Deep Feature Learning for Image Classification via Countering Over-fitting

QING YUANYUAN

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

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

May 22, 2021

.....

Date

.....

QING YUANYUAN

Supervisor Declaration Statement

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

January 7, 2021

.....

Date

.....

Prof. Huang Guang-Bin

Authorship Attribution Statement

This thesis contains material from 3 paper(s) published in the following peer-reviewed journal(s) in which I am listed as an author.

Chapter 3 is published as Yuanyuan Qing, Yijie Zeng, Yue Li, and Guang-Bin Huang, "Deep and wide feature based extreme learning machine for image classification", en, *Neurocomputing*, vol. 412, pp. 426–436, Oct. 2020. DOI: 10.1016/j.neucom.2020.06.110.

The contributions of the co-authors are as follows:

- Professor Huang Guang-Bin pointed out the initial research direction, provided guidance for the work and reviewed the manuscript.
- I proposed and designed the methodology, conducted all the experiments and analyzed the experiment results.
- I drafted the manuscript. The draft was revised together with Dr. Zeng Yijie and Dr. Li Yue.

Chapter 4 is published as Yuanyuan Qing, Yijie Zeng, and Guang-Bin Huang, "Label propagation via local geometry preserving for deep semi-supervised image recognition", *Neural Networks*, 2020, Under review.

The contributions of the co-authors are as follows:

- Professor Huang Guang-Bin pointed out the initial research direction, provided guidance for the work and reviewed the manuscript.
- I proposed and designed the methodology, conducted all the experiments and analyzed the experiment results.
- I drafted the manuscript. The draft was revised together with Dr. Zeng Yijie.

Chapter 5 is published as Yuanyuan Qing, Yijie Zeng, Qi Cao, and Guang-Bin Huang, "End-to-end novel visual categories learning via auxiliary self-supervision", *Neural Networks*, vol. 139, pp. 24–32, 2021.

The contributions of the co-authors are as follows:

- Professor Huang Guang-Bin pointed out the initial research direction, provided guidance for the work and reviewed the manuscript.
- I proposed and designed the methodology, conducted all the experiments and analyzed the experiment results.
- I drafted the manuscript. The draft was revised together with Dr. Zeng Yijie, and Dr. Cao Qi.

May 22, 2021 .... 青媛媛 ....

Date QING YUANYUAN

Acknowledgements

I am sincerely grateful to my supervisor Professor Huang Guang-Bin for his supervision and support during my Ph.D. period. His guidance and encouragement have helped me overcome a lot of obstacles.

I would like to thank my seniors Dr. Liu Tianchi and Dr. Cui Dongshun, who inspired me and guided me how to be a researcher. I would like to thank Dr. Zeng Yijie, Dr. Li Yue and Dr. Cao Qi, with whom I have closely worked, for all the diligent and thoughtful discussions with them.

I would like to thank my colleagues, Ms. Jia Xiaofan, Ms. Mao Shangbo and Mr. Chen Jichao, for all the meaningful and joyful discussions and chats with them. I am thankful to the technical support from Delta-EEE Joint Research Lab in School of Electrical and Electronic Engineering and financial support from Nanyang Technological University.

I would like to thank my parents, my sister and all my friends for their love and support during these years. Last but not the least, I would like to thank Dr. Zeng Yijie for his companionship and love, which light up my life.

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

— John von Neumann

Contents

Acknowledgements v

List of Figures x

List of Tables xiv

Summary xv

Symbols and Acronyms xvii

1 Introduction 1
- 1.1 Background and Motivation . . . 2
- 1.2 Objectives and Contributions . . . 4
- 1.3 Thesis Organization . . . 6

2 Literature Review 7
- 2.1 Extreme Learning Machine . . . 8
  - 2.1.1 Basic ELM Algorithm . . . 8
  - 2.1.2 Regularized ELM Algorithm . . . 10
- 2.2 Wide Residual Networks . . . 13
  - 2.2.1 Residual Network . . . 13
  - 2.2.2 Wide Residual Networks . . . 16
- 2.3 Deep Semi-Supervised Learning . . . 19
  - 2.3.1 Consistency-based Algorithms . . . 19
    - 2.3.1.1 Π-Model & Temporal Ensemble . . . 20
    - 2.3.1.2 Mean Teacher . . . 20
    - 2.3.1.3 Virtual Adversarial Training . . . 21
  - 2.3.2 Pseudo-Labeling . . . 21
    - 2.3.2.1 Pseudo Labels from Network . . . 22
    - 2.3.2.2 Pseudo Labels from Transductive Inference . . . 22
- 2.4 Self-Supervised Learning . . . 24
  - 2.4.1 Self-Supervised Learning in Transductive Learning . . . 26
  - 2.4.2 Self-Supervised Learning in Novel Visual Categories Learning 27
- 2.5 Novel Visual Categories Learning . . . 28

2.5.1 Clustering . . . 29
2.5.2 Zero-Shot Learning . . . 29

3 Deep and Wide Feature based Extreme Learning Machine for Image Classification 30
3.1 Background and Motivations . . . 31
3.2 Proposed Algorithm . . . 33
3.2.1 Feature Extraction . . . 33
3.2.2 Classification . . . 35
3.2.3 Complexity Analysis . . . 36
3.2.4 DW-ELM . . . 37
3.2.4.1 Training of Feature Extractor and Classifier of DW-ELM . . . 37
3.2.4.2 Testing of DW-ELM . . . 37
3.3 Performance and Comparisons . . . 39
3.3.1 Datasets . . . 39
3.3.1.1 CIFAR-10 and CIFAR-100 . . . 39
3.3.1.2 STL-10 Dataset . . . 39
3.3.1.3 Fashion-MNIST . . . 40
3.3.1.4 102 Category Flower . . . 40
3.3.2 Implementation . . . 40
3.3.3 Experiment Results . . . 42
3.3.3.1 Performance Improvement . . . 42
3.3.3.2 Stability . . . 45
3.3.3.3 Parameter Analysis . . . 46
3.3.3.4 Ablation Study . . . 46
3.4 Summary . . . 55

4 Label Propagation via Local Geometry Preserving for Deep Semi-Supervised Image Recognition 56
4.1 Background . . . 57
4.2 Proposed Algorithm . . . 61
4.2.1 Problem Formulation . . . 61
4.2.2 Motivations . . . 61
4.2.2.1 Self-Supervision and Full-Supervision . . . 61
4.2.2.2 Local Geometry Preserving . . . 62
4.2.3 Label Propagation via Local Geometry Preserving . . . 63
4.2.3.1 Phase 1 – Self-Supervised Feature Learning . . . 63
4.2.3.2 Phase 2 – Semi-Supervised Label Propagation . . . 63
4.2.3.3 Complexity Analysis . . . 69
4.2.4 Combining with Consistency-based algorithms . . . 69
4.3 Experiments . . . 70
4.3.1 Datasets . . . 70
4.3.2 Implementation . . . 70

4.3.3 Experiment Results . . . 71
4.3.3.1 Ablation Study . . . 72
4.3.3.2 Proposed Algorithm . . . 79
4.4 Summary . . . 85

5 End-to-end Novel Visual Categories Learning via Auxiliary Self-Supervision 86
5.1 Background and Motivations . . . 87
5.2 Proposed Algorithm . . . 90
5.2.1 Problem Formulation . . . 90
5.2.2 End-to-end Novel Visual Categories Learning via Auxiliary Self-Supervision . . . 90
5.2.2.1 Pairwise Similarity Learning . . . 90
5.2.2.2 Mixed Label Classification . . . 93
5.2.2.3 Self-Supervised Learning . . . 94
5.3 Experiments . . . 96
5.3.1 Datasets . . . 96
5.3.2 Implementation . . . 96
5.3.3 Evaluation . . . 99
5.3.4 Experiment Results . . . 99
5.3.4.1 Clustering accuracy . . . 99
5.3.4.2 Mixed classification performance . . . 102
5.3.4.3 Ablation Study . . . 104
5.4 Summary . . . 113

6 Conclusions and Future Works 114
6.1 Conclusions . . . 115
6.2 Future Works . . . 119

List of Publications 121

Bibliography 122

List of Figures

2.1 Single-hidden Layer Feed Forward Neural Network 8
2.2 Feature extracted by VGG-16 at different layers 13
2.3 Three-layer Feed Forward Neural Network 14
2.4 Rectified linear unit function 15
2.5 Residual Unit in ResNet 16
2.6 Consistency-based Semi-Supervised Learning Algorithm 20
2.7 Pseudo-Labeling based Semi-Supervised Learning Algorithm 22
2.8 Self-Supervised Learning Framework 25
2.9 Network Architecture of NIN and AlexNet 26

3.1 Generalization gap of ELM with features extracted by two different models on CIFAR-10 during learning process 33
3.2 Residual Unit in WRN 34
3.3 Training of proposed hybrid DW-ELM model 37
3.4 Testing of proposed hybrid DW-ELM model 38
3.5 Sample Images from Benchmark Datasets 41
3.6 Learning rate annealing schedule 42
3.7 Testing curves of the proposed model DW-ELM and WRN on CIFAR-100 43
3.8 Testing curves of the proposed model DW-ELM and WRN on CIFAR-10 43
3.9 Testing curves of the proposed model DW-ELM and WRN on Flower-102 44
3.10 Testing curves of the proposed model DW-ELM and WRN on Fashion-MNIST 44
3.11 Testing curves of the proposed model DW-ELM and WRN on STL-10 45
3.12 Accuracy of CIFAR-10 with various WRN models 47
3.13 Accuracy of CIFAR-100 with various WRN models 48
3.14 Parameter Analysis: Accuracy of proposed DW-ELM model on CIFAR-10 with varying number of hidden nodes 49
3.15 Parameter Analysis: Accuracy of proposed DW-ELM model on CIFAR-100 with varying number of hidden nodes 49
3.16 Parameter Analysis: Accuracy of proposed DW-ELM model on Fashion-MNIST with varying number of hidden nodes 50

3.17 Parameter Analysis: Accuracy of proposed DW-ELM model on STL-10 with varying number of hidden nodes 50
3.18 Parameter Analysis: Accuracy of proposed DW-ELM model on Flower-102 with varying number of hidden nodes 51

4.1 Overall framework of the proposed algorithm. Here both labeled data X_l and unlabeled data X_u are utilized in phase 1 via self-supervised learning. Each image is rotated by $(0°, 90°, 180°, 270°)$ first and assigned pseudo label $(0, 1, 2, 3)$ respectively to train the network. After the training in phase 1 is done, convolutional layers will be used to extract features in phase 2. In phase 2, label propagation will be performed in feature space after feature extraction and the inferred pseudo labels p_i for unlabeled data will be used together with Y_l of labeled data to calculate the final loss, which will back-propagate through the whole network (including the feature extractor and linear layer). During the training process, updated feature extractor will give new label propagation results and therefore the inferred pseudo labels p_i for unlabeled data will also be updated. 59
4.2 Testing error rate for same vanilla label propagation method with two different feature learning schemes for CIFAR-10 dataset with 500 labels. 62
4.3 Label propagation results with two different graph learning methods given the same data features. The dataset used is CIFAR-10 with 500 labels and t-SNE is used for dimension reduction. The figures here only show 100 sampled data points from the same class for visualization purpose. 64
4.4 Learning rate annealing schedule with initial learning rate of 0.1 . . 71
4.5 Labeled/unlabeled dataset split scheme 72
4.6 Testing error rate for CIFAR-10 with 500, 1000, 2000, and 4000 labels with different feature learning schemes (self-supervision or full-supervision) and different graph construction methods in label propagation (preserve local geometry or not) 73
4.7 Performance of label propagation for CIFAR-100 and *mini*ImageNet with varying number of labeled data under different settings. For each dataset, the labeled data split for each configuration (#. of labeled data) is the same for fair comparisons. It can be observed that how different feature learning schemes (self-supervision with $\{X_l, X_u\}$ or full-supervision with $\{X_l, Y_l\}$) and graph construction methods (preserve local geometry via reconstructing feature vectors as in Equation (4.5) or without geometry preserving by constructing graph with pairwise inner product) will affect the final classification performance. The green-triangle line corresponds to the proposed algorithm and the orange-cross line corresponds to the method in [59]. 74

4.8 2D t-SNE visualizations of embeddings before last linear layer for CIFAR-10 testing data under different training schemes (all trained with the same 500 labeled data split): (A) fully-supervised feature learning w/o geometry preserving, (B) self-supervised feature learning w/ geometry preserving and (C) the proposed algorithm. Each point represents one testing image and colors correspond to ground-truth labels (class 0 to 9 correspond to airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). Recommend to view in color version. 80
4.9 Two challenging pairs of visual categories in CIFAR-10 dataset: airplane & ship, dog & cat. 81

5.1 Labeled/unlabeled dataset split scheme 87
5.2 The illustration of the proposed end-to-end novel visual categories learning algorithm. For both labeled data X_l and unlabeled data X_u, the images are rotated by {0°, 90°, 180°, 270°}. All rotated images are fed into the convolutional neural network to get the corresponding feature vectors ϕ. The feature vectors for 0° rotated images, i.e., the original images, are used to calculated two losses: binary cross-entropy loss $\mathcal{L}_{BCE}$ for X_u and cross-entropy loss $\mathcal{L}_{CE}$ for both X_l and X_u. Feature vectors for all {0°, 90°, 180°, 270°} rotated images are used to calculate the third loss: rotation loss $\mathcal{L}_{Rot}$. f_{c1}, f_{c2} and f_{c3} represent three different linear classifiers (i.e., fully-connected layers), that are employed to map feature vectors to the target space for different loss calculations respectively. 91
5.3 Dataset split setting of CIFAR-10 for novel categories clustering task. The first five classes, i.e., airplane, automobile, bird, cat and deer, are labeled data X_l with Y_l and the rest five classes, i.e., dog, frog, horse, ship and truck, are used as unlabeled data X_u. 97
5.4 Ramp-up function for CIFAR-100 98
5.5 Ablation study: Self-supervised learning. For variant (7), the pre-training and fine-tuning schemes in [81] are followed. 105
5.6 Ablation study: Pairwise similarity learning. For variant (5), $\mathcal{L}_{BCE}$ is not included for the model optimization. For variant (6a), the proposed conditional probability is replaced with cosine similarity. For variant (6b), the proposed conditional probability is replaced with rank statistics in [81]. 108
5.7 Ablation study: Mixed Classification. For variant (2), $\mathcal{L}_{MSE}$ is not included for the model optimization. For variant (3), $\mathcal{L}_l$ is not included for the model optimization. For variant (4), $\mathcal{L}_u$ is not included for the model optimization. 109
5.8 3D t-SNE visualizations of embeddings ϕ for CIFAR-100 unlabeled training data under different training schemes (A). Each single point represents one unlabeled training image and different colors correspond to ground-truth labels. Recommend to view in color version. 110

5.9 3D t-SNE visualizations of embeddings ϕ for CIFAR-100 unlabeled training data under different training schemes (B). Each single point represents one unlabeled training image and different colors correspond to ground-truth labels. Recommend to view in color version. 111
5.10 3D t-SNE visualizations of embeddings ϕ for CIFAR-100 unlabeled training data under different training schemes (C). Each single point represents one unlabeled training image and different colors correspond to ground-truth labels. Recommend to view in color version. 112

6.1 Overall structure of this thesis 115

List of Tables

2.1 Testing Accuracy of WRN and ResNets on CIFAR datasets (%) . . 17

3.1 WRN Architecture 35
3.2 Datasets Used For Evaluating the Proposed Algorithm 40
3.3 Implementation details of the experiments 41
3.4 Testing Accuracy of WRN and Proposed Model (%) 45
3.5 Ablation Study: Testing Accuracy of pre-activation resnet with ELM classifier (%) 52
3.6 Ablation Study: Testing Accuracy of pre-activation resnet and WRN with Kernel SVM classifier and the proposed method (%) 53

4.1 Implementation details of the experiments 72
4.2 Ablation Study Results: Testing error rate 75
4.3 Performance improvement Δ given by self-supervision under different graph construction methods 76
4.4 Performance improvement Δ given by feature space geometry preserving under different feature learning schemes 77
4.5 Top-1 error rate on CIFAR datasets (%) 82
4.6 Top-1 error rate on *mini*ImageNet dataset (%) 83

5.1 Different linear layers of the proposed algorithm 94
5.2 Implementation details of the experiments 99
5.3 Clustering Accuracy on unlabeled training data (%). w/ SSL denotes self-supervised learning is used in the pre-training stage of the network. All reported results are using the same network architecture and averaged over ten runs. 101
5.4 Mixed Classification Performance on CIFAR datasets (%). The trained network is evaluated on testing data for each dataset. "Old" refers to testing images from classes that have been trained by training images with label information, while "Novel" refers to testing images from classes that no label information is used during training. "All" refers to all testing images in the dataset. 103
5.5 Mixed Classification Performance on SVHN dataset (%) 104
5.6 Ablation Study (%) 106

Summary

The great success of deep neural networks on visual recognition has inspired numerous real-world applications. However, such superior performance is closely related to model complexity and the amount of annotated data. Over-deepened networks and lack of data annotation will degrade generalization capability of the model as over-fitting problems arise. In this thesis, the focus is on extracting robust semantic features in image data by alleviating over-fitting problems under different learning frameworks.

For the first work in this thesis, the over-fitting problem of Extreme Learning Machine (ELM) classifier when combined with convolutional neural network (CNN) for supervised learning is studied. To remedy the over-fitting issue while still utilizing excellent feature extraction capability of deep neural network, a novel deep and wide feature based ELM (DW-ELM) is proposed by employing wide architecture design of residual networks (ResNets) for feature extraction. The empirical study has demonstrated that when combined with ELM that serves as a classifier, using wide ResNets (WRNs) for feature extraction can greatly compress the generalization gap. Extensive experiments on five visual benchmark datasets have shown that the proposed DW-ELM is able to boost and stabilize the generalization capability of the original backbone CNN model to a great extent.

For the second work in this thesis, scarce annotation problem of semi-supervised learning is studied. Label propagation is commonly utilized to provide information flow from labeled data to unlabeled data as an transductive learning algorithm for pseudo-labeling purpose. Two limitations of previous algorithms that ultimately lead to noisy and incomplete information flow are addressed in this thesis. The first limitation is that the learned feature mapping is highly likely to be biased and can easily over-fit noise as only labeled data are used for feature learning. The second limitation is the loss of local geometry information in feature space during label propagation. This thesis proposes a novel algorithm to alleviate the above mentioned issues by incorporating self-supervised learning into feature learning phase

and utilizing reconstruction concept to preserve local geometry. Extensive experiments conducted on three visual benchmark datasets have verified the effectiveness of the proposed algorithm and the empirical results show that the proposed algorithm consistently outperforms most of the state-of-the-art semi-supervised learning algorithms.

For the third work in this thesis, the focus is on novel visual categories learning, which is a clustering problem with certain prior knowledge. The task can also be considered as a special type of semi-supervised learning where the categories of unlabeled data and labeled data are disjoint from each other. The main challenge is how to effectively leverage knowledge in labeled data to unlabeled data when they are independent from each other, and not belonging to the same set of categories. Two issues commonly inherent in previous algorithms: 1) All of previous algorithms are comprised of multiple training phases, which makes it difficult to train the model in an end-to-end fashion. 2) Strong dependence on the quality of pairwise similarity pseudo labels limits the performance as pseudo labels are vulnerable to noise and bias. This thesis proposes an end-to-end novel visual categories learning algorithm via auxiliary self-supervision tasks, such that labeled data and unlabeled data will share the same set of surrogate labels and overall supervising signals can have strong regularization. Moreover, local structure information in feature space is utilized for pairwise pseudo label construction as local properties are more robust to noise. Experiments conducted on three visual benchmark datasets have indicated the effectiveness of the proposed algorithms and new state-of-the-art performances have been achieved.

Overall, this thesis discussed the over-fitting problem of deep learning-based feature learning in visual understanding from two perspectives : 1) Over-fitting problem of supervised learning due to network architecture. 2) Over-fitting problem in semi-supervised and unsupervised learning due to the lack of data annotation.

Symbols and Acronyms

Symbols

X	training data
$\mathbf{a_i}$	weight of hidden node i
b_i	bias of hidden node i
$\mathbf{H}$	output function of hidden layer in matrix form
$\boldsymbol{\beta}$	output weight for hidden layer in matrix form
$\mathbf{T}$	target value in matrix form
$\mathbf{H}^{\dagger}$	Moore–Penrose generalized inverse of matrix $\mathbf{H}$
X_l	labeled training data
Y_l	label of labeled training data
X_u	unlabeled training data
$\mathcal{L}_{X_l}$	loss on labeled data
$\mathcal{L}_{X_u}$	loss on unlabeled data

Acronyms

BN	Batch Normalization
CNN	Convolutional neural network
DMT	Deep Metric Transfer
DNN	Deep Neural Network
DW-ELM	Deep and Wide feature based ELM
E2E	End-to-end
ELM	Extreme Learning Machine
EMA	Exponential Moving Average
LLE	Locally Linear Embedding

LP	Label Propagation
LPLGP	Label Propagation via Local Geometry Preserving
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
ResNets	Residual networks
SGD	Stochastic Gradient Descent
SLFNs	Single-Layer Feedforward Neural Networks
SNE	Stochastic Neighbor Embedding
SPN	Similarity Prediction Network
SVHN	Street View House Numbers
SVM	Support-Vector Machine
WRNs	Wide ResNets

s.t.	subject to

Chapter 1

Introduction

Chapter 1 briefly introduces main topic of this thesis, i.e. over-fitting problems of deep learning-based feature learning in visual understanding. Research background and motivations are discussed in Section 1.1. Research objectives and main contributions are presented in Section 1.2. Section 1.3 summarizes the overall structure of this thesis.

1.1 Background and Motivation

Deep learning has achieved great success on visual recognition in recent years. Two crucial components determine the ultimate performance: network architecture and data.

With increasing demand for neural network model on complex visual tasks, structure of the model becomes more complicated. Depth of the neural network is a very important network dimension to determine model's ability. Generally, the deeper the network, the better expressivity or representation capability the model will have [1]. As the result, neural networks become deeper and depth of the model grows from several layers [2] to over hundreds of layers [3]. Exceptional feature extracting capability of deep CNN model has attracted researchers to combine CNN and powerful classifier, e.g. ELM classifier, to further boost the overall performance [4]–[7]. Extreme Learning Machine (ELM) [8] is a variant of single-layer feedforward neural networks (SLFNs) with randomly assigned and fixed neurons between the input and hidden layer. As a robust classifier with fast training speed and very little human intervention, ELM classifier is a preferable option to replace the fully-connected layer in CNN model to do classification. However, features extracted from over-deepened CNN model may suffer from severe over-fitting problem when fed into an ELM classifier, which is seldom addressed by previous research works.

While the main focus has been placed on depth of the network for quite a long time, some research works [9]–[12] also explored the importance of another network dimension: width of model. Width of the network means the number of neurons or the number of kernels for convolutional neural network at each layer. By increasing the width of neural network, more features at each layer can be learned such that representational ability of the model can be enhanced. Wider networks are shown to demonstrate better generalization ability in several empirical studies [9], [11], [13], [14]. Moreover, extra computation arising from larger width (more feature maps at each layer) will be in favor of GPU as computation in the same layer can be done in parallel. Therefore, wider neural networks exhibit superior computational efficiency to the deeper ones, whose structure is more apt to a sequential manner. Inspired by such favorable property of wide neural networks, this thesis explores the wide architecture design of CNN model as feature extractor for ELM classifier.

While the network architecture is growing more and more complex, the corresponding training data are more demanding than ever before. To be more specific, the superior performance of deep models heavily relies on large amount of annotated data, which can be expensive, time-consuming and impractical for real-world applications [15]. Semi-supervised learning [16] is utilized to alleviate the strong dependence of data annotation by exploiting structure information in unlabeled data. The nature of semi-supervised learning determines that over-fitting on labeled data is unavoidable under such learning framework. Therefore, how to alleviate the problem such that meaningful information of labeled data can be leveraged to unlabeled data is important to improve the algorithm. Two different scenarios of scarce data annotation problem are studied in this thesis. The first one is the standard semi-supervised learning where a common assumption is adopted, i.e., there is always labeled data from the same class of unlabeled data. The second one is more like a clustering task on unlabeled data as the assumption adopted is that the categories of unlabeled data and labeled data are disjoint from each other.

Self-supervised learning is one prominent direction in deep learning for image recognition tasks in recent years [17]–[20], which is largely due to its good generalization. Pretext task replaces the role of ground-truth label to provide supervision signal, e.g. colorization, solving jigsaw puzzles, and predicting rotations. As no ground-truth label information is required, self-supervised learning is applicable to all data no matter whether labeled or not. Such property of self-supervised learning is desirable for semi-supervised learning to counter over-fitting problems as extra supervising signal can provide strong regularization. Moreover, as local data geometry/structure information has been observed to be beneficial for manifold learning [21]–[23], it is promising to further study its application in low-dimension feature space for complex visual tasks such that more robust model can be learned. Therefore, this thesis investigates the use of self-supervised learning together with local data geometry information in semantic feature learning when data annotation is scarce.

1.2 Objectives and Contributions

The objectives of this thesis are defined as follows:

1. To alleviate over-fitting problem of ELM classifier for image classification while still utilizing the preferable feature extraction capability of deep neural neural networks.
2. To better leverage the knowledge in labeled data to unlabeled data when data annotation is scarce for visual recognition.

The contributions of this thesis to achieve the objectives are presented in the following part.

To achieve objective 1, a novel deep and wide feature based Extreme Learning Machine (DW-ELM) taking advantages from ELM classifier and "widened" convolutional neural networks is proposed in the first work. By taking advantage of wide architecture design of CNN model, over-fitting problem of ELM classifier can be largely alleviated and meanwhile the preferable feature extraction capability of deep neural networks can also be exploited. Experiment results has demonstrated that the generalization performance of the proposed DW-ELM is stable and effective. Moreover, the effect of width and depth of CNN model as a feature extractor on ELM classifier for image classification is explored. Insights that CNN model as a feature extractor for ELM image classifier should be deep and also wide are given. This work has been published in a journal, with details given as follows:

- Yuanyuan Qing, Yijie Zeng, Yue Li, and Guang-Bin Huang, "Deep and wide feature based extreme learning machine for image classification", en, *Neurocomputing*, vol. 412, pp. 426–436, Oct. 2020. DOI: 10.1016/j.neucom.2020.06.110

To achieve objective 2, two works has been conducted for different tasks.

The assumption adopted in the second work is that labeled data and unlabeled data belong to the same set of categories. A novel transductive pseudo-labeling based algorithm for deep semi-supervised image classification is proposed in the second work. The pseudo label construction of unlabeled data is performed by

label propagation, which propagates label information in latent feature space via similarity graph. Self-supervised learning is incorporated into the feature extraction phase such that limitations of scarce ground-truth label information, i.e., the learned feature mapping is highly likely to over-fit noise, can be avoided and cleaner information flow in subsequent label propagation is achieved. Moreover, local geometry information of data in feature space is preserved via reconstructing feature vector by its neighbors to build similarity graph. Experiment results have verified the effectiveness of the proposed algorithm and the results show that the proposed algorithm consistently outperforms most of the state-of-the-art semi-supervised learning methods under the same network architecture. This work has been submitted to a journal, with details given as follows:

- Yuanyuan Qing, Yijie Zeng, and Guang-Bin Huang, "Label propagation via local geometry preserving for deep semi-supervised image recognition", *Neural Networks*, 2020, Under review

The assumption adopted in the third work is that labeled data and unlabeled data belong to disjoint categories. Different from the second work, pairwise similarity pseudo labels are constructed for unlabeled data. An end-to-end novel visual categories learning algorithm is proposed in the third work by utilizing self-supervision signals simultaneously with pairwise similarity information. By doing so, the labeled data and unlabeled data will share the same set of surrogate labels and strong regularization can be imposed in the overall supervising signals. Furthermore, robust local structure properties in noisy feature space are enforced for the construction of pairwise similarity pseudo labels to capture data relationship more accurately. Experiment results have demonstrated the effectiveness of the proposed design of the algorithm and it has been observed that the proposed algorithm has outperformed other state-of-the-art methods. This work has been submitted to a journal, with details given as follows:

- Yuanyuan Qing, Yijie Zeng, Qi Cao, and Guang-Bin Huang, "End-to-end novel visual categories learning via auxiliary self-supervision", *Neural Networks*, vol. 139, pp. 24–32, 2021

1.3 Thesis Organization

The remainder of this thesis is organized as follows:

- Chapter 2 gives an extensive review for related works, including ELM algorithm, WRNs, deep semi-supervised learning for visual tasks, self-supervised learning and novel visual categories learning.
- Chapter 3 explores the effect of width and depth of CNN model as a feature extractor on ELM classifier and presents the proposed deep and wide feature based extreme learning machine (DW-ELM) algorithm for image classification.
- Chapter 4 explores the use of self-supervised learning to counter over-fitting problems due to lack of data annotation and presents the proposed label propagation algorithm via local geometry preserving for deep semi-supervised image recognition.
- Chapter 5 explores end-to-end training for novel visual categories learning and presents the proposed algorithm where auxiliary self-supervision task is exploited to impose strong regularization constraint.
- Chapter 6 concludes this thesis and discusses future research directions.

Chapter 2

Literature Review

Chapter 2 gives an extensive literature review of related works. Extreme Learning Machine (ELM) and wide ResNets (WRNs) are reviewed in Section 2.1 and Section 2.2 respectively. Previous state-of-the-art deep semi-supervised learning algorithms for visual tasks are presented in Section 2.3. Lastly, introductions of self-supervised learning and novel visual categories learning are given in Section 2.4 and Section 2.5 respectively.

2.1 Extreme Learning Machine

2.1.1 Basic ELM Algorithm

Extreme Learning Machines (ELM) [8] , introduced by Huang et al. in 2005, is an algorithm for single-hidden layer feed forward neural networks (SLFNs). This algorithm adopts random hidden nodes and calculates the output mapping analytically under optimization constraints. A typical single-hidden layer feed forward neural network (SLFN) is shown in Figure 2.1.

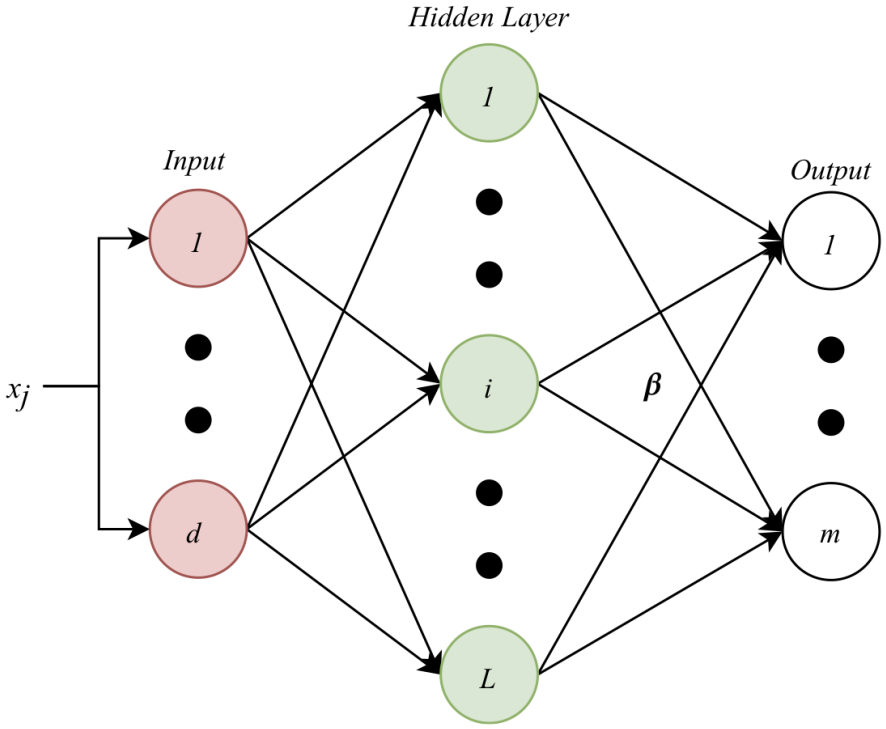


FIGURE 2.1: Single-hidden Layer Feed Forward Neural Network

The input $\mathbf{x}$ is assumed to be of dimension d and there are L hidden nodes. The output function of a typical addictive hidden node i can be expressed as:

$$g_i = G\left(\mathbf{a}_i, b_i, \mathbf{x}\right) = g\left(\mathbf{a}_i \cdot \mathbf{x} + b_i\right), \mathbf{a}_i \in \mathbf{R}^d, b_i \in R \tag{2.1}$$

where $\mathbf{a_i}$, b_i are the weight and bias of hidden node i respectively. The activation function g can take different kinds of form, such as:

1. sigmoid function:

$$g_i = \frac{1}{1 + \exp\left(-(\mathbf{a}_i \cdot \mathbf{x} + b_i)\right)} \tag{2.2}$$

2. hard-limit function:

$$g_i = \begin{cases} 1 & if\ \mathbf{a}_i \cdot \mathbf{x} + b_i \geq 0 \\ 0 & otherwise \end{cases} \tag{2.3}$$

3. Gaussian function:

$$g_i = \exp\left(-b_i ||\mathbf{x} - \mathbf{a}_i||^2\right) \tag{2.4}$$

4. multiquadric function:

$$g_i = \sqrt{(||\mathbf{x} - \mathbf{a}_i||^2 + b_i^2)} \tag{2.5}$$

The final output of this SLFN then can be expressed as:

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i g_i(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G\left(\mathbf{a}_i, b_i, \mathbf{x}\right) \tag{2.6}$$

where $\boldsymbol{\beta}_i$ is the output weight for hidden node i. For N samples $(\mathbf{x}_j, \mathbf{t}_j) \in \mathbf{R}^d \times \mathbf{R}^m$, the output of the network can be written as:

$$\boldsymbol{o}_j = \sum_{i=1}^{L} \boldsymbol{\beta}_i g_i\left(\mathbf{x}_j\right) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G\left(\mathbf{a}_i, b_i, \mathbf{x}_j\right), j = 1, \ldots, N \tag{2.7}$$

According to the interpolation theorem [24], for N distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in \mathbf{R}^d \times \mathbf{R}^m$, there exists $\mathbf{a_i}$, b_i and $\boldsymbol{\beta}_i$ such that the approximation error of this SLFN is zero, which means that

$$\boldsymbol{o}_j = \boldsymbol{t}_j, j = 1, \ldots, N \tag{2.8}$$

if $L = N$. After rewriting the above equations in a more compact matrix form, they become to

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2.9}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}\left(\mathbf{x}_1\right) \\ \vdots \\ \mathbf{h}\left(\mathbf{x}_N\right) \end{bmatrix} = \begin{bmatrix} G\left(\mathbf{a}_1, b_1, \mathbf{x}_1\right) & \ldots & G\left(\mathbf{a}_L, b_L, \mathbf{x}_1\right) \\ \vdots & \vdots & \vdots \\ G\left(\mathbf{a}_1, b_1, \mathbf{x}_N\right) & \cdots & G\left(\mathbf{a}_L, b_L, \mathbf{x}_N\right) \end{bmatrix} \tag{2.10}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L\times m} \quad and \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N\times m} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & \vdots \\ t_{N1} & \cdots & t_{Nm} \end{bmatrix} \tag{2.11}$$

However, in most cases, the number of hidden nodes is much smaller than the number of training samples, i.e., $L < N$ or even $L \ll N$. Then, in order to minimize the error function, the aim of ELM is to find the optimal $\widehat{\boldsymbol{\beta}}$ such that

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| \tag{2.12}$$

Then the optimal value of $\widehat{\boldsymbol{\beta}}$ can be directly calculated by least-square solution:

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{2.13}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$.
The overall basic ELM algorithm is summarized in Algorithm 1.

Algorithm 1 Basic ELM algorithm

1: **Input**: N training samples $(\mathbf{x}_j, \mathbf{t}_j) \in \mathbf{R}^d \times \mathbf{R}^m$
2: **for** $i \leftarrow 1$ to L **do**
3: randomly generate parameters $\mathbf{a}_i \in \mathbf{R}^d, b_i \in R$ for hidden node i
4: calculate the output function $\mathbf{H}$ of hidden nodes according to Equation (2.10)
5: calculate the hidden node mapping matrix $\boldsymbol{\beta} \leftarrow \mathbf{H}^{\dagger}\mathbf{T}$
6: the final network mapping function $\boldsymbol{f}(\mathbf{x}) \leftarrow \mathbf{h}(\mathbf{x})\boldsymbol{\beta}$
7: **Output**: The input to output mapping function $\boldsymbol{f} : \mathbf{R}^d \rightarrow \mathbf{R}^m$

2.1.2 Regularized ELM Algorithm

According to Equation (2.13), $\mathbf{H}^{\dagger}$ is required to calculate the hidden node mapping matrix. $\mathbf{H}^{\dagger}$ can be calculated by orthogonal projection formula:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \tag{2.14}$$

or

$$\mathbf{H}^{\dagger} = \mathbf{H}^{T}(\mathbf{H}\mathbf{H}^{T})^{-1} \tag{2.15}$$

if $(\mathbf{H}^{T}\mathbf{H})$ is non-singular in Equation (2.14) or $(\mathbf{H}\mathbf{H}^{T})$ is non-singular in Equation (2.15).

According to ridge regression (Tikhonov regularization), to improve the stability and reliability of the matrix inverse calculation, a positive bias term $\frac{\mathbf{I}}{\lambda}$ can be added in the calculation of $\mathbf{H}^{\dagger}$. Then the resulting calculated hidden node mapping matrix $\boldsymbol{\beta}$ will be:

$$\boldsymbol{\beta} = \left(\mathbf{H}^{T}\mathbf{H} + \frac{\mathbf{I}}{\lambda}\right)^{-1}\mathbf{H}^{T}\mathbf{T} \tag{2.16}$$

or

$$\boldsymbol{\beta} = \mathbf{H}^{T}\left(\mathbf{H}\mathbf{H}^{T} + \frac{\mathbf{I}}{\lambda}\right)^{-1}\mathbf{T} \tag{2.17}$$

Equation (2.16) and (2.17) are actually proved by Huang et al. [25] that these two equations are equivalent to find the optimal value of $\boldsymbol{\beta}$ that minimizes:

$$\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^{2} + \lambda\|\boldsymbol{\beta}\|^{2} \tag{2.18}$$

The additional term $\lambda\|\boldsymbol{\beta}\|^{2}$ will compress the norm of the weights in hidden node mapping and therefore a better generalization performance can be expected.
Then the regularized basic ELM algorithm is summarized in Algorithm 2.

Algorithm 2 Regularized Basic ELM algorithm

1: **Input**: N training samples $(\mathbf{x}_j, \mathbf{t}_j) \in \mathbf{R}^d \times \mathbf{R}^m$
2: **for** $i \leftarrow 1$ to L **do**
3: randomly generate parameters $\mathbf{a}_i \in \mathbf{R}^d, b_i \in R$ for hidden node i
4: calculate the output function $\mathbf{H}$ of hidden nodes according to Equation (2.10)
5: calculate the hidden node mapping matrix:

$$\boldsymbol{\beta} \leftarrow \begin{cases} \left(\mathbf{H}^{T}\mathbf{H} + \frac{\mathbf{I}}{\lambda}\right)^{-1}\mathbf{H}^{T}\mathbf{T} & if\ L < N \\ \mathbf{H}^{T}\left(\mathbf{H}\mathbf{H}^{T} + \frac{\mathbf{I}}{\lambda}\right)^{-1}\mathbf{T} & otherwise \end{cases}$$

6: the final network mapping function $\boldsymbol{f}(\mathbf{x}) \leftarrow \mathbf{h}(\mathbf{x})\boldsymbol{\beta}$
7: **Output**: The input to output mapping function $\boldsymbol{f} : \mathbf{R}^d \rightarrow \mathbf{R}^m$

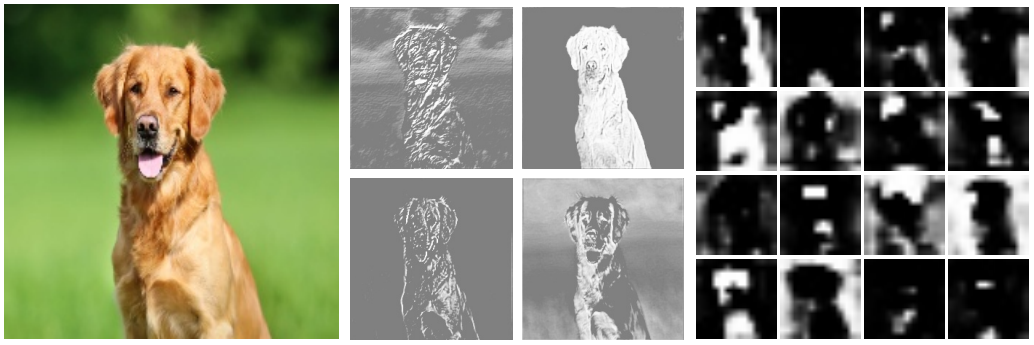
Apart from the basic ELM, there are many ELM variants: kernel-based ELM [25], fully complex ELM [26], on-line sequential ELM [27], and ELM based Auto Encoder [28]. ELM is widely applied in both supervised and unsupervised learning, such as classification [25], [29], [30], feature learning [31], [32] and clustering [33].

As ELM doesn't need any tuning in the hidden layer, which avoids spending a large amount of time and computational resources on iterative gradient-based learning, training speed of ELM can be extremely fast. Moreover, compared with other frequently-used classifiers, such as support-vector machines (SVM) [34], ELM requires less human intervention while being able to provide a comparable or even better generalization capability [8], [25], [29].

2.2 Wide Residual Networks

2.2.1 Residual Network

Convolutional neural network model is extensively applied in the field of visual imagery as the spatial information of image can be preserved by CNN model. Compared with traditional multi-layer neural networks, CNN model greatly reduces the number of parameters due to the use of pooling layers, shared weights of filters and the local connectivity of convolution. For CNN model, compared with shallow features extracted in the superficial layers, features extracted in deeper layers are more complex and abstract, which make the network feasible for complicated tasks. By increasing the depth of network, the model is expected to learn more latent and abstract features, thus a better feature representation can be achieved.



(a) (b) (c)

FIGURE 2.2: Feature extracted by VGG-16 at different layers

As shown in Figure 2.2, the feature maps of the original dog picture extracted by a pre-trained VGG-16 [35] model in different layers are compared. Figure 2.2b shows the feature maps in shallow layers, and Figure 2.2c shows the feature maps in deep layers. The difference between high-level information (features in deep layers) and low-level information (features in shallow layers) is obvious. The abstract feature extraction capability of deep CNN models makes the networks feasible for complicated tasks, such as to classify whether the image is a dog or cat.

Figure 2.3 shows a general three-layer feed-forward neural network. The output of such network can be expressed as:

$$f_l = a\left(\mathbf{x}_l, \mathbf{W}_l\right) \tag{2.19}$$

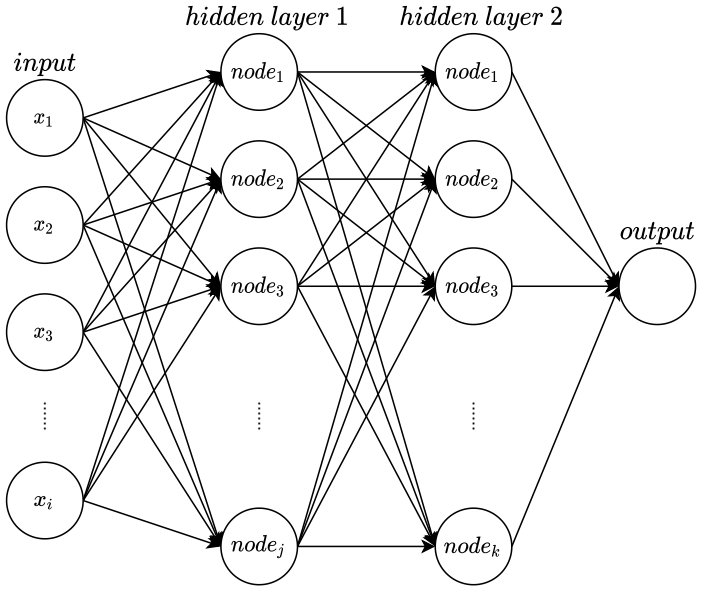


FIGURE 2.3: Three-layer Feed Forward Neural Network

where f_l, a, $\mathbf{x}_l$ and $\mathbf{W}_l$ are the output, activation function, input and weight matrix for l-th layer. To calculate the gradient of weights, chain rule is used:

$$\Delta w_1 = \frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial f_3}\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial w_1} \tag{2.20}$$

where w_1 is a weight parameter from input layer to hidden layer 1.

Form Equation (2.20) it can be observed that with more and more hidden layers, the right-hand side of the equation will have more terms like $\frac{\partial f_4}{\partial f_3}$, $\frac{\partial f_5}{\partial f_4}$, and $\frac{\partial f_l}{\partial f_{l-1}}$. And such derivatives are actually the gradients of the activation function used at each layer. If the derivative value is smaller than 1, then the final multiplication result of many small values will be extremely small such that the weight cannot learn anything from the back-propagation process. This is the gradient vanishing problem, which causes immense difficulty on training deep neural networks, including deep CNN models.

There are many alternatives to try to alleviate this problem. One approach is to use an activation function such that the gradient will never be smaller than 1. ReLU (rectified linear unit) function was firstly introduced in 2011 [36] to meet this requirement:

$$a(x) = \max(0, x) \tag{2.21}$$

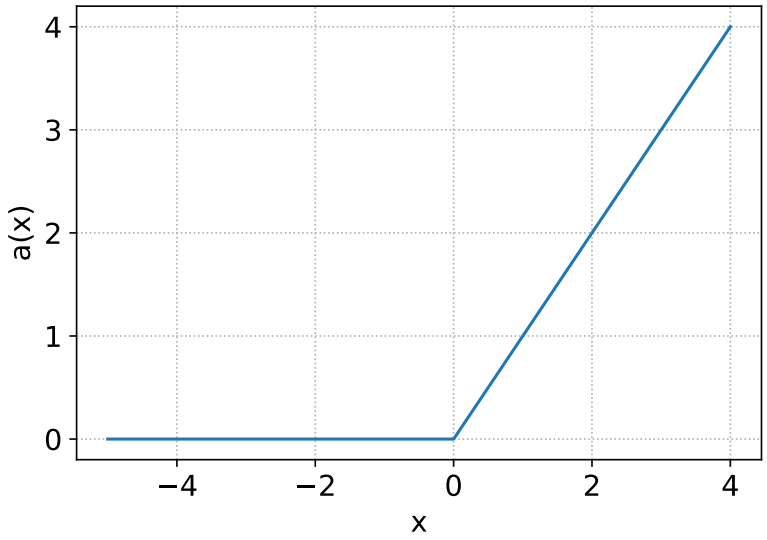


FIGURE 2.4: Rectified linear unit function

For positive value of x, the derivative will always be 1, which avoids gradient vanishing problem to a large extent. Another notable approach is the use of Batch Normalization (BN) introduced in 2015 [37]. The commonly used normalization was only for the input layer, where the input data will be normalized to have zero mean and a unit variance. Such pre-processing can help to accelerate the training convergence [38]. Inspired by this idea, the batch normalization method regards all the outputs from every intermediate layer as the inputs for the subnetworks consisting of the subsequent layers and normalization is performed for all of them before next layer's non-linear activation function. After such normalization, the input data for all layers in the network will have the same zero mean, unit variance normal distribution, which can help the activation function to respond more sensitively to the input data, therefore the resulting calculated gradients will stay at a relatively high level.

Techniques such as ReLU [36] and Batch Normalization (BN) [37] have successfully alleviated the problem of gradient vanishing when the network is getting deeper. However, the degradation problem, i.e., optimization difficulty in deep network, comes up with growing number of layers. Therefore, He et al. [3] have proposed the idea that instead of learning the original underlying mapping $\mathcal{H}(\mathbf{x})$, the residual

mapping $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$ should be easier to optimize. Then the network actually becomes: $\mathcal{F}(\mathbf{x}) + \mathbf{x}$, which can be achieved by the use of shortcut connection [39].

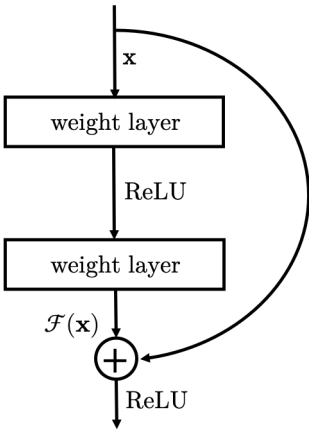


FIGURE 2.5: Residual Unit in ResNet

Figure 2.5 shows the shortcut connection that bypasses two layers. The shortcut connection can be identity mapping or projection of the input $\mathbf{x}$, and the number of layers bypassed can be two, three or even more. By making use of this connection, what is desired for the two layers to learn is no longer the original mapping $\mathcal{H}(\mathbf{x})$, the residual function $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$ is expected and now let $\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathbf{x}$.

In later experiments [3], it has been proved that such residual network will not experience the degradation problem and it can gain significant performance improvement on multiple tasks and datasets by stacking more layers to it. ResNets have successfully solved the degradation problem and therefore the depth of the neural network can be increased without concerning the degradation problem. As a result, extremely deep neural networks become possible and the proposed 152-layer ResNet, whose depth is eight times of VGG, has achieved very successful results on multiple competitions [3].

2.2.2 Wide Residual Networks

From LeNet-5 [2] to AlexNet [40], VGG [41] and ResNet [3], neural network architecture is growing deeper and deeper. For an extreme case, the depth of ResNet can even be increased to more than one thousand layers. The motivation behind all above-mentioned designs is to overcome obstacles when stacking more layers to the model such that a very deep neural network can be better trained. When the

community is digging into the effect of depth on model performance, the wide version of residual network proposed in [11] explored a different approach to improve feature representation capability of the model. Although increasing the width of network will result in quadratic growth of parameters while only linear growth for increasing the depth, wider network structure is more practically favorable as GPU is more adept at parallel computing, which makes it better utilized in the training of wide networks than deep networks. The empirical studies in [11] have shown that a wide residual network can achieve a better generalization performance and much more faster training speed than the original thin and deep model. Experiment results of WRN and ResNets on CIFAR datasets are shown in Table 2.1. In addition, several research works [9], [13], [14] further studied the effect of width on model generalization and suggested that wider networks can generalize better.

TABLE 2.1: Testing Accuracy of WRN and ResNets on CIFAR datasets (%)

model	depth	CIFAR-10	CIFAR-100
ResNet[3]	110	93.57	74.84
	1202	92.07	72.18
Preact-ResNet[42]	110	93.63	-
	164	94.54	75.67
	1001	95.08	77.29
WRN(10 times wider)[11]	28	**96.00**	**80.75**

Feature extraction and classification are key components to tackle most computer vision tasks and neither is dispensable. A powerful classifier is able to well discriminate the features of data while the quality of feature largely depends on feature extractor. Convolutional neural network model has achieved great success due to its exceptional feature extracting ability and since then the community seldom keeps an eye on classifier. Conventional classifier used in most neural networks is one fully-connected layer, which is handicapped by suboptimal problem brought by back-propagation and limited classification capability. Even though some research works [43], [44] tried to boost the overall performance by combining CNN and SVM classifier, the required hyper-parameter tuning of SVM is time-consuming. ELM, as a robust classifier with fast training speed and very little human intervention, turns out to be more preferable. Therefore, several works [4]–[7] that attempted to utilize the superiority of ELM classifier and CNN models were done. However, most of them are just limited to a specific application and requiring special algorithm design to achieve good results. As a result, the over-fitting problem of ELM

classifier when combined with deep CNN model as feature extractor is seldom observed. In the first work of this thesis, such over-fitting issue is well addressed by evaluating the generalization capability of "widened" convolutional neural networks as feature extractor for ELM classifier.

2.3 Deep Semi-Supervised Learning

For deep semi-supervised learning in visual recognition, most of the algorithms fall into a common framework, which can be described by:

$$loss = \mathcal{L}_{X_l} + \lambda \mathcal{L}_{X_u} \tag{2.22}$$

where $\mathcal{L}_{X_l}$ and $\mathcal{L}_{X_u}$ represent the loss relating to labeled data and unlabeled data respectively, and the coefficient λ controls the relative importance between them. $\mathcal{L}_{X_l}$ commonly takes the form of cross-entropy loss between network prediction and the ground truth label for labeled data, same as what would be done in supervised learning, while $\mathcal{L}_{X_u}$, which is used to explore useful structure information in the large amount of unlabeled data, may be of various forms. Therefore, starting from this equation, two branches of research work will be discussed: one is focusing on expanding the applicable scope of $\mathcal{L}_{X_l}$, i.e., produce pseudo labels for unlabeled data such that semi-supervised learning can be transformed to virtual supervised learning; another one is to make use of the smoothness assumption [16], [45] to impose consistency-based constraints on unlabeled data.

2.3.1 Consistency-based Algorithms

According to smoothness assumption, the outputs of mapping function $f(x_i)$ and $f(x_j)$ should be close to each other if the input data x_i and x_j are close. Therefore, the input data are assumed to lie on a smooth enough surface such that the prediction of data after applied with a small perturbation should not change too much. Consistency-based algorithms aim to learn a model robust to such perturbations by minimizing the difference between model predictions of original data and the data after perturbation. Such difference can be calculated in different forms, e.g., KL-divergence [46], [47] and mean square error [48]–[52]. Algorithms under this category mainly vary in the way how perturbation is created and imposed to the data.

As shown in Figure 2.6, $\widehat{X}_u$ refers to unlabeled data after perturbation. The $\mathcal{L}_{X_u}$ measures the difference between the network predictions $\widehat{y}_{X_u}$ and $\widehat{y}_{\widehat{X}_u}$. The loss

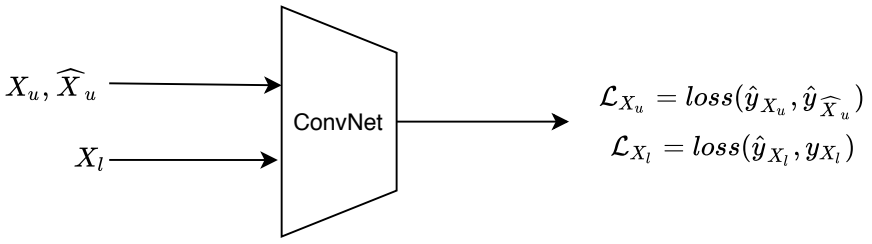


FIGURE 2.6: Consistency-based Semi-Supervised Learning Algorithm

function of $\mathcal{L}_{X_u}$ can be KL-divergence:

$$D_{KL}(\widehat{y}_{X_u} \| \widehat{y}_{\widehat{X}_u}) = -\sum_{X_u} \widehat{y}_{X_u} \log(\frac{\widehat{y}_{\widehat{X}_u}}{\widehat{y}_{X_u}}) \tag{2.23}$$

or MSE:

$$MSE(\widehat{y}_{X_u}, \widehat{y}_{\widehat{X}_u}) = \sum_{X_u} \|\widehat{y}_{\widehat{X}_u} - \widehat{y}_{X_u}\|^2 \tag{2.24}$$

According to [51], MSE is a stronger requirement than KL divergence for semi-supervised learning.

2.3.1.1 Π-Model & Temporal Ensemble

Π-Model [51] measures the difference between the predictions of the same data by the same network at each iteration, and the perturbation is given by random dropout and randomness in data augmentation (e.g., random translation) when the same data passes the network twice. Temporal Ensemble [51] measures the difference between the current prediction of data and the exponential moving average prediction of the same data in the past.

2.3.1.2 Mean Teacher

Mean Teacher proposed in [52] is mainly inspired by Temporal Ensemble in [51]. Instead of accumulating the past network predictions, Mean Teacher applies exponential moving average (EMA) on the network parameters during the training process so that the EMA model can be updated after each iteration, which is much faster than updating after each epoch in Temporal Ensemble. Therefore, Mean

Teacher model imposes the consistency constraint by minimizing the difference between the predictions of the current network and the EMA model.

2.3.1.3 Virtual Adversarial Training

Random perturbation such as Gaussian noise and random dropout is effective to smooth the data manifold, but it has been found that the learned model will be sensitive to changes in specific directions, i.e., the adversarial direction [53]. Therefore, the adversarial training is proposed in [53] to alleviate such problem by adding the adversarial perturbation to the input for supervised learning. Inspired by this work, VAT proposed in [46] applies the adversarial training for unlabeled data by approximating the virtual label with the current network prediction, such that it can be applied for semi-supervised learning.

Besides the above research works, there are many other semi-supervised learning algorithms have also achieved great progress by making use of the consistency constraints. ICT [48] makes use of the argumentation algorithm Mixup on unlabeled data and minimizes the difference between the model prediction of the mixed data and the mixed prediction of the original data. VAdV proposed in [50] follows the same philosophy in VAT, but shifts the focus to network dropout that can influence the model prediction the most. Fast-SWA [54] further analyzes consistency-based algorithms and proposes to average model weights along SGD trajectory in order to alleviate diversity problems during late stage of training.

2.3.2 Pseudo-Labeling

As the main issue for semi-supervised learning is the lack of labels, a natural idea will be to assign pseudo labels to unlabeled data based on labeled data information and then transform the problem into a supervised learning scenario. Therefore the effectiveness of such algorithms primarily depends on the quality of pseudo labels. The main difference among algorithms in this category lies in how labeled data information is utilized to create pseudo labels.

As shown in Figure 2.7, p_{X_u} refers to pseudo labels for unlabeled data. The use of p_{X_u} is similar with y_{X_l}, i.e., the ground-truth labels for X_l. Therefore, the loss function commonly taken by $\mathcal{L}_{X_u}$ is cross-entropy loss:

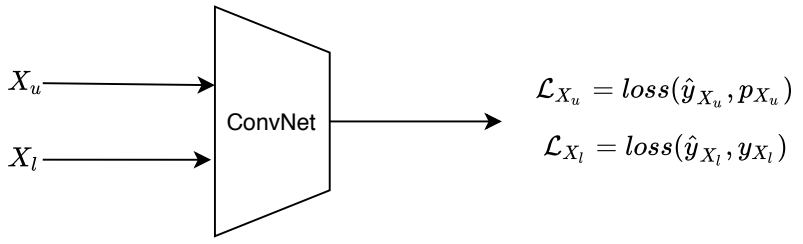


FIGURE 2.7: Pseudo-Labeling based Semi-Supervised Learning Algorithm

$$H(\widehat{y}_{X_u}, p_{X_u}) = -\sum_{X_u} p_{X_u} \log(\widehat{y}_{X_u}) \tag{2.25}$$

2.3.2.1 Pseudo Labels from Network

Algorithms under this category utilize labeled data information in an indirect way to infer pseudo labels. Pseudo-Label proposed in [55] trains a classifier with labeled data first and then assigns the current network prediction as the target label for unlabeled data to calculate cross-entropy loss, which is similar with the idea of entropy minimization [56], i.e., to encourage the model to give predictions with high confidence. [57] adds additional contrastive constraints on top of pseudo labels produced by the current network and similar idea is used in [58] to regularize the feature space. From empirical results in [59], the quality of pseudo labels inferred from network is suboptimal compared with transductive inference results performed in latent feature space. The reason behind such gap is believed to be the indirect information flow from labeled data to unlabeled data, which causes the trained model much more vulnerable to noise and information loss as noise is highly possible to be learned by the classifier (over-fitting problem) and meaningful feature information will be missed.

2.3.2.2 Pseudo Labels from Transductive Inference

Algorithms under this category utilize labeled data information in a direct way to infer pseudo labels by explicitly exploring the relationship between labeled and unlabeled data. Transductive learning aims to given predictions for specific data points, which is different from inductive learning whose purpose is to study a general rule such that new unseen data points can be predicted. Algorithms under

this category are non-parametric and the training data will be needed during inference stage. Graph-based learning is commonly used and often termed as label propagation [60]. The framework of label propagation is to construct a graph to characterize the similarity between data points first and then propagate the label information from labeled data to unlabeled data based on the graph. Traditional algorithms [60]–[62] construct the graph in data space and the graph will not change or update once it has been calculated. [59] proposed to combine traditional label propagation and deep learning in an iterative learning manner. Instead of measuring the similarity in noisy data space, [59] performs label propagation in feature space trained with labeled data and updates the similarity graph during training.

As shown in [57], [59], pseudo-labeling based algorithms can be combined with consistency-based algorithms to further boost model performance, indicating the two branches of research work are complementary to each other. The algorithm proposed in the second work of this thesis falls into the category of pseudo labeling via transductive inference, where similar philosophy in [59] is applied, i.e., to perform label propagation in feature space and keep similarity graph updated. The proposed algorithm differs from [59] in two major aspects: feature extraction scheme and similarity graph construction algorithm, which will be discussed in detail in Chapter 4.

2.4 Self-Supervised Learning

Self-supervised learning [63]–[73] is getting increasingly favorable for unsupervised feature representation due to its simplicity and high efficiency. Data annotation is not required in self-supervised learning. Instead, pretext task is defined to give model surrogate supervision and algorithms vary based on the pretext task defined. RotNet proposed in [20] learns features via predicting the rotational transformation that has been applied to the image. Jigsaw puzzle solver proposed in [19] targets to predict relative spatial position of each puzzle tile extracted from the original image. [17], [18] focuses on colorization, i.e., to predict the color with intensity information of the original image. For video data, cross-modal self-supervised learning is one dominant research direction. As multiple modalities, including RGB, optical flow, audio and visual information, are naturally accessible for video data, self-supervised learning aims to extract features by utilizing the correspondence between them. The correspondence between optical flow and RGB information is used in the research works of [71], [72], [74]. In the works of [70], [75]–[77], the corresponding relationship between audio signals and video frames are exploited. The surrogate supervision is freely available and high-level features are expected to be captured during such training process. Self-supervised learning is commonly used as a pre-training process and the trained model will be further utilized for downstream tasks, including semantic segmentation, object detection and classification.

Figure 2.8 gives an illustration for the common framework of self-supervised learning algorithm. Given the original training images without label information, pretext task will be defined and the corresponding surrogate labels will be used for objective function optimization. After the optimization of the neural network (deep CNN model), classifier, i.e., the linear layer, of the model will be discarded and the remaining convolutional blocks will be used as a feature extractor for subsequent target task training. In previous research works on self-supervised learning, the architecture of the network affects the effectiveness of the algorithm. Moreover, the commonly used features for down-stream tasks may not be the embeddings from the last convolution layer. Instead, RotNet achieves the highest accuracy with feature generated from the second convolutional block for NIN (network in network) [78] on CIFAR-10 dataset and features generated from conv4 for AlexNet [40] on ImageNet with non-linear classifiers. The architecture illustrations for NIN

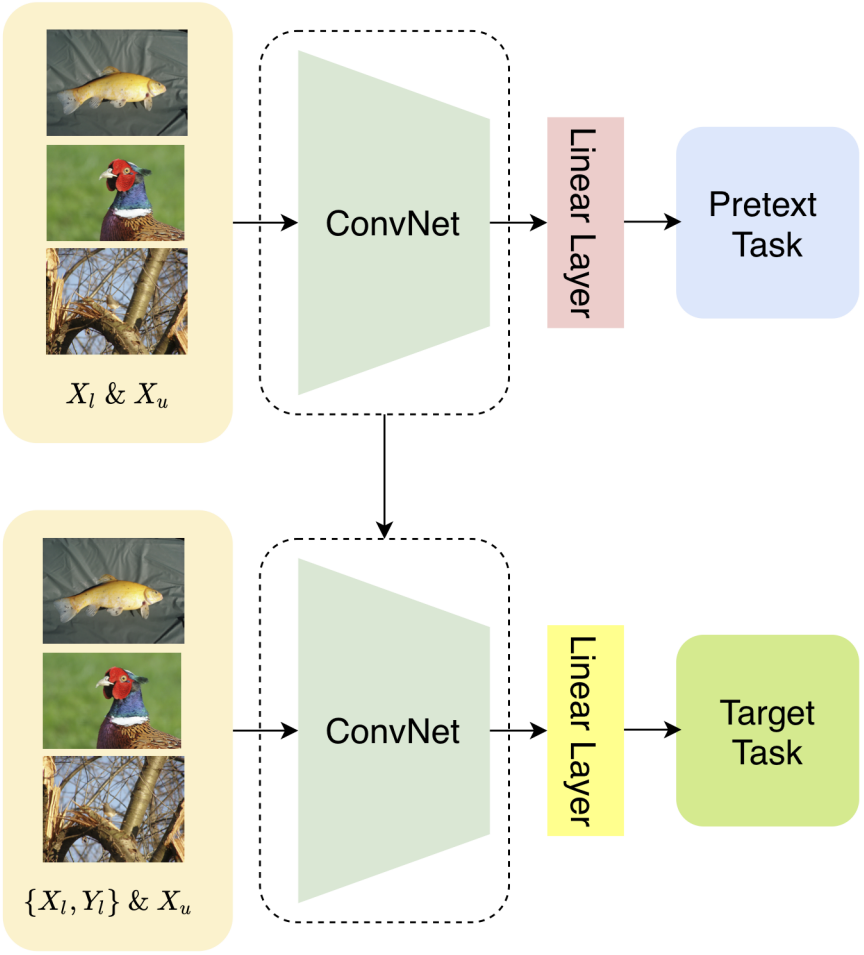
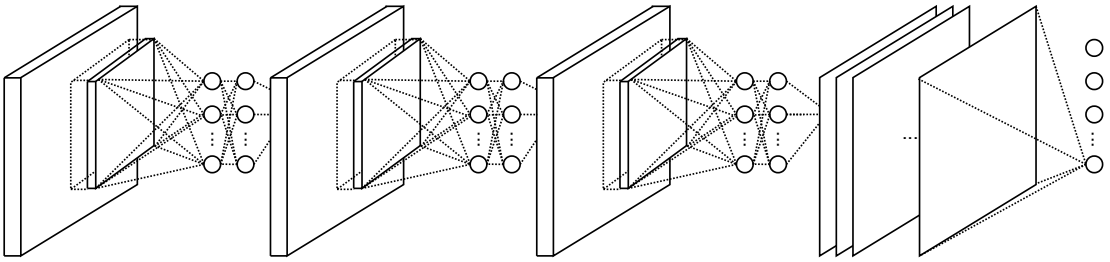


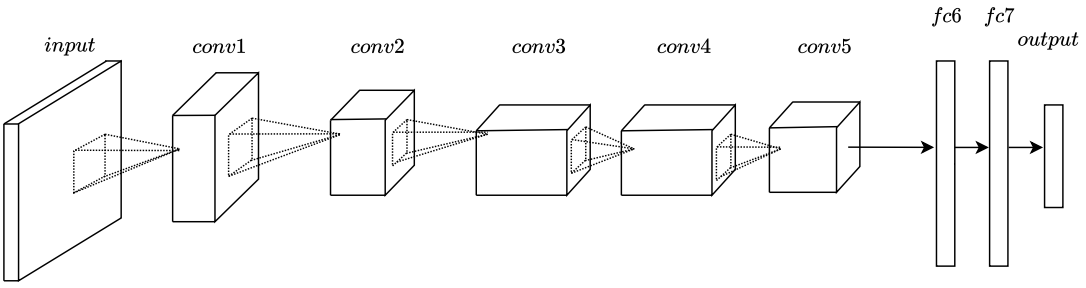
FIGURE 2.8: Self-Supervised Learning Framework

and AlexNet are given in Figure 2.9. For CNN models without residual connections, the performance of the features learned from self-supervised learning is going to degrade gradually if the convolutional layer is getting closer to the final linear layer. According to the works in [79], for CNN models with residual connections, e.g., ResNet, the degradation problem will go away and the optimal features for downstream tasks are from the last convolutional block.

Self-supervised learning is exploited for transductive learning in Chapter 4 and novel visual categories learning in Chapter 5. Therefore, the related works reported in literature for the two tasks are reviewed in the remaining part.



(a) NIN Architecture



(b) AlexNet Architecture

FIGURE 2.9: Network Architecture of NIN and AlexNet

2.4.1 Self-Supervised Learning in Transductive Learning

Deep Metric Transfer (DMT) proposed in [80] firstly proposes to utilize label propagation via metric learning for deep semi-supervised visual recognition. In [80], image colorization [18] is explored to be used as a self-supervised similarity metric learning method and label propagation is performed by making use of the learned similarity metric to generate pseudo labels for unlabeled data. In DMT, the learned metric after the self-supervised pre-training stage will be fixed for subsequent training. As ground-truth label information is not utilized in the pre-training stage, discriminative features are not captures by the learned metric and therefore the pseudo labels generated via label propagation is noisy and unreliable. Inspired by [55], [59], the proposed algorithm in this thesis incorporate ground-truth label information into the similarity metric learning to study discriminative features by adopting an iterative metric training manner.

2.4.2 Self-Supervised Learning in Novel Visual Categories Learning

AutoNovel proposed in [81] has explored the use of self-supervised learning in novel categories learning by initializing the model with trained RotNet [20]. However, self-supervision signals are utilized independently during pre-training and most blocks of the trained model are frozen in subsequent training process to avoid over-fitting problem in [81], which limits the learning capability of the model. Conversely, the proposed algorithm in this thesis is to train the model end-to-end by using self-supervision signals with other supervisions jointly, which provides strong regularization along the training process. As such, the whole network can be trained without concerns on over-fitting issues.

2.5 Novel Visual Categories Learning

A common assumption adopted by most semi-supervised learning algorithm is that there is always labeled data available from the same class of unlabeled data. However, this assumption cannot be always satisfied especially in real-world situations. More practical scenarios are that there are distinguishing categories for unlabeled data, which means unlabeled data are from separate classes without overlapping with existing classes of labeled data. This notable problem in most semi-supervised learning algorithms is not well addressed with very few research works on solutions reported in literature.

If labeled and unlabeled data belong to the same set of categories, label information can be efficiently extracted to supervise the model to learn discriminative features. On the contrary, label information from labeled data is not directly related to unlabeled data if they belong to disjoint categories, which suggests that label information is not sufficient as supervising signals. The next research question is what else can play the role of supervision and simultaneously apply to all data. Several research works [81]–[83] have been done recently for novel categories learning and their answer to the preceding question is the pairwise similarity information.

It requires weaker supervision for predicting whether two images are similar/belonging to the same class or not, compared to that of classifying images into multiple categories. Pairwise similarity pseudo labels are employed to supervise model training and the quality of pseudo labels determines the ultimate performance of the trained model. Research works in [82], [83] train a similarity prediction network (SPN) with labeled data first and then the predictions of trained SPN on unlabeled data will be used as pairwise pseudo labels. In [81], pseudo labels are generated with rank statistics of feature vectors that get updated during the training process. All of those previous algorithms are comprised of multiple training stages, making it difficult to implement in an end-to-end fashion. Moreover, their performances are constrained by strong dependence on the quality of pairwise similarity pseudo labels that can be noisy and biased.

As novel visual categories learning is closely related to clustering and zero-shot learning, brief reviews on them are given in the remaining part.

2.5.1 Clustering

Clustering as a classic unsupervised learning problem has been studied in many research works [84]–[91]. In recent years, deep learning based clustering studies [88], [89], [91]–[94] have been conducted due to favorable feature extraction capability of neural networks. Most algorithms in this category can simultaneously learn feature embeddings and cluster assignments. In [88], the model is first pre-trained with deep autoencoder [95] and then optimized with pseudo targets computed with current soft cluster assignments. [89] reduces the clustering problem into binary pairwise-classification problem and similar to [88], pairwise pseudo labels are calculated with current network predictions. However, the criteria used for clustering are ambiguous due to the lack of prior knowledge. Novel categories learning is free from such dilemma as some labeled data are given, which is able to provide categorical information.

2.5.2 Zero-Shot Learning

Zero-shot learning [96]–[102] also deals with classification problem of images from unseen categories during training. Unlike common classification problem, extra information is required in zero-shot learning other than training data and testing data, i.e., side information. Class attributes information is one popular option in most zero-shot learning algorithms [97], [103]–[105]. The class attributes include high-level semantic information, e.g., color, shape and texture, for each class in training data and testing data. Another form of side information exploited by previous researchers is textual information [106]–[108], which can be extracted from Wikipedia text via Word2Vec [109]. Novel categories learning problem discusses in this thesis is different from zero-shot learning as above-mentioned additional information is not required in novel categories learning.

Chapter 3

Deep and Wide Feature based Extreme Learning Machine for Image Classification

Chapter 3 investigates the effect of width and depth of CNN model as a feature extractor on ELM classifier and introduces a novel deep and wide feature based extreme learning machine (DW-ELM) algorithm for image classification to remedy over-fitting problem of ELM classifier and meanwhile utilize the preferable feature extraction capability of deep neural networks. Section 3.1 discusses the research background and motivations for this work. Section 3.2 presents detailed explanations of the proposed algorithm. Section 3.3 evaluates the proposed algorithm on benchmark datasets.

3.1 Background and Motivations

Extreme Learning Machine (ELM) [8] is an algorithm for single-layer feedforward neural networks (SLFNs) where neurons between input and hidden layer are randomly assigned. Compared with conventional deep learning techniques [2], ELM is much faster in training as it doesn't need any gradient-based iterative tuning. Once the input weights and hidden node biases are randomly generated, the optimal solution for the output mapping matrix can be directly calculated. Its universal approximation capability was proved theoretically in [25], [110]. ELM has demonstrated its effectiveness in both supervised learning [25], [29], [30] and unsupervised learning [33], [111], [112].

However, the shallow structure of ELM makes it infeasible to achieve good performance when solving complex visual tasks, especially on raw pixel space. Therefore, several works [28], [113]–[116] proposed deep ELM models by stacking ELM-AE (ELM based Auto Encoder) or its kernel version and ELM based supervised autoencoders for better representation learning. To further boost the performance, shortcut connection and residual learning are also combined with ELM based models in [117]–[119]. However, due to limited depth of these models, there is still a significant gap between ELM based deep models and commonly used deep neural networks (DNNs) [3]. As a consequence, extracting high-level feature representations first and then feeding those features to ELM for final classification has become one promising direction.Accordingly, several works that combined deep neural network and ELM classifier were applied successfully to various applications, such as age classification [4], [5], document image classification [6], 3D shape recognition [7], etc.

However, features extracted from over-deepened CNN model are vulnerable to over-fitting problem when ELM classifier is employed. Such problem is seldom addressed in literature due to several reasons. The first reason is that most works related to CNN+ELM hybrid model are limited to relatively shallow CNN architecture as residual connection is not utilized. The second reason is that well-designed algorithms for specific applications have neglected such over-fitting issues for general computer vision tasks. Therefore, this chapter aims to analyze and provide solutions for the over-fitting problem under a general-purpose visual learning framework.

To alleviate such over-fitting problem, the naive idea is to replace the deep CNN model with a shallow one. By doing so, the problem returns to the initial dilemma, i.e., model with shallow structure has poor performance on high-level semantic feature extraction. To handle this, another network dimension, i.e., width of model, is worth exploration. Width of model refers to the number of kernels for CNN model at each layer. More feature maps can be learned such that the extracted features are more diverse with wider CNN model. Research works [9], [13], [14] have explored the effect of width on generalization capability of the model and encouraging empirical results are demonstrated. In [11], wide residual network with 40 layers is eight times more efficient in the aspect of training time than ResNet-1001 [42] with comparable accuracy. While the wide design of residual networks has been shown to benefit image classification in terms of accuracy and efficiency, its application for feature extraction is not fully investigated. A novel deep and wide feature based Extreme Learning Machine (DW-ELM) is proposed in this chapter and the contributions can be summarized as follows:

1. A novel deep and wide feature based Extreme Learning Machine (DW-ELM) taking advantages from ELM classifier and "widened" convolutional neural networks is proposed. Wide architecture design of CNN model is exploited to remedy over-fitting of ELM classifier and meanwhile the preferable feature extraction capability of deep neural networks can be utilized.

2. Five benchmark datasets CIFAR-10, CIFAR-100, STL-10, Flower-102 and Fashion-MNIST are used to verify generalization performance of the proposed DW-ELM model. The experimental results show that DW-ELM is able to achieve better and more stable generalization performance than the backbone CNN model through out the whole training process.

3. This is the first work exploring the effect of width and depth of CNN model as a feature extractor on ELM classifier for image recognition. Insights that CNN model as a feature extractor for ELM image classification should be deep and also wide are given.

3.2 Proposed Algorithm

To solve complex visual tasks, deep neural networks (DNNs) are frequently used to extract features first before ELM classification for the raw image data. However, features extracted from over-deepened neural network are highly likely to cause over-fitting problem during ELM classification. Therefore, the proposed algorithm is aimed to alleviate the over-fitting problem of ELM classifier by deploying "widened" neural networks as feature extractor. For a better illustration, Figure 3.1 shows the generalization gap of an ELM classifier during the training process of different feature extractors. As shown in the Figure, the generalization gap, i.e. training accuracy minus testing accuracy, arising from features extracted from a deep and thin network (PreAct-ResNet110, red dotted line) can be greatly compressed by making use of a wider network (WRN, blue line) in the whole training process.

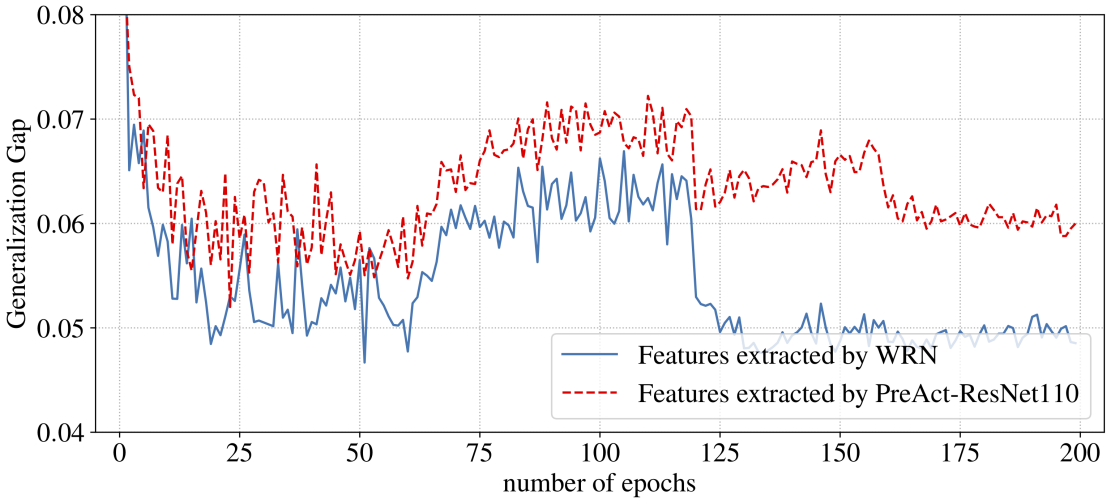


FIGURE 3.1: Generalization gap of ELM with features extracted by two different models on CIFAR-10 during learning process

3.2.1 Feature Extraction

Wide residual network (WRN) is used as feature extractor in the proposed hybrid model. WRN is built by stacking multiple units with similar structure, which are called residual units. Figure 3.2 shows the residual unit adopted in WRN. $\boldsymbol{x}_l$ is the input for l-th unit and $\boldsymbol{x}_{l+1}$ is the output for l-th unit and also the input for

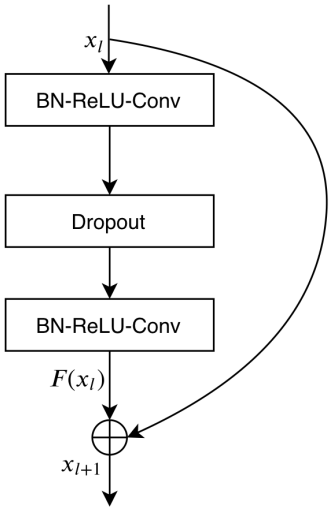


FIGURE 3.2: Residual Unit in WRN

$(l+1)$-th unit. $\boldsymbol{F}$ is the residual function. The residual unit can expressed as:

$$\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + \boldsymbol{F}(x_l) \tag{3.1}$$

As shown in Figure 3.2, pre-activation [42] is used, i.e., activation function ReLU is before the convolution, which is employed for smoother information flow such that better generalization performance and easier training can be achieved. For each residual unit, there are two 3×3 convolutions and dropout [120] is in between of them to prevent over-fitting of the network.

Table 3.1 shows the architecture of WRN. As shown in the table, there are totally four convolutional groups: conv1, conv2, conv3 and conv4. The input data will first go through conv1, which only consists of one 3×3 convolutional layer with 16 output channels and then feed in the following three groups. For group conv2, conv3 and conv4, the number of residual units in each group is the same, denoted by N. For WRN with depth 16, 22, 28, 40, the value of N is 2, 3, 4 and 6 respectively. The number of kernels used in each residual unit is given by the product of widening factor k and $\{16, 32, 64\}$ for group conv2, conv3 and conv4 respectively. Downsampling is done by group conv3 and conv4. After group conv4, a global average pooling layer followed by a fully-connected layer is used.

The architecture shown in table 3.1 is trained based on back-propagation with training dataset from scratch. After the training, extracted features before the last fully-connected layer is collected.

TABLE 3.1: WRN Architecture

Layers	**Output Channel**	**Output Size**	**Residual Unit**
conv1	16	32×32	3×3 conv
conv2	$16 \times k$	32×32	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times N$
conv3	$32 \times k$	16×16	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times N$
conv4	$64 \times k$	8×8	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times N$
GAP Layer	$64 \times k$	1×1	8×8 pooling
FC Layer	10 or 100		

3.2.2 Classification

Regularized basic ELM is used as classifier to discriminate extracted features from WRN for final classification. Only convolutional part of trained WRN is kept and the fully-connected layer is discarded. The extracted feature from each training data, i.e., feature vector ϕ_j for input data $\mathbf{x}_j, j \in \{1, 2, ..., N\}$, is fed into ELM classifier as input. ELM classifier will make use of these features to produce the output function $\mathbf{H}$ of hidden layer with random weights $\mathbf{a}_i \in \mathbf{R}^{d'}$ and bias $b_i \in R$ for each hidden node $i \in \{1, 2, ..., L\}$, where d' is the dimension of the feature vector ϕ_j.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\phi_1) \\ \vdots \\ \mathbf{h}(\phi_N) \end{bmatrix} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \phi_1) & \dots & G(\mathbf{a}_L, b_L, \phi_1) \\ \vdots & \vdots & \vdots \\ G(\mathbf{a}_1, b_1, \phi_N) & \cdots & G(\mathbf{a}_L, b_L, \phi_N) \end{bmatrix} \tag{3.2}$$

where

$$G(\mathbf{a}_i, b_i, \phi_j) = \frac{1}{1 + \exp(-(\mathbf{a}_i \cdot \phi_j + b_i))} \tag{3.3}$$

i.e., sigmoid function is used as the non-linear activation function in the hidden layer of ELM classifier.

Later according to ELM optimization algorithm (Algorithm 2), the hidden node mapping matrix $\boldsymbol{\beta} \in \mathbf{R}^{L \times m}$, where m is the number of categories to classify, is calculated as follows:

- For large datasets with $N > L$:

$$\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H} + \frac{\mathbf{I}}{\lambda})^{-1}\mathbf{H}^T\mathbf{T} \tag{3.4}$$

- For small datasets with $N \leq L$:

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T + \frac{\mathbf{I}}{\lambda}\right)^{-1}\mathbf{T} \tag{3.5}$$

with

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N\times m} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & \vdots \\ t_{N1} & \cdots & t_{Nm} \end{bmatrix} \tag{3.6}$$

where $\mathbf{t}_j \in \mathbf{R}^m, j \in \{1, 2, ..., N\}$ is the one-hot ground-truth label for input data $\mathbf{x}_j$. The extra term $\frac{\mathbf{I}}{\lambda}$ is for regularization purpose, where $\mathbf{I}$ is the identity matrix of size $\min(L, N)$ and $\frac{1}{\lambda}$ is the regularization coefficient.

The hidden node mapping matrix $\boldsymbol{\beta}$ together with hidden node parameters, i.e., weights $\mathbf{a}_i$ and biases b_i, give the input to output mapping function $\boldsymbol{f}$ accordingly:

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{h}(\phi)\boldsymbol{\beta} \tag{3.7}$$

3.2.3 Complexity Analysis

The computational complexity for classification is determined by Equation (3.4) and (3.5).

For large datasets with $N > L$, generating $\mathbf{H}^T\mathbf{H}$ has complexity of $\mathcal{O}(L^2N)$ as $\mathbf{H} \in \mathbf{R}^{N\times L}$. The matrix inversion in Equation (3.4) has complexity of $\mathcal{O}(L^3)$, given $\mathbf{H}^T\mathbf{H} \in \mathbf{R}^{L\times L}$. Matrix multiplication for calculating $\boldsymbol{\beta}$ in Equation (3.4) has complexity of $\mathcal{O}(L^2m + LNm)$, given $\mathbf{T} \in \mathbf{R}^{N\times m}$. Therefore, the summed complexity is $\mathcal{O}(L^2N + L^3 + L^2m + LNm)$. Due to the fact that $N > L$ and $L > m$, the overall computational complexity is $\mathcal{O}(L^2N)$.

For small datasets with $N \leq L$, generating $\mathbf{H}\mathbf{H}^T$ has complexity of $\mathcal{O}(N^2L)$ as $\mathbf{H} \in \mathbf{R}^{N\times L}$. The matrix inversion in Equation (3.5) has complexity of $\mathcal{O}(N^3)$,

given $\mathbf{H}\mathbf{H}^T \in \mathbf{R}^{N\times N}$. Matrix multiplication for calculating $\boldsymbol{\beta}$ in Equation (3.5) has complexity of $\mathcal{O}(N^2m + LNm)$, given $\mathbf{T} \in \mathbf{R}^{N\times m}$. Therefore, the summed complexity is $\mathcal{O}(N^2L + N^3 + N^2m + LNm)$. Due to the fact that $L \geq N$ and $N > m$, the overall computational complexity is $\mathcal{O}(N^2L)$.

3.2.4 DW-ELM

3.2.4.1 Training of Feature Extractor and Classifier of DW-ELM

The proposed DW-ELM model first makes use of WRN to learn how to extract features of the training data. In order to back-propagate the errors and update the network via gradients, WRN with fully-connected layer is trained from scratch with the training dataset, as shown in step 1 of Figure 3.3. In this training process, the original architecture of WRN will be kept and training data are used to train this network iteratively by making use of gradient-based optimization method. After training of WRN is finished, the extracted feature vectors are used to train a regularized basic ELM classifier, as shown in step 2 of Figure 3.3.

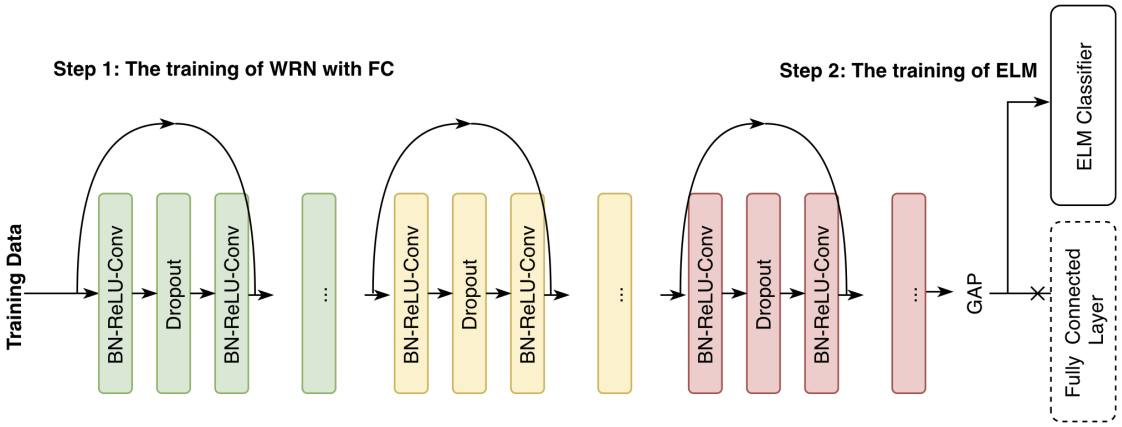


FIGURE 3.3: Training of proposed hybrid DW-ELM model

3.2.4.2 Testing of DW-ELM

After the training of ELM classifier, the input to output mapping function, and more precisely the feature space to label space mapping, can be computed. Moreover, the convolutional part of the trained WRN is capable of approximating the input space to feature space mapping. Therefore, when the trained WRN as a

feature extractor and trained ELM classifier are combined together, as shown in Figure 3.4, the overall framework is able to fit the mapping function from raw data space to label space based on training data. The testing dataset is used to evaluate the proposed framework. In evaluation stage, trained WRN is used to extract features of the testing data first and then the trained ELM classifier is applied for the final classification with extracted features from WRN.

For the proposed DW-ELM model, the most complex and time-consuming part is the first stage, which is the training of WRN, as the structure is much more complicated than the ELM classifier and in this stage gradient-based optimization algorithm is used to update the parameters. For the second stage, the training of ELM classifier, it's much easier to implement and the time taken is negligible compared with the time used in the first stage.

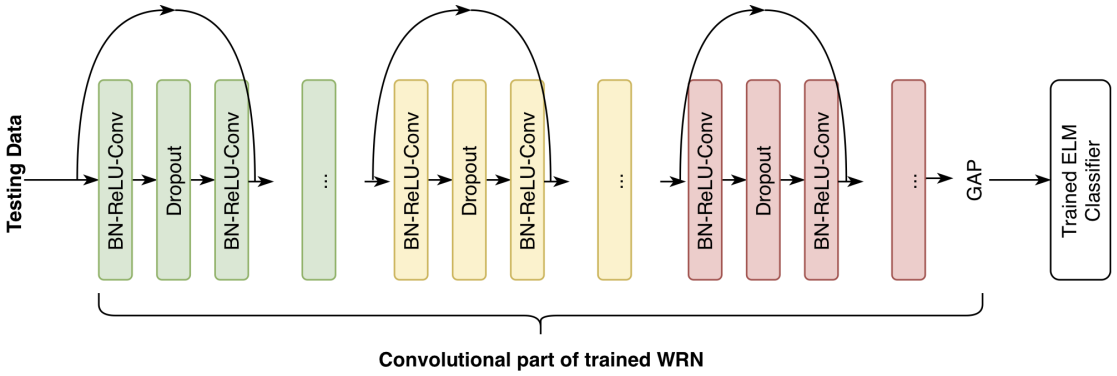


FIGURE 3.4: Testing of proposed hybrid DW-ELM model

3.3 Performance and Comparisons

3.3.1 Datasets

Total five different image datasets are used to evaluate the proposed model: CIFAR-10, CIFAR-100, STL-10, Fashion-MNIST and Flower-102. A brief summary for all datasets used is given in Table 3.2.

3.3.1.1 CIFAR-10 and CIFAR-100

CIFAR-10 and CIFAR-100 [121] are commonly used classification datasets in the field of machine learning. CIFAR-10 dataset has a total number of 60,000 colorful images of size 32×32, 50,000 of which are training data and the remaining 10,000 are testing data. There are total 10 classes in CIFAR-10 dataset, including airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each class in CIFAR-10 has 6000 images, of which 5000 images are in training dataset and 1000 images are in testing dataset. Similar with CIFAR-10, CIFAR-100 also has a total number of 60,000 colorful images of size 32×32, 50,000 of which are training data and the remaining 10,000 are testing data. Difference between CIFAR-10 dataset and CIFAR-100 dataset is that CIFAR-100 dataset has 100 classes. Each class in CIFAR-100 has 600 images, of which 500 images are in training dataset and 100 images are in testing dataset.

3.3.1.2 STL-10 Dataset

STL-10 [122] dataset is similar with CIFAR datasets but with smaller number of labeled data and higher resolution. STL-10 dataset has a total number of 13,000 colorful images of size 96×96, 5000 of which are training data and the remaining 8000 are testing data. There are total 10 classes in STL-10 dataset, including airplane, bird, car, cat, deer, dog, horse, monkey, ship, and truck. Each class in STL-10 has 1300 images, of which 500 images are in training dataset and 800 images are in testing dataset. Unlabeled data in this dataset is discarded as focus of this work is on supervised learning.

3.3.1.3 Fashion-MNIST

Fashion-MNIST [123] is similar with MNIST dataset [124] as both of them are in grayscale and with low resolution. Fashion-MNIST dataset has a total number of 70,000 grayscale images of size 28×28, 60,000 of which are training data and the remaining 10,000 are testing data. There are total 10 classes in Fashion-MNIST, covering 10 fashion objects.

3.3.1.4 102 Category Flower

102 Category Flower [125] is a classification dataset for colorful images of flowers. There are total 102 flower categories and each category consists of 40 to 258 images with different scales. The sizes for training data and testing data are 6552 and 819 respectively.

Dataset	#. Classes	#. Training Data	#. Testing Data	Dimension	Color
CIFAR-10	10	50,000	10,000	32×32	RGB
CIFAR-100	100	50,000	10,000	32×32	RGB
STL-10	10	5000	8000	96×96	RGB
Flower-102	102	6552	819	224×224	RGB
FMNIST	10	60,000	10,000	28×28	Gray

TABLE 3.2: Datasets Used For Evaluating the Proposed Algorithm

3.3.2 Implementation

Simple data augmentations including random cropping and horizontal flipping are used on all datasets. For the training of WRN, optimizer used is SGD (Stochastic Gradient Descent) with cross entropy loss, the weight decay value is set to be 5×10^{-4} and momentum value is 0.9. Learning rate annealing used in [11] is adopted in the experiments, where the initial learning rate is set to be 0.1 and will be divided by 5 at epoch 60, 120 and 160. The total number of training epochs is 200. The illustration for the learning rate annealing schedule adopted is shown in Figure 3.6 and implementation details are given in Table 3.3.

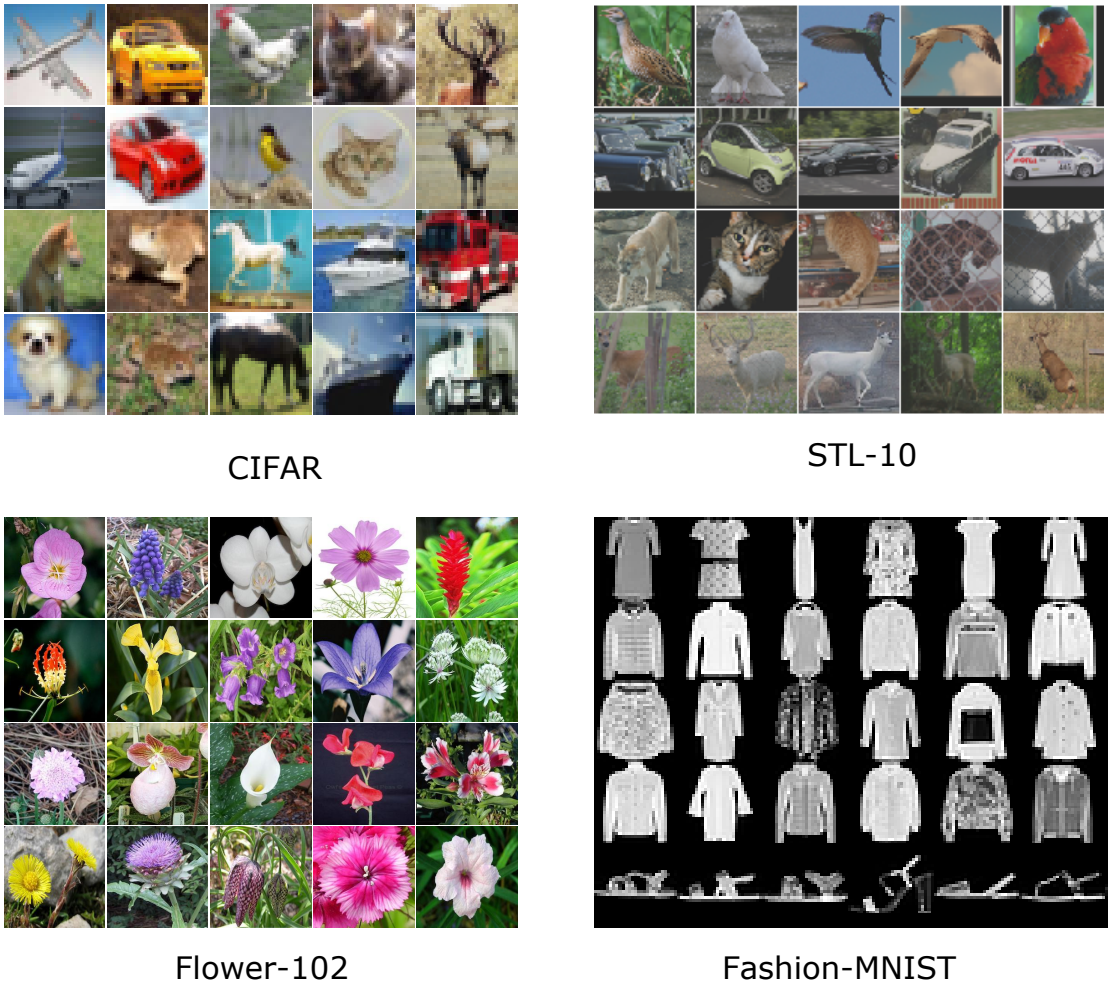


FIGURE 3.5: Sample Images from Benchmark Datasets

TABLE 3.3: Implementation details of the experiments

Optimizer	SGD
Initial Learning Rate	0.1
SGD Weight Decay	5×10^{-4}
SGD Momentum	0.9
LR Annealing Scheme	stepwise
Total Training Epochs	200

For CIFAR datasets, WRN architecture specified in Table 3.1 is used. For Fashion-MNIST dataset, all others remain the same except that the global average pooling will change to size 7×7. For STL10, downsampling is performed in conv2, conv3 and conv4 and the global average pooling will change to size 12×12. For 102 Category Flower dataset, all input images are resized to 224×224 first and 7×7 kernel is used in conv1. Downsampling is performed in conv1, conv2, conv3 and

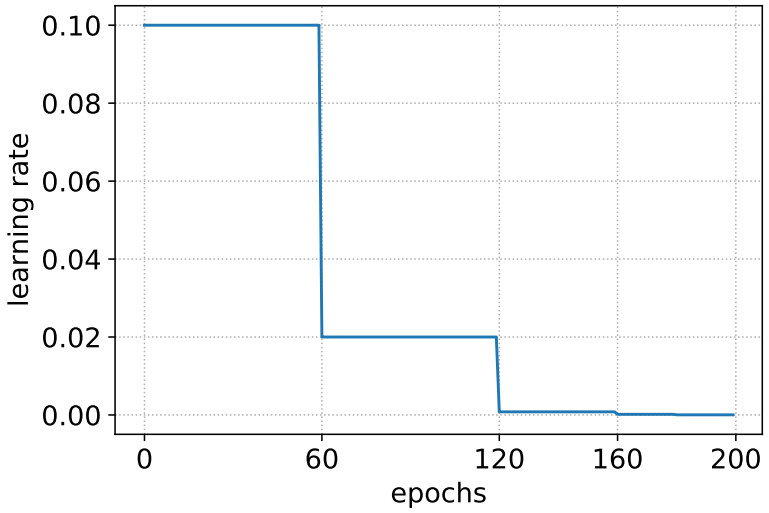


FIGURE 3.6: Learning rate annealing schedule

conv4 and the global average pooling will change to size 14×14. The dropout rate is set to be 30% according to [11].

3.3.3 Experiment Results

3.3.3.1 Performance Improvement

Testing curves for the proposed DW-ELM model with 8000-hidden-node ELM classifier and original WRN with depth 22 and widening factor 8 are presented in Figure 3.7 to Figure 3.11 for CIFAR-100, CIFAR-10, Flower-102, FMNIST and STL-10 respectively. It can be observed that the proposed DW-ELM model can exhibit distinct and consistent performance improvement in the whole training process on all datasets. In the early training phase, testing accuracy gain brought by DW-ELM is very significant. For CIFAR-100, Flower-102 and STL-10 datasets, the accuracy gain is over 10% in the first 60 epochs and over 4% for CIFAR-10 and FMNIST datasets. After the first learning rate annealing point, the accuracy gain is still obvious and such accuracy gain has maintained all the way till the end of whole training process. The final testing accuracy of the proposed DW-ELM model and original WRN is shown in Table 3.4. For CIFAR-10 and Fashion-MNIST datasets, DW-ELM has increased testing accuracy by 0.58% and 0.44% respectively. For CIFAR-100, STL-10 and Flower-102 datasets, testing accuracy has been boosted by 2.13%, 1.85% and 8.91% respectively.

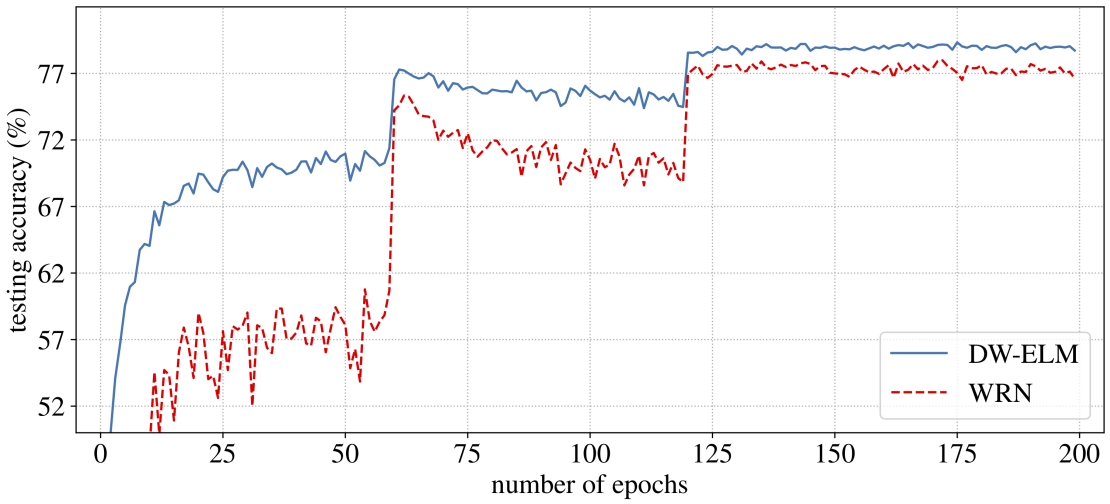


FIGURE 3.7: Testing curves of the proposed model DW-ELM and WRN on CIFAR-100

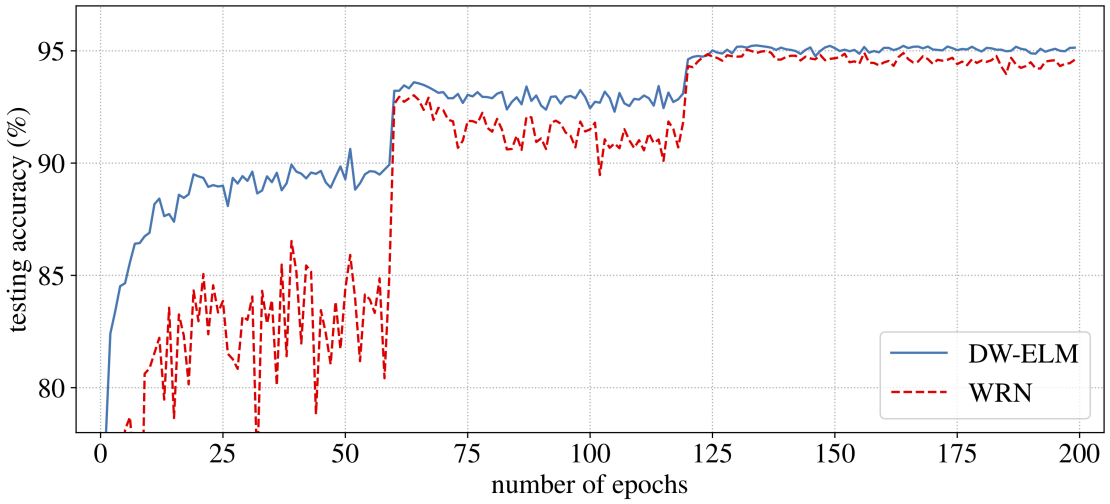


FIGURE 3.8: Testing curves of the proposed model DW-ELM and WRN on CIFAR-10

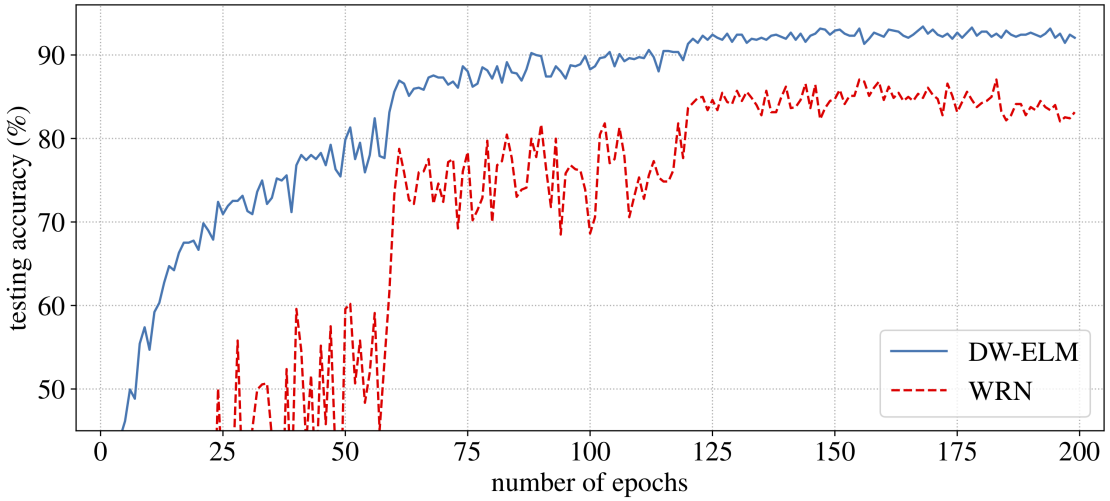


FIGURE 3.9: Testing curves of the proposed model DW-ELM and WRN on Flower-102

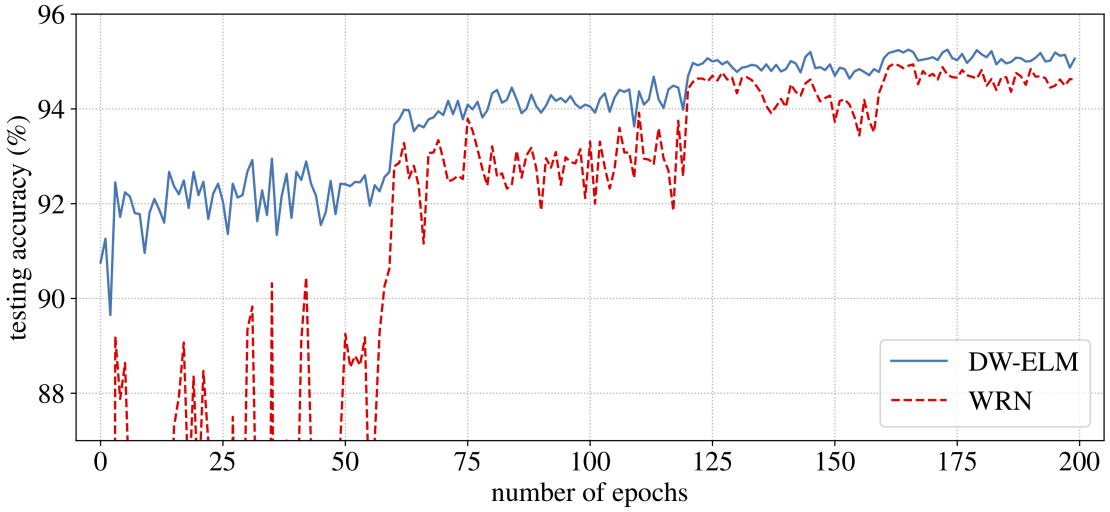


FIGURE 3.10: Testing curves of the proposed model DW-ELM and WRN on Fashion-MNIST

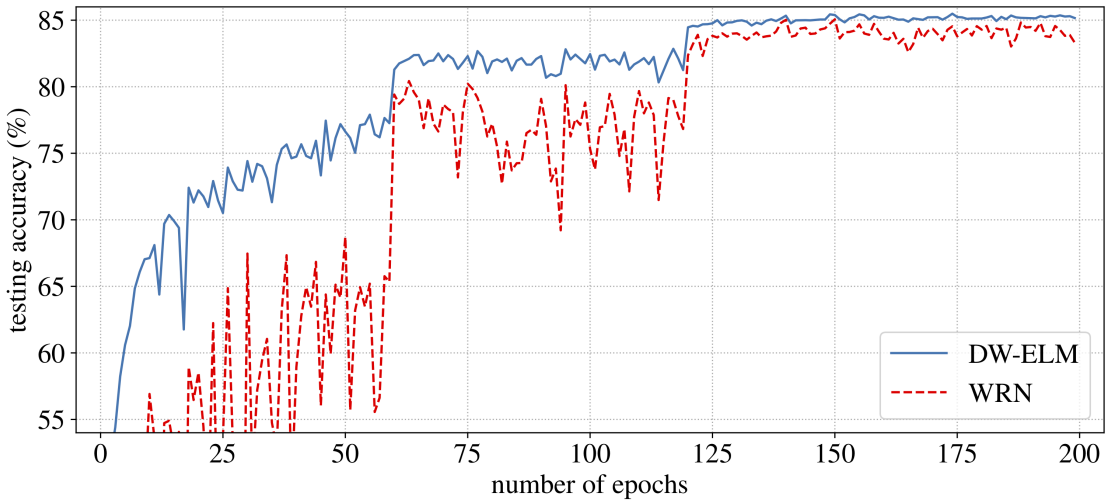


FIGURE 3.11: Testing curves of the proposed model DW-ELM and WRN on STL-10

TABLE 3.4: Testing Accuracy of WRN and Proposed Model (%)

Dataset	WRN	Proposed
CIFAR-10	94.60	**95.18**
CIFAR-100	76.59	**78.72**
STL-10	83.31	**85.16**
Flower-102	83.15	**92.06**
Fashion-MNIST	94.62	**95.06**

3.3.3.2 Stability

From Figure 3.7 to Figure 3.11, it can be observed that for all datasets, the learning curve of DW-ELM model has less noisy movements and fluctuations are greatly compressed compared with the backbone model, resulting in much more smoother curves. Such characteristic makes DW-ELM much more reliable than the original WRN model. For CIFAR datasets after epoch 60, accuracy of both models jumps to a higher level due to the change of learning rate but then a sudden drop is observed in the the backbone model between epoch 60 to epoch 120. DW-ELM greatly relieved this problem by maintaining accuracy in a comparably high level. Similar effect was also found by the use of dropout in [11], therefore the proposed DW-ELM model may have gained certain level of regularization power by the used of ELM classifier.

3.3.3.3 Parameter Analysis

Depth and width determine the capability of WRN as a feature extractor and the number of hidden nodes of ELM is the key parameter for its classification power. Therefore to better verify the effectiveness of proposed DW-ELM model, WRN model with various depth and width combinations together with ELM classifier with hidden nodes number ranging from 250 to 8000 are tested. Ten different WRN models and its corresponding DW-ELM models with 8000-hidden-node ELM classifier are compared on two CIFAR datasets. The results are shown in Figure 3.12 and Figure 3.13. The conventional notation used in [11] for WRN architecture is used in this section, where WRN-d-w denotes WRN with depth d and widening factor w. It can be seen that for all ten models with depth ranging from 16 to 40 and widening factor ranging from 4 to 12, the superiority of proposed DW-ELM model is consistent as significant accuracy gain is always present. Meanwhile, by comparing the results from models with different d and w, it can be observed that there is no trend indicating that models with larger w or smaller d will give a higher accuracy gain. Such observations suggest that width and depth of CNN model should be balanced carefully if optimal performance is desired.

DW-ELM models with WRN-16-04, WRN-22-08 and WRN-28-10 together with ELM classifier having different number of hidden nodes are also compared and the result is given in Figure 3.14 to Figure 3.18. For CIFAR-10 and Fashion-MNIST datasets, performance of DW-ELM model is pretty insensitive to the number of hidden nodes on all three WRN models, which suggests that a comparable performance with much less hidden nodes can be achieved. For the other three datasets, it can be observed that with increasing complexity of WRN, proposed DW-ELM model tends to be less sensitive to the number of hidden nodes. Therefore, proposed DW-ELM model is able to exhibit its effectiveness across a wide range of parameter values, indicating that little human intervention regarding parameter tuning is required.

3.3.3.4 Ablation Study

To further verify that the superiority of proposed DW-ELM model is coming from combination of WRN and ELM classifier, additional experiments with other residual networks and classifiers are also conducted. Pre-activation resnet with 110

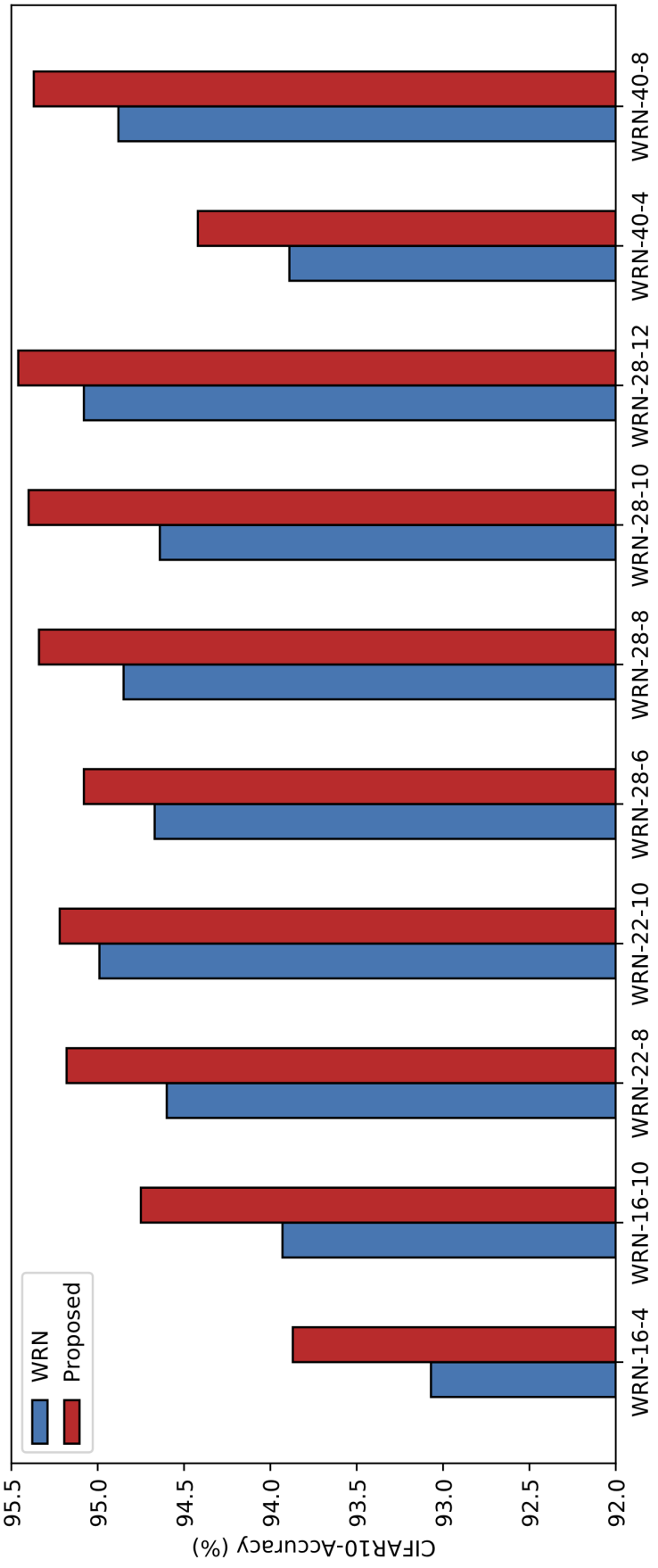


FIGURE 3.12: Accuracy of CIFAR-10 with various WRN models

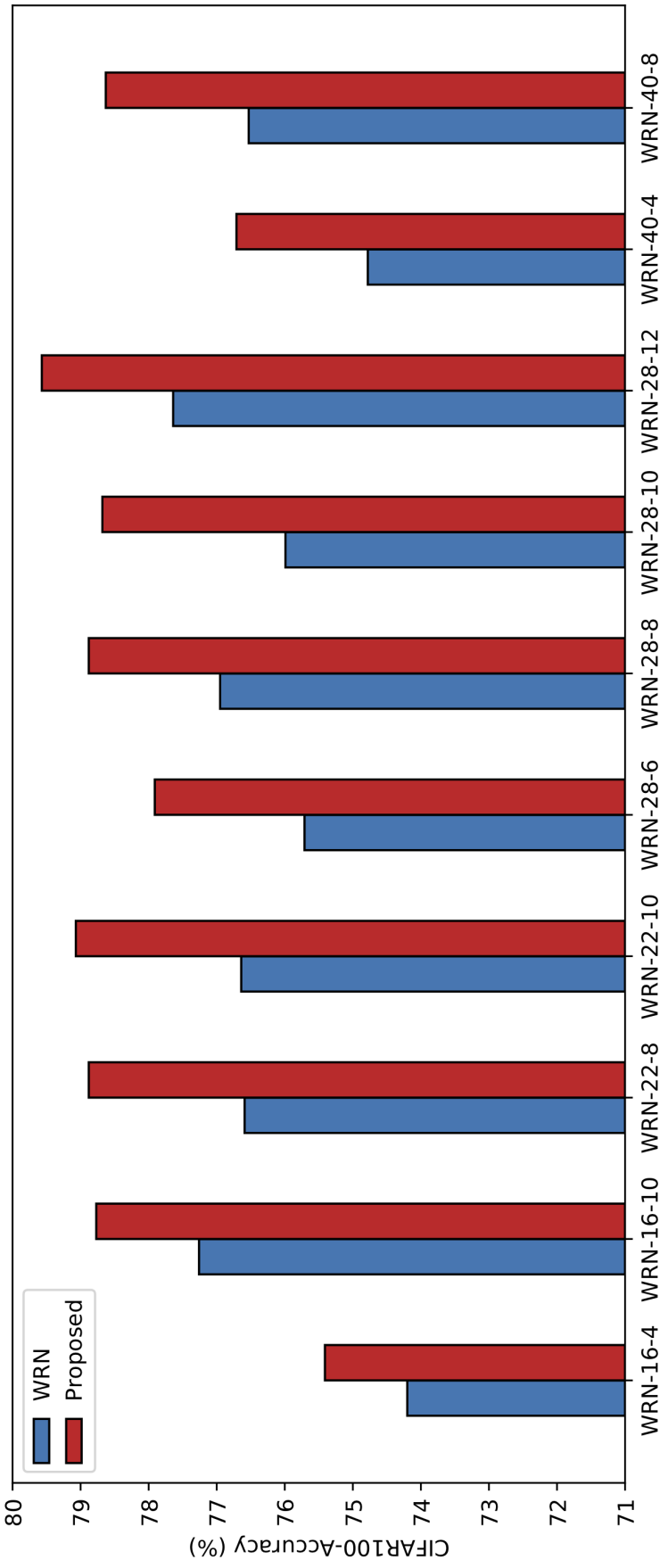


FIGURE 3.13: Accuracy of CIFAR-100 with various WRN models

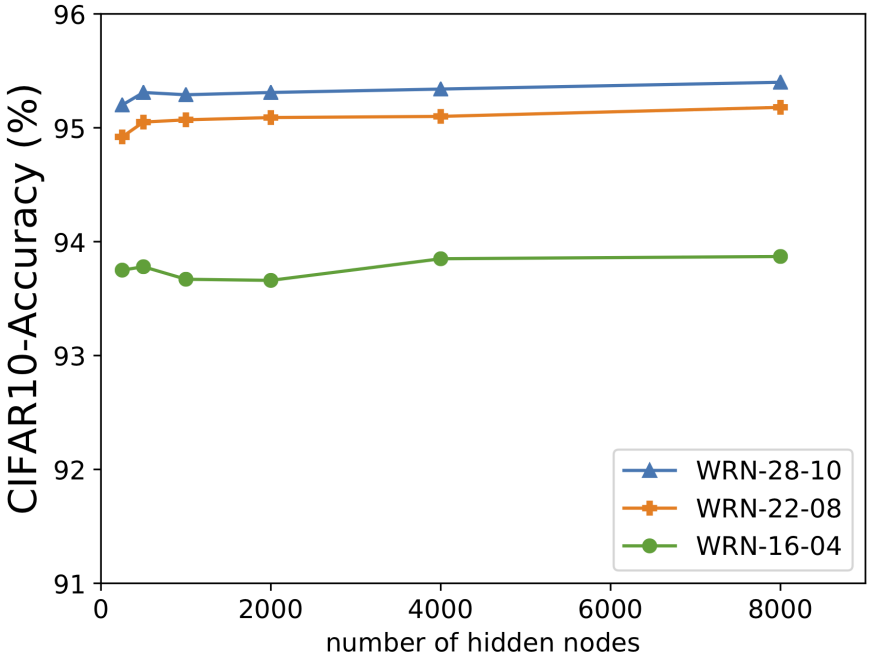


FIGURE 3.14: Parameter Analysis: Accuracy of proposed DW-ELM model on CIFAR-10 with varying number of hidden nodes

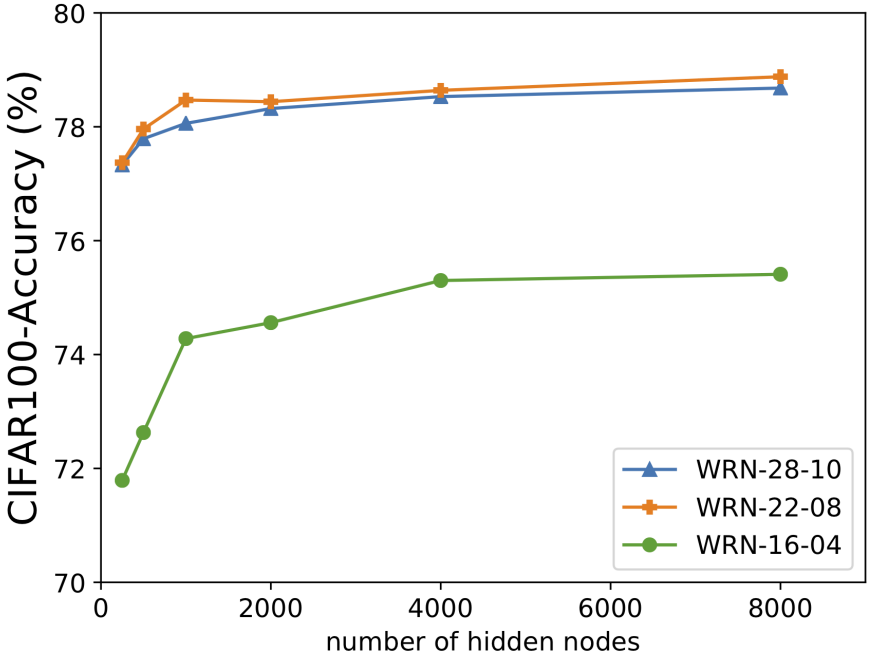


FIGURE 3.15: Parameter Analysis: Accuracy of proposed DW-ELM model on CIFAR-100 with varying number of hidden nodes

layers and 164 layers have very similar network architecture with WRN adopted in the proposed DW-ELM model. The only difference between them is that WRN is a widened and shallower version of them. These two very deep networks are

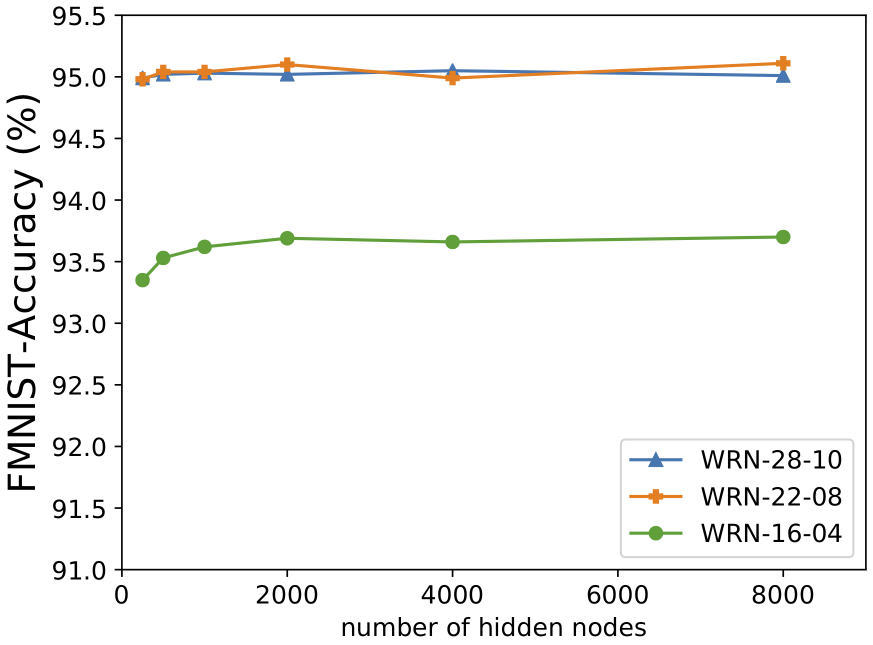


FIGURE 3.16: Parameter Analysis: Accuracy of proposed DW-ELM model on Fashion-MNIST with varying number of hidden nodes

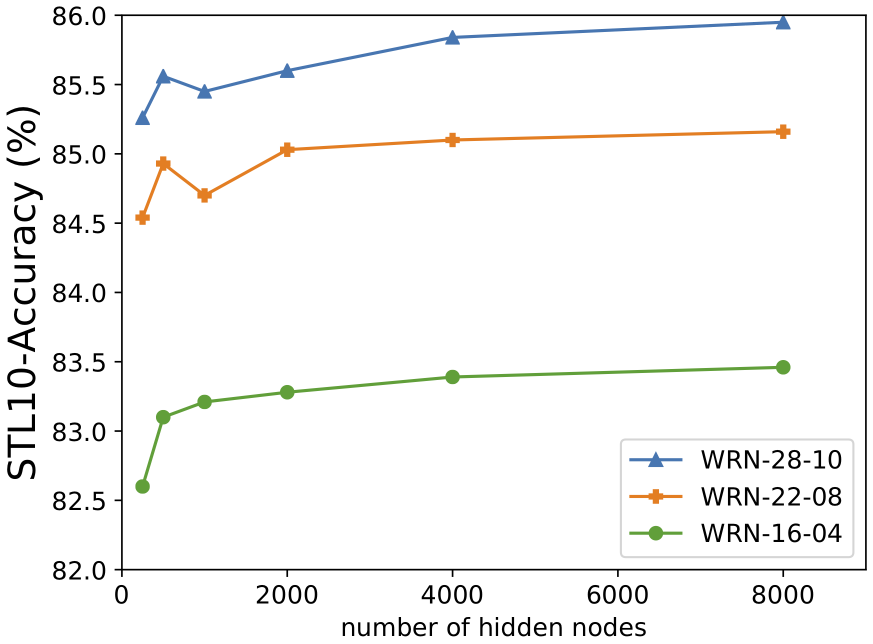


FIGURE 3.17: Parameter Analysis: Accuracy of proposed DW-ELM model on STL-10 with varying number of hidden nodes

used to do feature extraction and similar with the proposed model, the extracted features are fed into ELM as input data for classification. The testing accuracy on five datasets are given in Table 3.5. Among five datasets, only one shows positive

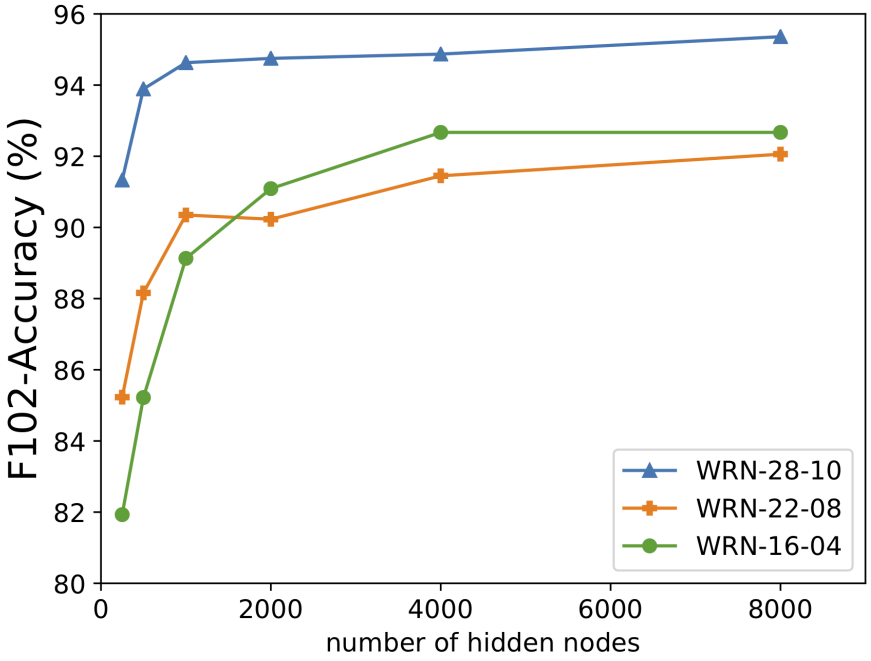


FIGURE 3.18: Parameter Analysis: Accuracy of proposed DW-ELM model on Flower-102 with varying number of hidden nodes

but very insignificant accuracy gain, while all other four datasets indicate negative accuracy gain for pre-activation resnet with 110 layers. For pre-activation resnet with 164 layers, two datasets show a positive accuracy gain but still very limited and all the others give negative gain. Therefore, the accuracy improvement in proposed DW-ELM model where WRN is used as features extractor will vanish if WRN is replaced by a thinner and deeper residual network (like Preact-110 and Preact-164).

TABLE 3.5: Ablation Study: Testing Accuracy of pre-activation resnet with ELM classifier (%)

Dataset	Preact-110	Preact-110 +ELM	**Accuracy Gain**	Preact-164	Preact-164 +ELM	**Accuracy Gain**
CIFAR-10	94.07	94	**-0.07**	94.79	94.7	**-0.09**
CIFAR-100	73.6	73.5	**-0.1**	76.74	76.63	**-0.11**
STL-10	81.13	80.76	**-0.37**	83.18	83.25	**0.07**
Flower-102	93.9	93.53	**-0.37**	93.29	93.28	**-0.01**
FMNIST	94.77	94.85	**0.08**	94.65	94.72	**0.07**

WRN-22-08 as feature extractor with RBF-kernel SVM classifier are also tested on five datasets and the results are listed in the third column of Table 3.6. ELM classifier outperforms kernel SVM on all five datasets with the same features extracted from WRN-22-08.

TABLE 3.6: Ablation Study: Testing Accuracy of pre-activation resnet and WRN with Kernel SVM classifier and the proposed method (%)

Dataset	Preact-110 +SVM	Preact-164 +SVM	WRN-22-08 +SVM	Proposed
CIFAR-10	94	94.84	94.99	**95.18**
CIFAR-100	73.46	76.8	78.52	**78.72**
STL-10	81.1	83.36	85	**85.16**
Flower-102	91.7	**92.43**	90.84	92.06
FMNIST	94.88	94.83	95.04	**95.06**

Go a step further, Preact-110 and Preact-164 as feature extractor with RBF-kernel SVM classifier are also tested on all datasets and the results are listed in the first two columns of Table 3.6. Among five datasets, the proposed DW-ELM achieves the best results on four of them, except for Flower-102. For dataset Flower-102, both of Preact-ELM and Preact-SVM models can surpass the proposed DW-ELM, which indicates that the main reason for relatively low accuracy of the proposed DW-ELM lies in the low accuracy of WRN on dataset Flower-102. From Table 3.4, the testing accuracy of WRN on Flower-102 is only 83.15%, which is much lower than 93.9% of Preact-110 and 93.29% of Preact-164 according to Table 3.5. Flower-102 is one unbalanced dataset, where the number of images from the same class ranges from 40 to 258. Therefore, the training is highly likely biased and according to the experiment results, WRN seems to be more susceptible to it. Although 83.15% accuracy on Flower-102 from vanilla WRN is much lower than the results from Preact-110 and Preact-164, the accuracy of the proposed DW-ELM can be greatly improved to 92.06%, such boosted performance can also justify the effectiveness of the proposed model. One more thing can also be observed in Table 3.6 is that WRN is a better choice than pre-activation resnets as feature extractor for SVM classifier as features extracted from WRN achieve the highest accuracy among all SVM-based methods (the first three columns) on four out of five datasets. So, WRN will benefit more than pre-activation resnets for both ELM and SVM classifier, and ELM will be a better choice than SVM classifier when the same WRN is used as feature extractor. Therefore, even thought SVM performs slightly better

than ELM classifier when Preact-164 is used as feature extractor, it doesn't have much realistic implications as Preact-164 itself is a suboptimal feature extractor for both classifiers.

To summarize, two key factors that contribute to the superior performance of proposed DW-ELM to others are firstly it can fully exploit classifying power of ELM classifier and secondly it's able to alleviate over-fitting by making use of deep and wide networks, i.e., WRNs.

3.4 Summary

This chapter presents the first work of this thesis, which has addressed the over-fitting problem of ELM classifier when combined with deep CNN model as feature extractor for supervised visual recognition. The motivations for this work are presented in Section 3.1. A novel deep and wide feature based extreme learning machine (DW-ELM) algorithm is proposed in Section 3.2. In Section 3.3, DW-ELM is validated on five commonly used public datasets, CIFAR-100, CIFAR-10, STL-10, Flower-102 and Fashion-MNIST. The proposed model has shown its significant performance enhancement and stabilization mechanism in the whole training phase. Moreover, experiments regarding parameter analysis show that proposed DW-ELM is able to maintain its effectiveness on various width and depth selection of WRN and hidden nodes number of ELM, which makes DW-ELM require very little hyper-parameter tuning. Additionally, ablation study provides evidence that features extracted by wider networks is likely to generalize better with ELM classification than over-deepened networks. Therefore it suggests that proposed DW-ELM model is capable of utilizing advantages of ELM classifier and WRN such that deficiency of them can be alleviated greatly by each other. Extra experiments comparing the effect of ELM classifier and kernel SVM classifier on features extracted by WRN further demonstrate superior classification power of proposed DW-ELM model on all datasets.

Chapter 4

Label Propagation via Local Geometry Preserving for Deep Semi-Supervised Image Recognition

Chapter 4 introduces a novel transductive pseudo-labeling based algorithm for deep semi-supervised image recognition. Section 4.1 reviews related works in the field of deep semi-supervised learning for visual tasks and discusses the limitations of previous research works. Section 4.2 gives detailed explanations of the proposed algorithm. Section 4.3 shows extensive experimental results and analysis, together with comparisons with other previous state-of-the-art semi-supervised learning algorithms.

4.1 Background

Deep convolutional neural network has achieved great success in the field of visual tasks, conventionally termed as computer vision. However, such superior performance is primarily based on the use of a large amount of annotated data, which sets a pretty high bar for realistic applications as annotation can be time-consuming, expensive and require expert knowledge under certain scenarios [15].

Semi-supervised learning [16] aims to alleviate such demands for labeled data by utilizing additional unlabeled data, which is much easier to get without the need for annotation, with the hope to achieve comparable or even better performance than fully supervised learning. One branch of research work for semi-supervised learning is mainly to add certain kinds of regularization to the network based on the smoothness assumption [16], [45], which can be summarized as consistency-based algorithms. Consistency-based algorithms enforce the network to minimize the difference between predictions of original data and data with noise or perturbation injected. Such noise or perturbation can be provided by directly injecting to the data [46], [47] or by the network [50], [51]. Another branch of research work can be summarized as pseudo-labeling based algorithms, where pseudo labels are produced for unlabeled data as if they were the ground-truth labels and then the problem can be tackled under fully-supervised learning framework. The reliability of pseudo labels will be a crucial factor to determine the efficacy of the model. According to how pseudo labels are inferred, there are two main approaches: inferring from network and inferring from transductive learning.

Inferring pseudo labels from network normally will train a network with labeled data first and then unlabeled data can be classified by such network so that inferred labels will be treated as true labels of unlabeled data for subsequent training together with labeled data [55]. Pseudo labels will be updated as the classification capability of network is evolving during the training process. While for inferring pseudo labels from transductive learning, a similarity graph describing the relationship between all data points will be constructed first, and then label information will be directly propagated from labeled data to unlabeled data accordingly, which is also names as label propagation [60]. In [59], the similarity graph is built in low-dimensional feature space, which is learned with labeled data, and similar with

inferring from network, similarity graph will be updated to give better pseudo labels during training process.

From empirical results in [59], the quality of pseudo labels inferred from network is suboptimal compared with label propagation results performed in latent feature space. The reason behind such gap is believed to be that label propagation utilizes labeled data information in a direct way such that potential noise and information loss are minimized. While for inference via network, the information flow of labeled data is indirect, i.e., labeled data are used to train a classifier first and then use the trained classifier to predict labels for unlabeled data, which is much more vulnerable to noise and information loss as noise is highly possible to be learned by the classifier and meaningful feature information will be missed. Therefore, this work adopts similar philosophy, i.e., to infer pseudo labels directly by graph-based method, and on top of it to further explore information flow with less noise and preserve local geometry information in feature space, which is commonly ignored.

It has been observed that fully-supervised learning adopted at early stage in [59] and other pseudo-labeling based research works [55], [57] only uses label information from labeled data, which can cause learned model highly likely to be biased and overfit to noise. Therefore, this work proposes to apply self-supervised learning in the proposed algorithm for feature extraction at phase 1 (shown in Figure 4.1). Self-supervised learning is one encouraging direction in deep feature learning for image recognition tasks in recent years [17], [19], [20]. Pretext task replaces ground-truth label to provide supervision signal, including colorization, solving jigsaw puzzles, predicting geometric transformations, etc. Self-supervised learning applies to both labeled and unlabeled data as ground-truth label information is not required, which will be desirable for more noise-robust feature learning. By making use of self-supervised learning, the learned feature space is expected to provide a cleaner information flow during subsequent label propagation.

In label propagation, similarity graph construction is an important starting point as it guides the direction of propagation. The similarity graph is expected to reveal the relationship between data points and it can be calculated in different ways, e.g., Euclidean distance and inner product [59], [80]. However, local geometry structure information is lost if the similarity is solely depending on two data points. Therefore, the idea of reconstruction in [21] is applied in the proposed algorithm to preserve local geometry information in feature space, i.e., each feature vector

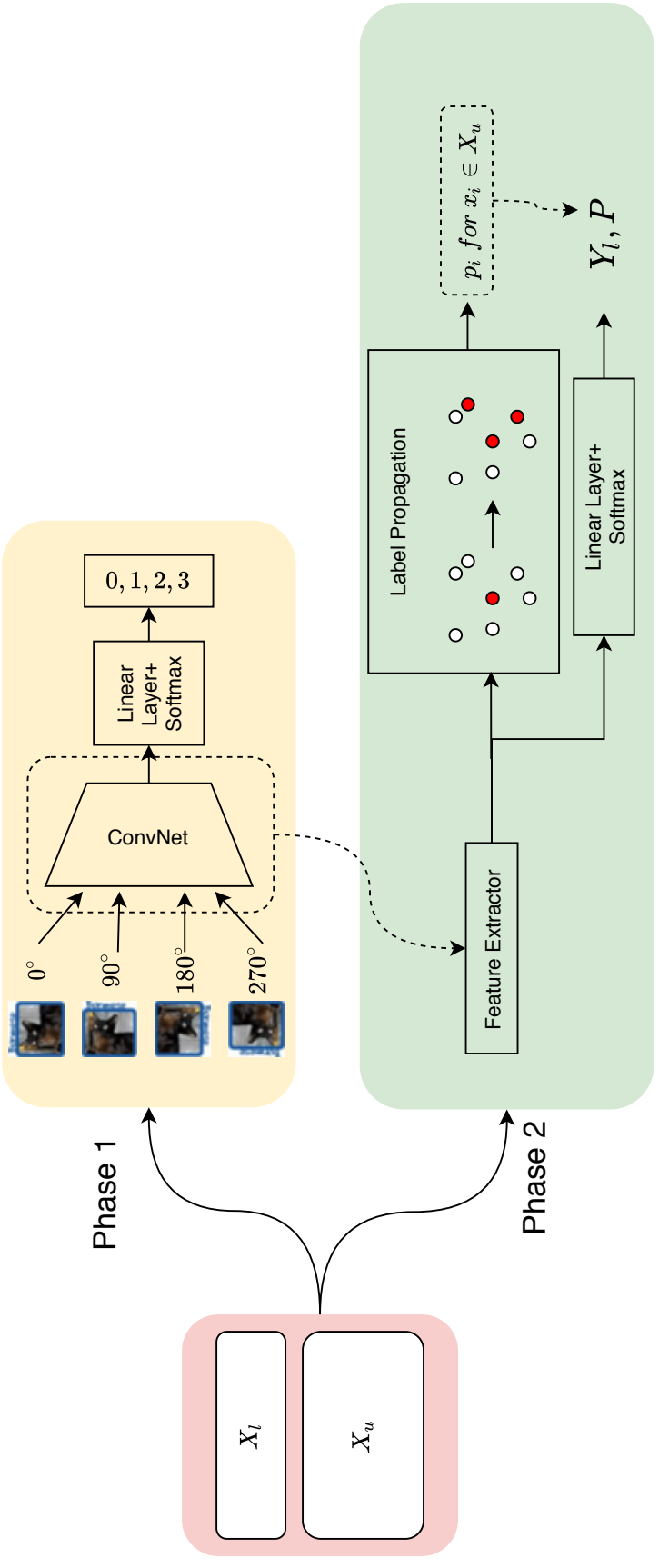


FIGURE 4.1: Overall framework of the proposed algorithm. Here both labeled data X_l and unlabeled data X_u are utilized in phase 1 via self-supervised learning. Each image is rotated by $(0°, 90°, 180°, 270°)$ first and assigned pseudo label $(0, 1, 2, 3)$ respectively to train the network. After the training in phase 1 is done, convolutional layers will be used to extract features in phase 2. In phase 2, label propagation will be performed in feature space after feature extraction and the inferred pseudo labels p_i for unlabeled data will be used together with Y_l of labeled data to calculate the final loss, which will back-propagate through the whole network (including the feature extractor and linear layer). During the training process, updated feature extractor will give new label propagation results and therefore the inferred pseudo labels p_i for unlabeled data will also be updated.

can be reconstructed by a weighted sum of its neighbors and such weights will be generalized as the similarity between feature vector and its neighbors. By doing so, the local geometry information embedded in feature space can be preserved in information flow during label propagation.

In this Chapter, a novel transductive pseudo labeling based algorithm for deep semi-supervised image recognition is presented, and the overall framework of the proposed algorithm is illustrated in Figure 4.1. Inspiration from the superiority of pseudo labels from label propagation compared with those from network, i.e., information flow from labeled data to unlabeled data should be kept noiseless and with minimum loss as much as possible, is taken into consideration. Accordingly, self-supervised learning is incorporated in feature extraction in phase 1 and local geometry is preserved in feature space during graph construction before propagating label information in phase 2. In the later ablation study, "whole is greater than the sum" is observed from experimental results as feature space learned from self-supervision and local geometry preserving scheme in label propagation are found to be mutually reinforcing. The contributions can be summarized as follows:

1. Self-supervised learning is incorporated into the feature extraction phase such that limitations of scarce ground-truth label information can be avoided and cleaner information flow in subsequent label propagation is achieved. This is the first work that utilizes self-supervision in iterative graph-learning based algorithms for image recognition.

2. Local geometry information of each data in feature space is preserved during label propagation via reconstructing feature vector by its neighbors to build similarity graph. Moreover, ablation study shows that such geometry preserving scheme is synergistic with features learned with self-supervision in the proposed algorithm.

3. Extensive experiments conducted on three image datasets (CIFAR-10, CIFAR-100, *mini*ImageNet) have verified the effectiveness of the proposed algorithm and the results show that the proposed algorithm consistently outperforms most of the state-of-the-art semi-supervised learning methods under the same network architecture.

4.2 Proposed Algorithm

4.2.1 Problem Formulation

Given a set of training data $X = \{x_1, x_2, ..., x_n\}$, where $x_i \in R^d$, the corresponding labels Y are only available for a limited number of them, denoted by Y_l, and $Y_l = \{y_1, y_2, ..., y_l\}$, where $y_i \in R^c$ and c is the number of all possible classes. Therefore, training data $X = X_l \cup X_u$ and there are l labeled samples in X_l and u unlabeled samples in X_u such that $l+u = n$. Semi-supervised learning aims to find a mapping function $f_\theta(\cdot)$ from input data space to target label space by making use of X and Y_l, such that it can predict the label of new data.

4.2.2 Motivations

4.2.2.1 Self-Supervision and Full-Supervision

One crucial part of label propagation in deep learning is the learning of feature mapping. Instead of directly propagating labels in data space [60]–[62], extracting features from the raw data first and then performing label propagation in feature space are more capable of handling complex visual tasks. Such feature mapping can be learned with deep neural networks by utilizing X and Y_l in different ways. Fully-supervised learning with X_l and Y_l tends to have inferior generalization capability as noise is easily picked up by the model due to over-fitting problem, especially when the number of labeled data is scarce. While for self-supervised learning, both X_l and X_u will be used and no ground-truth label information is needed. As a result, much less biased and more noise-robust feature learning can be achieved because all training data are utilized. Figure 4.2 shows the testing error rate curve for the same vanilla label propagation method with different feature learning schemes. Same label propagation method is used to infer pseudo labels for unlabeled data and the only difference between them is the way how feature space is learned. The orange-triangle one propagates labels in feature space learned from X_l and Y_l, while the blue-dot one propagates labels in feature space learned from X_l, X_u and their pseudo labels given by surrogate signals, e.g., here the degree of rotational transformation that has been applied to raw images is used as supervision. It

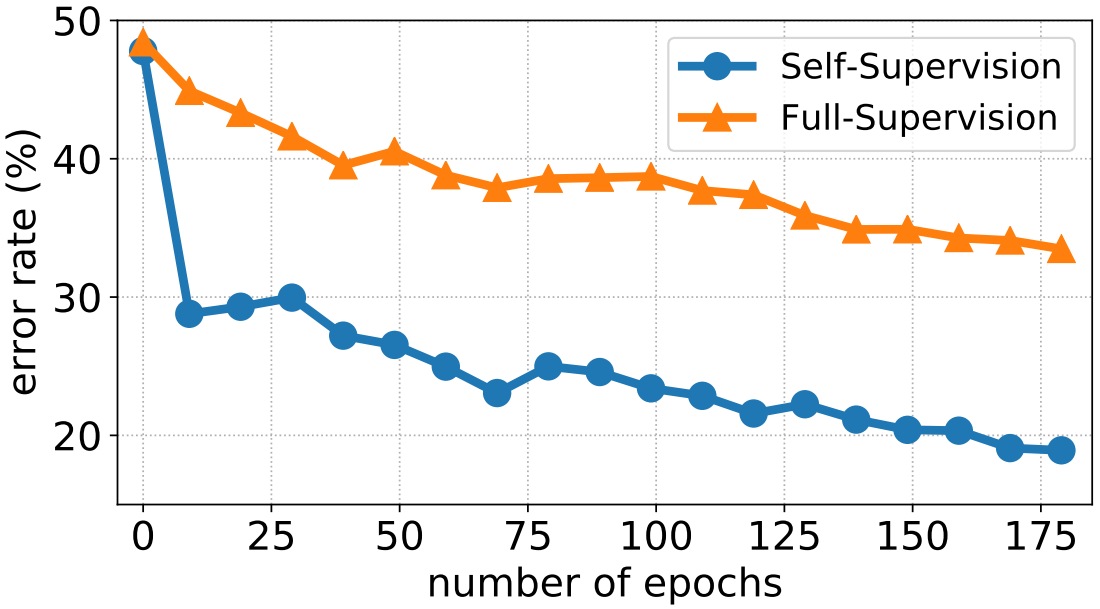


FIGURE 4.2: Testing error rate for same vanilla label propagation method with two different feature learning schemes for CIFAR-10 dataset with 500 labels.

can be observed that self-supervision based feature learning scheme can largely outperform the full-supervision one for label propagation in deep learning.

4.2.2.2 Local Geometry Preserving

A graph capable of capturing the relationship between data points is required for label propagation in data space. As the proposed algorithm is to perform label propagation in feature space, a graph is constructed to characterize the relationship between feature vectors. Such a graph can be represented by a similarity matrix of size $n \times n$ with each element representing the similarity between two feature vectors. Most of the works [59], [61], [80] define the similarity as pairwise Euclidean distance (in Gaussian kernel) or pairwise inner product. However, neither Euclidean distance nor inner product can characterize or preserve the local geometry as only two individual feature vectors are considered when computing their similarity. By making use of the reconstruction concept, which will be discussed in detail in Section 4.2.3, the local geometry in feature space can be preserved. Figure 4.3 shows the label propagation results in the same feature space but with different similarity matrix construction methods for CIFAR-10 dataset with 500 labels. It can be observed that preserving local geometry can boost the label propagation performance by a large margin. In Figure 4.3, the feature space is learned

with self-supervision and the inference accuracy of pseudo labels produced by label propagation for X_u can be improved from 24.21% to 32.83%.

4.2.3 Label Propagation via Local Geometry Preserving

In this section, detailed explanations for the proposed algorithm are presented. There are two phases in the proposed algorithm: self-supervised feature learning phase and semi-supervised label propagation phase.

4.2.3.1 Phase 1 – Self-Supervised Feature Learning

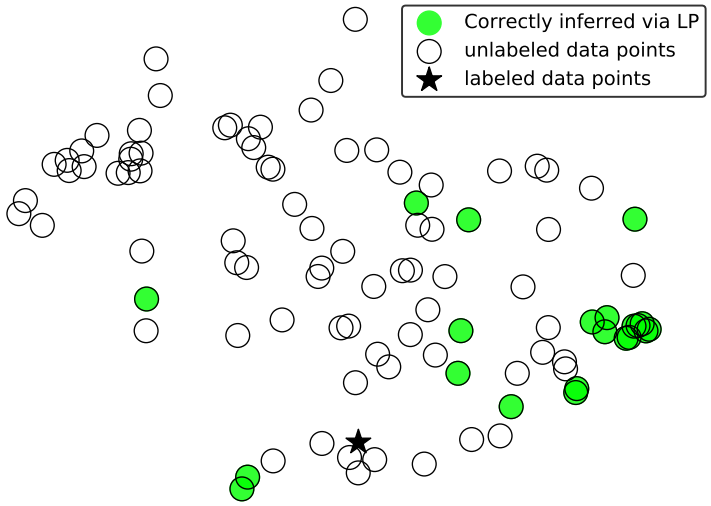
Different from [59], where supervised learning with labeled training data is used to learn a feature representation, the superior generalization capability of self-supervision is exploited. In the proposed algorithm, RotNet [20] is used for self-supervised feature learning. Given the original training image X, four different rotation transformations $(0°, 90°, 180°, 270°)$ are performed on each x_i. Then for the transformed images, a set of pseudo labels $\widehat{P} = \{\widehat{p}_1, \widehat{p}_2, ..., \widehat{p}_n\}$ is created according to the transformation that has been applied to them, i.e., $\widehat{p}_i = [0, 1, 2, 3]$ for transformed images after rotation of $0°, 90°, 180°, 270°$, respectively. Then a neural network will be trained to map the transformed training data $\widehat{X}$ to the pseudo labels $\widehat{P}$. By doing so, all training data will be used during the feature learning phase, which can produce a less noisy feature representation as over-fitting problem is greatly alleviated, as opposed to fully supervised learning with limited labeled training data in [59]. Training scheme for phase 1 is summarized in Algorithm 3.

4.2.3.2 Phase 2 – Semi-Supervised Label Propagation

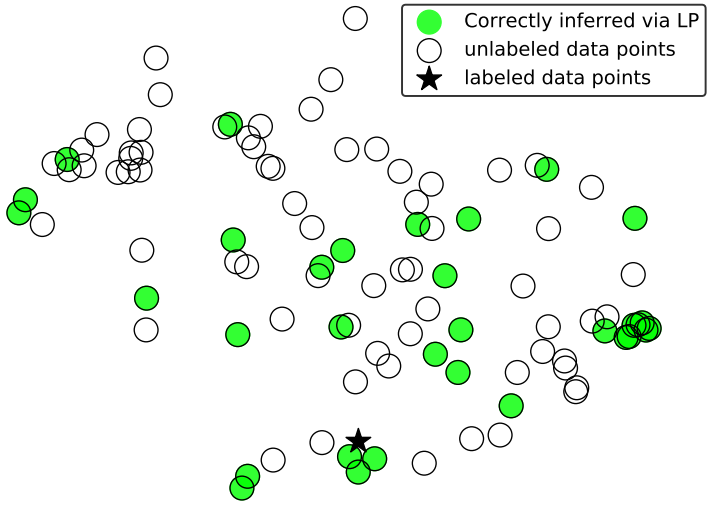
After the first phase, a feature mapping function $\phi(\cdot)$ given by the trained neural network can project all training data to a feature space, as shown in Equation (4.1).

$$z_i = \phi(x_i) \tag{4.1}$$

In the feature space, a sparse similarity matrix W of size $n \times n$ will be constructed. The proposed algorithm makes use of the idea from locally linear embedding (LLE)



(a) Label propagation result with graph constructed by pairwise inner product, inference accuracy is 24.21%



(b) Label propagation result with graph constructed by preserving local geometry, inference accuracy is 32.83%

FIGURE 4.3: Label propagation results with two different graph learning methods given the same data features. The dataset used is CIFAR-10 with 500 labels and t-SNE is used for dimension reduction. The figures here only show 100 sampled data points from the same class for visualization purpose.

[21] that a data point can be reconstructed by a weighted sum of its neighbors to learn a similarity matrix, which can preserve the intrinsic geometry information.

In [21], a data point x_i can be reconstructed by a weighted linear combination of its neighbors:

$$\begin{aligned} & x_i \approx \textstyle\sum_{j:x_j \in \mathcal{N}(x_i)} w_{ij} x_j \\ s.t. \quad & w_{ij} \geq 0, \textstyle\sum_{j:x_j \in \mathcal{N}(x_i)} w_{ij} = 1 \end{aligned} \tag{4.2}$$

where $\mathcal{N}(x_i)$ is the collection of x_i's k nearest neighbors. Weight w_{ij} indicates the contribution of data point x_j to reconstruct data point x_i if x_j is one of x_i's k nearest neighbors, and a large weight value means a large portion of x_i can be solely reconstructed by x_j, implying two data sample points are very similar to each other. In the proposed algorithm, the same reconstruction rule is applied to measure the similarity of feature vectors in the feature space, as shown in Equation (4.3).

$$z_i \approx \sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} z_j \tag{4.3}$$

The reconstruction error function to be minimized will be

$$\begin{aligned} & \varepsilon = \textstyle\sum_i \| z_i - \sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} z_j \|^2 \\ s.t. \quad & w_{ij} \geq 0, \textstyle\sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} = 1 \end{aligned} \tag{4.4}$$

which can be further rewritten as

$$\begin{aligned} \varepsilon &= \sum_i \| \sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} z_i - \sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} z_j \|^2 \\ &= \sum_i \| \sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} (z_i - z_j) \|^2 \\ &= \sum_i W_i^T G^i W_i \\ s.t. \quad & w_{ij} \geq 0, \sum_{j:z_j \in \mathcal{N}(z_i)} w_{ij} = 1 \end{aligned} \tag{4.5}$$

where $W_i = [w_{ij_1}, w_{ij_2}, ..., w_{ij_k}]^T$, similarity vector of z_i with its k nearest neighbors $\mathcal{N}(z_i)$, and element in a-th row and b-th column of matrix G^i is

$$g^i_{a,b} = (z_i - z_{j_a})^T (z_i - z_{j_b}) \tag{4.6}$$

where $a, b = 1, ..., k$. For each feature vector z_i, constructing the similarity vector W_i requires to solve quadratic programming problem in Equation (4.5), which can be efficiently solved via existing solvers, such as `CVXOPT` [126].

After solving the optimization problem, the sparse weight matrix W ($w_{ij} = 0$ if $z_j \notin \mathcal{N}(z_i)$) can be calculated, but it can be observed that w_{ij} may not be the same as w_{ji}, which makes weight matrix W non-symmetric. One more step is added to fix this by

$$\mathbf{W} = W + W^T \tag{4.7}$$

By doing so, the weight matrix $\mathbf{W}$ now becomes symmetric and all diagonal elements are zero. Then the weight matrix is normalized to $\mathcal{W} = D^{-0.5}\mathbf{W}D^{-0.5}$ for the convergence of label propagation [61], where D is a diagonal matrix with $d_i = \sum_j \mathbf{w}_{ij}$.

After weight matrix $\mathcal{W}$ is calculated, similar approach in [59], [61] is followed to propagate label information from X_l to X_u in feature space. The label propagation rule is to propagate label information in an iterative manner, as shown in Equation (4.8).

$$F^t = \alpha\mathcal{W}F^{t-1} + (1-\alpha)\mathbf{Y} \tag{4.8}$$

where F^t denotes the label information at t-th iteration, $\alpha \in (0, 1)$ is the coefficient controlling the amount of label information to propagate and $\mathbf{Y}$ represents initial label information, i.e., $F^0 = \mathbf{Y}$. $\mathbf{Y}$ is a matrix of size $n \times c$ with each element defined as

$$\mathbf{y}_{ij} = \begin{cases} 1 & if\ x_i \in X_l\ and\ y_i = j \\ 0 & otherwise \end{cases} \tag{4.9}$$

Equation (4.8) can be rewritten as

$$F^t = (\alpha\mathcal{W})^{t-1}\mathbf{Y} + (1-\alpha)\sum_{i=0}^{t-1}(\alpha\mathcal{W})^i\mathbf{Y} \tag{4.10}$$

As $0 < \alpha < 1$ and $\mathcal{W} = D^{-0.5}\mathbf{W}D^{-0.5}$, the following two limits can hold:

$$\begin{aligned} &\lim_{t\to\infty}(\alpha\mathcal{W})^{t-1} = 0 \\ &\lim_{t\to\infty}\sum_{i=0}^{t-1}(\alpha\mathcal{W})^i = (I - \alpha\mathcal{W})^{-1} \end{aligned} \tag{4.11}$$

Therefore, Equation (4.8) will eventually converge to

$$\mathbf{F} = F^{\infty} = (1-\alpha)(I - \alpha\mathcal{W})^{-1}\mathbf{Y} \tag{4.12}$$

Matrix $\mathbf{F}$ is of size $n \times c$ and the pseudo label for $x_i \in X_u$ is inferred as

$$p_i = \arg\max_j \mathbf{f}_{ij} \tag{4.13}$$

where $\mathbf{f}_{ij}$ is the element of matrix $\mathbf{F}$.

Let $P = \{p_{l+1}, p_{l+2}, ..., p_n\}$, so now labeled data X_l and unlabeled data X_u have true label Y_l and pseudo label P respectively. A unified single weight value to compress the pseudo label loss is not reasonable as pseudo labels for each unlabeled data don't have equal uncertainty. The uncertainty weighting scheme in [59] is employed to give different weight for each pseudo label:

$$\mu_i = 1 - \frac{-\sum_j \widehat{\mathbf{f}}_{ij} \log(\widehat{\mathbf{f}}_{ij})}{\log(c)} \tag{4.14}$$

where $\widehat{\mathbf{f}}_{ij}$ is row-normalized counterpart of $\mathbf{f}_{ij}$ such that each row of matrix $\mathbf{F}$ can be viewed as a probability distribution. One more weighting coefficient δ_i to tackle imbalanced dataset issue originating from pseudo labels is introduced:

$$\delta_i = \frac{n/c}{|X^{c_i}|} \tag{4.15}$$

where n is the size of dataset X, c is the number of all possible classes and $X^{c_i} = \{x_i \in X_l : y_i = c_i\} \cup \{x_i \in X_u : p_i = c_i\}$.
The final loss function to be minimized will be:

$$loss = \frac{1}{l}\sum_{i=1}^{l} L(\widehat{y}_i, y_i) + \frac{1}{u}\sum_{i=l+1}^{n} \mu_i \delta_i L(\widehat{y}_i, p_i) \tag{4.16}$$

where $\widehat{y}_i = f_c(z_i) = f_\theta(x_i)$ and $f_c(\cdot)$ is the classifier. The model is trained in an iterative way such that after pseudo labels P inferred and the loss in Equation (4.16) back propagated to the whole network $f_\theta(\cdot)$, the feature mapping function $\phi(\cdot)$ in Equation (4.1) will also be updated and then new pseudo labels will be generated. Training scheme for phase 2 is summarized in Algorithm 4.

Algorithm 3 Self-Supervised Feature Learning

Input: Training image data $X = \{x_1, x_2, ..., x_n\}$
for $i \leftarrow 1$ to n **do**
  $\widehat{x}_i \leftarrow$ rotate x_i by $0°, 90°, 180°, 270°$
  $\widehat{p}_i \leftarrow [0, 1, 2, 3]$
$\widehat{X} \leftarrow \{\widehat{x}_1, \widehat{x}_2, ..., \widehat{x}_n\}$
$\widehat{P} \leftarrow \{\widehat{p}_1, \widehat{p}_2, ..., \widehat{p}_n\}$
Train the network $f_\theta(\cdot)$ with $\widehat{X}$ and $\widehat{P}$ to minimize:
$loss \leftarrow \sum_{i=1}^{n} L(f_\theta(\widehat{x}_i), \widehat{p}_i)$
Output: Feature extractor parameters θ for $\phi(\cdot)$

Algorithm 4 Semi-Supervised Label Propagation

Input: Labeled training image data and the corresponding labels $\{X_l, Y_l\}$, unlabeled training image data X_u, trained feature mapping function $\phi(\cdot)$, number of nearest neighbors k, label propagation coefficient α
for *epoch* $\leftarrow 1$ to T **do**
  for $i \leftarrow 1$ to n **do**
    $z_i \leftarrow \phi(x_i)$
    find k nearest neighbors $\mathcal{N}(z_i)$
    solve Equation (4.5) to get W_i
  $\mathbf{W} \leftarrow W + W^T$
  $\mathcal{W} \leftarrow D^{-0.5}\mathbf{W}D^{-0.5}$, where D is a diagonal matrix with $d_i = \sum_j \mathbf{w}_{ij}$
  $\mathbf{F} \leftarrow (1-\alpha)(I - \alpha\mathcal{W})^{-1}\mathbf{Y}$, where $\mathbf{Y}$ is constructed by Equation (4.9)
  for $i \leftarrow 1$ to n **do**
    $p_i \leftarrow \arg\max_j \mathbf{f}_{ij}$
    $\mu_i \leftarrow 1 - \frac{-\sum_j \widehat{\mathbf{f}}_{ij} \log(\widehat{\mathbf{f}}_{ij})}{\log(c)}$
  for $i \leftarrow 1$ to n **do**
    calculate δ_i by Equation (4.15)
  $loss \leftarrow \sum_{i=1}^{l} L(f_c(z_i), y_i) + \sum_{i=l+1}^{n} \mu_i \delta_i L(f_c(z_i), p_i)$
  update $f_c(\cdot)$ and $\phi(\cdot)$ ▷ back-propagation
Output: Model parameters θ for $f_\theta(\cdot)$

4.2.3.3 Complexity Analysis

Similarity graph is updated per epoch, which means the calculation for the similarity weight matrix is performed after each training epoch. Therefore, the computational complexity for similarity matrix construction is analyzed.

The optimization problem in Equation (4.5) determines the complexity of similarity matrix construction. Searching for z_i's k nearest neighbors $\mathcal{N}(z_i)$ has complexity of $\mathcal{O}(n^2)$, where n is the total number of training data. Solving the quadratic programming problem in Equation (4.5) has complexity of $\mathcal{O}(nk^3)$ [21]. Therefore, the overall complexity for similarity matrix construction is $\mathcal{O}(n^2 + nk^3)$ for each training epoch.

For datasets used in the experiments in Section 4.3, $n = 50,000$ and $k = 50$. The computational complexity is acceptable for such relatively small datasets. However, large datasets with over millions of training data will have large computational burden due to $\mathcal{O}(n^2 + nk^3)$ complexity. Therefore, the proposed similarity graph construction algorithm is not feasible for large datasets from the perspective of computational efficiency.

4.2.4 Combining with Consistency-based algorithms

As aforementioned in Section 2.3.1, consistency-based algorithm is one promising direction for semi-supervised learning and consistency constraints are complementary with pseudo-labeling based algorithms, to which the proposed algorithm belongs. Therefore, consistency-based regularization is integrated into the proposed approach described in Algorithm 3 and 4, i.e., one more consistency loss is added as regularization to Equation (4.16). Inspired by [47], AutoAugment [127] is used during phase 2 to augment input data x_i to get $\widetilde{x}_i$ and then optimize the final full loss function:

$$\begin{aligned} loss &= \frac{1}{l}\sum_{i=1}^{l} L(f_\theta(x_i), y_i) + \frac{1}{u}\sum_{i=l+1}^{n} \mu_i \delta_i L(f_\theta(x_i), p_i) \\ &+ \frac{1}{u}\sum_{i=l+1}^{n} \mathcal{D}_{KL}(f_\theta(\widetilde{x}_i) || f_\theta(x_i)) \end{aligned} \tag{4.17}$$

4.3 Experiments

4.3.1 Datasets

- **CIFAR-10**[121] A dataset comprising 60,000 colorful images with size 32×32. There are 10 different classes in the whole dataset and images for each class are evenly distributed, i.e., 6000 images for each class. For each class, 5000 images belong to training dataset and the remaining belong to testing dataset.
- **CIFAR-100**[121] Similar with CIFAR-10, there are 60,000 colorful images with size 32×32. Total 100 different classes are in this dataset and images for each class are evenly distributed, i.e., 600 images for each class. For each class, 500 images belong to training dataset and the remaining belong to testing dataset.
- ***mini*ImageNet**[128] A lightweight version of ImageNet, which comprises 60,000 colorful images with size 84×84. There are 100 different classes in the whole dataset and images for each class are evenly distributed, i.e., 600 images for each class. The train/test split in [59] is adopted, where 500 images are assigned to training dataset and the remaining are assigned to testing dataset for each class.

4.3.2 Implementation

For most of the experimental settings, similar approaches used in [59] is followed. The 13-layer CNN model [51], [52], [59] is used for CIFAR-10 and CIFAR-100 datasets and ResNet-18 [3] for *mini*ImageNet. The classifier used is the last fully-connected layer followed by a softmax function, i.e., features before the last fully-connected layer for each data will be z_i in Algorithm 4. The initial learning rate is 0.1 for CIFAR-100 and *mini*ImageNet, 0.05 for CIFAR-10, which will follow a cosine annealing schedule [129] with a period of 210 epochs. The illustration for the learning rate annealing schedule adopted in this work with initial learning rate of 0.1 is shown in Figure 4.4.

SGD optimizer is used with a momentum of 0.9 and weight decay of 0.0002. The total number of training epochs is 180 in phase 1 and 200 in phase 2. For training

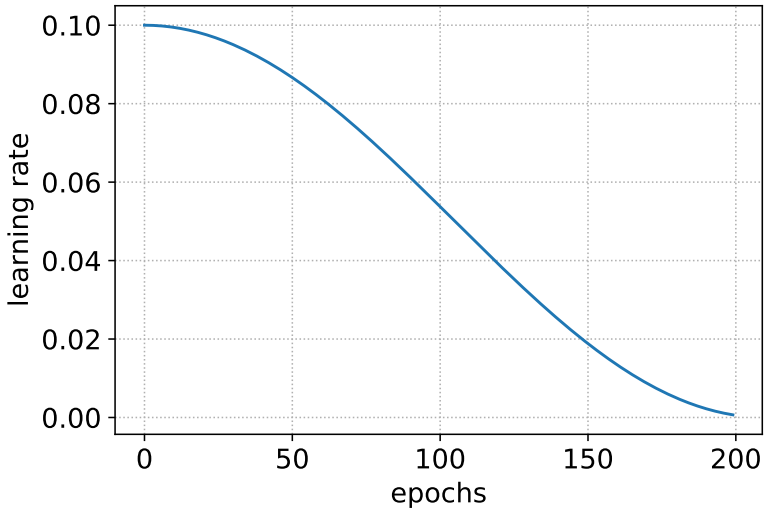


FIGURE 4.4: Learning rate annealing schedule with initial learning rate of 0.1

in phase 2, both labeled data and unlabeled data will be seen in each batch. The number of labeled data in each batch is 31 for CIFAR-100 and *mini*ImageNet and 50 for CIFAR-10. The batch size is 128 for CIFAR-100 and *mini*ImageNet and 100 for CIFAR-10. For every dataset, all training images are augmented by standard random cropping and horizontal flipping followed by channel-wise normalization to have zero mean and unit variance, unlabeled data of which will be augmented one more time with policies from AutoAugment [127]. The number of nearest neighbors k and label propagation coefficient α in Algorithm 4 is set to be 50 and 0.99 respectively. The implementation details are given in Table 4.1.

For the labeled/unlabeled dataset split, scheme from previous research works [51], [52], [59] is adopted. As shown in Figure 4.5, the unlabeled data are evenly distributed among all classes. For the number of labeled data in the following parts, all stated values are the total number of labeled data for the whole dataset.

4.3.3 Experiment Results

In this section, an ablation study is done to analyze the effects of individual components in the proposed algorithm first and followed by discussions and comparisons with previous methods in literature for the experiment results on benchmark datasets.

TABLE 4.1: Implementation details of the experiments

	CIFAR-10	CIFAR-100	*mini*ImageNet
Network	13-layer CNN	13-layer CNN	ResNet-18
Initial Learning Rate	0.05	0.1	0.1
Batch Size	100	128	128
#. of Labeled Data in Each Batch	50	31	31
Optimizer	SGD		
SGD Weight Decay	2×10^{-4}		
SGD Momentum	0.9		
LR Annealing Scheme	cosine		
k	50		
α	0.99		
Total Training Epochs	180 in phase 1 and 200 in phase 2		

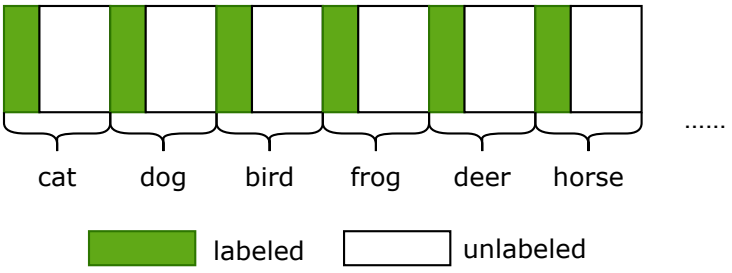


FIGURE 4.5: Labeled/unlabeled dataset split scheme

4.3.3.1 Ablation Study

As discussed in Section 4.2.2, self-supervision tends to provide better feature learning capability and preserving local geometry during graph construction in label propagation will benefit the inference performance for unlabeled data, which are the two main motivations of the proposed algorithm. In the proposed algorithm, these two ideas are combined in order to have information flow from X_l to X_u with less noise and less information loss, which is expected to boost the final classification performance. To verify and analyze the effect from each of them, experiments are conducted for all three datasets with different numbers of labeled data under four different settings, and the results are shown in Figure 4.6, Figure 4.7 and Table 4.2. Moreover, the performance gain due to self-supervision for feature extraction and geometry preserving in graph construction is listed in Table 4.3 and Table 4.4, respectively. It should be noted that all experiments conducted for ablation study

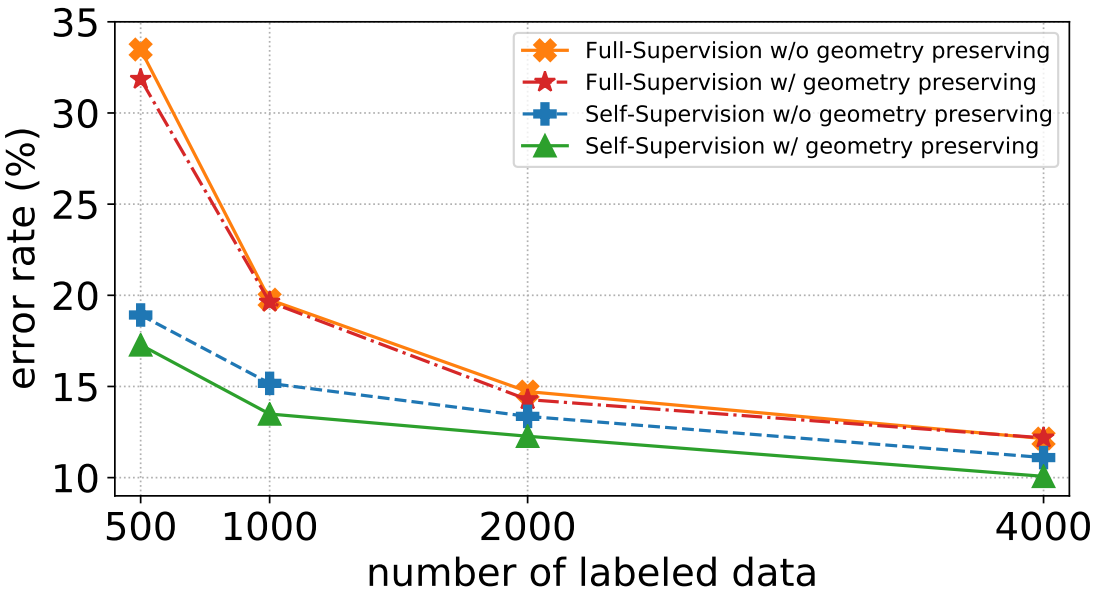
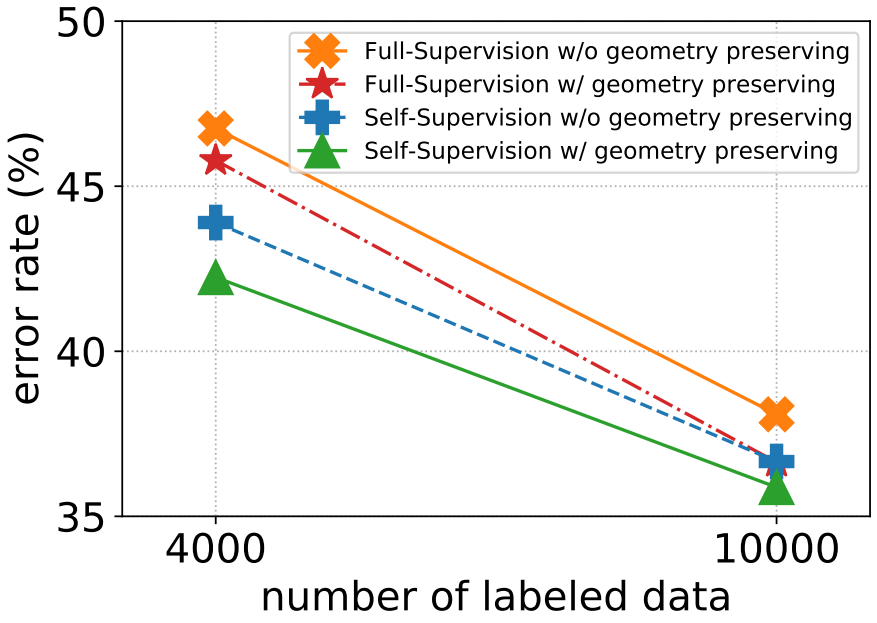


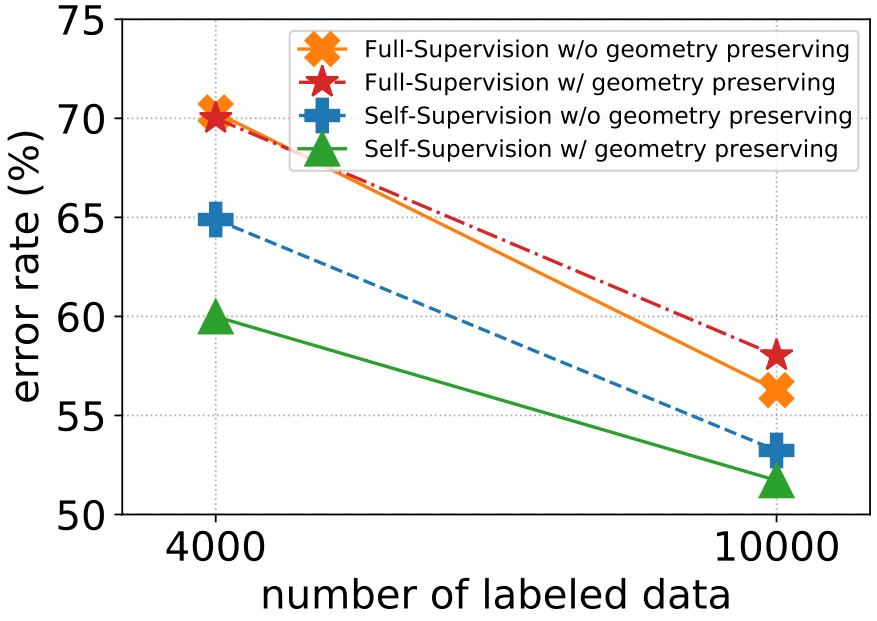
FIGURE 4.6: Testing error rate for CIFAR-10 with 500, 1000, 2000, and 4000 labels with different feature learning schemes (self-supervision or full-supervision) and different graph construction methods in label propagation (preserve local geometry or not)

purpose in this part are using a simplified version of the proposed algorithm, i.e., here the focus is label propagation performance alone so the consistency loss is not included. By doing so, it can be observed from Figure 4.6 and Figure 4.7 that the label propagation method (green-triangle line) of the proposed algorithm outperforms the label propagation method in [59] (orange-cross line) by a large margin as here the effects of consistency loss are excluded for both methods.

CIFAR-10: For CIFAR-10 dataset, it can be observed that the use of self-supervision largely diminishes the impact of labeled data size, as the curves of self-supervision based algorithms are much flatter than the full-supervision based ones in Figure 4.6. The favorable generalization capability of self-supervised features is very pronounced during low labeled data regime, as more than 14% performance improvement can be achieved for 500 labeled data regardless of the method used for graph construction (in Table 4.3). Then for the use of geometry preserving based graph construction, the results in Table 4.4 are interesting. When full-supervision is used to extract features, preserving local geometry or not doesn't seem to give very different results. While self-supervision is used for feature extraction, preserving geometry can provide a stable and sustainable performance gain across different labeled data regimes. It is believed that such behavior difference is



(a) Testing error rate for CIFAR-100 with 4000 and 10000 labels with different feature learning schemes (self-supervision or full-supervision) and different graph construction methods in label propagation (preserve local geometry or not)



(b) Testing error rate for *mini*ImageNet with 4000 and 10000 labels with different feature learning schemes (self-supervision or full-supervision) and different graph construction methods in label propagation (preserve local geometry or not)

FIGURE 4.7: Performance of label propagation for CIFAR-100 and *mini*ImageNet with varying number of labeled data under different settings. For each dataset, the labeled data split for each configuration (#. of labeled data) is the same for fair comparisons. It can be observed that how different feature learning schemes (self-supervision with $\{X_l, X_u\}$ or full-supervision with $\{X_l, Y_l\}$) and graph construction methods (preserve local geometry via reconstructing feature vectors as in Equation (4.5) or without geometry preserving by constructing graph with pairwise inner product) will affect the final classification performance. The green-triangle line corresponds to the proposed algorithm and the orange-cross line corresponds to the method in [59].

TABLE 4.2: Ablation Study Results: Testing error rate

Dataset	CIFAR-10				CIFAR-100		*mini*ImageNet	
#. of labeled data	500	1000	2000	4000	4000	10000	4000	10000
Self-supervision w/ geometry preserving	**17.28**%	**13.49**%	**12.27**%	**10.07**%	**42.24**%	**35.88**%	**59.99**%	**51.72**%
Full-supervision w/ geometry preserving	31.85%	18.63%	14.28%	12.18%	45.78%	36.64%	70%	58.02%
Self-supervision w/o geometry preserving	18.92%	15.17%	13.36%	11.1%	43.9%	36.66%	64.89%	53.23%
Full-supervision w/o geometry preserving	33.48%	19.76%	14.72%	12.14%	46.75%	38.09%	70.3%	56.28%

TABLE 4.3: Performance improvement Δ given by self-supervision under different graph construction methods

Dataset	CIFAR-10			CIFAR-100			*mini*ImageNet	
#. of labeled data	500	1000	2000	4000	4000	10000	4000	10000
w/ geometry preserving	+14.57%	+6.14%	+2.01%	+2.11%	+3.54%	+0.76%	+10.01%	+6.3%
w/o geometry preserving	+14.56%	+4.59%	+1.36%	+1.04%	+2.85%	+1.43%	+5.41%	+3.05%

TABLE 4.4: Performance improvement Δ given by feature space geometry preserving under different feature learning schemes

Dataset	CIFAR-10			CIFAR-100		*mini*ImageNet	
#. of labeled data	500	1000	2000	4000	4000	10000	4000	10000
self-supervision	+1.64%	+1.68%	+1.09%	+1.03%	+1.66%	+0.78%	+4.9%	+1.51%
full-supervision	+1.63%	+0.13%	+0.44%	−0.04%	+0.97%	+1.45%	+0.3%	−1.74%

due to that feature space geometry learned from full-supervision is more noisy and biased than the one learned from self-supervision. Therefore, even if the geometry is preserved, such information will be less helpful in label propagation.

CIFAR-100: For CIFAR-100 dataset, self-supervision shows its superiority at both low and hight labeled data regimes regardless of the method used for graph construction, and larger gain is observed for low labeled data regime (3.54% and 2.85% for 4000 labeled data compared with 0.76% and 1.43% for 10000 labeled data from Table 4.3). By preserving local geometry for graph construction in label propagation, the performance is boosted for both self-supervised and fully-supervised features, which suggests that CIFAR-100 dataset is less susceptible to noisy feature space geometry during label propagation, as the geometry information can always provide performance gain under both feature learning schemes.

***mini*ImageNet:** For *mini*ImageNet, it can be observed that the effects of self-supervision and geometry preserving are very noticeable, especially at low labeled data regime. Self-supervision can improve the performance under both graph construction methods. However, the gain of 10.01% and 6.3% when geometry preserving based graph construction is used is obviously higher than 5.41% and 3.05% without geometry preserving for 4000 labels and 10000 labels respectively (shown in Table 4.3), which indicates that self-supervised features are better utilized by geometry preserving based label propagation. The use of geometry preserving based graph construction gives very different results under the two feature learning schemes in Table 4.4. Preserving local geometry gives 4.9% and 1.51% performance improvement when self-supervision is used, while only 0.3% and even -1.74% when full-supervision is used, for 4000 and 10000 labeled data respectively. Preserving the local geometry is not only helpless but also harmful to the performance under full-supervised feature learning scheme, which implies that *mini*ImageNet dataset is more sensitive and susceptible to noisy feature space geometry during label propagation.

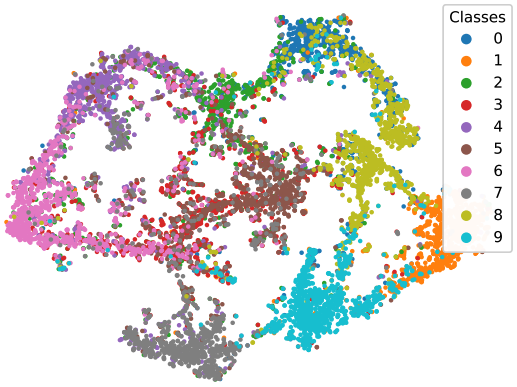
To summarize, the effectiveness of self-supervision for feature extraction and geometry preserving based graph construction for label propagation are verified on all datasets across all labeled data regimes and each of them plays a vital role in the proposed algorithm. Moreover, geometry preserving during label propagation and self-supervision based feature learning scheme are mutually reinforcing, as neither

can benefit performance the most without the other, especially for dataset sensitive to noisy feature space geometry.

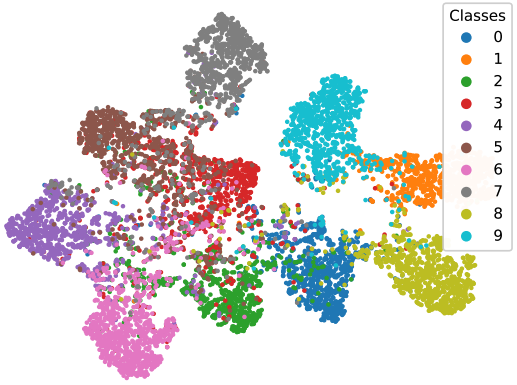
4.3.3.2 Proposed Algorithm

Here the results from the proposed algorithm optimized by the full loss function in Equation (4.17) is reported and discussed .

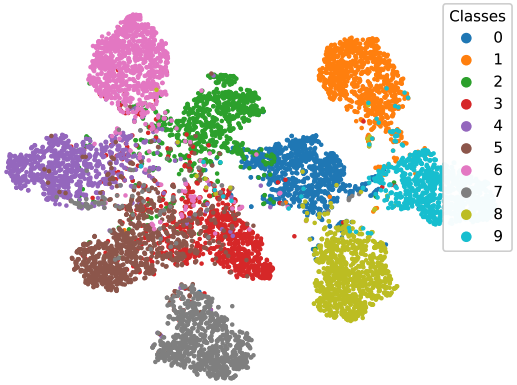
Visualizations: 2D t-SNE visualizations of embeddings before the last linear layer for CIFAR-10 testing data are shown in Figure 4.8. From Figure 4.8a to Figure 4.8b, the effectiveness of the proposed local geometry preserving scheme with feature learning under self-supervision can be clearly demonstrated. In Figure 4.8a, only 3 classes (class '7', '9', '1') can be relatively separated from other classes and the worst performance is on two pairs: class '0' airplane and class '8' ship, class '5' dog and class '3' cat. It can be seen that points belonging to class '5' almost fully overlap with points belonging to class '3', which indicates that the model cannot distinguish images of dog and cat, and a similar case for class '0' and class '8'. Sample images from these two challenging pairs are shown in Figure 4.9, which can be observed that indeed the contrast between them is not significant, especially for dog and cat. In Figure 4.8b, a very pronounced improvement can be observed that almost all ten classes can be separated from each other. For the two challenging pairs, class '0' and class '8' can be separated now and the overlapping between points of class '3' and class '5' is greatly alleviated. Moving to Figure 4.8c, i.e., visualization for embeddings from the proposed algorithm where consistency constraints are utilized on top of the pseudo-labeling approach in Figure 4.8b. Although there are still some overlapping points from class '3' and class '5', the embeddings now are closer to a perfect clustering compared with Figure 4.8b, which can confirm that the proposed pseudo-labeling scheme is complementary to consistency-based algorithms.



(a)

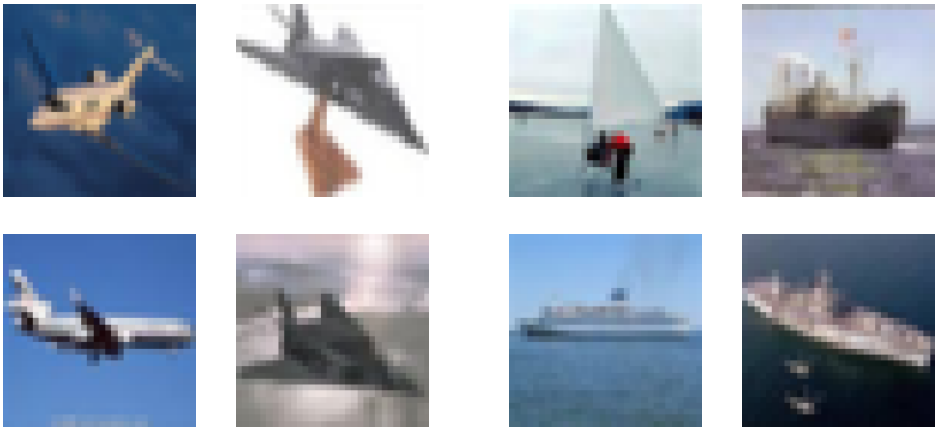


(b)



(c)

FIGURE 4.8: 2D t-SNE visualizations of embeddings before last linear layer for CIFAR-10 testing data under different training schemes (all trained with the same 500 labeled data split): (A) fully-supervised feature learning w/o geometry preserving, (B) self-supervised feature learning w/ geometry preserving and (C) the proposed algorithm. Each point represents one testing image and colors correspond to ground-truth labels (class 0 to 9 correspond to airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). Recommend to view in color version.



(a) Samples images of airplane and ship from CIFAR-10 testing dataset.

(b) Samples images of dog and cat from CIFAR-10 testing dataset.

FIGURE 4.9: Two challenging pairs of visual categories in CIFAR-10 dataset: airplane & ship, dog & cat.

TABLE 4.5: Top-1 error rate on CIFAR datasets (%)

Dataset	CIFAR-10				CIFAR-100	
#. of labeled data	500	1000	2000	4000	4000	10000
Π-Model[51]†	-	-	-	12.36 ± 0.31	39.19 ± 0.36	-
Temporal Ensemble[51]†	-	-	-	12.16 ± 0.24	38.65 ± 0.51	-
Mean Teacher[52]*	27.45 ± 2.64	19.04 ± 0.51	14.35 ± 0.31	11.41 ± 0.25	45.36 ± 0.49	36.08 ± 0.51
VAT+EntMin[46]†	-	-	-	10.55 ± 0.05	-	-
SNTG[58]†	-	18.41 ± 0.52	13.64 ± 0.32	10.93 ± 0.14	-	37.97 ± 0.29
VAdD[50]†	-	-	-	9.22 ± 0.10	-	-
Fast-SWA+MT[54]†	-	15.58 ± 0.12	11.02 ± 0.23	9.05 ± 0.21	-	33.62 ± 0.54
ICT[48]†	-	15.48 ± 0.78	9.26 ± 0.09	7.29 ± 0.02	-	-
MixMatch[49]†	-	-	-	6.84	-	-
AutoAugment[127]	20.51 ± 0.47	14.94 ± 0.12	10.21 ± 0.23	8.08 ± 0.05	39.64 ± 0.62	31.35 ± 0.67
TSSDL[57]†	-	18.41 ± 0.92	13.54 ± 0.32	9.3 ± 0.55	-	-
LP [59]†	24.02 ± 2.44	16.93 ± 0.70	13.22 ± 0.29	10.61 ± 0.28	43.73 ± 0.20	35.92 ± 0.47
MUSCLE [130]⋆	16.64 ± 0.43	13.29 ± 0.36	-	-	42.34 ± 0.62	35.21 ± 0.25
Proposed	$\mathbf{9.89 \pm 0.55}$	$\mathbf{8.38 \pm 0.47}$	$\mathbf{7.55 \pm 0.10}$	$\mathbf{6.65 \pm 0.09}$	$\mathbf{35.05 \pm 0.24}$	$\mathbf{29.18 \pm 0.13}$

† denotes numbers are results reported in their original research works. ∗ For Mean Teacher algorithm, the numbers are from [59] as their reproduced results are better than the original work of Mean Teacher in [52] and they reported results on more datasets. For AutoAugment algorithm, this result is generated empirically. ⋆ MUSCLE [130] is from a recent research work in late 2020, which is published after the proposed algorithm in this chapter.

TABLE 4.6: Top-1 error rate on *mini*ImageNet dataset (%)

Dataset	*mini*ImageNet	
#. of labeled data	4000	10000
Mean Teacher[52]*	72.51 ± 0.22	57.55 ± 1.11
AutoAugment[127]	71.76 ± 0.97	56.71 ± 0.47
LP[59]†	72.78 ± 0.15	57.35 ± 1.66
MUSCLE [130]⋆	62.65 ± 0.25	52.70 ± 1.12
Proposed	$\mathbf{49.21 \pm 0.59}$	$\mathbf{41.68 \pm 0.23}$

† denotes numbers are results reported in their original research works. ∗ For Mean Teacher algorithm, the numbers are from [59] as their reproduced results are better than the original work of Mean Teacher in [52] and they reported results on more datasets. For AutoAugment algorithm, this result is generated empirically. ⋆ MUSCLE [130] is from a recent research work in late 2020, which is published after the proposed algorithm in this chapter.

Comparisons with the state-of-the-art: In Table 4.5 and Table 4.6, the proposed algorithm is compared with previous state-of-the-art semi-supervised learning methods. All methods in Table 4.5 use the same 13-layer CNN network architecture and all methods in Table 4.6 use the same ResNet-18 network architecture for fair comparisons. The results averaged over three different data splits for each configuration are reported. As AutoAugment [127] is used in consistency constraints for the proposed algorithm, a baseline model is generated by using it alone and also report its performance in Table 4.5 and Table 4.6. AutoAugment is used for data augmentation in fully-supervised training in its original paper, here it is utilized in consistency-based semi-supervised learning by optimize the loss function in Equation (4.17) without the pseudo-labeling loss, i.e., the second term, and all other settings remain the same. The top part of Table 4.5 and Table 4.6 summarizes consistency-based semi-supervised learning algorithms and the bottom part summarizes pseudo-labeling based algorithms. MUSCLE [130] is from a recent research work in late 2020, which is published after the proposed algorithm in this chapter. MUSCLE is included in Table 4.5 and Table 4.6 to further strengthen the comparisons with related research works. TSSDL [57], LP [59] and MUSCLE [130] have incorporated consistency loss in their works, similar with the proposed algorithm. Among all algorithms, the proposed algorithm achieves the best results on all three datasets from low labeled data regime to high labeled data regime.

CIFAR-Datasets: Among pseudo-labeling based algorithms, the improvement that the proposed algorithm has made over the previous state-of-the-art [59] is ranging from 3.96% to 14.13%. Comparing with AutoAugment baseline model, it can be observed that the proposed pseudo-labeling scheme is capable of boosting the performance by a large margin, especially in low labeled data regime (up to 10.62% performance gain). Such behavior once again verifies the efficacy of the proposed pseudo-labeling scheme for semi-supervised learning, as the model capability in extreme low labeled data regime reveals its core competence.

***mini*ImageNet**: The proposed algorithm achieves very notable performance on *mini*ImageNet dataset by giving an error rate lower than 50% and 42% with only 4000 labels and 10000 labels respectively. Performance gains of 21.08% and 15.67% have been made over the previous state-of-the-art [59] with 4000 labels and 10000 labels respectively. 49.21% error rate with 4000 labels has been achieved, which is lower than 57.35% with 10000 labels by [59], meaning that the proposed algorithm can beat previous state-of-the-art with 2.5× fewer labels by 8.14%.

4.4 Summary

This chapter presents the second work of this thesis, i.e., a novel transductive pseudo-labeling based semi-supervised learning algorithm, which has addressed the biased and over-fitted feature mapping problem during label propagation for pseudo label inference. Related research works and their limitations are discussed in Section 4.1. Existing label propagation based methods fail to extract clean semantic features due to noisy fully-supervised training with limited labeled data and commonly ignore local geometry information in latent feature space. In Section 4.2, a novel pseudo-labeling scheme via label propagation for deep semi-supervised visual recognition is proposed. The proposed scheme incorporates self-supervised learning into feature extraction to utilize all training data so that unbiased and clean information flow can be achieved in subsequent label propagation. Local geometry information is preserved in the proposed scheme by reconstructing feature vector with its neighbors during similarity graph construction. In ablation study of Section 4.3, synergistic effects are observed on features learned with self-supervision and local geometry preserved label propagation. Such empirical results confirm that information flow from labeled data to unlabeled data should be kept noiseless and with minimum loss for semi-supervised learning, which may inspire future research works. The proposed pseudo-labeling scheme is applied together with consistency constraints in the proposed algorithm as they are complementary to each other. In Section 4.3, experiments conducted on three benchmark datasets demonstrate the effectiveness of the proposed algorithm and results show that the proposed algorithm consistently outperforms most of the state-of-the-art semi-supervised learning methods under the same network architecture.

Chapter 5

End-to-end Novel Visual Categories Learning via Auxiliary Self-Supervision

Chapter 5 explores the use of self-supervised learning in novel visual categories learning as an auxiliary task and introduces an end-to-end novel visual categories learning algorithm by utilizing self-supervision signals simultaneously with pairwise similarity information. Section 5.1 reviews the limitation of previous related works in novel visual categories learning and presents the research motivations. Detailed explanations of the proposed end-to-end novel visual categories learning algorithm via auxiliary self-supervision are given in Section 5.2. Section 5.3 validates the proposed algorithm on three benchmark datasets for both novel categories clustering task and mixed classification task.

5.1 Background and Motivations

Semi-supervised learning has largely alleviated the strong demand for large amount of annotations in deep learning. Many state-of-the-art semi-supervised learning methods are capable of achieving good performance with much less labels than those of supervised baseline [46], [47], [49], [51], [52], [55], [59], [131], [132]. Under a common assumption adopted by semi-supervised learning algorithm, there is always labeled data available from the same class of unlabeled data. However, it is difficult to satisfy such strict assumption in most real-world situations. The unlabeled data may come from novel categories, which means that no overlapping between existing classes of labeled data and unknown classes of unlabeled data. In this work, the focus is on semi-supervised learning when the categories of unlabeled data and labeled data are disjoint from each other. As shown in Figure 5.1, the labeled/unlabeled dataset split scheme for novel categories learning is different from the one in Chapter 4 (Figure 4.5).

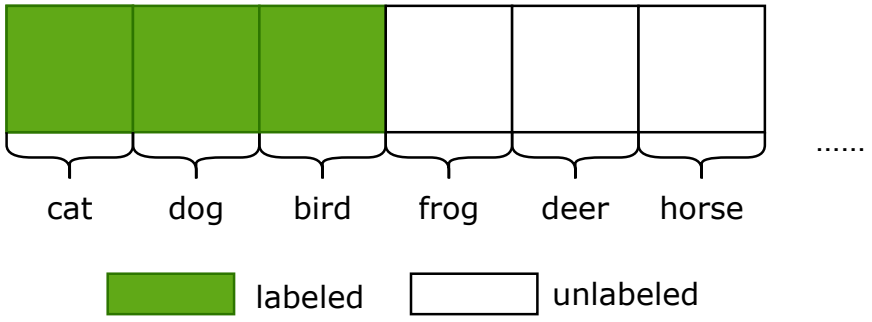


FIGURE 5.1: Labeled/unlabeled dataset split scheme

One example in real-world applications is e-commerce platform. For the e-commerce platform, products need to be categorized for the convenience of users. However, a flux of new products come in everyday and even every second. How to do the classification for those new products that belong to unseen categories? This is the question that this chapter is going to discuss.

When labeled and unlabeled data belong to disjoint categories, label information from labeled data is not related to unlabeled data, which means that discriminative information has difficulty to transfer from labeled data to unlabeled data. Previous research works [81]–[83] have utilized pairwise similarity information to handle such problem. Pairwise similarity refers to identifying whether two data are similar or dissimilar and it is assumed that two images belong to the same category if they are

similar, otherwise they belong to different categories. Pairwise similarity information is a weaker supervision signal compared with multi-class label information, and the empirical results [81]–[83] suggest that such weak supervision signal is more apt to novel visual categories learning than the other. KCL proposed in [82] and MCL proposed in [83] generate pairwise similarity pseudo labels for unlabeled data by a similarity prediction network (SPN), which is trained by labeled data at the first stage. As the SPN is solely learning on the pairwise relationship of labeled data, the model is biased and can easily over-fit noises. AutoNovel proposed in [81] utilizes RotNet for feature learning and feature vectors are compared via rank statistics to construct pairwise similarity pseudo labels. The use of self-supervised learning (RotNet) in AutoNovel has alleviated the over-fitting problem on labeled data, but problems still remain. In AutoNovel, most convolutional blocks of the model are frozen after the RotNet pre-training stage and only one single convolutional block together with the linear layer are updating in the subsequent discriminative training stage, which is to avoid the forgetting issue as the network may once again over-fit to labeled data if the whole network is updated.

In this work, an end-to-end novel visual categories learning algorithm is proposed by introducing auxiliary task via self-supervision. Self-supervised learning has demonstrated very prominent results in the field of unsupervised feature learning [133]. Pretext tasks are defined in self-supervised learning and the required supervising signals are free to get, e.g., angle of rotational transformation that has been applied [20], color of the image [17] and order of patches from the same image [19]. Such self-supervision signals apply to both labeled and unlabeled data, and are argued to be able to teach models on studying high-level features. Instead of utilizing self-supervision as a pre-training stage, which is commonly adopted for semi-supervised learning purpose [20], [81], [134], self-supervision signals are incorporated with pairwise similarity information to supervise the model simultaneously in the proposed algorithm. Thus, supervising signals injected to the model will have strong regularization so that performance degradation caused by noisy and biased pairwise pseudo labels can be largely alleviated.

Moreover, local data structure information in feature space is exploited to further improve the quality of pairwise pseudo labels. Methods proposed in [82] and [83] generate pseudo labels with static function/network, whereas [81] constructs pseudo labels on-the-fly with rank statistics. Inspired by the superior performance

of the latter one in [81], similar philosophy is adopted, i.e., producing pairwise pseudo labels by comparing feature vectors. However, feature space trained without ground-truth label information is noisy. Vanilla similarity metrics, e.g., cosine similarity, may fail to capture correct relationship between feature vectors. Therefore, conditional probability in symmetric Stochastic Neighbor Embedding (SNE) [22] is proposed to be used as pairwise similarity measure to enforce local properties, which are more robust to noisy feature space.
The main contributions are summarized as follows:

1. An end-to-end novel visual categories learning algorithm is proposed by utilizing self-supervision signals simultaneously with pairwise similarity information, which is the first research work that does not require model pre-training such that end-to-end learning can be achieved for novel visual categories learning.

2. Robust local structure properties in noisy feature space are utilized for the construction of pairwise similarity pseudo labels. In the ablation study, experiment results have verified that pairwise similarity pseudo labels constructed by the proposed algorithm can capture data relationship more accurately than that of commonly-used cosine similarity method.

3. Extensive experiments conducted for three commonly-used visual datasets, i.e., CIFAR-10, CIFAR-100 and SVHN, have demonstrated the effectiveness of the proposed algorithm as it is observed that the proposed algorithm has outperformed other state-of-the-art methods on benchmark datasets.

5.2 Proposed Algorithm

5.2.1 Problem Formulation

Given a set of image data $X = \{x_1, x_2, ..., x_N\}$, labels are only known for part of the classes, denoted by $\{X_l, Y_l\}$, and $Y_l = \{y_1, y_2, ..., y_l\}$ are the corresponding labels for labeled data X_l. The number of all possible classes in Y_l is denoted as c_l. For remaining data in X, the corresponding labels are unknown, denoted by $\{X_u, Y_u\}$. The number of all possible classes in Y_u is denoted as c_u, which is assumed to be known. $Y_u \bigcap Y_l = \varnothing$, i.e., the label classes in X_u are disjoint from label classes in X_l. X_l and X_u are assumed to share certain common knowledge such that they do not differ too much. The objective is to correctly cluster X_u into the number of c_u classes as much as possible with given $\{X_l, Y_l\}$.

5.2.2 End-to-end Novel Visual Categories Learning via Auxiliary Self-Supervision

The illustration of the proposed algorithm is shown in Figure 5.2. Predicting the angle of rotational transformation that has been applied to original images proposed in [20] is utilized in the proposed algorithm as auxiliary self-supervision task. Three different linear classifiers are used to map data points from sharing feature space to three target spaces, i.e., pairwise similarity pseudo label space, categorical label space and self-supervision label space respectively, which correspond to three modules in the proposed algorithm. In the following subsections, detailed explanations are presented for each module.

5.2.2.1 Pairwise Similarity Learning

Without any pre-training or fine-tuning being performed, the proposed algorithm starts with a random initialized CNN network. Image data in X is fed into the network and the corresponding feature vector ϕ_i before linear layers for each $x_i \in X_u$ is collected to compute the pairwise similarity. As there is no label information for X_u, the learned feature space is vulnerable to noise. Therefore, robust local data structure information in the feature space is proposed to be exploited to mitigate

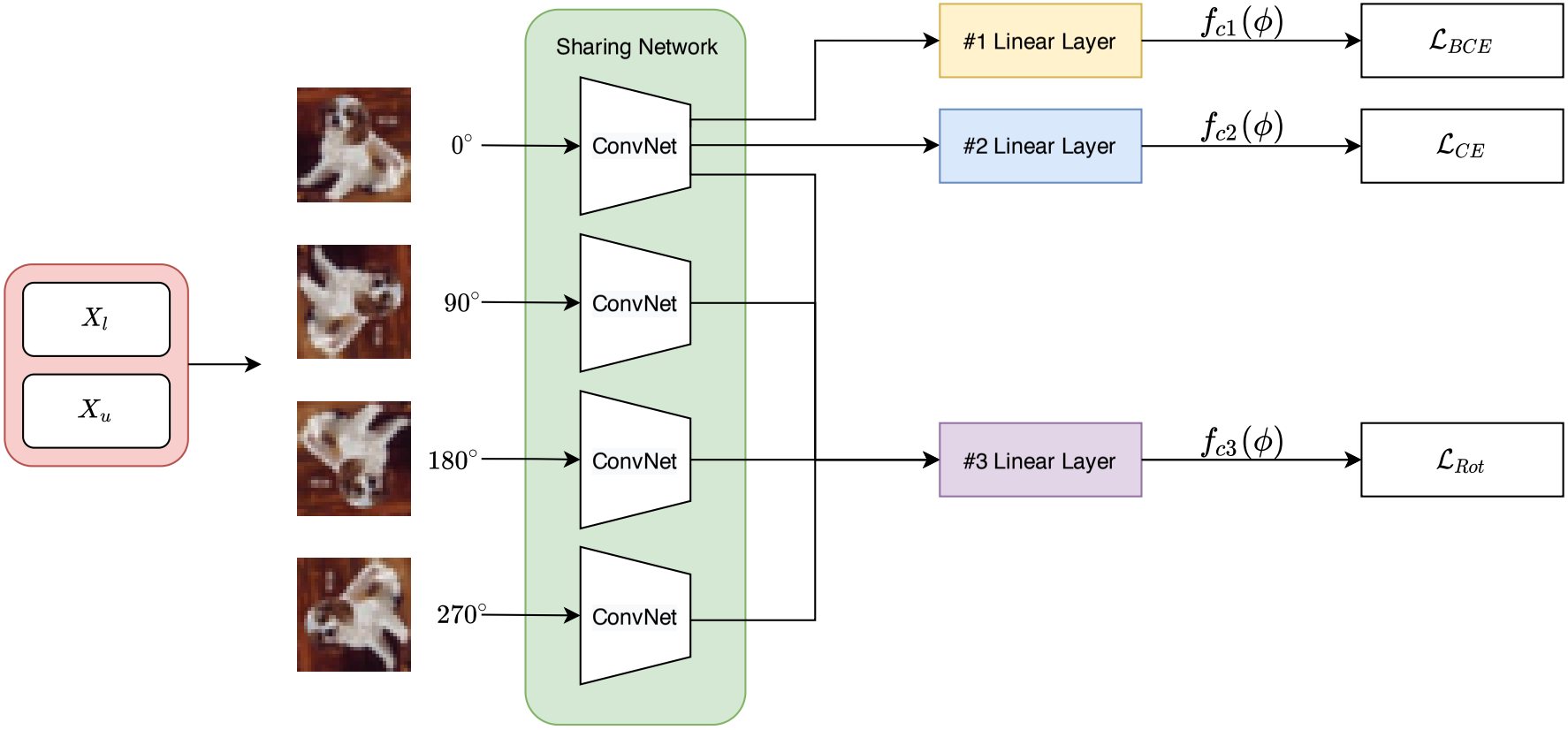


FIGURE 5.2: The illustration of the proposed end-to-end novel visual categories learning algorithm. For both labeled data X_l and unlabeled data X_u, the images are rotated by {0°, 90°, 180°, 270°}. All rotated images are fed into the convolutional neural network to get the corresponding feature vectors ϕ. The feature vectors for 0° rotated images, i.e., the original images, are used to calculated two losses: binary cross-entropy loss $\mathcal{L}_{BCE}$ for X_u and cross-entropy loss $\mathcal{L}_{CE}$ for both X_l and X_u. Feature vectors for all {0°, 90°, 180°, 270°} rotated images are used to calculate the third loss: rotation loss $\mathcal{L}_{Rot}$. f_{c1}, f_{c2} and f_{c3} represent three different linear classifiers (i.e., fully-connected layers), that are employed to map feature vectors to the target space for different loss calculations respectively.

such impact. Conditional probability proposed in symmetric SNE [22] is used to represent the similarity s_{ij} between ϕ_i and ϕ_j, computed with Equation (5.1) and Equation (5.2).

$$p_{i|j} = \frac{\exp(-||\phi_j - \phi_i||^2/T^2)}{\sum_{k \neq j} \exp(-||\phi_j - \phi_k||^2/T^2)} \tag{5.1}$$

$$s_{ij} = \frac{1}{2}(p_{i|j} + p_{j|i}) \tag{5.2}$$

where $p_{i|j}$ represents the conditional probability of ϕ_i being selected as a neighbor of ϕ_j, given by a Gaussian distribution centered at ϕ_j, as shown in Equation (5.1). If two feature vectors are not close enough to each other, the probability will be small. Temperature coefficient T controls the number of neighbors being considered.

After the computation of similarity s_{ij}, binary cross-entropy loss is calculated accordingly to enforce the network giving similar predictions for image data with similar feature vectors, as shown in Equation (5.3) and Equation (5.4).

$$\mathcal{L}_{BCE} = -\frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \{A_{ij} \log(\widehat{\mathbf{y}}_\mathbf{i}^\top \cdot \widehat{\mathbf{y}}_\mathbf{j}) + (1 - A_{ij}) \log(1 - \widehat{\mathbf{y}}_\mathbf{i}^\top \cdot \widehat{\mathbf{y}}_\mathbf{j})\} \tag{5.3}$$

$$A_{ij} = \begin{cases} 1 & if\ s_{ij} > \tau \\ s_{ij} & otherwise \end{cases} \tag{5.4}$$

where $\widehat{\mathbf{y}}_\mathbf{i} = softmax(f_{c1}(\phi_i)) \in R^{c_u}$, the output from linear layer f_{c1} for feature vector ϕ_i followed by a *softmax* function. In the proposed algorithm, the network is expected to be updated once per batch. Therefore, $\mathcal{L}_{BCE}$ is calculated for image data in a batch and m is the number of unlabeled data in a batch. Similarly, the calculation of s_{ij} is done for each batch, instead of the whole dataset. A_{ij} equals to 1 if two image data x_i and x_j are considered to be similar, i.e., s_{ij} is larger than the threshold value τ. Here the inner product between $\widehat{\mathbf{y}}_\mathbf{i}$ and $\widehat{\mathbf{y}}_\mathbf{j}$ measures the similarity between network predictions for image data x_i and x_j.

As the calculation of pairwise pseudo label A_{ij} is performed for every batch, the corresponding computational complexity is $\mathcal{O}(m^2)$ according to Equation (5.1). The computational complexity is independent with the training data size, which makes the proposed algorithm more applicable to large datasets from the perspective of computational efficiency.

5.2.2.2 Mixed Label Classification

Besides pairwise similarity learning for unlabeled data, discriminative feature learning for classification is essential as label information is known for X_l. Firstly cross-entropy loss $\mathcal{L}_l$ between the network predictions and Y_l is calculated for all data in X_l. For cross-entropy loss $\mathcal{L}_u$ on unlabeled data, predictions $f_{c1}(\phi_i)$ produced by the network in the previous pairwise similarity learning can be used to infer pseudo labels. Therefore, cross-entropy loss is computed between the network predictions and true labels for X_l together with pseudo labels for X_u. One more linear layer f_{c2} is added to the network to map feature vectors to label space of dimension $c_l + c_u$. The computation of $\mathcal{L}_{CE}$ is shown in Equation (5.5).

$$\begin{aligned} \mathcal{L}_{CE} &= \quad \mathcal{L}_l \quad + \quad \mathcal{L}_u \\ &= -\frac{1}{n}\Big(\sum_{i:x_i \in X_l} \log(\widehat{\mathbf{y}}_\mathbf{i})_{y_i} + \lambda_{pl} \sum_{i:x_i \in X_u} \log(\widehat{\mathbf{y}}_\mathbf{i})_{p_i}\Big) \end{aligned} \tag{5.5}$$

where n is the batch size, i.e., the total number of labeled data and unlabeled data in a batch, $\widehat{\mathbf{y}}_\mathbf{i} = softmax(f_{c2}(\phi_i)) \in R^{c_l+c_u}$, the output from linear layer f_{c2} for feature vector ϕ_i followed by a *softmax* function, y_i is the ground-truth label, λ_{pl} is the weight coefficient for pseudo labels and the pseudo label p_i for each unlabeled data is inferred with Equation (5.6).

$$p_i = c_l + \arg\max\{softmax(f_{c1}(\phi_i))\} \tag{5.6}$$

Pseudo label p_i is inferred by the output from the first linear layer f_{c1} incremented by c_l, as pseudo labels are mixed together with ground-truth labels to calculate the loss. Similar to [81], consistency loss is added on the output from linear layer f_{c1} and f_{c2} to enforce the stability and robustness of the proposed algorithm, as shown in Equation (5.7).

$$\begin{aligned} \mathcal{L}_{MSE} = &\frac{1}{n-m} \sum_{i:x_i \in X_l} ||softmax(f_{c2}(\phi_i)) - softmax(f_{c2}(\widehat{\phi}_i))||^2 \\ &+\frac{1}{m} \sum_{i:x_i \in X_u} ||softmax(f_{c1}(\phi_i)) - softmax(f_{c1}(\widehat{\phi}_i))||^2 \end{aligned} \tag{5.7}$$

where $\widehat{\phi}_i$ denotes the feature vector of $\widehat{x}_i$, a random-augmented counterpart of x_i.

TABLE 5.1: Different linear layers of the proposed algorithm

Linear Layer	Output Dimension	Loss Function
f_{c1}	C_u	$\mathcal{L}_{BCE}$
f_{c2}	$C_u + C_l$	$\mathcal{L}_{CE}$
f_{c3}	4	$\mathcal{L}_{Rot}$

5.2.2.3 Self-Supervised Learning

Pairwise similarity learning builds the pairwise relationship between unlabeled data based on the learned feature space, which is primarily supervised by the label information of X_l, shown in Equation (5.5). However, the extracted features solely depending on image data from labeled classes are highly likely to be biased and its generalization capability for unlabeled data will be degraded. To alleviate such problem, the proposed algorithm gives additional supervision signal other than the label information, which is only available for limited number of data. In [20], the network is trained to predict rotational geometric transformation that has been applied to the image data, serving the role of self-supervision. The rotation-based self-supervision signal is incorporated into the proposed algorithm by requiring the network to classify image categories and rotational transformations that has been applied simultaneously.

Every image $x_i \in X$ is rotated by $\{0°, 90°, 180°, 270°\}$, and the rotated images $\mathbb{X}$ are fed into the same network to get feature vectors, as shown in Figure 5.2. The third linear layer f_{c3} is added to map feature vectors of rotated images to surrogate label space of dimension 4. Rotation loss $\mathcal{L}_{Rot}$ is calculated by computing the cross-entropy loss between network prediction $\widehat{\mathbf{y}}_\mathbf{i}$ and rotational surrogate label $r = \{0, 1, 2, 3\}$ for images after rotational transformation of $\{0°, 90°, 180°, 270°\}$ respectively, as shown in Equation (5.8).

$$\mathcal{L}_{Rot} = -\frac{1}{n} \sum_{i:x_i \in \mathbb{X}} \log(\widehat{\mathbf{y}}_\mathbf{i})_{r_i} \tag{5.8}$$

where $\widehat{\mathbf{y}}_\mathbf{i} = softmax(f_{c3}(\phi_i)) \in R^4$, the output from linear layer f_{c3} for feature vector ϕ_i of all rotated images followed by a *softmax* function. The summary of the three different linear layers f_{c1}, f_{c2} and f_{c3} is given in Table 5.1.

The overall objective function is computed in Equation (5.9).

$$\mathcal{L}_{all} = (1 - \lambda_{Rot})(\mathcal{L}_{BCE} + \mathcal{L}_{CE} + \beta\mathcal{L}_{MSE}) + \lambda_{Rot}\mathcal{L}_{Rot} \tag{5.9}$$

where λ_{Rot} controls the relative importance between auxiliary self-supervision and explicit label information, and β determines the weight of consistency loss. The pseudocode for the proposed algorithm is summarized in Algorithm 5.

Algorithm 5 End-to-end Novel Visual Categories Learning via Auxiliary Self-Supervision

Input: Labeled training images from known classes and the corresponding labels $\{X_l, Y_l\}$, unlabeled training images X_u from novel classes, the number of all possible labeled classes c_l, the number of all possible novel classes c_u, temperature coefficient T, threshold value τ, weight coefficient λ_{pl}, λ_{Rot} and β, convolutional layers of random initialized neural network $g_\theta(\cdot)$, three linear layers $f_{c1}(\cdot), f_{c2}(\cdot), f_{c3}(\cdot)$
for *epoch* $\leftarrow 1$ to # of training epochs **do**
  for sampled minibatch B **do**
    $\phi_{i\in B} \leftarrow g_\theta(x_i)$
    for $i, j \in B \cap X_u$ **do**
      $p_{i|j} \leftarrow \frac{\exp(-||\phi_j - \phi_i||^2/T^2)}{\sum_{k\neq j}\exp(-||\phi_j-\phi_k||^2/T^2)}$
      $s_{ij} \leftarrow \frac{1}{2}(p_{i|j} + p_{j|i})$
      $\mathcal{L}_{BCE} \leftarrow \sum L(A_{ij}, f_{c1}(\phi_i), f_{c1}(\phi_j))$, where A_{ij} is defined in Equation (5.4)
      $p_i \leftarrow c_l + \arg\max\{softmax(f_{c1}(\phi_i))\}$
    $\mathcal{L}_{CE} \leftarrow \sum_{i\in B\cap X_l} L(y_i, f_{c2}(\phi_i)) + \lambda_{pl}\sum_{i\in B\cap X_u} L(p_i, f_{c2}(\phi_i))$
    $\widehat{x}_{i\in B} \leftarrow$ randomly augment x_i
    $\widehat{\phi}_i \leftarrow g_\theta(\widehat{x}_i)$
    $\mathcal{L}_{MSE} \leftarrow \sum_{i\in B\cap X_l} L(f_{c2}(\phi_i), f_{c2}(\widehat{\phi}_i)) + \sum_{i\in B\cap X_u} L(f_{c1}(\phi_i), f_{c1}(\widehat{\phi}_i))$
    $x^r_{i\in B} \leftarrow$ rotate x_i by $0°, 90°, 180°, 270°$
    $\phi^r_i \leftarrow g_\theta(x^r_i)$
    $r_{i\in B} \leftarrow \{0, 1, 2, 3\}$
    $\mathcal{L}_{Rot} \leftarrow \sum_{i\in B} L(r_i, f_{c3}(\phi^r_i))$
    $loss \leftarrow (1-\lambda_{Rot})(\mathcal{L}_{BCE} + \mathcal{L}_{CE} + \beta\mathcal{L}_{MSE}) + \lambda_{Rot}\mathcal{L}_{Rot}$
    update $f_{c1}(\cdot), f_{c2}(\cdot), f_{c3}(\cdot)$ and $g_\theta(\cdot)$ ▷ back-propagation
Output: Model parameters θ of $g_\theta(\cdot)$ and $f_{c1}(\cdot), f_{c2}(\cdot), f_{c3}(\cdot)$

5.3 Experiments

5.3.1 Datasets

- **CIFAR-10** dataset [121] is one commonly-used benchmark dataset for visual classification. There are total 60,000 color images of size 32×32, evenly distributed among ten different classes. For each class, 5000 images belong to training data and the rest 1000 images belong to testing data.
- **CIFAR-100** dataset [121] is similar to CIFAR-10 in terms of dataset size, but CIFAR-100 is much more challenging than CIFAR-10 as there are more diverse images in CIFAR-100. There are total 60,000 colorful images of size 32×32, evenly distributed among 100 different classes. For each class, 500 images belong to training data and the rest 100 images belong to testing data.
- **SVHN** (Street View House Numbers) dataset [135] is a real-world digit number recognition dataset, comprising 73,257 and 26,032 colorful images of size 32×32 for training and testing data respectively. There are total 10 classes in SVHN dataset, where each digit number represents one class.

5.3.2 Implementation

The network architecture used is ResNet-18 [3]. For CIFAR-10 and SVHN dataset, half of the training data are used as labeled data and the other half are used as unlabeled data, i.e., $c_l = c_u = 5$. An illustration for CIFAR-10 dataset split is given in Figure 5.3.

For CIFAR-100, 80 classes of training images are used as labeled data and the rest 20 classes of training images are utilized as unlabeled images from novel categories, i.e., $c_l = 80$ and $c_u = 20$. For the hyper-parameter setting, λ_{Rot} is set to be 0.8 for all datasets and the settings reported in [81] are followed for the weight coefficient λ_{pl} of pseudo labels and the weight β of consistency loss. λ_{pl} and β take the form of a ramp-up function $w(t)$ along the training process, computed with Equation

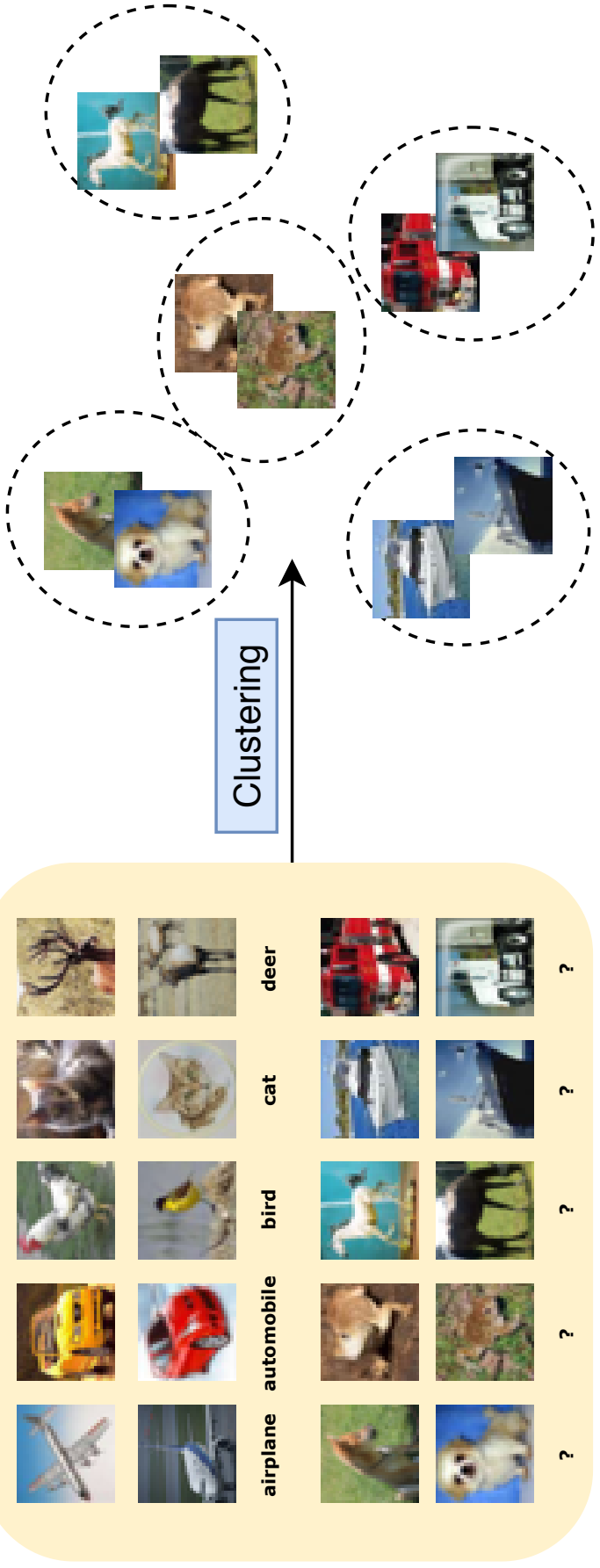


FIGURE 5.3: Dataset split setting of CIFAR-10 for novel categories clustering task. The first five classes, i.e., airplane, automobile, bird, cat and deer, are labeled data X_l with Y_l and the rest five classes, i.e., dog, frog, horse, ship and truck, are used as unlabeled data X_u.

(5.10).

$$
\begin{aligned}
w(t) &= \exp(-5(1-\frac{t}{\mathbb{T}})^2) \\
\lambda_{pl} &= A_{pl} \cdot w(t), \quad \beta(t) = A_{mse} \cdot w(t)
\end{aligned}
\tag{5.10}
$$

where t is the current training epoch and $\mathbb{T}$ is the total number of epochs in one period. $\mathbb{T}$ is set to be $\{50, 300, 80\}$, $A_{mse}$is set to be $\{5, 25, 50\}$ and $A_{pl} = 0.05$ for {CIFAR-10, CIFAR-100, SVHN}. The total number of training epochs is set to be $\{200, 400, 200\}$ and initial learning rate is set to be $\{0.05, 0.1, 0.05\}$ for {CIFAR-10, CIFAR-100, SVHN}. The illustration for the ramp-up function $w(t)$ for CIFAR-100 is shown in Figure 5.4.

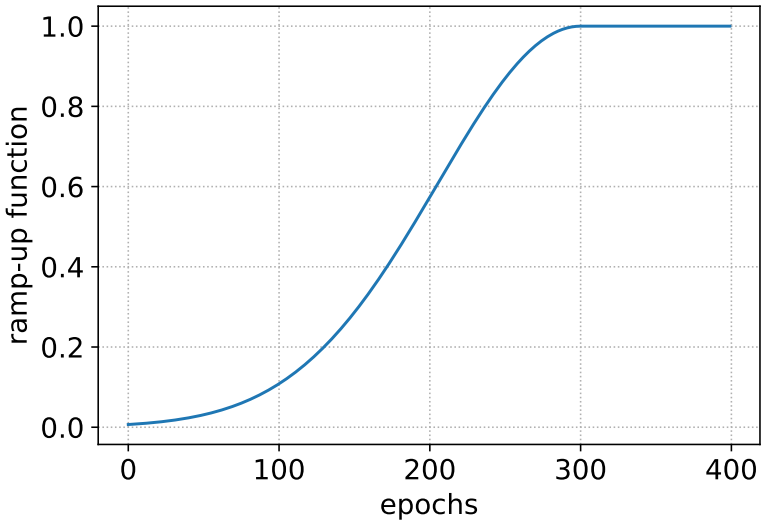


FIGURE 5.4: Ramp-up function for CIFAR-100

For all datasets, Stochastic Gradient Descent (SGD) optimizer is used for back-propagation and the learning rate follows a cosine annealing scheme [129]. The batch size is set to be $\{128, 256, 128\}$, temperature coefficient T in Equation (5.3) is set to be $\{1, 0.5, 1\}$, and threshold value τ in Equation (5.4) is set to be 0.01 for {CIFAR-10, CIFAR-100, SVHN}. The implementation details are given in Table 5.2.

TABLE 5.2: Implementation details of the experiments

	CIFAR-10	CIFAR-100	SVHN
Initial Learning Rate	0.05	0.1	0.05
Batch Size	128	256	128
Total Training Epochs	200	400	200
c_l	5	80	5
c_u	5	20	5
$\mathbb{T}$	50	300	80
A_{mse}	5	25	50
T	1	0.5	1
Network		ResNet-18	
Optimizer		SGD	
LR Annealing Scheme		cosine	
λ_{Rot}		0.8	
A_{pl}		0.05	
τ		0.01	

5.3.3 Evaluation

As the objective is to cluster X_u, the proposed algorithm is evaluated with clustering accuracy ACC metric in Equation (5.11):

$$ACC = \max_{perm \in P} \frac{1}{N-l} \sum \delta_{\widehat{y}_i, perm(y_i)} \tag{5.11}$$

where $\widehat{y}_i$ is the network prediction for unlabeled data $x_i \in X_u$, i.e., $\widehat{y}_i = \arg\max\{softmax(f_{c1}(\phi_i))\}$ and y_i is the ground-truth label. P denotes all possible permutations for the indices of c_u clusters. δ denotes Kronecker delta function.

5.3.4 Experiment Results

5.3.4.1 Clustering accuracy

The clustering capability of the proposed algorithm is compared with previous state-of-the-art methods in Table 5.3. JRLNCD [136] is from a recent research work in 2021, which is published after the proposed algorithm in this chapter. JRLNCD is included in Table 5.3 to further strengthen the comparisons with related research works. Seven methods listed in the bottom part of Table 5.3 including the proposed

algorithm have utilized self-supervised learning, whereas those listed in the upper part haven't. Methods in the bottom part of Table 5.3 generally perform better than those in the upper part, which indicates that it is beneficial to use self-supervised learning for the clustering task of concern. Among all methods that have utilized self-supervised learning, the proposed algorithm achieves the best results on all three datasets. The results of the proposed algorithm surpass the previous state-of-the-art results of AutoNovel [81] by 1.9%, 6.1% and 0.2% on all three datasets. Moreover, it can be observed that the proposed algorithm has more stable performance compared with AutoNovel [81]. For CIFAR-100 dataset, the clustering accuracy of the proposed algorithm has a standard deviation of 1.9%, which is much smaller than that of AutoNovel (4.2%).

TABLE 5.3: Clustering Accuracy on unlabeled training data (%). w/ SSL denotes self-supervised learning is used in the pre-training stage of the network. All reported results are using the same network architecture and averaged over ten runs.

Dataset	CIFAR-10	CIFAR-100	SVHN
k-means [137]	65.5 ± 0.0	56.6 ± 1.6	42.6 ± 0.0
KCL[82]	66.5 ± 3.9	14.3 ± 1.3	21.4 ± 0.6
MCL[83]	64.2 ± 0.1	21.3 ± 3.4	38.6 ± 10.8
DTC[92]	87.5 ± 0.3	56.7 ± 1.2	60.9 ± 1.6
k-means [137] w/ SSL	72.5 ± 0.0	56.3 ± 1.7	46.7 ± 0.0
KCL[82] w/ SSL	72.3 ± 0.2	42.1 ± 1.8	65.6 ± 4.9
MCL[83] w/ SSL	70.9 ± 0.1	21.5 ± 2.3	53.1 ± 0.3
DTC[92] w/ SSL	88.7 ± 0.3	67.3 ± 1.2	75.7 ± 0.4
AutoNovel [81]	91.7 ± 0.7	75.2 ± 4.2	95.2 ± 0.2
JRLNCD [136]*	93.4 ± 0.6	76.4 ± 2.8	-
Proposed	$\mathbf{93.6 \pm 0.6}$	$\mathbf{81.3 \pm 1.9}$	$\mathbf{95.4 \pm 0.3}$

$\star$ JRLNCD [136] is from a recent research work in 2021, which is published after the proposed algorithm in this chapter.

5.3.4.2 Mixed classification performance

Besides the clustering capability of the trained model on unlabeled training images, classifying new and unseen images from either "old" classes or "novel" classes is of great significance for practical applications. Therefore, the proposed algorithm is further evaluated together with other state-of-the-art ones on classification performance on testing images in these three datasets. In this case, the output from linear layer f_{c2} is evaluated, and the results are listed in Table 5.4 and Table 5.5. On CIFAR datasets in Table 5.4, the proposed algorithm achieves the best results for all categories. The proposed algorithm has outperformed others on both "old" and "novel" classes. It indicates that the proposed algorithm is not biased to either category, as the trained network generalizes equally well on both. For SVHN dataset in Table 5.5, the proposed algorithm achieves the best result on "old" classes, but AutoNovel [81] beats the proposed algorithm on "novel" classes. As classifying SVHN dataset is a relatively easier task compared with CIFAR datasets, the reason is deduced to be that over-fitting has degraded the performance. In AutoNovel [81], label information is only used to tune the last block of network while most parameters are frozen after self-supervised learning, which limits the learning capability but meanwhile largely avoids over-fitting. Comparing with AutoNovel [81], the proposed algorithm is able to capture more information as the whole network is updating during the training process, whereas over-fitting problem may arise when dealing with relatively easy datasets. Despite its suboptimal performance on "novel" classes, the proposed algorithm still achieves the best result on the whole testing dataset including all images from both "old" and "novel" classes.

TABLE 5.4: Mixed Classification Performance on CIFAR datasets (%). The trained network is evaluated on testing data for each dataset. "Old" refers to testing images from classes that have been trained by training images with label information, while "Novel" refers to testing images from classes that no label information is used during training. "All" refers to all testing images in the dataset.

Dataset	CIFAR-10			CIFAR-100		
Category	Old	Novel	All	Old	Novel	All
KCL[82] w/ SSL	79.3 ± 0.6	60.1 ± 0.6	69.8 ± 0.1	23.4 ± 0.3	29.4 ± 0.3	24.6 ± 0.2
MCL[83] w/ SSL	81.4 ± 0.4	64.8 ± 0.4	73.1 ± 0.1	18.2 ± 0.3	18.0 ± 0.1	18.2 ± 0.2
DTC[92] w/ SSL	58.7 ± 0.6	78.6 ± 0.2	68.7 ± 0.3	47.6 ± 0.2	49.1 ± 0.2	47.9 ± 0.2
AutoNovel [81]	90.6 ± 0.2	88.8 ± 0.2	89.7 ± 0.1	71.2 ± 0.1	56.8 ± 0.3	68.3 ± 0.1
Proposed	$\mathbf{94.6 \pm 0.4}$	$\mathbf{90.4 \pm 0.4}$	$\mathbf{92.5 \pm 0.2}$	$\mathbf{76.0 \pm 0.2}$	$\mathbf{66.7 \pm 1.8}$	$\mathbf{74.1 \pm 0.5}$

TABLE 5.5: Mixed Classification Performance on SVHN dataset (%)

Dataset	SVHN		
Category	Old	Novel	All
KCL[82] w/ SSL	90.3 ± 0.3	65.0 ± 0.5	81.0 ± 0.1
MCL[83] w/ SSL	94.0 ± 0.2	48.6 ± 0.3	77.2 ± 0.1
DTC[92] w/ SSL	90.5 ± 0.3	72.8 ± 0.2	84.0 ± 0.1
AutoNovel [81]	96.3 ± 0.1	$\mathbf{96.1 \pm 0.0}$	96.2 ± 0.1
Proposed	$\mathbf{97.6 \pm 0.2}$	94.8 ± 0.3	$\mathbf{96.5 \pm 0.1}$

5.3.4.3 Ablation Study

To evaluate each module of the proposed algorithm, ablation study is performed on CIFAR-100 dataset and the results are listed in Table 5.6.

Self-supervised learning: First of all, the bias between mixed classification performance on "old" classes and "novel" classes are largely alleviated due to the use of self-supervision, as the discrepancy, i.e., the classification accuracy gap between "old" classes and "novel" classes, 8.90% of proposed and 12.54% of variant (7) are lower than 18.96% of variant (1). Secondly, it is observed that self-supervised learning can always boost the performance on both clustering and mixed classification for "novel" classes by comparing variant (7) and proposed with variant (1). Lastly, the results have verified the argument that leveraging self-supervision simultaneously with pairwise similarity information can further boost model's learning capability as the proposed algorithm outperforms variant (7) by a significant margin on all evaluation metrics. The difference between the proposed algorithm and variant (7) is the way how self-supervision is utilized: self-supervision and pairwise similarity information are jointly training the network in the proposed algorithm while self-supervision is only used for pre-training in variant (7).

The learning curves for the proposed algorithm and variant (7) are presented in Figure 5.5. It can be observed that at the initial training stage, variant (7) outperforms the other, which is reasonable as variant (7) is fine-tuned with labeled data after pre-training. As the training continues, the lead of variant (7) shrinks quickly, especially during the last 150 epochs. On the contrary, the clustering accuracy of the proposed algorithm is improving significantly with more training epochs. As

the whole network is updating in the proposed algorithm, the learning capability of the model is better than the one in variant (7). For variant (7), error signals only back-propagate to the final convolutional block and the linear layer (the parameters for the rest of the network are frozen) to avoid the forgetting issue, which greatly limits the learning capability of the model.

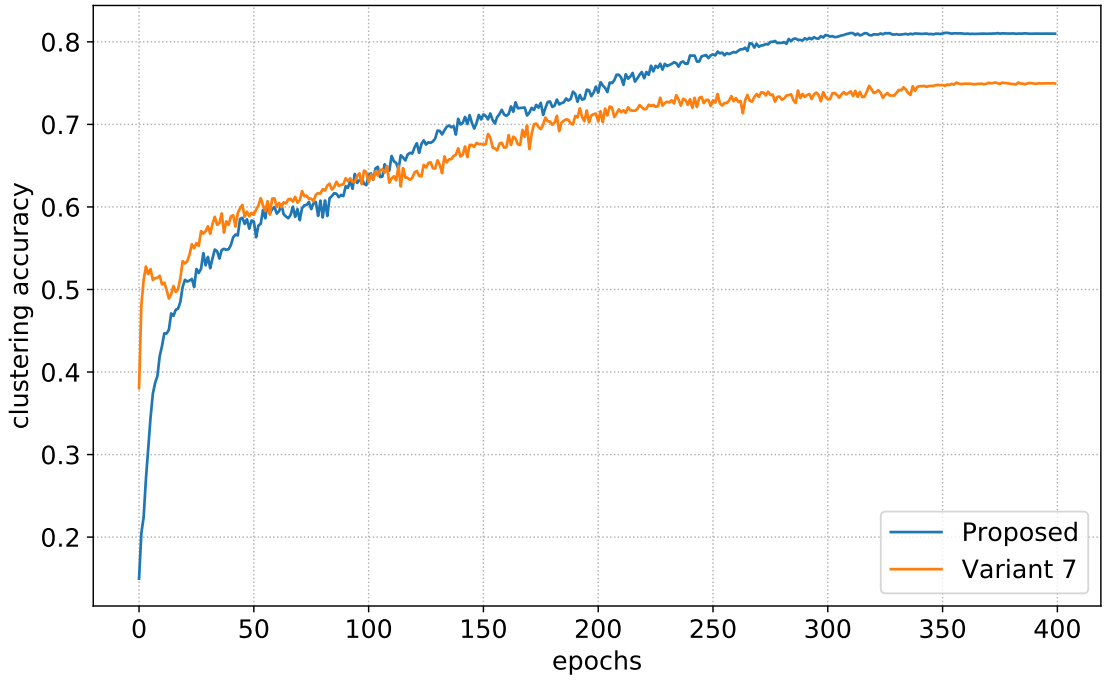


FIGURE 5.5: Ablation study: Self-supervised learning. For variant (7), the pre-training and fine-tuning schemes in [81] are followed.

TABLE 5.6: Ablation Study (%)

		Clustering	**Mixed**	**Classification**	
	Ablation	ACC	Old	Novel	All
	Proposed	**80.99**	75.90	**67.00**	**74.09**
Variant (1)	Proposed without $\mathcal{L}_{Rot}$	70.98	76.81	57.85	72.92
Variant (2)	Proposed without $\mathcal{L}_{MSE}$	79.97	75.09	65.45	73.16
Variant (3)	Proposed without $\mathcal{L}_{l}$	51.00	4.11	39.45	8.34
Variant (4)	Proposed without $\mathcal{L}_{u}$	73.49	**77.75**	47.90	62.20
Variant (5)	Proposed without $\mathcal{L}_{BCE}$	33.51	72.01	31.40	63.52
Variant (6a)	Proposed with cosine similarity†	75.56	75.39	63.45	72.96
Variant (6b)	Proposed with rank statistics⋆	74.14	75.59	61.75	72.79
Variant (7)	Proposed with SSL as initialization*	74.95	71.29	58.75	68.75

† The proposed conditional probability is replaced with cosine similarity, i.e., $s_{ij} = \frac{\phi_i \cdot \phi_j}{||\phi_i|| \times ||\phi_j||}$, as similarity measure for the calculation of $\mathcal{L}_{BCE}$.

⋆ The proposed conditional probability is replaced with rank statistics in [81].

* The pre-training and fine-tuning schemes in [81] are followed, i.e., self-supervised learning is only utilized independently for the network initialization.

Pairwise similarity learning: Firstly, it is observed that pairwise similarity learning is the most crucial module for novel categories learning as only 33.51% and 31.40% accuracies have been achieved by variant (5) for clustering and mixed classification on "novel" classes respectively, which are the worst results among all variants. Secondly, the use of conditional probability as pairwise similarity measurement can capture such pairwise relationship much more accurately in the feature space than commonly-used cosine similarity. It can be observed by comparing proposed with variant (6a) that the proposed algorithm has outperformed variant (6a) by significant margins of 5.43% and 3.55% for clustering and mixed classification on "novel" classes respectively. Moreover, the rank statistics proposed in [81] are also compared with the proposed one, as shown in variant (6b). The result indicates that the rank statistics are less robust to noises in the feature space trained with the proposed learning framework, as the clustering and mixed classification accuracy of variant (6b) are less than 75% and 62% on "novel" classes respectively, which are even poorer than those of variant (6a).

The learning curves for the proposed algorithm, variant (5), variant (6a) and variant (6b) are presented in Figure 5.6. First of all, the effectiveness of pairwise similarity learning is further verified as the clustering accuracy of variant (5) remains at pretty low level compared with others. Secondly, it can be observed that the learning curves for the proposed algorithm, variant (6a) and variant (6b) almost overlap with each other during the first 100 epochs and they can only be differentiated clearly during later training epochs. The reason is believed to be that the noise in the feature space during early training stage is relatively high such that different pseudo label construction schemes perform similarly and the superiority of effective construction scheme can only be revealed during late training stage as noise is reduced after certain training epochs.

Mixed Classification: By comparing variant (3) and (4) with the proposed algorithm, it can be noticed that mixed classification plays an important role for both "old" and "novel" classes learning. Without $\mathcal{L}_u$, the pseudo labels inferred by output from f_{c1} are not utilized. Both clustering accuracy and mixed classification performance on "novel" classes have dropped and the trained network is largely biased to "old" classes during mixed classification, as the highest mixed classification accuracy on "old" classes has been achieved but it only reaches 47.90% accuracy

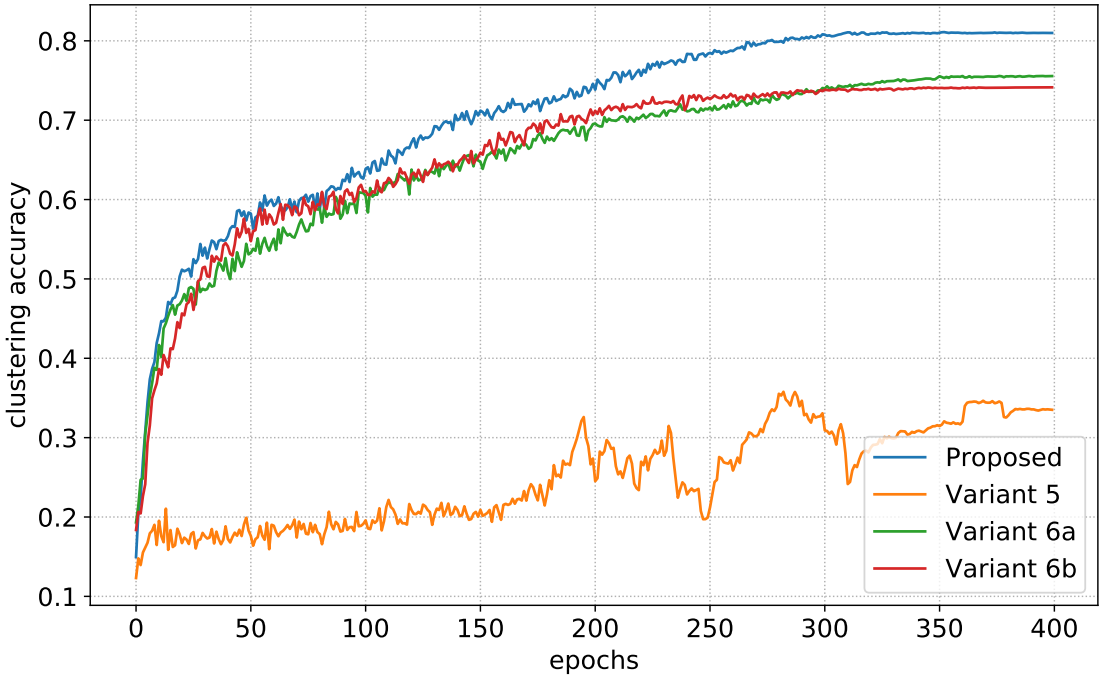


FIGURE 5.6: Ablation study: Pairwise similarity learning. For variant (5), $\mathcal{L}_{BCE}$ is not included for the model optimization. For variant (6a), the proposed conditional probability is replaced with cosine similarity. For variant (6b), the proposed conditional probability is replaced with rank statistics in [81].

on "novel" classes for variant (4). Without $\mathcal{L}_l$, i.e., the ground-truth label information for "old" classes is removed, all evaluation metrics have dropped a lot. The effectiveness of consistency loss $\mathcal{L}_{MSE}$ has been demonstrated by comparing the proposed algorithm with variant (2).

The learning curves for the proposed algorithm, variant (2), variant (3) and variant (4) are presented in Figure 5.7. As the loss term $\mathcal{L}_u$ contributed by pseudo labels of X_u and the consistency loss term $\mathcal{L}_{MSE}$ have weight coefficient in the form of ramp-up functions (as shown in Figure 5.4), the effectiveness of the two loss terms is gradually demonstrated along the training process (corresponds to the three learning curves of the proposed algorithm, variant (2) and variant (4) in Figure 5.7). For the learning curve of variant (3), although the clustering accuracy is improving with more training epochs, the performance is left behind by others with a significant margin. Such results indicate that the use of prior knowledge, i.e., the labeled data for novel categories learning, is beneficial for clustering.

To summarize, the proposed algorithm achieves best results on both clustering and

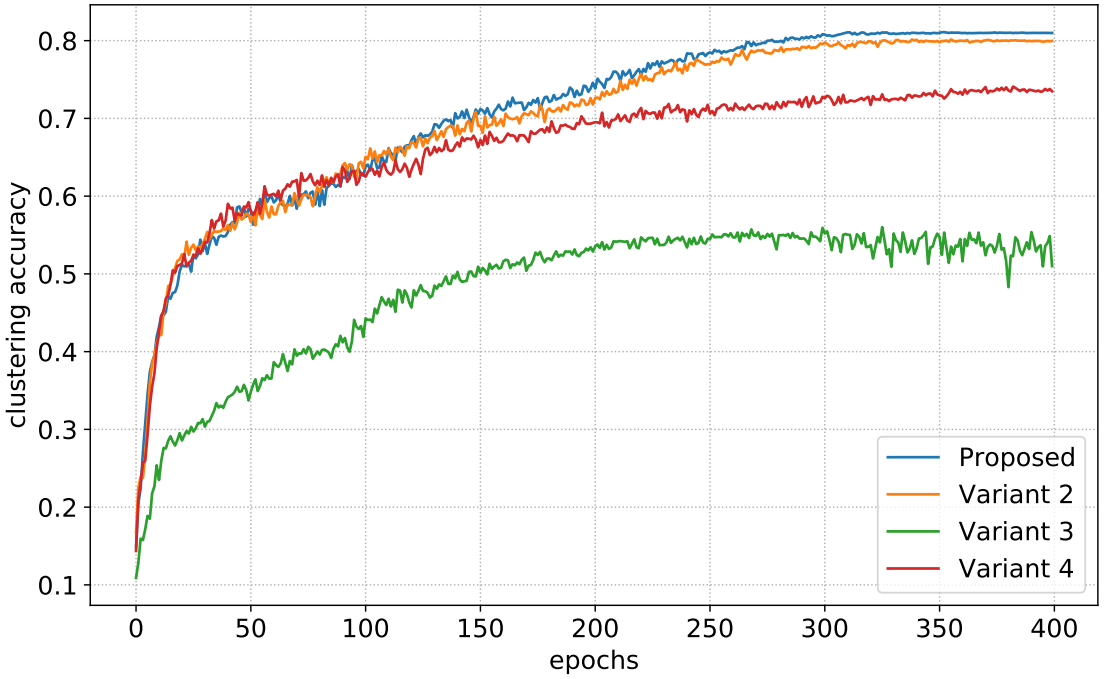
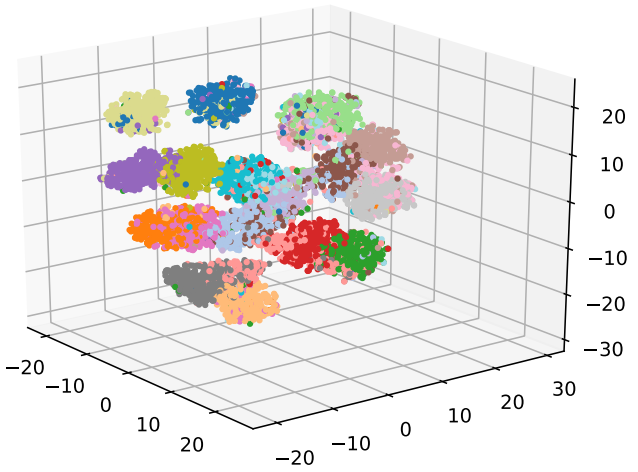


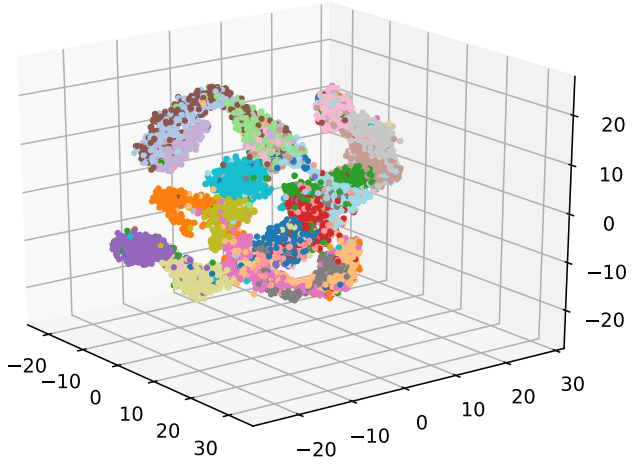
FIGURE 5.7: Ablation study: Mixed Classification. For variant (2), $\mathcal{L}_{MSE}$ is not included for the model optimization. For variant (3), $\mathcal{L}_l$ is not included for the model optimization. For variant (4), $\mathcal{L}_u$ is not included for the model optimization.

mixed classification tasks on "novel" classes. Although the highest mixed classification accuracy on "old" classes is achieved by variant (4), the price is to sacrifice the classification capability on "novel" classes. The effectiveness of proposed pairwise similarity pseudo label construction scheme has also been demonstrated by the comparisons with variant (6a) and (6b). Moreover, the superiority of the proposed end-to-end training via auxiliary self-supervision is clearly validated by the comparison with variant (7).

From Figure 5.8 to Figure 5.10, 3D t-SNE visualizations for feature vectors ϕ of training images from novel classes are demonstrated for models with different training schemes. Without $\mathcal{L}_{Rot}$, i.e., only pairwise similarity information and ground-truth label information are supervising the model, feature vectors belonging to different classes are not well separated and some parts are connected with each other as shown in Figure 5.8. For the case when pairwise similarity is measured by cosine similarity or rank statistics, there are more feature vectors clustered to the wrong groups compared with the proposed algorithm as shown in Figure 5.9 and Figure 5.10, indicating using conditional probability as pairwise similarity criterion is more accurate than cosine similarity and rank statistics.

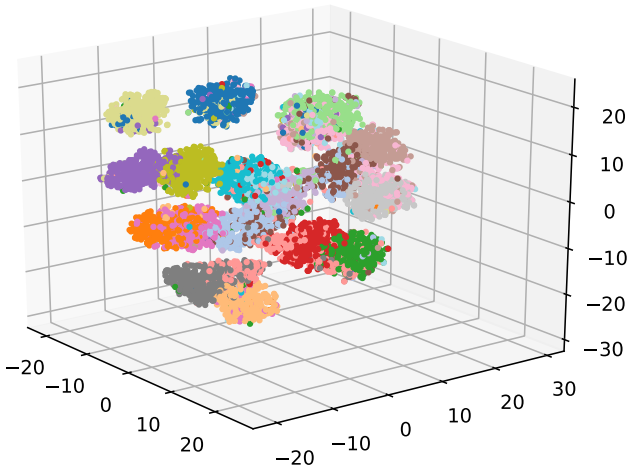


(a) The proposed algorithm

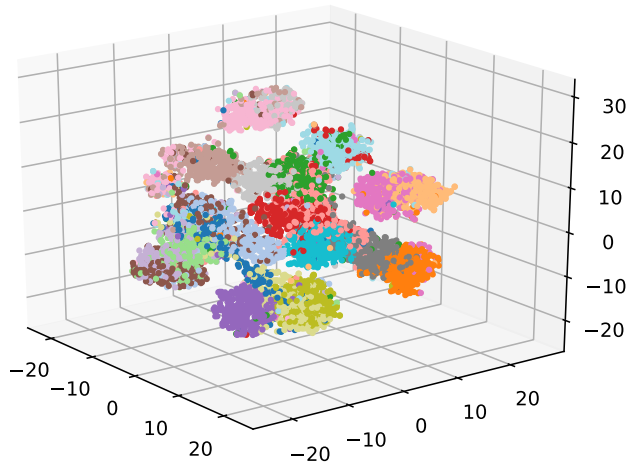


(b) The proposed algorithm w/o $\mathcal{L}_{Rot}$

FIGURE 5.8: 3D t-SNE visualizations of embeddings ϕ for CIFAR-100 unlabeled training data under different training schemes (A). Each single point represents one unlabeled training image and different colors correspond to ground-truth labels. Recommend to view in color version.

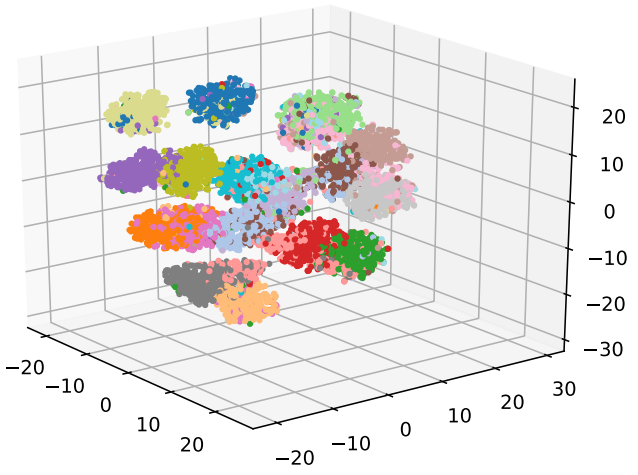


(a) The proposed algorithm

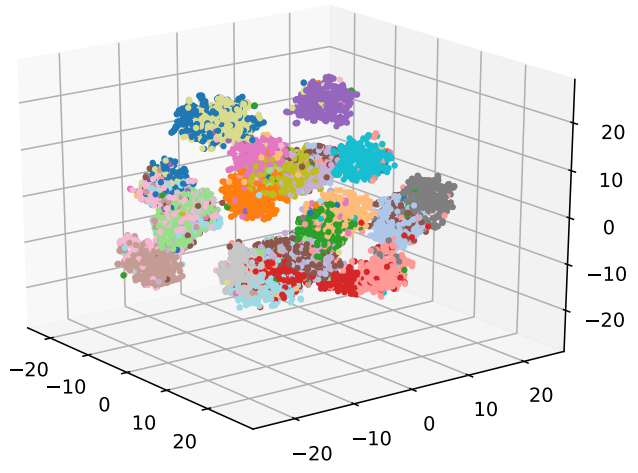


(b) The proposed algorithm w/ cosine similarity

FIGURE 5.9: 3D t-SNE visualizations of embeddings ϕ for CIFAR-100 unlabeled training data under different training schemes (B). Each single point represents one unlabeled training image and different colors correspond to ground-truth labels. Recommend to view in color version.



(a) The proposed algorithm



(b) The proposed algorithm w/ rank statistics

FIGURE 5.10: 3D t-SNE visualizations of embeddings ϕ for CIFAR-100 unlabeled training data under different training schemes (C). Each single point represents one unlabeled training image and different colors correspond to ground-truth labels. Recommend to view in color version.

5.4 Summary

This chapter presents the third work of this thesis, i.e., a new end-to-end novel visual categories learning algorithm, which has addressed the learning difficulty of novel categories due to over-fitting problems on known categories. In Section 5.1, previous research works are reviewed and motivations of this work are discussed. Two major issues inherent in prior relevant methods reported in literature are limitation of learning capability due to the difficulty for end-to-end training and pairwise similarity pseudo labels' vulnerability to noise and bias. In Section 5.2, the end-to-end novel visual categories learning algorithm where self-supervision is utilized as an auxiliary task is proposed. To address the limitations of previous methods, surrogate labels from self-supervision are proposed to be used together with pairwise similarity pseudo labels to train the model simultaneously, which is expected to provide extra supervision and regularization for end-to-end learning. Moreover, robust local data structure information in noisy feature space is proposed to be exploited for the construction of pairwise similarity pseudo labels. As such, the quality of pairwise pseudo labels can be further improved. The ablation study in Section 5.3 has demonstrated the contribution from each single module and has shown the efficacy of proposed designs. Experiments have been conducted on three commonly-used visual datasets i.e., CIFAR-10, CIFAR-100 and SVHN. The results have indicated the effectiveness of the proposed algorithm as the proposed algorithm has outperformed previous state-of-the-art algorithms significantly.

Chapter 6

Conclusions and Future Works

Chapter 6 gives a summary for this thesis based on Chapter 3, Chapter 4 and Chapter 5 in Section 6.1 , followed by a discussion on future research directions in Section 6.2.

6.1 Conclusions

This thesis investigates the over-fitting problem in deep feature learning for visual recognition under different frameworks. Figure 6.1 shows the overall structure for the main contents in this thesis. Starting from the over-fitting problem, two major perspectives are defined, i.e., the model architecture and data annotation. The first work in Chapter 3 is focusing on the over-fitting problem due to model architecture. To be more specific, the over-fitting problem arising from over-deepened CNN model as the feature extractor for ELM classifier is addressed in Chapter 3. The second and the third work are focusing on the over-fitting problem due to the lack of data annotation. The second work in Chapter 4 has addressed the over-fitting problem of feature mapping during semi-supervised learning, where a large amount of the data are unlabeled. The third work in Chapter 5 has explored the novel visual categories learning, i.e., the categories of labeled data and unlabeled data are disjoint from each other, which highly likely gives rise to overfitting on labeled categories.

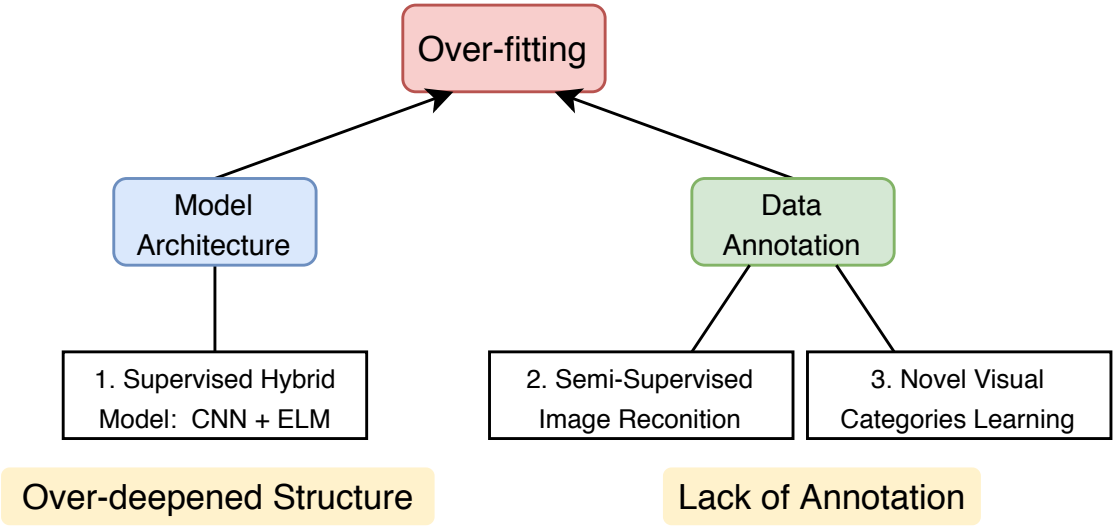


FIGURE 6.1: Overall structure of this thesis

The main conclusions and insights of this thesis are summarized as follows:

(1) *CNN model as a feature extractor for ELM classifier on image classification should be deep and also wide.*

ELM is an efficient and powerful classifier, as reviewed in Section 2.1. For solving complex visual tasks, CNN model as a deep feature extractor is a perfect choice

to combine with ELM classifier to form a hybrid model. The overfitting problem addressed in Chapter 3 suggests that such hybrid model should not be constructed blindly as ELM classifier will over-fit to training data when the fed-in features are learned by over-deepened CNN models. The proposed DW-ELM in Chapter 3 aims to alleviate such over-fitting problem by focusing on another network dimension, i.e., width of the model. The proposed DW-ELM utilizes the good generalization capability of widened residual network to remedy the over-fitting problem of features extracted by deepened residual networks. Extensive experiments have been conducted on five benchmark datasets to explore the effects of network depth and width on ELM classifier for image classification tasks and the empirical results have verified that wider network architecture design is beneficial for the performance of CNN-ELM hybrid models.

(2) *Self-supervised learning as a feature learning scheme is beneficial for label propagation in feature space on semi-supervised visual recognition task.*

For pseudo-labeling based semi-supervised learning, label propagation is one transductive method to provide information flow from labeled data to unlabeled data. One main objective is to minimize noise in such information flow. For image data, propagating label information in latent feature space is more helpful to avoid redundant and noisy information, compared with that in raw data space. Therefore, learning a good feature mapping is crucial for the final performance of label propagation. Chapter 4 have discussed the limitation of previous research works, i.e., the biased and over-fitted feature mapping due to limited number of labeled data. The proposed algorithm in Chapter 4 has well addressed such problem by exploiting the use of self-supervised learning for feature learning. Instead of learning feature mapping by labeled data, the proposed algorithm takes advantages of RotNet, a self-supervised learning framework, to train a feature mapping with both labeled and unlabeled data. By doing so, the learned feature mapping is capable of studying from all data such that over-fitting problem arising from scarce data annotation is greatly alleviated. The ablation study in Section 4.3.3.1 have empirically demonstrated the performance gain from self-supervised feature learning scheme, which shows very promising results (e.g., more than 14% improvement has been achieved with only 500 labeled data for CIFAR-10 dataset).

(3) *Local geometry preserving during graph construction provides more complete and meaningful information flow from labeled data to unlabeled data, which can*

improve the quality of pseudo labels from label propagation.

The other main objective of label propagation for pseudo-labeling based semi-supervised learning is to minimize information loss during the label knowledge transfer process, i.e., flow of information from labeled data to unlabeled data. The information flow is characterized by the similarity graph of the data, which defines the relationship between data points. Previous methods commonly ignore the local geometry information, which results in incomplete and inaccurate label propagation. The proposed algorithm in Chapter 4 preserves the local geometry information by employing the reconstruction idea in LLE, i.e., to rebuild the data/feature vector by weighted sum of its neighbors. The ablation study in Section 4.3.3.1 suggests that preserving local geometry is capable of boosting the quality of pseudo labels if the geometry information is measured in the feature space learned with self-supervised learning. Such synergistic effect indicates that the two main objectives are key components that determine the final quality of the proposed semi-supervised learning algorithm, and neither one is dispensable.

(4) *Joint and simultaneous supervision from surrogate signals of self supervision and pairwise similarity information makes end-to-end training feasible, which can boost the overall learning capability of the model for novel visual categories learning.*

Novel visual categories learning has been introduced and explored in Chapter 5. The task handles visual recognition problem where labeled data and unlabeled data belong to disjoint classes. Previous methods are limited by two main problems, i.e., the over-fitting problem on labeled classes and the restricted learning capability due to multi-stage training framework. The proposed algorithm in Chapter 5 aims to achieve end-to-end training and give strong regularization to the model by introducing an auxiliary task via self-supervision. The proposed algorithm exploits the effect of joint training, i.e., the network is supervised simultaneously by pairwise similarity information in feature space and surrogated labels from self-supervision (e.g., the rotation angle of the transformed image). The proposed algorithm has been validated on three visual benchmark datasets in Section 5.3. The experiment results indicate that: 1) The utilization of surrogate labels from self-supervision can largely alleviate the over-fitting problem on labeled classes for novel categories learning. 2) The proposed end-to-end training framework outperforms the multi-stage training framework. The effectiveness of the proposed algorithm has been demonstrated in Section 5.3 by comparing with previous state-of-the-art methods.

(5) *Local property in feature space is robust to noises, which helps to capture pairwise data relationship more accurately.*

Besides the surrogate labels from self-supervision, pairwise similarity pseudo labels are the main supervising signals for novel categories learning. The pairwise pseudo labels are constructed by comparing the data points in feature space, which is vulnerable to noises due to lack of annotation for images from novel categories. Local data structure is enforced in the proposed algorithm in order to capture more accurate data relationship. The experiments conducted in Section 5.3.4.3 compares the effects of different pairwise pseudo label construction schemes and the empirical results indicate that the proposed scheme where local property in the feature space is emphasized can outperform others by a significant margin.

6.2 Future Works

After investigation of various topics in this thesis for countering over-fitting problems under different learning frameworks, some related research directions are worth for further exploration.

1. In Chapter 3, the effect of width and depth of CNN model as a feature extractor on ELM classifier is analyzed. The insight given is that CNN model as a feature extractor for ELM image classification should be deep and also wide. However, such claim is vague for practical deployment as a quantitative metric for optimal CNN model architecture is not given. The empirical results in Section 3.3.3.3 indicate that width and depth of CNN model should be balanced carefully for optimal performance. Therefore, a quantitative metric, which takes both width and depth of CNN model into consideration, deserves further research on.

   For the proposed DW-ELM hybrid model in Chapter 3, CNN model and ELM classifier is training separately such that the superiority of ELM classifier in the early training phase cannot benefit back to the CNN model. Multi-head CNN architecture where both fully-connected layer and ELM classifier are attached to the last convolutional block is one possible direction. The double-head architecture is capable of back-propagating error signals via the fully-connected layer and meanwhile the ELM classifier after the calculation of hidden layer output weight can be treated as a SLFN with fixed weight parameters. As shown in Equation (6.1), the objective function is to minimize the loss from fully-connected layer and ELM classifier jointly. By doing so, end-to-end learning framework of ELM classifier and CNN model can be achieved.

   $$\min_{\theta} \sum L(f_{\theta}(x_i), y_i) + \sum L(\mathbf{h}(\phi_i)\boldsymbol{\beta}, y_i) \tag{6.1}$$

2. For the proposed algorithm in Chapter 4, the local geometry preserving is achieved by the use of reconstruction idea, i.e., the feature vector is reconstructed by its neighbors in the feature space. According to Equation (4.5), the calculation of similarity weight matrix W_i requires to solve a quadratic programming problem. For industrial applications, massive datasets with over millions and even billions of data are commonly used [138]. Even though there are existing efficient solvers, e.g., `CVXOPT` [126], the computational cost

is expected to be pretty high for massive datasets, e.g., ImageNet [15] and Places 205 [139]. Moreover, the collection of k nearest neighbors $\mathcal{N}(x_i)$ for each data point x_i is required for the calculation in Equation (4.5), which poses computational burden for large datasets. In Section 4.2.3.3, the corresponding computational complexity is analyzed to be $\mathcal{O}(n^2+nk^3)$. Therefore, to preserve the local geometry and meanwhile lower the computational requirement, e.g., to avoid the quadratic programming problem during each training epoch and exploit the use of approximate nearest neighbors (ANN) search algorithms, need to be addressed for large datasets.

3. The novel visual categories learning is discussed in Chapter 5. The proposed algorithm utilizes self-supervision as an auxiliary task to overcome the overfitting problem. The motivation behind this work is to alleviate the bias arising from lack of annotations for images from novel categories. According to [81], self-supervised learning does not demonstrate its effectiveness on ImageNet and OmniGlot [140] for novel categories learning. The reason behind is claimed to be that the diversity and abundance of those large datasets have already reduced the bias between different categories. Therefore, the focus of novel categories learning problem for such kind of datasets may not be on eliminating the bias between various categories. To further explore effective algorithms of novel visual categories learning for abundant and diverse datasets is one future research direction.

   Moreover, the novel categories learning problem discussed in this thesis assumes that the number of novel categories is known, which may not be satisfied for realistic applications. In a more challenging setting, the number of novel categories/number of clusters for unlabeled data is unknown and needs to be approximated. For the e-commerce platform example given in Section 5.1, the total number of unseen categories is highly likely to be unknown. How to determine the number of clusters for novel categories learning is an important question worth further research on. According to the research work in [92], approximating the number of novel categories with prior information in labeled data is one possible direction. In [92], probe dataset X_{probe} consisting of unlabeled data X_u and a subset of labeled data X_l is constructed. Constrained k-means clustering is performed on X_{probe} with different k and the optimal value of k is determined by the clustering accuracy on labeled data in X_{probe}.

List of Publications

Journal Articles

- **Yuanyuan Qing**, Yijie Zeng, Yue Li, and Guang-Bin Huang, "Deep and wide feature based extreme learning machine for image classification", en, *Neurocomputing*, vol. 412, pp. 426–436, Oct. 2020. DOI: 10.1016/j.neucom.2020.06.110.

- **Yuanyuan Qing**, Yijie Zeng, and Guang-Bin Huang, "Label propagation via local geometry preserving for deep semi-supervised image recognition", *Neural Networks*, 2020, Under review.

- **Yuanyuan Qing**, Yijie Zeng, Qi Cao, and Guang-Bin Huang, "End-to-end novel visual categories learning via auxiliary self-supervision", *Neural Networks*, vol. 139, pp. 24–32, 2021.

- Yijie Zeng, Jichao Chen, Yue Li, **Yuanyuan Qing**, and Guang-Bin Huang, "Clustering via Adaptive and Locality-constrained Graph Learning and Unsupervised ELM", en, *Neurocomputing*, vol. 401, pp. 224–235, Aug. 2020. DOI: 10.1016/j.neucom.2020.03.045.

- Yue Li, Yijie Zeng, **Yuanyuan Qing**, and Guang-Bin Huang, "Learning local discriminative representations via extreme learning machine for machine fault diagnosis", en, *Neurocomputing*, vol. 409, pp. 275–285, Oct. 2020. DOI: 10.1016/j.neucom.2020.05.021.

Bibliography

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks", en, *Pattern Recognition*, vol. 77, pp. 354–377, May 2018. DOI: 10.1016/j.patcog.2017.10.013.

[2] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network", in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed., Morgan-Kaufmann, 1990, pp. 396–404.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[4] M. Duan, K. Li, C. Yang, and K. Li, "A hybrid deep learning CNN–ELM for age and gender classification", en, *Neurocomputing*, vol. 275, pp. 448–461, Jan. 2018. DOI: 10.1016/j.neucom.2017.08.062.

[5] F. Gurpinar, H. Kaya, H. Dibeklioglu, and A. Salah, "Kernel ELM and CNN Based Facial Age Estimation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 80–86.

[6] A. Kölsch, M. Z. Afzal, M. Ebbecke, and M. Liwicki, "Real-Time Document Image Classification Using Deep CNN and Extreme Learning Machines", in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov. 2017, pp. 1318–1323. DOI: 10.1109/ICDAR.2017.217.

[7] Z.-X. Yang, L. Tang, K. Zhang, and P. K. Wong, "Multi-View CNN Feature Aggregation with ELM Auto-Encoder for 3D Shape Recognition", en, *Cognitive Computation*, vol. 10, no. 6, pp. 908–921, Dec. 2018. DOI: 10.1007/s12559-018-9598-1.

[8] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications", en, *Neurocomputing*, Neural Networks, vol. 70, no. 1, pp. 489–501, Dec. 2006. DOI: 10.1016/j.neucom.2005.12.126.

[9] D. S. Park, J. Sohl-Dickstein, Q. V. Le, and S. L. Smith, "The Effect of Network Width on Stochastic Gradient Descent and Generalization: An Empirical Study", *arXiv:1905.03776 [cs, stat]*, May 2019. arXiv: 1905.03776 [cs, stat].

[10] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent", *arXiv:1902.06720 [cs, stat]*, Dec. 2019. arXiv: 1902.06720 [cs, stat].

[11] S. Zagoruyko and N. Komodakis, "Wide Residual Networks", *arXiv:1605.07146 [cs]*, Jun. 2017. arXiv: 1605.07146 [cs].

[12] Z. Wu, C. Shen, and A. van den Hengel, "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition", en, *Pattern Recognition*, vol. 90, pp. 119–133, Jun. 2019. DOI: 10.1016/j.patcog.2019.01.006.

[13] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep Neural Networks as Gaussian Processes", *arXiv:1711.00165 [cs, stat]*, Mar. 2018. arXiv: 1711.00165 [cs, stat].

[14] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "The role of over-parametrization in generalization of neural networks", in *International Conference on Learning Representations*, Sep. 2018.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[16] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, en, ser. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006.

[17] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a Proxy Task for Visual Understanding", *arXiv:1703.04044 [cs]*, Aug. 2017. DOI: 10.1109/cvpr.2017.96. arXiv: 1703.04044 [cs].

[18] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization", en, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9907, Cham: Springer International Publishing, 2016, pp. 649–666. DOI: 10.1007/978-3-319-46487-9_40.

[19] M. Noroozi and P. Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles", *arXiv:1603.09246 [cs]*, Aug. 2017. DOI: 10.1007/978-3-319-46466-4_5. arXiv: 1603.09246 [cs].

[20] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations", *arXiv:1803.07728 [cs]*, Mar. 2018. arXiv: 1803.07728 [cs].

[21] L. Saul and S. Roweis, "An introduction to locally linear embedding", *Journal of Machine Learning Research*, vol. 7, Jan. 2001.

[22] J. Cook, I. Sutskever, A. Mnih, and G. E. Hinton, "Visualizing Similarity Data with a Mixture of Maps", in *AISTATS*, 2007.

[23] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[24] G.-B. Huang, "What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle", en, *Cognitive Computation*, vol. 7, no. 3, pp. 263–278, Jun. 2015. DOI: 10.1007/s12559-015-9333-0.

[25] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, Apr. 2012. DOI: 10.1109/TSMCB.2011.2168604.

[26] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes", en, *Neurocomputing*, Neural Networks: Algorithms and Applications, vol. 71, no. 4, pp. 576–583, Jan. 2008. DOI: 10.1016/j.neucom.2007.07.025.

[27] N.-y. Liang, G.-b. Huang, S. Member, P. Saratch, S. Member, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks", *IEEE Trans. Neural Netw*, pp. 1411–1423, 2006.

[28] L. Kasun, H. Zhou, G.-B. Huang, and C.-M. Vong, "Representational Learning with ELMs for Big Data", *IEEE Intelligent Systems*, vol. 28, pp. 31–34, Nov. 2013.

[29] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification", en, *Neurocomputing*, Artificial Brains, vol. 74, no. 1, pp. 155–163, Dec. 2010. DOI: 10.1016/j.neucom.2010.02.019.

[30] Y. Zeng, X. Xu, Y. Fang, and K. Zhao, "Traffic Sign Recognition Using Deep Convolutional Networks and Extreme Learning Machine", en, in *Intelligence Science and Big Data Engineering. Image and Video Data Engineering*, X. He, X. Gao, Y. Zhang, Z.-H. Zhou, Z.-Y. Liu, B. Fu, F. Hu, and Z. Zhang, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 272–280. DOI: 10.1007/978-3-319-23989-7_28.

[31] Y. Li, C. K. L. Lekamalage, T. Liu, P. Chen, and G. Huang, "Learning Representations With Local and Global Geometries Preserved for Machine Fault Diagnosis", *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2360–2370, Mar. 2020. DOI: 10.1109/TIE.2019.2905830.

[32] D. Cui, G. Zhang, W. Han, L. Lekamalage Chamara Kasun, K. Hu, and G.-B. Huang, "Compact Feature Representation for Image Classification Using ELMs", in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1015–1022.

[33] T. Liu, C. K. Liyanaarachchi Lekamalage, G.-B. Huang, and Z. Lin, "Extreme Learning Machine for Joint Embedding and Clustering", en, *Neurocomputing*, Hierarchical Extreme Learning Machines, vol. 277, pp. 78–88, Feb. 2018. DOI: 10.1016/j.neucom.2017.01.115.

[34] C. Cortes and V. Vapnik, "Support-vector networks", en, *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. DOI: 10.1007/BF00994018.

[35] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv:1409.1556 [cs]*, Apr. 2015. arXiv: 1409.1556 [cs].

[36] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks", en, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, Jun. 2011, pp. 315–323.

[37] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *arXiv:1502.03167 [cs]*, Mar. 2015. arXiv: 1502.03167 [cs].

[38] S. Wiesler and H. Ney, "A Convergence Analysis of Log-Linear Training", en, *Advances in Neural Information Processing Systems*, vol. 24, pp. 657–665, 2011.

[39] C. M. Bishop, *Neural Networks for Pattern Recognition*, English, Illustrated edition. Oxford : New York: Oxford University Press, USA, Jan. 1996.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *arXiv:1409.0575 [cs]*, Jan. 2015. arXiv: 1409.0575 [cs].

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper With Convolutions", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks", en, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 630–645. DOI: 10.1007/978-3-319-46493-0_38.

[43] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits", en, *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012. DOI: 10.1016/j.patcog.2011.09.021.

[44] J. Kang, Y.-J. Park, J. Lee, S.-H. Wang, and D.-S. Eom, "Novel Leakage Detection by Ensemble CNN-SVM and Graph-Based Localization in Water Distribution Systems", *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4279–4289, May 2018. DOI: 10.1109/TIE.2017.2764861.

[45] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples", *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, Dec. 2006.

[46] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning", *arXiv:1704.03976 [cs, stat]*, Jun. 2018. DOI: 10.1109/tpami.2018.2858821. arXiv: 1704.03976 [cs, stat].

[47] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised Data Augmentation for Consistency Training", *arXiv:1904.12848 [cs, stat]*, Sep. 2019. arXiv: 1904.12848 [cs, stat].

[48] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation Consistency Training for Semi-Supervised Learning", *arXiv:1903.03825 [cs, stat]*, May 2019. DOI: 10.24963/ijcai.2019/504. arXiv: 1903.03825 [cs, stat].

[49] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "MixMatch: A Holistic Approach to Semi-Supervised Learning", *arXiv:1905.02249 [cs, stat]*, Oct. 2019. arXiv: 1905.02249 [cs, stat].

[50] S. Park, J.-K. Park, S.-J. Shin, and I.-C. Moon, "Adversarial Dropout for Supervised and Semi-supervised Learning", *arXiv:1707.03631 [cs]*, Sep. 2017. arXiv: 1707.03631 [cs].

[51] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning", *arXiv:1610.02242 [cs]*, Mar. 2017. arXiv: 1610.02242 [cs].

[52] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", *arXiv:1703.01780 [cs, stat]*, Apr. 2018. arXiv: 1703.01780 [cs, stat].

[53] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples", *arXiv:1412.6572 [cs, stat]*, Mar. 2015. arXiv: 1412.6572 [cs, stat].

[54] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average", *arXiv:1806.05594 [cs, stat]*, Feb. 2019. arXiv: 1806.05594 [cs, stat].

[55] D.-H. Lee, "Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks", *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, Jul. 2013.

[56] Y. Grandvalet and Y. Bengio, "Semi-supervised Learning by Entropy Minimization", in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., MIT Press, 2005, pp. 529–536.

[57] W. Shi, Y. Gong, C. Ding, Z. Ma, X. Tao, and N. Zheng, "Transductive Semi-Supervised Deep Learning Using Min-Max Features", en, in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11209, Cham: Springer International Publishing, 2018, pp. 311–327. DOI: 10.1007/978-3-030-01228-1_19.

[58] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth Neighbors on Teacher Graphs for Semi-Supervised Learning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8896–8905.

[59] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label Propagation for Deep Semi-Supervised Learning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5070–5079. DOI: 10.1109/cvpr.2019.00521.

[60] X. Zhu and Z. Ghahramani, "Learning from Labeled and Unlabeled Data with Label Propagation", Tech. Rep., 2002.

[61] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency", in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds., MIT Press, 2004, pp. 321–328.

[62] Fei Wang and Changshui Zhang, "Label Propagation through Linear Neighborhoods", en, *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, Jan. 2008. DOI: 10.1109/TKDE.2007.190672.

[63] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-Supervised Video Representation Learning With Odd-One-Out Networks", Jul. 2017. DOI: 10.1109/CVPR.2017.607.

[64] Z. Ren and Y. Lee, "Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery", Nov. 2017.

[65] X. Wang, K. He, and A. Gupta, "Transitive Invariance for Self-Supervised Visual Representation Learning", in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 1338–1347. DOI: 10.1109/ICCV.2017.149.

[66] C. Doersch and A. Zisserman, "Multi-task Self-Supervised Visual Learning", in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2070–2079. DOI: 10.1109/ICCV.2017.226.

[67] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, *Boosting Self-Supervised Learning via Knowledge Transfer*. May 2018.

[68] L. Jing, X. Yang, J. Liu, and Y. Tian, "Self-Supervised Spatiotemporal Feature Learning via Video Rotation Prediction", *arXiv:1811.11387 [cs]*, Apr. 2019. arXiv: 1811.11387 [cs].

[69] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization", in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18, Montréal, Canada: Curran Associates Inc., Dec. 2018, pp. 7774–7785.

[70] A. Owens and A. A. Efros, "Audio-Visual Scene Analysis with Self-Supervised Multisensory Features", en, in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 639–658. DOI: 10.1007/978-3-030-01231-1_39.

[71] N. Sayed, B. Brattoli, and B. Ommer, "Cross and Learn: Cross-Modal Self-Supervision", *GCPR*, 2018. DOI: 10.1007/978-3-030-12939-2_17.

[72] A. Mahendran, J. Thewlis, and A. Vedaldi, "Cross Pixel Optical-Flow Similarity for Self-supervised Learning", in, May 2019, pp. 99–116. DOI: 10.1007/978-3-030-20873-8_7.

[73] K. Dahun, D. Cho, and S.-O. Kweon, "Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8545–8552, Jul. 2019. DOI: 10.1609/aaai.v33i01.33018545.

[74] S. Purushwalkam and A. Gupta, "Pose from Action: Unsupervised Learning of Pose Features based on Motion", *arXiv:1609.05420 [cs]*, Sep. 2016. arXiv: 1609.05420 [cs].

[75] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient Sound Provides Supervision for Visual Learning", en, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 801–816. DOI: 10.1007/978-3-319-46448-0_48.

[76] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning Sound Representations from Unlabeled Video", *arXiv:1610.09001 [cs]*, Oct. 2016. arXiv: 1610.09001 [cs].

[77] R. Arandjelović and A. Zisserman, "Look, Listen and Learn", *arXiv:1705.08168 [cs]*, Aug. 2017. arXiv: 1705.08168 [cs].

[78] M. Lin, Q. Chen, and S. Yan, "Network In Network", *arXiv:1312.4400 [cs]*, Mar. 2014. arXiv: 1312.4400 [cs].

[79] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting Self-Supervised Visual Representation Learning", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 1920–1929. DOI: 10.1109/CVPR.2019.00202.

[80] B. Liu, Z. Wu, H. Hu, and S. Lin, "Deep Metric Transfer for Label Propagation with Limited Annotated Data", en, Dec. 2018.

[81] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, "Automatically Discovering and Learning New Visual Categories with Ranking Statistics", in *International Conference on Learning Representations*, Sep. 2019.

[82] Y.-C. Hsu, Z. Lv, and Z. Kira, "Learning to cluster in order to transfer across domains and tasks", *arXiv:1711.10125 [cs]*, Mar. 2018. arXiv: 1711.10125 [cs].

[83] Y.-C. Hsu, Z. Lv, J. Schlosser, P. Odom, and Z. Kira, "Multi-class Classification without Multi-class Labels", *arXiv:1901.00544 [cs, stat]*, Jan. 2019. arXiv: 1901.00544 [cs, stat].

[84] J. MacQueen, "Some methods for classification and analysis of multivariate observations", EN, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, The Regents of the University of California, 1967.

[85] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000. DOI: 10.1109/34.868688.

[86] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", en, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002. DOI: 10.1109/34.1000236.

[87] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm", in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., MIT Press, 2002, pp. 849–856.

[88] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis", en, in *International Conference on Machine Learning*, PMLR, Jun. 2016, pp. 478–487.

[89] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep Adaptive Image Clustering", en, in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 5880–5888. DOI: 10.1109/ICCV.2017.626.

[90] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features", en, in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 139–156. DOI: 10.1007/978-3-030-01264-9_9.

[91] N. Mrabah, N. M. Khan, R. Ksantini, and Z. Lachiri, "Deep clustering with a Dynamic Autoencoder: From reconstruction towards centroids construction", en, *Neural Networks*, vol. 130, pp. 206–228, Oct. 2020. DOI: 10.1016/j.neunet.2020.07.005.

[92] K. Han, A. Vedaldi, and A. Zisserman, "Learning to Discover Novel Visual Categories via Deep Transfer Clustering", *arXiv:1908.09884 [cs]*, Aug. 2019. arXiv: 1908.09884 [cs].

[93] M. Śmieja, Ł. Struski, and M. A. T. Figueiredo, "A classification-based approach to semi-supervised clustering with pairwise constraints", en, *Neural Networks*, vol. 127, pp. 193–203, Jul. 2020. DOI: 10.1016/j.neunet.2020.04.017.

[94] S.-A. Rebuffi, S. Ehrhardt, K. Han, A. Vedaldi, and A. Zisserman, "LSD-C: Linearly Separable Deep Clusters", *arXiv:2006.10039 [cs]*, Jun. 2020. arXiv: 2006.10039 [cs].

[95] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[96] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks", in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'08, Chicago, Illinois: AAAI Press, Jul. 2008, pp. 646–651.

[97] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-Based Classification for Zero-Shot Visual Object Categorization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, Mar. 2014. DOI: 10.1109/TPAMI.2013.140.

[98] M. Rohrbach, M. Stark, and B. Schiele, "Evaluating knowledge transfer and zero-shot learning in a large-scale setting", en, in *CVPR 2011*, Colorado Springs, CO, USA: IEEE, Jun. 2011, pp. 1641–1648. DOI: 10.1109/CVPR.2011.5995627.

[99] X. Yu and Y. Aloimonos, "Attribute-Based Transfer Learning for Object Categorization with Zero/One Training Example", en, in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2010, pp. 127–140. DOI: 10.1007/978-3-642-15555-0_10.

[100] X. Xu, F. Shen, Y. Yang, D. Zhang, H. T. Shen, and J. Song, "Matrix Tri-Factorization with Manifold Regularizations for Zero-Shot Learning", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2007–2016. DOI: 10.1109/CVPR.2017.217.

[101] Z. Ding, M. Shao, and Y. Fu, "Low-Rank Embedded Ensemble Semantic Dictionary for Zero-Shot Learning", en, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6005–6013. DOI: 10.1109/CVPR.2017.636.

[102] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-Shot Learning – A Comprehensive Evaluation of the Good, the Bad and the Ugly", *arXiv:1707.00600 [cs]*, Sep. 2020. arXiv: 1707.00600 [cs].

[103] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 951–958. DOI: 10.1109/CVPR.2009.5206594.

[104] B. Romera-Paredes and P. H. S. Torr, "An Embarrassingly Simple Approach to Zero-Shot Learning", en, in *Visual Attributes*, R. S. Feris, C. Lampert, and D. Parikh, Eds., Cham: Springer International Publishing, 2017, pp. 11–30. DOI: 10.1007/978-3-319-50077-5_2.

[105] Y. Atzmon and G. Chechik, "Probabilistic AND-OR Attribute Grouping for Zero-Shot Learning", *arXiv:1806.02664 [cs]*, Jul. 2018. arXiv: 1806.02664 [cs].

[106] Z. Akata, S. Reed, D. Walter, Honglak Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification", en, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 2927–2936. DOI: 10.1109/CVPR.2015.7298911.

[107] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-Shot Learning Through Cross-Modal Transfer", in *Advances in Neural Information Processing Systems*, 2013, pp. 935–943.

[108] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings", English (US), in *2nd International Conference on Learning Representations, ICLR 2014*, Jan. 2014.

[109] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality", in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, Lake Tahoe, Nevada: Curran Associates Inc., Dec. 2013, pp. 3111–3119.

[110] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes", *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, Jul. 2006. DOI: 10.1109/TNN.2006.875977.

[111] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-Supervised and Unsupervised Extreme Learning Machines", *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014. DOI: 10.1109/TCYB.2014.2307349.

[112] L. L. C. Kasun, Y. Yang, G.-B. Huang, and Z. Zhang, "Dimension Reduction With Extreme Learning Machine", *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3906–3918, Aug. 2016. DOI: 10.1109/TIP.2016.2570569.

[113] J. Zhang, W. Xiao, Y. Li, S. Zhang, and Z. Zhang, "Multilayer probability extreme learning machine for device-free localization", en, *Neurocomputing*, vol. 396, pp. 383–393, Jul. 2020. DOI: 10.1016/j.neucom.2018.11.106.

[114] C. M. Wong, C. M. Vong, P. K. Wong, and J. Cao, "Kernel-Based Multilayer Extreme Learning Machines for Representation Learning", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 757–762, Mar. 2018. DOI: 10.1109/TNNLS.2016.2636834.

[115] C.-M. Vong, C. Chen, and P.-K. Wong, "Empirical kernel map-based multilayer extreme learning machines for representation learning", en, *Neurocomputing*, vol. 310, pp. 265–276, Oct. 2018. DOI: 10.1016/j.neucom.2018.05.032.

[116] M. D. Tissera and M. D. McDonnell, "Deep extreme learning machines: Supervised autoencoding architecture for classification", en, *Neurocomputing*, vol. 174, pp. 42–49, Jan. 2016. DOI: 10.1016/j.neucom.2015.03.110.

[117] M. D. Tissera and M. D. McDonnell, "Modular expansion of the hidden layer in Single Layer Feedforward neural Networks", in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 2939–2945. DOI: 10.1109/IJCNN.2016.7727571.

[118] T. Wang, J. Cao, X. Lai, and B. Chen, "Deep Weighted Extreme Learning Machine", en, *Cognitive Computation*, vol. 10, no. 6, pp. 890–907, Dec. 2018. DOI: 10.1007/s12559-018-9602-9.

[119] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression", en, *Neurocomputing*, vol. 311, pp. 126–136, Oct. 2018. DOI: 10.1016/j.neucom.2018.05.057.

[120] N. Srivastava, G. 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. 56, pp. 1929–1958, 2014.

[121] A. Krizhevsky, "Learning multiple layers of features from tiny images", Tech. Rep., 2009.

[122] A. Coates, A. Ng, and H. Lee, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning", *Journal of Machine Learning Research - Proceedings Track*, vol. 15, pp. 215–223, Jan. 2011.

[123] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms", *arXiv:1708.07747 [cs, stat]*, Sep. 2017. arXiv: 1708.07747 [cs, stat].

[124] *MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges*, http://yann.lecun.com/exdb/mnist/.

[125] M.-E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes", in *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, Dec. 2008, pp. 722–729. DOI: 10.1109/ICVGIP.2008.47.

[126] M. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A python package for convex optimization (version 1.2)", 2020.

[127] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning Augmentation Policies from Data", *arXiv:1805.09501 [cs, stat]*, Apr. 2019. arXiv: 1805.09501 [cs, stat].

[128] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning", *arXiv:1606.04080 [cs, stat]*, Dec. 2017. arXiv: 1606.04080 [cs, stat].

[129] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts", *arXiv:1608.03983 [cs, math]*, May 2017. arXiv: 1608.03983 [cs, math].

[130] H. Xie, M. E. Hussein, A. Galstyan, and W. Abd-Almageed, "MUSCLE: Strengthening Semi-Supervised Learning Via Concurrent Unsupervised Learning Using Mutual Information Maximization", *arXiv:2012.00150 [cs]*, Nov. 2020. arXiv: 2012.00150 [cs].

[131] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations", *arXiv:2002.05709 [cs, stat]*, Jun. 2020. arXiv: 2002.05709 [cs, stat].

[132] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4L: Self-Supervised Semi-Supervised Learning", *arXiv:1905.03670 [cs]*, Jul. 2019. DOI: 10.1109/iccv.2019.00156. arXiv: 1905.03670 [cs].

[133] L. Jing and Y. Tian, "Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey", *arXiv:1902.06162 [cs]*, Feb. 2019. DOI: 10.1109/tpami.2020.2992393. arXiv: 1902.06162 [cs].

[134] S.-A. Rebuffi, S. Ehrhardt, K. Han, A. Vedaldi, and A. Zisserman, "Semi-Supervised Learning with Scarce Annotations", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 3294–3302. DOI: 10.1109/CVPRW50498.2020.00389.

[135] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning", in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[136] X. Jia, K. Han, Y. Zhu, and B. Green, "Joint Representation Learning and Novel Category Discovery on Single- and Multi-modal Data", *arXiv:2104.12673 [cs]*, Apr. 2021. arXiv: 2104.12673 [cs].

[137] L. M. L. Cam and J. Neyman, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability: Weather Modification*, en. University of California Press, 1967.

[138] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin, and P. Bojanowski, "Self-supervised Pretraining of Visual Features in the Wild", *arXiv:2103.01988 [cs]*, Mar. 2021. arXiv: 2103.01988 [cs].

[139] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database", en, *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[140] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction", en, *Science*, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015. DOI: 10.1126/science.aab3050.