standard font, the model fails to learn this complicated mapping relationship. In such extreme circumstances, the provided single prototype font in this research might be an inappropriate choice. In this case, it could be a good idea to learn additional mappings which can transform the original prototype to suitable latent features. It can be better to handle free writing styles in real scenarios.

A simple way to fulfill the task is to introduce multiple content prototypes, posing as those additional mapping relationships. The generated character is assumed to be synthesized from those multiple contents. Those over one single content prototypes provide a better possibility to synthesize a more reasonable generated character for those over-illegible handwriting styles or printing fonts.



Fig. 5.14: Unsatisfied generated examples: In each figure: 1st row: generated characters; 2nd row: corresponding style references (ground truth characters).

### 5.2.4 Possible Statistical Evaluation Procedures

An appropriate measuring metric on the synthesizing performance of the generative models is still an open question in the relevant research community. The W-Net architecture is only evaluated with subjective criterion [101, 162, 163], e.g., by asking human annotators to determine whether the generated character is similar enough with the corresponding real one. However, it would be affected by sudden and random incidents, resulting in misleading and time-consuming evaluating procedures [164].

On the contrast, there are also proposals which engage the objective evaluation procedures, including [105, 47, 48, 165, 102, 164]. Particularly, [48, 165, 102] evaluated the performance with the classification performance by feeding the *generated character* into a pre-trained classifier. [105, 47] were tested by introducing the distance-based assessment between the *generated character* and the corresponding real one. It was determined by pixel-level discrepancy including  $L_1$  variation, Mean Square Error (MSE), and the Pixel Disagreement Ratio [47]. The Wasserstein distance was reviewed in [164] by specifying the logit difference in the discriminator of a *generated character* and the real one. It is adversarially trained with the generator by the W-GAN-GP framework [24].

the pixel-level discrepancy might not be robust to mis-alignment and random perturbation. The Wasserstein distance is not appropriate to evaluate the model performance



between different synthesization frameworks since each discriminator shall be trained along with the specific generator. Based on the non-universal nature, the distance computed by various discriminators would not represent the real synthesization performance among comparing models. The classification-based evaluation might not be an appropriate option since complicated specification will be involved when new-coming contents or styles that are out of the training set are present.

The deep perceptual loss based on a pre-trained Vgg-16 Network [7] on a relevant classification task was engaged in [18] to assist to minimize the discrepancy between the *generated character* and the real one in a structured fashion. Effectiveness in promoting the generation performance has been readily shown. In this sense, some of the distance-based evaluating metrics based on the Vgg-16 deep perceptual features would be a proper candidate to perform the evaluation.

A simple comparison on the example characters (Fig. 5.15) is depicted in Fig. 5.16. The relative discrepancy is calculated by the ratio of the specific metrical variation occupied on the corresponding upper limit. The limit is defined by the difference between two characters with totally different style (Fig. 5.15(b) and Fig. 5.15(a)), while the contents between them are kept. The measuring difference is due to the variation brought by some small perturbation (rotation in Fig. 5.15(c) or translation in Fig. 5.15(d) and Fig. 5.15(e)) on the specific character (Fig. 5.15(b)). The involving distance-based evaluating metrics include the pixel-level difference ( $L_1$ , MSE, and PDAR), and the deep feature divergence (based on a pre-trained Vgg-16 [7]) computed by MSE and VN [159].<sup>19</sup> From the illustra-

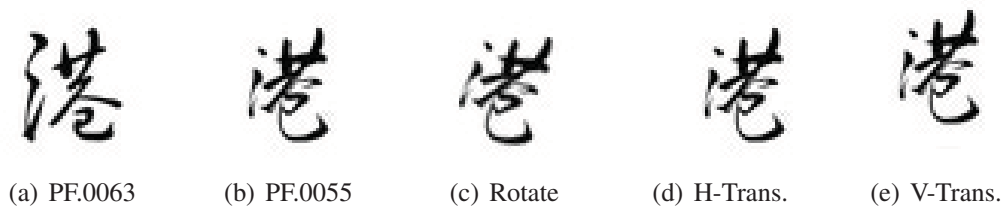


Fig. 5.15: Example printed font characters for Metric Comparison: H-Trains represents horizontal translation, and W-Trans denotes the vertical one.

tion, one can conclude that pixel-based difference including  $L_1$ , MSE, and the PDAR are easily affected by image variation. MSE-based middle feature measurement (B3-L3-MSE and B4-L3-MSE) are more robust to rotational variation. VN-based ones performs better (B3-L3-VN and B4-L3-VN) by maintaining the variation in a relatively small value when the perturbation becomes greater. On the contrast, deeper MSE variation (B5-L3-MSE) tends to be more consistent in translating difference, The VN divergence in B5-L3-VN seems to be more consistent against translating variation.

As a summary, feature divergence of both MSE and VN in B3-L3, B4-L3, B5-L3 will be involved in future objective evaluations.<sup>20</sup>

<sup>19</sup>Deeper features from the fully-connected layers of the Vgg-16 model are omit since the the structural information is lost during the reshaping operation.

<sup>20</sup>Such objective evaluations will be seen as a part of the future work, as will be demonstrated in Chapter 6. No objective evaluation will be specified in this thesis.

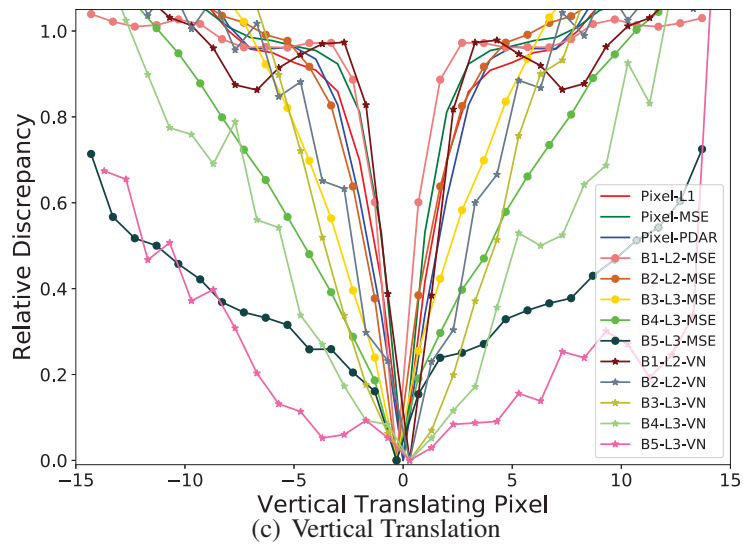
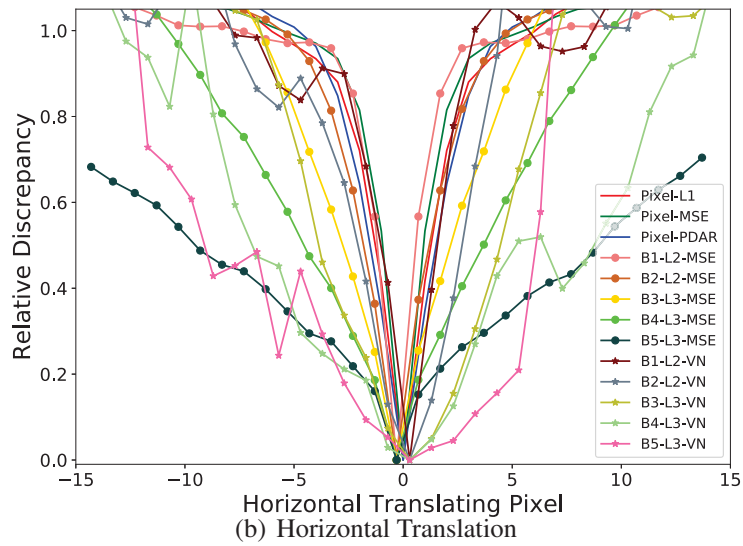
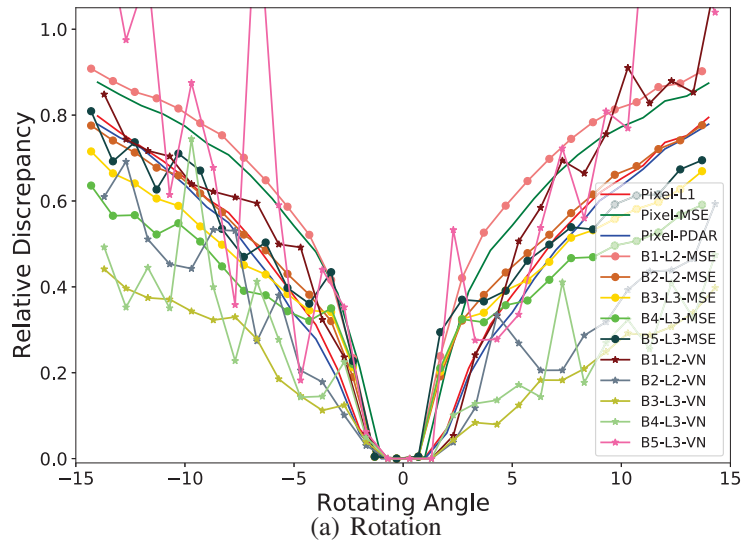


Fig. 5.16: Evaluation of metrics comparison:  $BM-LN$  represents the  $\phi_{M-N}$  layer in the Vgg-16 Network.

## 5.2.5 Further Studies for Other Eastern Asian Characters

Most of the characters in the eastern Asian languages including Chinese (traditional or simplified), Korean, and Japanese are constructed by rectangular shapes (known as the *block characters* [166]). In this sense, it is an interesting evaluation to make the well-optimized W-Net model to generate characters of these three kinds of languages. The training of the W-Net follows the description in Section 5.2.3 where only the simplified Chinese characters are available. For each of the generation processes of paragraphs in these kinds of languages, one randomly selected brush-written character listed in Fig. 5.17 will be specified as the one-shot style reference. Fig. 5.19, Fig. 5.20, and Fig. 5.21 illustrate the generated result of the corresponding a traditional Chinese poetry, a Korean lyric, and a Japanese speech. It can be obviously found that the style tendency given in Fig. 5.17 is well kept in all the three generated outputs.



Fig. 5.17: The input few brush-written examples of actual Chinese characters.<sup>21</sup>

A more interesting discovery is that the W-Net model trained with simplified Chinese characters is always capable to synthesize the *circular radicals* that are commonly seen in Korean but rarely found in Chinese. As seen in Fig. 5.18, the *circular radicals* are mostly preserved and recovered in the generated characters (Fig. 5.18(b)) when compared with the corresponding content prototypes shown in Fig. 5.18(a). It further demonstrates that the proposed W-Net is capable of being generalized to the useful knowledge in strokes and radicals that are absent from the training data.

## 5.2.6 Summary and Future Work

A novel generalized framework named W-Net is introduced in this section in order to achieve Few-shot Multi-content Arbitrary-style Chinese Character Generation (FMACCG) task. The special case of the FMACCG task, namely, the One-shot Single-content Arbitrary-style Chinese Character Generation (OSACCG) has been extensively studied and experimented. Specifically, the proposed model, composing of two encoders and one decoder with several layer-wised connections, is trained adversarially based on the Wasserstein GAN scheme with the gradient penalty. It enables synthesizing any arbitrary stylistic character by transferring the learned style information from one single style reference to the input single content prototype. Extensive experiments have demonstrated the reasonableness and effectiveness of the proposed W-Net model in the one-shot setting.

---

<sup>21</sup>These brush-written simplified Chinese character are written by Dr. Fei CHENG from seedeep.ai.

동 해 이 닳 하 이 우 하  
 우 위 에 을 함 은 우 상  
 일 을 하 공 한 없 이 은  
 은 우 일 일 이 기 상 이  
 으 충 성 을 하 여 우 우  
 랑 하 궁 화 화 강 한 한  
 으 길 이 하

(a) Content Prototypes

동 해 이 닳 하 이 무 하  
 무 위 에 을 함 은 무 상  
 일 을 하 공 한 없 이 은  
 은 무 일 일 이 기 상 이  
 으 충 성 을 하 여 무 무  
 랑 하 궁 화 화 강 한 한  
 으 길 이 하

(b) Generated Characters

Fig. 5.18: Some Korean characters with *circular radicals*. The characters with circular radicals are selected from the ones given in Fig. 5.20

### Future Work

Extensions to more proper mapping architectures for image reconstruction will be studied in the future so as to capture sufficiently complicated and free writing styles. In particular, the performance of the FMACCG setting will be further investigated. As have been analyzed for the failure examples in Section 5.2.3, it will provide more additional mapping relationship from multiple content prototypes to generate a more realistic character style. Meanwhile, practical applications are to be developed not only restricted in the character generation tasks, but also in other relevant arbitrary-style image generation frameworks. The AdaIN proposed in [95] will also be studied to be integrated into the W-Net framework in order to achieve better synthesizing performance.

Moreover, there is no universal objective criterion of the evaluation measurement on the generated samples of all the generative models [167]. Simple Turing test [168] based subjective evaluation by human annotators and evaluators are easy to be affected by sudden and random incidents to the examiners' emotions. In order to verify the effectiveness and reasonableness of the proposed W-Net model, means of objective measurement on the quality of the generated characters will be discussed in the future.

策至下桂命藩南於愚兵十城之處已金  
長履天為委守敢怨以之人為測之下固也  
振侯答以頸而不報言下金華不害天之業  
烈諸鞭地系城人而之天為踐臨要何中之  
余亡而之首長胡弓家收以後城守誰關世  
之而撲越俯築里營百傑鑄然之弩而為萬  
世周敲百君北余敢焚豪鎬民丈勁兵以王  
六二執取之恬百不道殺鋒之億將利自帝  
奮吞合南越蒙七士之城銷下據良陳心孫  
皇內六海百使奴馬王名陽天池固卒之子  
始宇制四郡乃匈牧先隳咸弱為為精皇里  
至禦而振象吏卻而廢首之以河以臣始千  
及而尊威林下籬下是黔聚二因淵信定城

Fig. 5.19: The W-Net (trained with only simplified Chinese characters) generated essay of traditional Chinese characters from the one selected style referenc shown in Fig. 5.17. The shown traditional Chinese paragraph is part of the Chinese ancient essay titled as *The crimes of the Qin Empire* written by Yi JIA in Chinese Han Dynasty (Approx. 200s to 100s B.C.)

동	해	물	과	백	두	산	이	마	르	고
뫑	도	록	하	느	님	이	보	우	하	사
우	리	나	라	만	세	남	산	위	메	저
소	나	무	철	갑	을	두	른	듯	바	람
서	리	불	변	함	은	우	리	기	상	일
세	가	을	하	늘	공	활	한	데	높	고
구	름	없	이	밖	은	달	은	우	리	가
슴	일	편	단	심	일	세	이	기	상	과
이	맘	으	로	총	성	을	다	하	여	괴
로	우	나	즐	거	우	나	나	라	사	랑
하	세	무	궁	화	삼	천	리	화	러	강
산	대	한	사	람	대	한	으	로	길	이
보	전	하	세							

Fig. 5.20: The W-Net (trained with only simplified Chinese characters) generated essay of Korean characters from the one selected style reference shown in Fig. 5.17. The shown Korean paragraph is part of national anthem of the Republic of Korea, titled as *The Patriotic Song* originally written in 1935 and adopted in Aug., 1948 [169].

朕四諾ノキル陸ノ必亦殘辜測交族入人  
 ハ國入主ハ二海勵最入我虐ヲル戰ノ延シ  
 帝二ル權固交將精善シ二ナ殺人ヲ滅テ  
 國對旨ヲヨ戰兵朕ヲ七利ル傷力繼亡人  
 政シ通排リ己ノ力盡好ア爆シヲ續ヲ類  
 府其告シ朕二勇一七轉ヲ彈慘サセ招ノ  
 ヲノセ領力四戰億ルセ入ヲ害ルム來文  
 シ共シ土志歳朕衆二入加使ノ二力入明  
 テ同メヲ二ヲ力庶拘世之用及至終ルヲ  
 米宣夕侵ア閱百ノヲ界敵シアル二七  
 英言リ入ヲシ僚奉スノハテ所而我ミ破  
 中ヲ他力入朕有公戰大新二真七力ナ却  
 蘇受國如然力司各局勢二無二尚民ヲ入

Fig. 5.21: The W-Net (trained with only simplified Chinese characters) generated essay of Japanese characters from the one selected style reference shown in Fig. 5.17. The shown Japanese paragraph is part of public speech titled as *Imperial Rescript on the Termination of the War* delivered by the Japanese Ruler of the time *Emperor Hirohito* to declare the unconditional surrender of the Japanese invading military forces, in 15th, Aug., 1945 [170].





# Chapter 6

## Conclusion

The conclusion of this thesis will be given in this chapter. It begins with one challenge in the Artificial Intelligence (AI) and the Machine Learning (ML) research communities. Namely, the *identical and independent distribution (i.i.d.)* assumption may not always be satisfied in real practice. In particular, style consistency among the patterns, obviously violating the *i.i.d.* assumption, would bring the degraded prediction performance directly.

Motivated by that challenge and the problem of such the style inconsistency issue, we propose several field prediction models. These models include pure image process based algorithms, the discriminative machine learning frameworks, and the generative deep machine learning architectures. These proposed models are proved effective in promoting the final prediction performance by taking the inconsistent style information into proper consideration. Moreover, the style information is also investigated to conduct the style data generation task. The Few-shot Multi-content Arbitrary-style Chinese Character Generation (FMACCG) task is investigated by making use of the style information in a generative machine learning deep architecture. The One-shot Single-content Arbitrary-style Chinese Character Generation (OSACCG), seen as a special case of FMACCG, is further implemented. It was less seen in the research literature.

After those discussions to draw the final conclusion of this thesis, future directions and perspectives of the relevant academic research will be also provided in the last part of this chapter.

### 6.1 Review of the Thesis

This thesis is majorly concentrating on the relevant models to make use of the inconsistent style information to perform both the prediction and the generation tasks. In the literature, most non-*i.i.d.* prediction tasks are solved by the Multi-task Learning (MTL) and the Field Prediction Models (FPM). These models and approaches are firstly reviewed and compared. Particularly, one specific class of the FPM-based frameworks, namely, the Field Averaging Models (FAM) to perform the Style Average Transformation (SAT), is further investigated and developed in the following part of this thesis.

One specific method to SAT is to introduce a pure image process based algorithm. Named as the Style Elimination Transformation (SET), the proposed algorithm is capable of recovering the corrupted pixels brought by inconsistent style information due to the sunglasses with diverse shapes and various luminance transmittance in a facial expression image. Decoupled into the sunglasses region detection and the histogram equalization

on the detected sunglasses region, the proposed SET eliminates such style inconsistency by producing the corresponding style-eliminated recovered facial portraits. The recognition performance has been promoted in several of the mainstream classifiers including the Support Vector Machine (SVM), the Linear Discriminant Analysis (LDA), and the  $k$ -Nearest Neighbour (KNN) thanks to those style-eliminated patterns. The proposed algorithm even enables the one-shot facial expression recognition with one single image for each expression of each individual involved in the training data. It brings the robustness in the facial expression recognition system.

Then, the SAT is implemented into a discriminative machine learning model, namely, the SVM model. It is designed to normalize the inconsistent style information to produce style-normalized patterns. Named as the Field-SVM (F-SVM), the Style Normalization Transformation (SNT) in it is learned simultaneously the SVM parameters are optimized. The kernelized representation is also deduced to represent complicated style information in the nonlinear kernel space. Such representation brings flexibility and reasonableness to perform the SNT. Furthermore, a self-learning strategy is further studied to perform the field prediction on unseen styles during training. The F-SVM model is deduced into the Field Support Vector Classification (F-SVC) and the Field Support Vector Regression (F-SVR) frameworks. Compared with several of the state-of-the-art baselines, both the F-SVC and the F-SVR improves the classification accuracy and declines the regression error with the extensive statistical experiments. The performance of the F-SVC can be further observed by several visualized performance achieved.

Moreover, the SAT is investigated into a generative machine learning model. Named as the Style Neutralization Generative Adversarial Classifier (SN-GAC), the inconsistent style information is neutralized with a neural network based generator. Trained with the Generative Adversarial Network (GAN) based adversarial training strategy, the upon-mentioned generative model is capable of producing high-quality style-neutralized human-readable examples when the style-discriminative ones are given. The attached classifier of the discriminative model in the GAN framework assigns the correct class labels when given such the generated images. Extensive experiments have demonstrated the effectiveness of the SN-GAC model against several of the relevant models. In particular, similar performance is achieved compared with the F-SVC model. However, it is obtained with no access to the testing data in the SN-GAC model, compared with that in the F-SVC framework based on the self-training strategy.

Finally, the Few-shot Multi-content Arbitrary-style Chinese Character Generation task (FMACCG) is studied. The proposed W-Net architecture includes the style reference encoder ( $Enc_r$ ), the content prototype encoder ( $Enc_p$ ), and the decoder ( $Dec$ ). A well-trained W-Net is capable of generating a character which is consistent in the style with those input references fed into  $Enc_r$ , and maintains the content with those input prototypes to  $Enc_p$ . Trained also with the GAN framework, the reasonableness and the effectiveness of the proposed W-Net has been extensively evaluated with the One-shot Single-content Arbitrary-style Chinese Character Generation setting with only one single style reference available. Furthermore, it also enables to generate any block characters in all the eastern Asian languages (traditional Chinese, Korean, and Japanese) when trained with only simplified Chinese characters. Some of the special radicals rarely occur in Chinese can also be effectively synthesized. Such an interesting property is rarely seen in the research literature.

## 6.2 Future Work

First, the SN-GAC model will be studied by conducting the classification on the larger number of classes. The primary target is to perform the classification on the GB-2312 L1 set [91], where in total 3,755 Chinese characters are involved. Then it will be evaluated with much greater numbers of classes. Such large-scale classification can hardly be fulfilled by the SVC-based model, including the F-SVC model demonstrated in Chapter 4 because of the voting strategy [126, 127] to extend the binary SVC model for multi-class classification scenarios [36].

Then, the W-Net model will also be developed to find a suitable metric to determine the quality of the generated character. There is no universal objective criterion of the evaluation measurement on the generated samples of all the generative models [167], including both the GAN-based and Variational Auto-encoder architectures. Simple Turing test [168] based subjective evaluation by human annotators and evaluators are easy to be affected by sudden changes and random incidents to the examiners' bodies and emotions. In order to verify the effectiveness and reasonableness of the proposed W-Net model, means of objective measurement on the quality of the generated characters will be studied in the future.

Apart from them, extensions to more proper mapping architectures for the W-Net will be studied in the future so as to capture sufficiently complicated and free writing styles, which has failed in the current network demonstrated in Section 5.2.3. In particular, the performance of the One-shot Multi-content Arbitrary-style Chinese Character Generation (OMACCG) setting will be further investigated. As have been analyzed, it will provide more additional mapping relationship from multiple content prototypes to generate a more realistic character style.

Finally, practical applications are to be developed in other relevant arbitrary-style image generation tasks. The AdaIN will also be studied to be integrated into the W-Net framework in order to achieve better synthesizing performance.



# Appendix: A list of Publications

Here is a brief list of my research publications during my Ph.D. studies:

- **Journal Papers:**

1. Haochuan JIANG, Kaizhu HUANG, Rui ZHANG, "Style Neutralized Pattern Classification based on Adversarially-trained Upgraded U-Net," *Cognitive Computation* (Invited for Publication), Springer, 2019 [27].
2. Jieming MA, Haochuan JIANG, Ziqiang BI, Kaizhu HUANG, Xingshuo LI, Huiqing WEN, "Maximum Power Point Estimation for Photovoltaic String subjected to Partial Shading Scenarios," *IEEE Transactions on Industry Applications*, 2018 [23].
3. Jieming MA, Haochuan JIANG, Kaizhu HUANG, Ziqiang BI and Ka Lok MAN, "Novel Field-Support Vector Regression-based Soft Sensor for Accurate Estimation of Solar Irradiance," *IEEE Transactions on Circuits and Systems I: Regular Papers* 64, no. 12, pp.3183-3191, 2017 [22].
4. Kaizhu HUANG, Haochuan JIANG, Xu-Yao ZHANG, "Field Support Vector Machines (initial version)," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 6, 2017 [14].
5. Haochuan JIANG, Kaizhu HUANG, Tingting MU, Rui ZHANG, Ting T.O., Chen WANG, "Robust One-Shot Facial Expression Recognition with Sunglasses," *International Journal of Machine Learning and Computing*, vol. 6, no. 2, 2016 [12].

- **Conference Papers:**

1. Haochuan JIANG, Guanyu YANG, Kaizhu HUANG, Rui ZHANG, "W-Net: One-Shot Arbitrary-Style Chinese Character Generation with Deep Neural Networks," in *Proceedings of the International Conference on Neural Information Processing*, Springer, 2018, pp. 483-493 [18].
2. Haochuan JIANG, Kaizhu HUANG, Rui ZHANG, "Style Neutralization Generative Adversarial Classifier," in *Proceedings of the International Conference on Brain Inspired Cognitive System*, Springer, 2018, pp. 3-13 [17].
3. Haochuan JIANG, Kaizhu HUANG, Rui ZHANG, "Field Support Vector Regression," in *Proceedings of the International Conference on Neural Information Processing*, Springer, 2017, pp. 699-708 [15].

4. Kaizhu HUANG, Haochuan JIANG, Xu-Yao ZHANG, "Field Support Vector Machines (extended version)," in Proceedings of the International Conference on Internet of Things and Machine Learning, ACM, 2017, pp. 72-83 [13].

- **Academic Book Chapters:**

1. Haochuan JIANG, Kaizhu HUANG, Xu-Yao ZHANG, Rui ZHANG, "Self-Training Field Pattern Prediction based on Kernel Methods," in Semi-Supervised Learning: Background, Applications and Future Directions (Guoqiang ZHONG, Kaizhu HUANG, 1st ed., pp. 123-169). 2018, New York, Nova Science Publisher Inc [16].

# Reference

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems (NeurIPS)*, 2014, pp. 2672–2680.
- [3] Quora, “Forbes this is the cutting edge of deep learning research,” 2016. [Online]. Available: <https://www.forbes.com/sites/quora/2016/08/05/this-is-the-cutting-edge-of-deep-learning-research/#2f71c7d451c8>
- [4] A. Clauset, “A brief primer on probability distributions,” in *Santa Fe Institute*, 2011.
- [5] S. Veeramachaneni and G. Nagy, “Analytical results on style-constrained bayesian classification of pattern fields,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 7, pp. 1280–1285, 2007.
- [6] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [7] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.
- [9] P. Baldi and R. Vershynin, “On neuronal capacity,” in *Advances in Neural Information Processing Systems (NeurIPS) 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7740–7749. [Online]. Available: <http://papers.nips.cc/paper/7999-on-neuronal-capacity.pdf>
- [10] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [11] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *arXiv preprint arXiv:1701.07875*, 2017.

- [12] H. Jiang, K. Huang, T. Mu, R. Zhang, T. Ting, and C. Wang, “Robust one-shot facial expression recognition with sunglasses,” *International Journal of Machine Learning and Computing*, vol. 6, no. 2, p. 80, 2016.
- [13] K. Huang, H. Jiang, and X.-Y. Zhang, “Field support vector machines,” in *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*. ACM, 2017, p. 72.
- [14] —, “Field support vector machines,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 6, pp. 454–463, 2017.
- [15] H. Jiang, K. Huang, and R. Zhang, “Field support vector regression,” in *International Conference on Neural Information Processing*. Springer, 2017, pp. 699–708.
- [16] H. Jiang, K. Huang, X.-Y. Zhang, and R. Zhang, *Self-training Field Pattern Prediction based on Kernel Methods*, G. Zhong and K. Huang, Eds. NOVA Science Publishers, 2018.
- [17] H. Jiang, K. Huang, R. Zhang, and A. Hussain, “Style neutralization generative adversarial classifier,” in *International Conference on Brain Inspired Cognitive Systems*. Springer, 2018, pp. 3–13.
- [18] H. Jiang, G. Yang, K. Huang, and R. Zhang, “W-net: One-shot arbitrary-style chinese character generation with deep neural networks,” in *International Conference on Neural Information Processing*. Springer, 2018, pp. 483–493.
- [19] P. Sarkar and G. Nagy, “Style consistent classification of isogenous patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 88–98, Jan 2005.
- [20] J. B. Tenenbaum and W. T. Freeman, “Separating style and content with bilinear models,” *Neural Computation*, vol. 12, no. 6, pp. 1247–1283, 2000.
- [21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *arXiv preprint*, 2017.
- [22] J. Ma, H. Jiang, K. Huang, Z. Bi, and K. L. Man, “Novel field-support vector regression-based soft sensor for accurate estimation of solar irradiance,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 12, pp. 3183–3191, 2017.
- [23] J. Ma, H. Jiang, K. Huang, Z. Bi, X. Li, and H. Wen, “Maximum power point estimation for photovoltaic strings subjected to partial shading scenarios,” *IEEE Transactions on Industry Applications*, 2018.
- [24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5767–5777.



- [25] I. Olkin and F. Pukelsheim, “The distance between two random vectors with given dispersion matrices,” *Linear Algebra and its Applications*, vol. 48, pp. 257–263, 1982.
- [26] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” *arXiv preprint arXiv:1610.09585*, 2016.
- [27] H. Jiang, K. Huang, R. Zhang, and A. Hussain, “Style neutralized pattern classification based on adversarially-trained upgraded u-net.” Springer, 2019.
- [28] R. Caruana, “Multitask learning,” *Machine Learning*, 1997.
- [29] X.-Y. Zhang, K. Huang, and C.-L. Liu, “Pattern field classification with style normalized transformation,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1621.
- [30] T. Evgeniou and M. Pontil, “Regularized multi-task learning,” in *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2004, pp. 109–117.
- [31] B. Bakker and T. Heskes, “Task clustering and gating for bayesian multitask learning,” *Journal of Machine Learning Research*, vol. 4, no. May, pp. 83–99, 2003.
- [32] A. Kumar and H. Daume III, “Learning task grouping and overlap in multi-task learning,” *arXiv preprint arXiv:1206.6417*, 2012.
- [33] F. Nielsen, “A family of statistical symmetric divergences based on jensen’s inequality,” *arXiv preprint arXiv:1009.4004*, 2010.
- [34] D. Reynolds, “Gaussian mixture models,” *Encyclopedia of biometrics*, pp. 827–832, 2015.
- [35] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [36] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [37] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [38] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 1998, pp. 200–205.
- [39] X. Wang, X. Liu, L. Lu, and Z. Shen, “A new facial expression recognition method based on geometric alignment and lbp features,” in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 1734–1737.

- [40] M. Dahmane and J. Meunier, "Emotion recognition using dynamic grid-based hog features," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 2011, pp. 884–888.
- [41] H. Cate, F. Dalvi, and Z. Hussain, "Deepface: face generation using deep learning," *ArXiv Preprint ArXiv:1701.01876*, 2017.
- [42] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.
- [43] V. Vapnik, E. Levin, and Y. L. Cun, "Measuring the vc-dimension of a learning machine," *Neural computation*, vol. 6, no. 5, pp. 851–876, 1994.
- [44] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.
- [45] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 521–528.
- [46] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [47] Y. Zhang, Y. Zhang, and W. Cai, "Separating style and content for generalized style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2018.
- [48] D. Sun, T. Ren, C. Li, H. Su, and J. Zhu, "Learning to write stylized chinese characters by reading a handful of examples," *arXiv preprint arXiv:1712.06424*, 2017.
- [49] J. Zhao, H. Tian, W. Xu, and X. Li, "A new approach to hand vein image enhancement," in *Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on*, vol. 1. IEEE, 2009, pp. 499–501.
- [50] W. Ting and L.-B. ZHANG, "Curvature analysis and visualization of dental mesh models," *DEStech Transactions on Computer Science and Engineering*, no. aice-ncs, 2016.
- [51] D. Zur, "A system and method for detection of suspicious tissue regions in an endoscopic procedure," 2018, uS Patent App. 15/758,679.
- [52] R. Memisevic, "An introduction to structured discriminative learning," Technical report, University of Toronto, Toronto, Canada, Tech. Rep., 2006.
- [53] X.-Y. Zhang and C. Liu, "Writer adaptation with style transfer mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1773–1787, 2013.
- [54] X.-Y. Zhang and C.-L. Liu, "Style transfer matrix learning for writer adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 393–400.

- [55] G. J. McLachlan, “Mahalanobis distance,” *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.
- [56] R. Salakhutdinov, “Learning deep generative models,” *Annual Review of Statistics and Its Application*, vol. 2, pp. 361–385, 2015.
- [57] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [58] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 2172–2180.
- [59] L. Zhang, Y. Ji, and X. Lin, “Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan,” *arXiv preprint arXiv:1706.03319*, 2017.
- [60] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [61] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2234–2242.
- [62] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2813–2821.
- [63] J.-F. Coeurjolly, R. Drouilhet, and J.-F. Robineau, “Normalized information-based divergences,” *Problems of Information Transmission*, vol. 43, no. 3, pp. 167–189, 2007.
- [64] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, “The gan landscape: Losses, architectures, regularization, and normalization,” *arXiv preprint arXiv:1807.04720*, 2018.
- [65] Y. Yoshida and T. Miyato, “Spectral norm regularization for improving the generalizability of deep learning,” *arXiv preprint arXiv:1705.10941*, 2017.
- [66] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.
- [67] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, “Variational autoencoder for deep learning of images, labels and captions,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 2352–2360.
- [68] C. Doersch, “Tutorial on variational autoencoders,” *arXiv preprint arXiv:1606.05908*, 2016.

- [69] L. Mescheder, S. Nowozin, and A. Geiger, “Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks,” *arXiv preprint arXiv:1701.04722*, 2017.
- [70] S. H. Khan, M. Hayat, and N. Barnes, “Adversarial training of variational autoencoders for high fidelity image generation,” *arXiv preprint arXiv:1804.10323*, 2018.
- [71] E. CK, “Towards Data Science what the heck are vae-gans,” 2017. [Online]. Available: <https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a?gi=27d281c19457>
- [72] OpenAI, “OpenAI generative models,” 2016. [Online]. Available: <https://blog.openai.com/generative-models/>
- [73] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.
- [74] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [75] M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, “The importance of skip connections in biomedical image segmentation,” in *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187.
- [76] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” *arXiv preprint arXiv:1711.11585*, 2017.
- [77] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” *arXiv preprint arXiv:1611.02200*, 2016.
- [78] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv preprint arXiv:1703.10593*, 2017.
- [79] Z. Yi, H. R. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation.” in *ICCV*, 2017, pp. 2868–2876.
- [80] Y. Hiasa, Y. Otake, M. Takao, T. Matsuoka, K. Takashima, A. Carass, J. L. Prince, N. Sugano, and Y. Sato, “Cross-modality image synthesis from unpaired data using cyclegan,” in *International Workshop on Simulation and Synthesis in Medical Imaging*. Springer, 2018, pp. 31–41.
- [81] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” *arXiv preprint arXiv:1703.05192*, 2017.

- [82] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” *arXiv preprint*, vol. 1711, 2017.
- [83] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, “Arbitrary facial attribute editing: Only change what you want,” *arXiv preprint arXiv:1711.10678*, 2017.
- [84] H. Chang, J. Lu, F. Yu, and A. Finkelstein, “Pairedcyclegan: Asymmetric style transfer for applying and removing makeup,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [85] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” *arXiv preprint arXiv:1711.04340*, 2017.
- [86] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” *arXiv preprint arXiv:1612.07828*, 2016.
- [87] C. Li, K. Xu, J. Zhu, and B. Zhang, “Triple generative adversarial nets,” *arXiv preprint arXiv:1703.02291*, 2017.
- [88] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning.” in *OSDI*, vol. 16, 2016, pp. 265–283.
- [89] J. Kim, “Github triplegan-tensorflow,” 2018. [Online]. Available: <https://github.com/taki0112/TripleGAN-Tensorflow>
- [90] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: a large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.
- [91] I. Archive, “Gb 2312-1980: Information technology—chinese ideogram coded character set for information interchange (basic set),” 2016. [Online]. Available: <https://archive.org/details/GB2312-1980/page/n17>
- [92] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, “Multi-content gan for few-shot font style transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 11, 2018, p. 13.
- [93] K. Cao, J. Liao, and L. Yuan, “Carigans: unpaired photo-to-caricature translation,” *arXiv preprint arXiv:1811.00222*, 2018.
- [94] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” *ArXiv Preprint ArXiv:1502.03167*, 2015.
- [95] X. Huang and S. J. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization.” in *ICCV*, 2017, pp. 1510–1519.
- [96] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” *arXiv preprint arXiv:1812.04948*, 2018.

- [97] V. L. D. U. A. Vedaldi, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.
- [98] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis.” in *CVPR*, vol. 1, no. 2, 2017, p. 3.
- [99] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 386–396.
- [100] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” *arXiv preprint arXiv:1804.04732*, 2018.
- [101] Y. Tian, “Github zi2zi-tensorflow,” 2017. [Online]. Available: <https://kaonashi-tyc.github.io/2017/04/06/zi2zi.html>
- [102] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, “Drawing and recognizing chinese characters with recurrent neural network,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 849–862, 2018.
- [103] Y. Jiang, Z. Lian, Y. Tang, and J. Xiao, “Dcfont: an end-to-end deep chinese font generation system,” in *SIGGRAPH Asia 2017 Technical Briefs*. ACM, 2017, p. 22.
- [104] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, “Google’s multilingual neural machine translation system: enabling zero-shot translation,” *arXiv preprint arXiv:1611.04558*, 2016.
- [105] Y. Zhang, Y. Zhang, and W. Cai, “A unified framework for generalizable style transfer: Style and content separation,” *arXiv preprint arXiv:1806.05173*, 2018.
- [106] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*. Ieee, 1999, pp. 41–48.
- [107] T. M. Cover, P. E. Hart *et al.*, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [108] H.-W. Kung, Y.-H. Tu, and C.-T. Hsu, “Dual subspace nonnegative graph embedding for identity-independent expression recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 626–639, 2015.
- [109] M. H. Siddiqi, R. Ali, A. M. Khan, Y.-T. Park, and S. Lee, “Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields,” *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1386–1398, 2015.

- [110] S. Milborrow and F. Nicolls, “Locating facial features with an extended active shape model,” in *European conference on computer vision*. Springer, 2008, pp. 504–513.
- [111] ———, “Active shape models with sift descriptors and mars,” in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2. IEEE, 2014, pp. 380–387.
- [112] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [113] R. Gonzalez, *Digital image processing*. Pearson Education India, 2002.
- [114] W. Liu, C. Song, Y. Wang, and L. Jia, “Facial expression recognition based on gabor features and sparse representation,” in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*. IEEE, 2012, pp. 1402–1406.
- [115] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.
- [116] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [117] D. Basak, S. Pal, and D. C. Patranabis, “Support vector regression,” *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.
- [118] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the 5th Annual Workshop on Computational Learning Theory*. ACM, 1992, pp. 144–152.
- [119] N. Gourier, D. Hall, and J. L. Crowley, “Estimating face orientation from robust detection of salient facial structures,” in *FG Net Workshop on Visual Observation of Deictic Gestures*, vol. 6, 2004.
- [120] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [121] C. L. Liu, F. Yin, D. H. Wang, and Q. F. Wang, “Casia online and offline chinese handwriting databases,” in *International Conference on Document Analysis and Recognition (ICDAR)*, Sept 2011, pp. 37–41.
- [122] E. Schöneburg, “Stock price prediction using neural networks: A project report,” *Neurocomputing*, vol. 2, no. 1, pp. 17–27, 1990.
- [123] D. L. Nuttall, H. Goldstein, R. Prosser, and J. Rasbash, “Differential school effectiveness,” *International Journal of Educational Research*, vol. 13, no. 7, pp. 769–776, 1989.

- [124] P. J. Lenk, W. S. DeSarbo, P. E. Green, and M. R. Young, “Hierarchical bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs,” *Marketing Science*, vol. 15, no. 2, pp. 173–191, 1996.
- [125] G. Baudat and F. Anouar, “Kernel-based methods and function approximation,” in *Neural Networks, 2001. Proceedings. IJCNN’01. International Joint Conference on*, vol. 2. IEEE, 2001, pp. 1244–1249.
- [126] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [127] C.-H. Hoi, C.-H. Chan, K. Huang, M. R. Lyu, and I. King, “Biased support vector machine for relevance feedback in image retrieval,” in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, vol. 4. IEEE, 2004, pp. 3189–3194.
- [128] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, “1-norm support vector machines.” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 15, 2003, pp. 49–56.
- [129] K. Huang, D. Zheng, J. Sun, Y. Hotta, K. Fujimoto, and S. Naoi, “Sparse learning for support vector classification,” *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1944–1951, 2010.
- [130] K. Huang, D. Zheng, I. King, and M. R. Lyu, “Arbitrary norm support vector machines,” *Neural Computation*, vol. 21, no. 2, pp. 560–582, Feb 2009.
- [131] C. Chang and C. Lin, “LIBSVM: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [132] J. Platt *et al.*, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [133] T.-F. Wu, C.-J. Lin, and R. C. Weng, “Probability estimates for multi-class classification by pairwise coupling,” *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.
- [134] G. H. Nguyen, S. L. Phung, and A. Bouzerdoum, “Efficient svm training with reduced weighted samples,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–5.
- [135] Y.-M. Huang and S.-X. Du, “Weighted support vector machine for classification with uneven training class sizes,” in *2005 International Conference on Machine Learning and Cybernetics*, vol. 7. IEEE, 2005, pp. 4365–4369.
- [136] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.



- [137] G.-B. Huang and C.-K. Siew, “Extreme learning machine: Rbf network case,” in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, vol. 2. IEEE, 2004, pp. 1029–1036.
- [138] T. Hastie and R. Tibshirani, “Discriminant adaptive nearest neighbor classification and regression,” in *Advances in Neural Information Processing Systems*, 1996, pp. 409–415.
- [139] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, “Modified quadratic discriminant functions and the application to chinese character recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 149–153, Jan 1987.
- [140] T. Mensink, J. Verbeek, G. Csurka, and F. Perronnin, “Metric learning for nearest class mean classifiers,” May 12 2015, uS Patent 9,031,331.
- [141] W. Shang, K. Sohn, D. Almeida, and H. Lee, “Understanding and improving convolutional neural networks via concatenated rectified linear units,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.
- [142] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, “Understanding deep neural networks with rectified linear units,” *arXiv Preprint arXiv:1611.01491*, 2016.
- [143] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
- [144] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [145] T. Xiao, H. Li, W. Ouyang, and X. Wang, “Learning deep feature representations with domain guided dropout for person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1249–1258.
- [146] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *British Machine Vision Conference (BMVC)*, vol. 1, no. 3, 2015, p. 6.
- [147] J. Cook and J. Ranstam, “Overfitting,” *British Journal of Surgery*, vol. 103, no. 13, pp. 1814–1814, 2016.
- [148] G. E. Hinton, A. Krizhevsky, I. Sutskever, and N. Srivastva, “System and method for addressing overfitting in a neural network,” Jul. 28 2016, uS Patent App. 15/222,870.
- [149] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” *ArXiv Preprint ArXiv:1611.03530*, 2016.

- [150] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, “Singular value decomposition and principal component analysis,” in *A Practical Approach to Microarray Data Analysis*. Springer, 2003, pp. 91–109.
- [151] C.-L. Liu and X.-D. Zhou, “Online japanese character recognition using trajectory-based normalization and direction feature extraction,” in *10th International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006.
- [152] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [153] ———, “Ridge regression: applications to nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970.
- [154] L. Van Der Maaten, “Accelerating t-sne using tree-based algorithms.” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.
- [155] K. Huang, H. Yang, I. King, and M. R. Lyu, “Maxi–min margin machine: learning large margin classifiers locally and globally,” *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 260–272, Feb 2008.
- [156] Y. Chang, C. Hsieh, K. Chang, M. Ringgaard, and C. Lin, “Training and testing low-degree polynomial data mappings via linear svm,” *Journal of Machine Learning Research*, vol. 11, pp. 1471–1490, 2010.
- [157] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [158] X.-Y. Jing, H.-S. Wong, and D. Zhang, “Face recognition based on 2d fisherface approach,” *Pattern Recognition*, vol. 39, no. 4, pp. 707–710, 2006.
- [159] X. Yang, K. Huang, R. Zhang, and A. Hussain, “Learning latent features with infinite nonnegative binary matrix trifactorization,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, no. 99, pp. 1–14, 2018.
- [160] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [161] A. Krogh and J. A. Hertz, “A simple weight decay can improve generalization,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 1992, pp. 950–957.
- [162] S. Yang, J. Liu, Z. Lian, and Z. Guo, “Awesome typography: Statistics-based text effects transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7464–7473.
- [163] Q. Lin, L. Liang, Y. Huang, and L. Jin, “Learning to generate realistic scene chinese character images by multitask coupled gan,” in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2018, pp. 41–51.

- [164] G. Huang, Y. Yuan, Q. Xu, C. Guo, Y. Sun, F. Wu, and K. Q. Weinberger, “An empirical study on evaluation metrics of generative adversarial networks,” *arXiv: Learning*, 2018.
- [165] B. Chang, Q. Zhang, S. Pan, and L. Meng, “Generating handwritten chinese characters using cyclegan,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 199–207.
- [166] J.-C. Shieh, “The unified phonetic transcription for teaching and learning chinese languages.” *Turkish Online Journal of Educational Technology-TOJET*, vol. 10, no. 4, pp. 355–369, 2011.
- [167] K. Shmelkov, C. Schmid, and K. Alahari, “How good is my gan?” *arXiv preprint arXiv:1807.09499*, 2018.
- [168] K. M. Colby, F. D. Hilf, S. Weber, and H. C. Kraemer, “Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes,” *Artificial Intelligence*, vol. 3, pp. 199–221, 1972.
- [169] I. Archive, “Aegukga,” 2007. [Online]. Available: <https://archive.org/details/RepublicOfKorea-NationalAnthem>
- [170] W. Machine, “Jowel voice broadcast,” 2012. [Online]. Available: <https://web.archive.org/web/20130910212019/http://www.airforcemag.com/MagazineArchive/Pages/2012/August%202012/0812keeper.aspx>